EXHIBIT E2

DECLARATION OF MICHAEL FRATTO

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No. 6,502,135	)	
	)	
Munger et al.	)	Group Art Unit: Central Reexamination Unit
	)	
Filed: February 15, 2000	)	
	)	
For: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY	)	

DECLARATION OF MICHAEL ALLYN FRATTO UNDER 37 C.F.R. § 1.132

I, MICHAEL ALLYN FRATTO, do hereby declare and state:

1. I am a citizen of the United States, and reside in Syracuse, New York. My c.v. is attached as Exhibit A.

2. I am presently Editor of the Network Computing magazine and website. In that position, I review and evaluate networking products, including network security products, and report on industry developments in the field of networking and network security. I also write articles about network infrastructure, data center, and network access control items which are published on the Network Computing website.

3. I presently serve as an adjunct faculty member of School of Information Studies at Syracuse University.

4. Since before 1999, I have had an extensive background and experience in network security systems, software and related technologies. I have been on staff of Network Computing conducting and writing comparative product reviews of networking and security products for the magazine, interviewing IT administrators and executives about networking and security issues trying to understand their needs. During the course of a review, I have to understand a problem set, understand technologies and standards that address a problem set, and create a set of comparative measures to asses a products ability to execute. I would set up a test network, verify its operation, conduct the tests, and ensure the results were accurate. In the 1997 to 2000 time frame, I focused on remote access products including modems, ISDN, and virtual private networking products, technologies, and standards as well as network and host-based firewalls.

5. I am being compensated for my time at a rate of $250.00 per hour.

A. Public Availability of Certain Aventail Documents

6. Between 1997 and 1999, I reviewed and published articles on a number of VPN products distributed by Aventail, Inc. I recall that Aventail distributed two series of VPN products during this period. One series included client software called "AutoSOCKS" and server software called "Aventail VPN Server" or "Aventail Mobile VPN." A later product was called the Aventail Extranet Center ("AEC"), which included client software called "Aventail Connect" and server software called "Aventail Extranet Server."

7. Aventail distributed several versions of the client and server products in each series during this period. I recall receiving and evaluating at least versions 2.1, 2.2 and 2. 6 of the AutoSOCKS product, versions 3.01/2.51 and 3.1/2.6 of the Aventail Connect product, and at least versions 3.01 and 3.1 of the AEC product.

1. Aventail AutoSOCKS v2.1 Administrator's Guide

8. Exhibit B is a copy of the Aventail AutoSOCKS v2.1 Administration & User's Guide ("AutoSOCKS"). Exhibit B was distributed with the AutoSOCKS v2.1 software product.

9. Aventail announced that they had shipped version 2.1 of AutoSOCKS in May of 1997. See Exhibit C (PR Newswire, "Aventail Ships the First Standards-Based Virtual Private Network Software Solution," May 2, 1997). On June 23, 1997, InfoWorld published a review of the AutoSOCKS v2.1 product. See Exhibit D (Infoworld, Vol. 19, Issue 25 (June 23, 1997) at page 70).

10. I recall receiving a copy of Exhibit B with Aventail Autosocks v2.1 no later than March of 1997. The copy of Exhibit B that I received in March of 1997 was not marked as being confidential, and no restrictions were imposed on my use of it or information in it.

11. I also recall receiving and reviewing subsequent versions of Aventail AutoSOCKS between September of 1997 and June of 1998. For example, I published an article evaluating the Aventail VPN Server version 2.5 and AutoSOCKS version 2.2 in Network Computing in October of 1997. See Exhibit E (Fratto, "Aventail VPN 2.5: Not Your Father's Socks," Network Computing, Vol. 8, No. 18 (October 1, 1997)). I also published a review of the Aventail VPN Server v2.6 in June of 1998. See, Exhibit F (Fratto, "Footlose and Fancy Free with Three SOCKS 5-based proxy servers," Network Computing, Vol. 9, Issue 11 (June 15, 1998)).

2. Aventail Connect v3.01/2.51 Administrator's Guide

12. Exhibit G is a copy of the Aventail Connect v3.01/2.51 Administrator's Guide ("Aventail Connect v3.01"). The Aventail Connect 3.01/2.51 Administrator's Guide was distributed with the AEC v3.0 product.

13. Aventail announced AEC v3.0 in August of 1998. See Exhibit H (PR Newswire, "Aventail Ships Directory-enabled Extranet Solution; Aventail Extranet Center V3.1 Available At www.aventail.com." (August 9, 1999)). The AEC v3.0 product was

distributed by Aventail in the fall of 1998. See, for example, Exhibit I ("Intranet Applications: Briefs," Network World, at page 55 (October 19, 1998)).

14. I recall receiving Exhibit G with the Aventail Extranet Center v3.0 product in approximately October of 1998. The copy of Exhibit G that I received in October of 1998 was not marked as being confidential, and no restrictions were imposed on my use of it or information in it.

3. Aventail Connect v3.1/2.6 Administrator's Guide

15. Exhibit J is a copy of the Aventail Connect v3.1/v2.6 Administrator's Guide ("Aventail Connect v3.1"). Exhibit J was distributed with the Aventail Extranet Center (AEC) v3.1 product.

16. Aventail announced that they had begun shipping the AEC v3.1 product in August of 1999. See, Exhibit K ("Aventail Ships Directory-enabled Extranet Solution; Aventail Extranet Center v3.1 available at www.aventail.com," Business Wire (August 9, 1999)).

17. I recall receiving a pre-release copy of the AEC v3.1 product sometime during April of 1999. The AEC v3.1 product I received included installation media for the Aventail Extranet Server, Aventail Connect and Aventail Management Server and Config Tool software packages. It also included printed administrator guides for the three software packages.

18. The AEC v3.1 product and the Aventail Connect v3.1 Administrator's Guide that I received in April of 1999 was not marked as being confidential, no restrictions were imposed on my use of it or the information in it.

19. On June 28, 1999, Network Computing published an article I wrote about the Aventail ExtraNet Center (AEC) v3.1 product. A copy of this article is provided as Exhibit K.

20. Before preparing Exhibit K, I oversaw the installation of the Extranet Server and Aventail Connect software on computers in our testing facility at Syracuse University in April of 1999. Between April and June of 1999, I directed or performed a series of tests and evaluations of the features and functionality of the AEC v3.1 product.

21. My June 28, 1999 Network Computing article indicates that I tested a "beta" version of the AEC 3.1 product. The versions of the three software packages that I tested between April and June of 1999 were stable and feature-complete. I am not aware of any significant differences between the versions of the products that I tested and those that were shipped to customers later that summer. For example, I note that Network Computing ordinarily would perform reviews on "pre-release" versions of products within about 30 days of the release of the final version of the product.

B. Discussion of the Aventail AutoSocks v2.1 Administrator's Guide

22. As I explained above, Aventail AutoSOCKS was the client component of a VPN solution that was distributed by Aventail Corporation. AutoSOCKS ran on client computers running the Windows operating systems.

23. AutoSOCKS would act on DNS requests made by applications running on the client computer, such as web browsers and email clients. When one of these applications made a DNS request (e.g., a user typed a hostname in a web browser), AutoSOCKS would intercept the DNS request, evaluate it and automatically establish the VPN if the client was authorized.

24. AutoSOCKS did this transparently – neither the requesting application nor the user would know that AutoSOCKS was functioning. See Exhibit B, page 1 ("AutoSOCKS transparently intercepts WinSock communication requests issued by TCP/IP applications and processes them based upon a set of routing directives (rules) assigned when AutoSOCKS is configured."); Exhibit B, page 6 ("AutoSOCKS is designed to run transparently on each workstation. In most cases, you'll interact with AutoSOCKS only when it prompts you to enter authentication information for a connection to a secure SOCKS server. You may also occasionally need to start and exit AutoSOCKS, although network administrators often configure it to run automatically at startup."); Exhibit B, page 7 ("With AutoSOCKS running, an application executes additional steps in order to connect to a remote host through WinSock. These steps must be transparent to the application so that it cannot differentiate between when AutoSOCKS is running and when it is not. The following three steps are identical to standard WinSock communications steps described above; however, nested inside them are additional actions and options introduced by AutoSOCKS.")(emphasis added).

25. Exhibit B, at pages 6 to 8, explains that AutoSOCKS intercepted DNS requests by working within the existing TCP/IP handling procedures of a client computer. This part of Exhibit B shows that AutoSOCKS would replicate the way that handle DNS requests were handled WinSock and the TCP/IP stack on client computers running Windows. On page 6, Exhibit B explained the general way DNS requests were handled by the Windows operating system:

> Via WinSock, an application goes through the following steps to connect to a remote host on the Internet or corporate intranet:
>
> 1. The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address. If the application already knows the IP address, this step is skipped.
>
> 2. The application requests a connection to the specified remote host. This causes the underlying stack to begin the TCP handshake. (The TCP handshake is the process by which two computers initiate communication with each other.) When the handshake is complete,

the application is notified that the connection is established, and that data may now be transmitted and received.

3. The application sends and receives data.

26. So, if an application on the client computer made a DNS request that contained a hostname (e.g., a domain name specifying a website), a DNS resolution step would be performed to determine the IP address of the target. If the application made a DNS request that included the "real" IP address (e.g., 1.2.3.4), there would be no need for resolution of the hostname.

27. Exhibit B on page 7 explains how AutoSOCKS worked in conjunction with the native TCP/IP handling procedures of the client computer:

> AutoSOCKS slips in between the Windows TCP/IP application and the single access point – WinSock. In simple terms, AutoSOCKS redirects WinSock calls (both parameters and data) and reroutes them through a SOCKS-based server when required. The routing is determined by the rules described in the configuration file created when AutoSOCKS is installed (See "Configuring AutoSOCKS").
>
> Because AutoSOCKS intercepts calls to Winsock, AutoSOCKS must duplicate WinSock functionality. Since AutoSOCKS also makes calls directly into WinSock, it must behave as a typical WinSock application as well. (See Figure 1.)

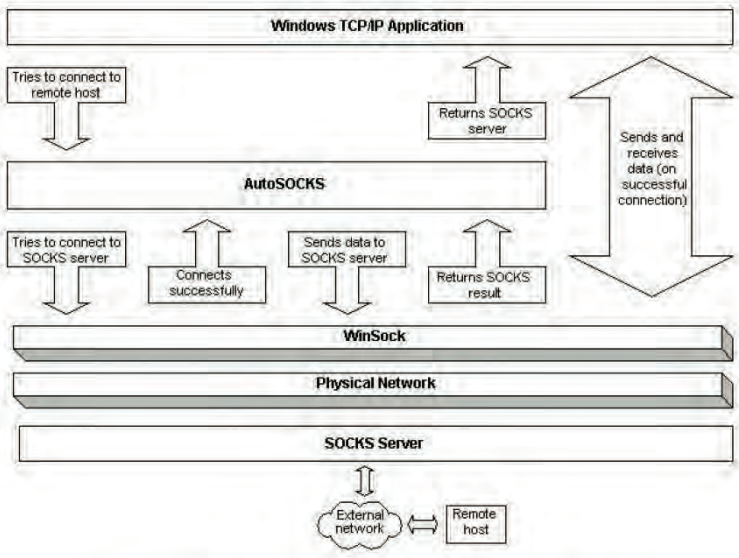
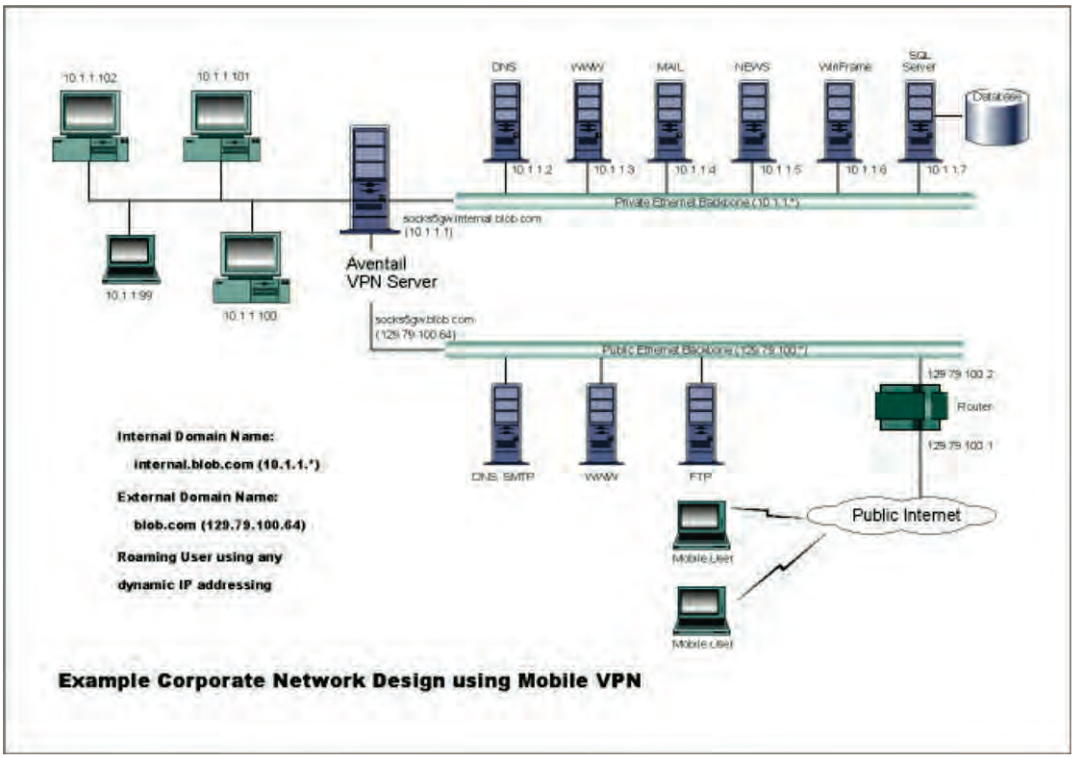


Figure 1. Network application calls intercepted by AutoSOCKS

28. Because AutoSOCKS had to replicate the WinSock functionality of the Windows OS, it inherently would be able to handle standard TCP/IP protocols for handshaking, routing and transmission that were defined in TCP/IP standards. For example, AutoSOCKS

would have to be able to recognize and handle standard error messages that are provided by an unsuccessful transmission under the TCP/IP protocol (e.g., "host not found").

29. Page 37 of Exhibit B describes a VPN based on the use of AutoSOCKS software on client computers (called "mobile users") and the Aventail VPN Server software (also called "Mobile VPN") running on a separate computer that sits between the private network and the public Internet and regulates access to the private network (a "gateway" computer).



AutoSOCKS in an Aventail Mobile VPN Environment

30. The following explanation of this VPN is provided on page 38 of Exhibit B:

> The mobile user workstations connected to the public Internet are the client workstations, onto which, AutoSOCKS will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail VPN Server unless they are running AutoSOCKS. Depending on the security policy and the Aventail VPN Server configuration, AutoSOCKS will automatically proxy their allowed application traffic into the private network. In this [sic] situation, AutoSOCKS will forward traffic destined for the private internal network to the Aventail VPN Server. Then, based on the security policy, the Aventail VPN Server will proxy mobile end user traffic into the private network but only to those resources allowed. (emphasis added)

31. In the VPN described on pages 37 to 39 of Exhibit B, communications between the mobile user and the target computer inside the private network are both authenticated and encrypted ("end user authentication and encryption has been enabled on the Aventail VPN Server, which will require all end users to use AutoSOCKS to enable authentication and encryption of their sessions before being allowed to have any connectivity to the internal private network(s).")

32. Client computers running AutoSOCKS in the VPN described on pages 37 to 39 of Exhibit B handle requests for non-secure destinations, such as a web site on the Internet, by simply passing the DNS request on to the client computer for local handling. As explained on page 39 of Exhibit B:

> Second, not all traffic is passed through to the Aventail VPN Server. Only traffic that is destined for the internal network is authenticated and encrypted, all other traffic passes through AutoSOCKS unchanged. For example, browsing the Internet from the mobile user workstation occurs as if AutoSocks was not even running in the background. (emphasis added)

33. This feature is explained in more detail on page 8 of Exhibit B, which specifies:

> If the hostname matches a local domain string or does not match a redirection rule, AutoSOCKS passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack then performs the lookup as if AutoSOCKS is not running.

34. A "redirection rule" is used by the AutoSOCKS software running on a client computer to determine if a destination specified in a DNS request requires authentication and/or encrypted communications. Redirection refers to the transfer of the DNS request to another computer for hostname resolution (if required) and handling.

35. A redirection rule identifies a target destination by hostname or IP address, and for each destination, defines how the client computer running AutoSOCKS would handle a DNS request containing that hostname or IP address. How redirection rules were implemented in AutoSOCKS is described in more detail on pages 20 to 27 of Exhibit B.

36. First, destinations would be entered using the configuration tool. See pages 20-23. Then, for each destination, options would be entered that told AutoSOCKS how to handle DNS entries matching that destination. See pages 23-27. This configuration process yielded a table of entries having hostnames and/or IP addresses, and policies that were to be followed for those entries (e.g., redirect DNS request to specified server, deny DNS request, route DNS request to destination).

37. For example, as shown on page 25 of Exhibit B, a client computer running AutoSOCKS v2.1 could be configured to route network traffic for a specified destination to a SOCKS server, directly to the specified destination or block access locally to that destination.

38. In addition, a client computer running AutoSOCKS could be configured in one of two ways. First, it could be configured to locally resolve DNS requests containing hostnames. As explained on page 26 of Exhibit B:

> Local Name Resolution instructs AutoSOCKS to resolve hostnames locally without needing to venture on to the Internet. This option feature offers you another level of control over how AutoSOCKS performs name resolution.
>
> The local workstation resolver is the name resolution component of the local TCP/IP stack. This feature acts as a shortcut; hostnames matching the strings defined in the Local Name Resolution dialog box are passed to the local resolver for name resolution instead of being proxied through the SOCKS v5 server.

39. In other words, AutoSOCKS could perform DNS resolution on the client computer to yield an IP address from a hostname in a DNS request without accessing the Internet using its local name resolution feature.

40. Pages 7 to 9 of Exhibit B describe how AutoSOCKS worked to determine if DNS requests contained destinations that required a VPN.

41. Initially, the client computer running AutoSOCKS would determine if a DNS request contained a hostname requiring resolution (i.e., a determination of the IP address associated with the hostname). If it did, then AutoSOCKS would do the following, as described on pages 7 and 8 of Exhibit B:

- If the client computer running AutoSOCKS determined that a "hostname matches a local domain string or does not match a redirection rule, AutoSOCKS passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack then performs the lookup as if AutoSOCKS is not running." A hostname would match a local domain string if the hostname specified a computer on the local network or was the same as an entry in a local resolution rule. See page 8 of Exhibit B, step 1.

- If the client computer running AutoSOCKS determined that the DNS request contained a hostname matching "a redirection rule domain name (i.e. the host is part of a domain we are proxying traffic to) then AutoSOCKS creates a false DNS entry (HOSTENT) that it can recognize during the connection request. AutoSOCKS will forward the hostname to the SOCKS server in step 2 and the SOCKS server performs the hostname resolution." See, page 8 of Exhibit B, step 1.

42. A second way in which client computers running AutoSOCKS could be configured was to send all DNS requests containing hostnames requiring resolution (other than those matching a local domain string) to another computer (i.e., a DNS proxy server) for resolution. In particular, the client computer would establish communication with the

Aventail VPN Server and, after being authenticated, would send the hostname from the initial DNS request to the Aventail VPN Server (the "SOCK server"). The Aventail VPN server would resolve the received hostname and then determine whether it was necessary to establish a VPN. Page 8 of Exhibit D describes the outset of the process:

- If the DNS proxy option is enabled and the domain cannot be looked up directly, AutoSOCKS creates a fake DNS entry that it can recognize later, and returns this to the calling application. The false entry tells AutoSOCKS that the DNS lookup should be proxied, and that it should send the fully qualified hostname to the SOCKS server with the SOCKS connection request.

43. In other words, if the DNS proxy option was enabled, all DNS requests containing hostnames matching a redirection rule and not matching a local domain string would be sent to the DNS proxy server for resolution. If the DNS proxy option were not enabled, then only DNS requests in the first category (i.e., containing hostnames matching a redirection rule) would be sent to the DNS proxy server for resolution.

44. Page 8 of Exhibit B explains that AutoSOCKS would flag DNS requests containing hostnames matching a redirection rule in the first step by inserting a false DNS entry that AutoSOCKS could recognize in the second step of the process AutoSOCKS followed. In that second step, AutoSOCKS would use the false DNS flag to identify those DNS requests containing a hostname that had to be proxied (i.e., those that required a VPN).

45. As explained on pages 8 and 9 of Exhibit B, AutoSOCKS monitored TCP/IP connection requests to determine if a request was seeking access to a destination that required authentication and/or encrypted communications (e.g., a secure website inside a private network in the VPN described on pages 37 to 39).

- If AutoSOCKS determined that connection request contained a false DNS entry that been inserted during a DNS resolution in step 1 (e.g., because the DNS request specified a hostname matching a redirection rule) or if the connection request contained a "real" IP address that matched a redirection rule, then the client computer running AutoSOCKS would call WinSock to begin the TCP handshake with a predefined server (e.g., a "gateway" or "proxy" computer hosting the VPN Server shown on pages 37 to 39).

- If AutoSOCKS determined that a connection request specified a destination that did not match a redirection rule (e.g., a non-secure web site on the Internet), AutoSOCKS would hand the connection request off to the client computer for handling by the TCP/IP stack on that computer. In that situation, if the DNS request contained a hostname, then, depending on the configuration of the AutoSOCKS client, either the client computer or a designated server computer would resolve the hostname, and the resolved IP address would then be handed off to the local

> computer. If the DNS request contained a "real" IP address, there would be no need to conduct step 1 (i.e., resolution of the hostname).

46. If AutoSOCKS determined that it needed to establish a VPN based on its evaluation of the connection request, it would perform the following steps as explained on page 8 of Exhibit B:

> When the connection is completed, AutoSOCKS begins the SOCKS negotiation.
>
> - It sends the list of authentication methods enabled in the configuration file.
> - Once the server chooses an authentication method, AutoSOCKS executes the specified authentication processing.
> - It then sends the proxy request to the SOCKS server. This includes either the IP address provided by the application or the DNS entry (hostname) provided in step 1.

47. In other words, AutoSOCKS would not send the proxy computer the information in the original DNS request (i.e., the hostname requiring DNS resolution or the real IP address) to the DNS proxy server until after the user had successfully authenticated with the server. If Step 2.b was successfully completed, "AutoSOCKS notifies the application" and then, in step 2.c, "the application transmits and receives data [from the secure destination]." See Exhibit B, pages 8 to 9.

48. I note that when Aventail Connect was configured to proxy all DNS requests to the proxy server for resolution, Aventail VPN Server rather than the client computer running AutoSOCKS would resolve the hostname (if necessary) and determine if the VPN had to be established. While Exhibit B does not explain that the VPN Server would perform both steps, this is inherent in the way that AutoSOCKS worked. For example, as I explain in paragraph 49, authentication of the client preceded the sending of the DNS request that contained the hostname. Once the client had been successfully authenticated, there would be no reason to send the resolved hostname back to the client so that the client could send that same information back to the Aventail VPN Server.

49. AutoSOCKS could be configured to use any of a number of different authentication procedures and techniques including as described on pages 27 to 35 of Exhibit B.

50. AutoSOCKS and the VPN Server required authentication to succeed before a VPN would be established and data communications could proceed. See Exhibit B, page 27 ("SOCKS v5 servers often require user authentication before allowing access. AutoSOCKS authentication modules facilitate this process by displaying dialog boxes which ask for username and password information as well as other authentication credentials."); page 37 ("End user authentication and encryption has been enabled on the Aventail VPN Server, which will require all end users to use AutoSOCKS to enable

authentication and encryption of their sessions before being allowed to have any connectivity to the internal private network(s).") (emphasis added)

51. So, in the VPN described on pages 37 to 39 of Exhibit B, a mobile user with a client computer running AutoSOCKS would have to successfully authenticate itself before the VPN Server would automatically establish the VPN and allow data transmissions between the client and the secure destination to proceed.

52. If authentication failed, an error would be returned to the client computer running Aventail AutoSOCKS, and depending on the configuration of the client, an error notification would be provided to the user. For example, if AutoSOCKS was configured to use a certificate-based authentication procedure, and the server certificate was suspect, AutoSOCKS would display the certificate to the user during the authentication process and, depending on the configuration of the client, reject the connection. See Exhibit B at page 34. AutoSOCKS could also be configured to simply reject the connection in this scenario. See Exhibit B at pages 33-34.

53. Client computers running AutoSOCKS could be configured to encrypt all data communications occurring after a TCP/IP connection was established and authenticated. See, Exhibit B at page 9 ("If an encryption module is enabled and selected by the SOCKS server, AutoSOCKS encrypts the data on its way to the server on behalf of the application. If the data is being returned, AutoSOCKS decrypts it so that the application sees clear text data.")

54. In the VPN described on pages 37 to 39 of Exhibit B, AutoSOCKS is configured to require authentication and encryption of all communications other than those to non-secure web sites:

> End user authentication and encryption has been enabled on the Aventail VPN Server, which will require all end users to use AutoSOCKS to enable authentication and encryption of their sessions before being allowed to have any connectivity to the internal private network(s). For this example, the Aventail VPN Server is configured to use SSL for encryption of all sessions. …
>
> Second, not all traffic is passed through to the Aventail VPN Server. Only traffic that is destined for the internal network is authenticated and encrypted, all other traffic passes through AutoSOCKS unchanged.

C. Discussion of the Aventail Connect v3.01/v2.51 Administrator's Guide

55. The Aventail Connect v3.01/v2.51 client shared much of the functionality of the Aventail AutoSOCKS client. Like the AutoSOCKS client described above, Aventail Connect v3.01/2.51 worked by automatically authenticating and encrypting communications between a client computer running Aventail Connect and a private network resource via a VPN server called the Aventail Extranet Server. How Aventail Connect did this is described on page 7 of Exhibit G:

> When you run Aventail Connect on your system, it automatically routes appropriate network traffic from a WinSock application to an extranet (SOCKS) server, or through successive servers. (WinSock is a Windows component that connects a Windows PC to the Internet using TCP/IP.) The SOCKS server then sends the traffic to the Internet or the external network. Network administrators can define a set of rules that route this traffic.
>
> Aventail Connect is designed to run transparently on each workstation, without adding overhead to the user's desktop. In most cases, users will interact with Aventail Connect only when it prompts them to enter authentication credentials for a connection to a secure extranet (SOCKS) server. Users may also occasionally need to start and exit Aventail Connect, although network administrators often configure it to run automatically at startup. Aventail Connect does not require administrators to manually establish an encrypted tunnel; Aventail Connect can establish an encrypted tunnel automatically. (emphasis added)

56. In other words, a client computer running Aventail Connect v3.01/2.51 (i) operated transparently to the user and the client computer, (ii) would automatically authenticate a user attempting to access a secure location, (iii) would automatically encrypt communications between a client computer and the secure network destination, and (iv) that network administrators could route the TCP/IP traffic through intermediary destinations between the client computer and the final secure network destination.

57. Aventail Connect worked with applications that communicate via TCP/IP, and was implemented in WinSock on client computers running Windows. Among other things, this meant that the client computer running Aventail Connect would act on DNS requests, which could contain either hostnames or IP addresses.

58. As explained on page 8 of Exhibit G, an application on a client computer running Windows, via WinSock, "goes through the following steps to connect to a remote host on the Internet or corporate network:

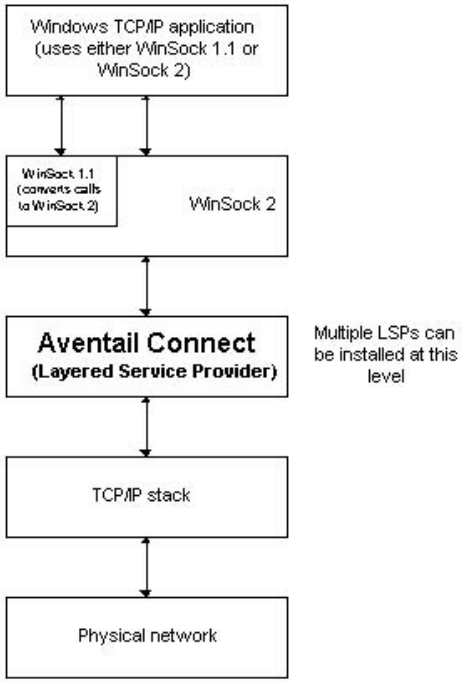
    1. The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address. If the application already knows the IP address, this step is skipped.

    2. The application requests a connection to the specified remote host. This causes the underlying stack to begin the TCP handshake, when two computers initiate communication with each other. When the handshake is complete, the application is notified that the connection is established, and data can then be transmitted and received.

    3. The application sends and receives data."

59. Pages 9 to 10 of Exhibit G explain how Aventail Connect functioned within these TCP/IP handling procedures of the client computer:

> Aventail Connect slips in between WinSock and the underlying TCP/IP stack. (See diagram below.) As an application that sits between WinSock and the TCP/IP stack, Aventail Connect 3.01 is a Layered Service Provider (LSP). Aventail Connect can change data (compressing it or encrypting it, for example) before routing it to the TCP/IP stack for transport over the network. The routing is determined by the rules described in the configuration file.



60. Page 10 of Exhibit B explains:

> When the Aventail Connect LSP receives a connection request, it determines whether or not the connection needs to be redirected (to an Aventail ExtraNet Server) and/or encrypted (in SSL). When redirection and encryption are not necessary, Aventail Connect simply passes the connection request, and any subsequent transmitted data, to the TCP/IP stack

61. Thus, if a client computer running Aventail Connect received a DNS request specifying a destination that did not match a redirection rule and thus did not need a VPN (e.g., a non-secure website on the Internet), it would simply pass that DNS request on to the TCP/IP stack of the client computer for handling. In that scenario, the client computer handled the DNS request as if Aventail Connect were not running on the client computer.

62. However, if the client computer running Aventail Connect determined that a DNS request matched a redirection rule requiring a VPN (e.g., a secure website inside a private network), it would automatically handle authentication of the user to the private network and encrypt the communications between the client computer and the private network.

The authentication and encryption steps were transparent both to the client computer and the user. See Exhibit G at page 7 ("Aventail Connect is designed to run transparently on each workstation, without adding overhead to the user's desktop. In most cases, users will interact with Aventail Connect only when it prompts them to enter authentication credentials for a connection to a secure extranet (SOCKS) server.")

63. How Aventail Connect did this is explained on pages 11 to 12 of Exhibit G. First, Aventail Connect would determine if the DNS request contained a hostname requiring resolution (e.g., "securenet.com"). If it did, Aventail Connect would do the following to resolve the hostname, depending on how Aventail was configured.

- "If the hostname matches a local domain string or does not match a redirection rule, Aventail Connect passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack performs the lookup as if Aventail Connect were not running."

- "If the destination hostname matches a redirection rule domain name (i.e., the host is part of a domain we are proxying traffic to) then Aventail Connect creates a false DNS entry (HOSTENT) that it can recognize during the connection request. Aventail Connect will forward the hostname to the extranet (SOCKS) server in step 2 and the SOCKS server performs the hostname resolution."

- "If the DNS proxy option is enabled and the domain cannot be looked up directly, Aventail Connect creates a fake DNS entry that it can recognize later, and returns this to the calling application. The false entry tells Aventail Connect that the DNS lookup must be proxied, and that it must send the fully qualified hostname to the SOCKS server with the SOCKS connection request."

64. In other words, depending on its configuration, Aventail Connect either would flag all DNS requests containing hostnames specifying non-local destinations (i.e., "if the DNS proxy option is enabled") or only those DNS requests with hostnames matching a redirection rule. Hostnames matching a redirection rule were destinations that required a VPN (i.e., authentication and encryption). Also, like the AutoSOCKS product, if the DNS proxy option were enabled, the client computer would establish communication with the Aventail VPN Server (Aventail Extranet Server) and, after being authenticated, would send the hostname from the initial DNS request to the Aventail VPN Server (the "SOCKS server"). The Aventail VPN server would resolve the received hostname and then determine whether it was necessary to establish a VPN.

65. After DNS resolution or in cases where a DNS request specified a "real" IP address, Aventail Connect would handle connection requests as described in step "2" of page 12 of Exhibit G. This section explains that after the TCP/IP handshake was completed (i.e., "by the underlying stack" on the client computer), the application on the client computer would be notified that the TCP/IP connection had been established and that data could

then be transmitted and received. At this point, Aventail Connect would evaluate the connection request, and do one of several things.

(a) If Aventail Connect determined that the connection request contained a "false DNS entry" (i.e., because the DNS request specified a hostname for a secure website in step 1 or because it could not be resolved locally on the client computer), the DNS request would be "proxied" (i.e., sent to another computer for resolution of the hostname).

(b) If Aventail Connect determined that the connection "request contains a routable IP address, and the rules in the configuration file say it must be proxied, Aventail Connect will call WinSock to begin the TCP handshake with the server specified in the configuration file." In other words, if the connection request contained a "real" IP address (e.g., "1.2.3.4") which specified a secure website on a private network which required authentication and encrypted communications, then Aventail Connect would send the connection request to the VPN server for handling (e.g., the Aventail Extranet Server).

(c) If the "request contains a real IP address and the configuration file rule says that it does not need to be proxied, the request will be passed to WinSock and processing jumps to step 3 as if Aventail Connect were not running."

66. As explained on page 12 of Exhibit G, if the hostname or real IP address specified a destination requiring authentication and/or encrypted communications (e.g., a secure website on a private network), Aventail Connect would cause the client computer to communicate with the "proxy" computer (e.g., a "gateway" computer running Aventail Extranet Server). The client computer and the proxy computer would then do the following:

- First, the client computer running Aventail Connect "sends the list of authentication methods enabled in the configuration file."

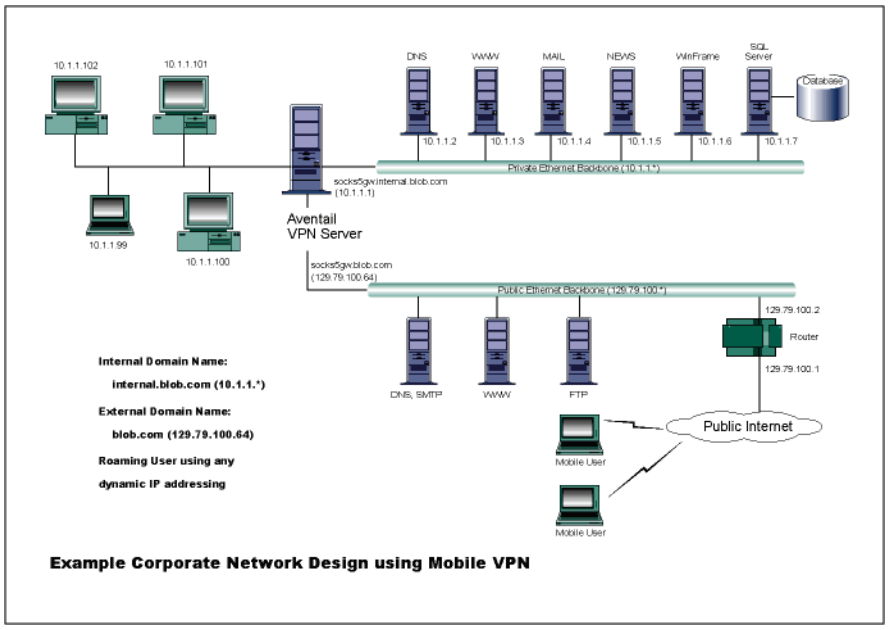
- Then, "once the server selects an authentication method, Aventail Connect executes the specified authentication processing."

- If the authentication step is successful, the client computer running Aventail Connect "then sends the proxy request to the extranet (SOCKS) server. This includes either the IP address provided by the application or the DNS entry (hostname) provided in step 1."

67. In other words, the client computer running Aventail Connect would not send the original hostname in the DNS request to the DNS proxy server (Aventail Extranet Server) for resolution until after the client had been successfully authenticated.

68. Pages 72 to 74 of Exhibit G describe a VPN implemented using the Aventail Connect v3.01/2.51 software running on client computers (called "mobile users") in conjunction

with an Aventail VPN Server (i.e., the Aventail Extranet Server v3.0) running on a running on a separate computer that sits between the private network and the public Internet and regulates access to the private network (a "gateway" computer). A figure describing this VPN is shown on page 73 of Exhibit G and is reproduced below:



69. Pages 72 to 73 of Exhibit G describe how Aventail Connect and Aventail Extranet Server could be configured to establish a VPN between a client computer and network resources on a private network:

> The design used in the example above consists of two individual Ethernet segments, one public and one private. The public segment is used to host anonymous services available to the general public. The public access is provided through a router that is connected to the public Internet. The private segment is used to house all of the corporation's private network resources and data to be used only by internal company employees. The Aventail ExtraNet Server depicted in this example is used to provide secure and monitored access to the private LAN for mobile employees and partners. For security reasons the Aventail ExtraNet Server is configured such that operating system routing is disabled. Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the Aventail ExtraNet Server.
>
> The mobile user workstations connected to the public Internet are the client workstations, onto which, Aventail Connect will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail ExtraNet Server unless they are running Aventail Connect. Depending on the security policy and the Aventail ExtraNet Server configuration, Aventail Connect will

> automatically proxy their allowed application traffic into the private network. In this situation, Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then, based on the security policy, the Aventail ExtraNet Server will proxy mobile user traffic into the private network but only to those resources allowed. The client workstations we focus on in this section are Microsoft Windows based PCs.
>
> User authentication and encryption on the Aventail ExtraNet Server require all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s). For this example, the Aventail ExtraNet Server encrypts all sessions with SSL.
>
> Installing and using Aventail Connect for remote access purposes differs a bit from its installation and use within a corporate network. First, configuration files need to be kept locally on the workstation or laptop. This is due to the inability to share a file server that allows direct access outside the perimeter of the private network. Second, not all traffic passes through to the Aventail ExtraNet Server. Only traffic destined for the internal network is authenticated and encrypted; all other traffic passes through Aventail Connect unchanged. For instance, browsing the Internet from the mobile user workstation occurs as if Aventail Connect is not even running in the background. Large sites with many mobile users will want to set up an internal file server for a network installation for all mobile users to easily install and configure Aventail Connect. For more information, refer to "Network Installation."

70. "Extranets" are functionally identical to VPNs – an "extranet" is simply a VPN that has been established between a non-employee's client computer and the private network. The same procedures and steps are followed regardless of whether a client connection being managed by Aventail Connect was connecting to a "VPN" or an "extranet."

71. Client computers running Aventail Connect could be configured to use a number of different authentication techniques. See Exhibit G, at pages 42 to 58. Logically, in any of these schemes, authentication must succeed before Aventail Connect and the Aventail Extranet Server would permit establishment of a secure connection and transmission of data. See, e.g., page 42 of Exhibit G, which explains:

> SOCKS v5 servers often require authentication before allowing access. Aventail Connect authentication modules display dialog boxes that prompt users to enter username and password information as well as other authentication credentials. (emphasis added)

72. If authentication failed, an error would be returned to the client computer running Aventail Connect, and depending on the configuration of the client, an error notification would be provided to the user. For example, an Aventail Connect client computer configured to use SSL for authentication and encryption would display to the user a

certificate it determined to be suspect, or would both display the certificate and reject a connection based on this authentication failure. See Exhibit G at pages 48 to 49.

73. Communications between a client computer running Aventail Connect and the server computer (e.g., the gateway computer running the Aventail Extranet Server) could be automatically encrypted. As explained on page 12 of Exhibit G:

> If an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server on behalf of the application. If data is being returned, Aventail Connect decrypts it so that the application sees cleartext data.

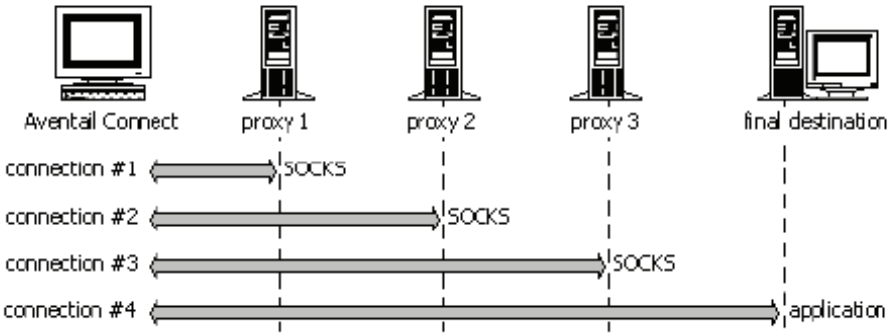
74. So, if a gateway computer running Aventail Extranet Server (the "proxy") was configured to require encrypted communications with client computers running Aventail Connect, then all outgoing and incoming TCP/IP communications would be automatically encrypted and decrypted.

75. On pages 59 to 67, Exhibit G explains that client computers running Aventail Connect and server computers running Aventail Extranet Server could be configured to route TCP/IP communications between the client and the server computers through intermediary destinations according to different routing schemes.

76. One routing scheme in Exhibit G was called "Aventail Multiproxy." This technique is generally described on page 59 of Exhibit B:

> The Aventail MultiProxy feature allows Aventail Connect to traverse multiple firewalls by making connections through successive proxy servers. Aventail Connect makes a connection with each proxy server individually. Each proxy server forms a link in a chain that connects Aventail Connect to the final destination. Any or all of the proxy servers can apply authentication and access control rules. Proxies can be Aventail ExtraNet Servers, other SOCKS 5 servers, SOCKS 4 servers, or HTTP proxies."

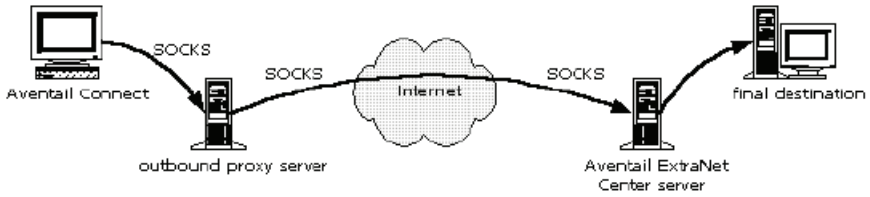
77. As explained on page 60 of Exhibit G, in the MultiProxy scheme, the client computer running Aventail Connect manages the routing of these communications, and handles authentication, encryption and access parameters to each of the intermediary proxy servers, which could be SOCKS servers or HTTP proxy servers.



78. How client computers running Aventail Connect made these connections is described on page 60 of Exhibit G.

> The steps for making a connection using MultiProxy are:
>
> 1. The client application requests access to the destination server.
>
> 2. Aventail Connect establishes a connection with the outbound server (SOCKS server or HTTP proxy). Aventail Connect then sends the access request to the outbound server, specifying the Aventail ExtraNet Server as the destination. The user authenticates with the outbound server, if necessary.
>
> 3. Aventail Connect instructs the outbound server to establish a connection with the Aventail ExtraNet Server on the specified port. The user authenticates with the Aventail ExtraNet Server, if necessary.
>
> 4. Aventail Connect instructs the Aventail ExtraNet Server to proxy its connection to the final destination.
>
> 5. Once the connection between the client and the Aventail ExtraNet Server is established, the outbound server simply relays the data."

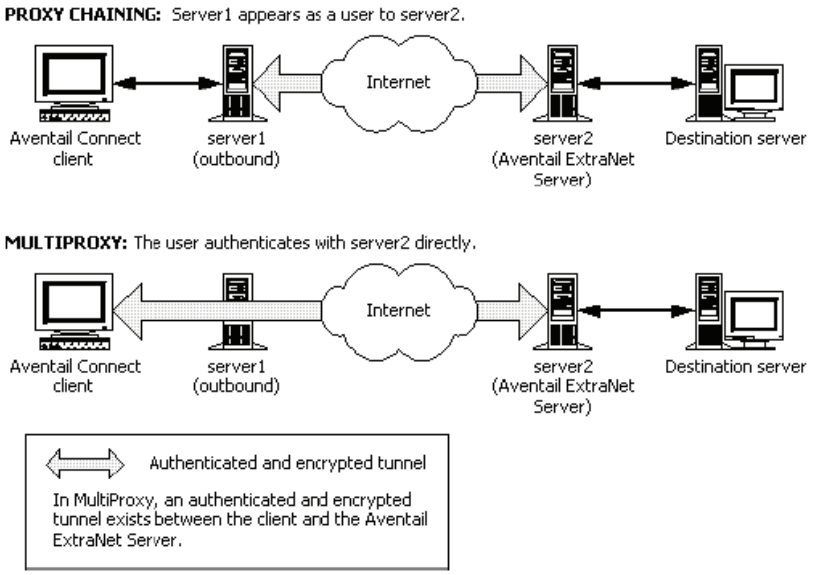
79. Also, as explained on page 60 of Exhibit G, the proxy server computer (i.e., the Aventail Extranet Server) "acts both as a destination and a server. It is a destination because a proxy server routes traffic to it. It is a server because it routes traffic to the final destination."



80. As explained on page 61 of Exhibit G, Aventail Connect could be configured to specify one or more intermediary proxies through which communications could be routed. After defining the final destination and the extranet server locations, one or more intermediary proxy servers could be added (a "destination"). All or some of the traffic from the client could be routed through each destination, based on how the client computer running Aventail Connect was configured.

81. As explained on page 64 of Exhibit G, another routing scheme used by client computers running Aventail Connect was called "proxy chaining." In this scheme, the client computer running Aventail Connect would be configured to send traffic to a specified intermediary proxy server. That server would then forward the traffic on to one or more subsequent proxy servers. The Aventail Connect client computer would authenticate the connection to the first intermediary. After that, the intermediary servers would authenticate to the next proxy server, and so on.

82. Exhibit G, at pages 63 to 64, a comparison is provided of the authentication and access rule variables for the "MultiProxy" and "Proxy Chaining" routing schemes:



83. Client computers running Aventail Connect v3.01/2.51 could dynamically browse and access resources within a private network once a secure connection had been established. As explained on pages 90 to 101 of Exhibit G, Aventail Connect v3.01/2.51 included a feature called "Secure Extranet Explorer (SEE)," which was described as follows:

> Secure Extranet Explorer (SEE) allows you to view your Extranet Neighborhood, which is accessed through the Extranet Neighborhood icon on your desktop. The Extranet Neighborhood user interface resembles that of Network Neighborhood. However, while Network Neighborhood displays all computers on your local network, Extranet Neighborhood allows you to browse, copy, move, and delete files from remote computers via the Aventail Connect extranet connection. With Extranet

> Neighborhood, all interaction with the remote server can be secured. Network administrators determine which local and remote computers are available to users.

84. This functionality in Aventail Connect v3.01/2.51 allowed a remote user who had successfully established a VPN to a private network to see and access all of the network resources which that user was authorized to access. The remote user would be equivalent to a local user in that user's ability to see and/or access network resource, such as a secure website on the private network.

85. The SEE capability was implemented in Aventail Connect v3.01/2.51 via a Windows Explorer shell extension ("Extranet Neighborhood") that enabled a Windows client to visually navigate resources on a private network to which the client has established a VPN connection. As explained on page 91 of Exhibit G:

> Extranet Neighborhood, a Windows Explorer shell extension, is a collection of Windows file servers and Windows NT domains. Network Neighborhood displays only those remote computers that the network administrator has specified. SEE requires a hosts file (SEEHosts) that determines which Windows file servers and NT domains are available. You can include a SEEHosts file with the Aventail Customizer tool. If users install a custom package that does not include a SEEHosts file, then the first time they open Extranet Neighborhood, SEE will create a SEEHosts file. For more information, see the "Customizer" section in the Administrator's Guide.
>
> Extranet Neighborhood offers Aventail Connect users a secure alternative to traditional file-browsing methods. Users can securely access computers from the desktop through Extranet Neighborhood (see icon below), or through Windows Explorer.
>
> Generally, you will use Extranet Neighborhood to connect to a remote network through Aventail Connect. For example, you will use Extranet Neighborhood when:
>
> - you are inside the office, on the corporate network, and you connect through an Aventail ExtraNet Server to your company's remote site, or to another company's network.
>
> - you are outside the office, and you connect your laptop through an Aventail ExtraNet Server to your internal company network, or to another company's network.

86. As explained on page 91 of Exhibit G, the Extranet Neighborhood functionality in Aventail Connect was implemented by redirecting Windows communications in NetBIOS (NBT) to WinSock. This allowed Aventail Connect to handle those communications in the same way it handles other DNS requests and TCP/IP traffic. See Exhibit G at page 91 ("To deliver a secured version of standard Windows browsing,

Aventail Connect redirects NBT calls to WinSock.") This also enabled a client computer running Aventail Connect to view and access a "dynamic list of available Windows hosts" (e.g., secure sites within the private network). Thus, a client computer running Aventail Connect via the Extranet Neighborhood feature would be able to see the same resources (subject to administrator defined limitations) that other users on the private network would see.

D. Discussion of the Aventail Connect v3.1/2.6 Administrator's Guide

87. Aventail Connect v3.1/2.6 was very similar in its functionality to Aventail Connect v3.01/2.51. Like the earlier product, Aventail Connect v3.1/2.6 worked by automatically authenticating and encrypting communications between a client computer running Aventail Connect and a private network resource via a VPN server (the Aventail Extranet Server v3.1). How Aventail Connect did this is described on page 7 of Exhibit J:

> Aventail Connect is the client component of the Aventail ExtraNet Center. Aventail Connect works with the Aventail ExtraNet Server, the SOCKS 5 server component of the Aventail ExtraNet Center. You can use Aventail Connect as a simple proxy client for managed outbound access, and for secure inbound access.
>
> Aventail Connect automates the "socksification" of Transmission Control Protocol/Internet Protocol (TCP/IP) client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol. When you run Aventail Connect on your system, it automatically routes appropriate network traffic from a WinSock (windows sockets) application to an extranet (SOCKS) server, or through successive servers. (WinSock is a Windows component that connects a Windows PC to the Internet using TCP/IP.) The SOCKS server then sends the traffic to the Internet or the external network. Network administrators can define a set of rules that route this traffic.
>
> Aventail Connect is designed to run transparently on each workstation, without adding overhead to the user's desktop. In most cases, users will interact with Aventail Connect only when it prompts them to enter authentication credentials for a connection to a secure extranet (SOCKS) server. Users may also occasionally need to start and exit Aventail Connect, although network administrators often configure it to run automatically at startup. Aventail Connect does not require administrators to manually establish an encrypted tunnel; Aventail Connect can establish an encrypted tunnel automatically. (emphasis added)

88. As explained on Pages 7 and 77 to 79 of Exhibit J, a client computer running Aventail Connect v3.1/2.6 (i) operated transparently to the user and the client computer, (ii) would automatically authenticate a user attempting to access a secure location, (iii) would automatically encrypt communications between a client computer and the secure network

destination, and (iv) that network administrators could route the TCP/IP traffic between the client computer and the secure network destination.

89. Aventail Connect v3.1/2.6 worked with applications that communicate via TCP/IP, and was implemented using the WinSock functionality in client computers running Windows. See, e.g., Exhibit J at page 8 ("Windows TCP/IP networking applications (such as telnet, e-mail, Web browsers and ftp) use WinSock to gain access to networks or the Internet.") Among other things, this meant that Aventail Connect v3.1/2.6 would act on DNS requests, which could contain either hostnames or IP address. As explained on page 8 of Exhibit J, an application on the client computer, via WinSock, "goes through the following steps to connect to a remote host on the Internet or corporate network:

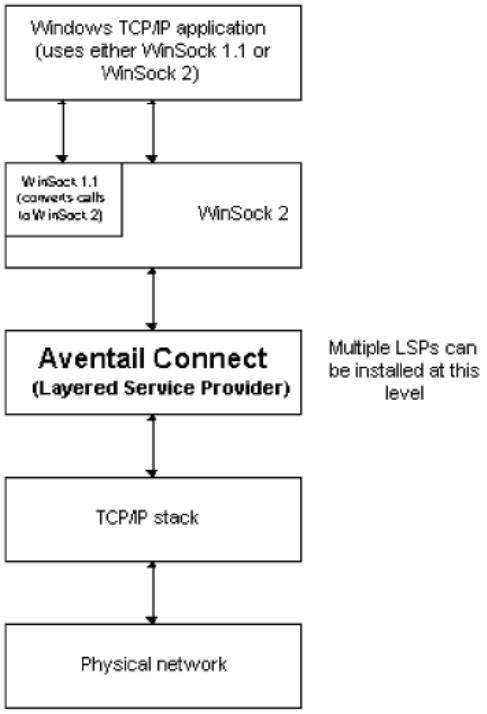
1. The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address or, in rare cases, it will do a reverse DNS lookup to convert the IP address into a hostname. If the application already knows the IP address, this step is skipped.

2. The application requests a connection to the specified remote host. This causes the underlying stack to begin the TCP handshake, when two computers initiate communication with each other. When the handshake is complete, the application is notified that the connection is established, and data can then be transmitted and received

3. The application sends and receives data."

90. Pages 9 to 10 of Exhibit G explain how Aventail Connect v3.1/2.6 functioned within the TCP/IP handling procedures of the client computer:

> **WHAT DOES AVENTAIL CONNECT DO?**
>
> Aventail Connect slips in between WinSock and the underlying TCP/IP stack. (See diagram below.) As an application that sits between WinSock and the TCP/IP stack, Aventail Connect 3.1 is a Layered Service Provider (LSP). Aventail Connect can change data (compressing it or encrypting it, for example) before routing it to the TCP/IP stack for transport over the network. The routing is determined by the rules described in the configuration file.



91. As explained on pages 10 of Exhibit J, a client computer running Aventail Connect v3.1 would evaluate DNS requests to determine if the request was seeking access to a destination requiring authentication and/or encrypted communication or a non-secure destination (e.g., a website on the Internet):

> When the Aventail Connect LSP receives a connection request, it determines whether or not the connection needs to be redirected (to an Aventail ExtraNet Server) and/or encrypted (in SSL). When redirection and encryption are not necessary, Aventail Connect simply passes the connection request, and any subsequent transmitted data, to the TCP/IP stack.

92. Thus, if a client computer running Aventail Connect v3.1/2.6 received a DNS request specifying a destination did not match a redirection rule and thus did not require a VPN (e.g., a non-secure website on the Internet), it would simply pass that DNS request on to the TCP/IP stack of the client computer for handling. In that scenario, the client computer handled the DNS request as if Aventail Connect were not running on the client computer.

93. However, if the client computer running Aventail Connect v3.1/2.6 determined that a DNS request contained a hostname that matched a redirection rule requiring a VPN (e.g., a secure website inside a private network), it would automatically handle authentication of the user to the private network and encrypt the communications between the client computer and the private network. The authentication and encryption steps were transparent both to the client computer and the user. See Exhibit J at page 7 ("Aventail Connect is designed to run transparently on each workstation, without adding overhead to

the user's desktop. In most cases, users will interact with Aventail Connect only when it prompts them to enter authentication credentials for a connection to a secure extranet (SOCKS) server.")

94. How Aventail Connect v3.1/2.6 did this is explained on pages 11 to 12 of Exhibit J. First, Aventail Connect v3.1/2.6 would determine if the DNS request contained a hostname requiring resolution (e.g., "securenet.com"). If so, Aventail Connect would do the following to resolve the hostname depending on how Aventail was configured:

1. The application does a DNS lookup to convert the hostname to an IP address or, in rare cases, it will do a reverse DNS lookup to convert the IP address to a hostname. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following:

    - If the hostname matches a local domain string or does not match a redirection rule, Aventail Connect passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack performs the lookup as if Aventail Connect were not running.

    - If the destination hostname matches a redirection rule domain name (i.e., the host is part of a domain we are proxying traffic to) then Aventail Connect creates a false DNS entry (HOSTENT) that it can recognize during the connection request. Aventail Connect will forward the hostname to the extranet (SOCKS) server in step 2 and the SOCKS server performs the hostname resolution.

    - If the DNS proxy option is enabled and the domain cannot be looked up directly, Aventail Connect creates a false DNS entry that it can recognize later, and returns this to the calling application. The false entry tells Aventail Connect that the DNS lookup must be proxied, and that it must send the fully qualified hostname to the SOCKS server with the SOCKS connection request

95. Thus, like Aventail Connect v3.01/2.51 and AutoSOCKS v2.1, Aventail Connect v3.1/2.6 would either would flag all DNS requests containing hostnames specifying non-local destinations (i.e., "if the DNS proxy option is enabled") or only those DNS requests with hostnames matching a redirection rule. Hostnames matching a redirection rule were destinations that required a VPN (i.e., authentication and encryption). Also, like the AutoSOCKS product, In particular, the client computer would establish communication with the Aventail VPN Server (Aventail Extranet Server) and, after being authenticated, would send the hostname from the initial DNS request to the Aventail VPN Server (the "SOCKS server"). The Aventail VPN server would resolve the received hostname and then determine whether it was necessary to establish a VPN.

96. After DNS resolution or in cases where a DNS request specified a "real" IP address, Aventail Connect v3.1/2.6 would handle connection requests as described in step "2" on page 12 to 13 of Exhibit J. This section explains that after the TCP/IP handshake was

completed (i.e., "by the underlying stack" on the client computer), the application on the client computer would be notified that the TCP/IP connection had been established and that data could be transmitted and received. At this point, Aventail Connect v3.1/2.6 would evaluate the connection request, and do one of the following:

a. Aventail Connect checks the connection request.

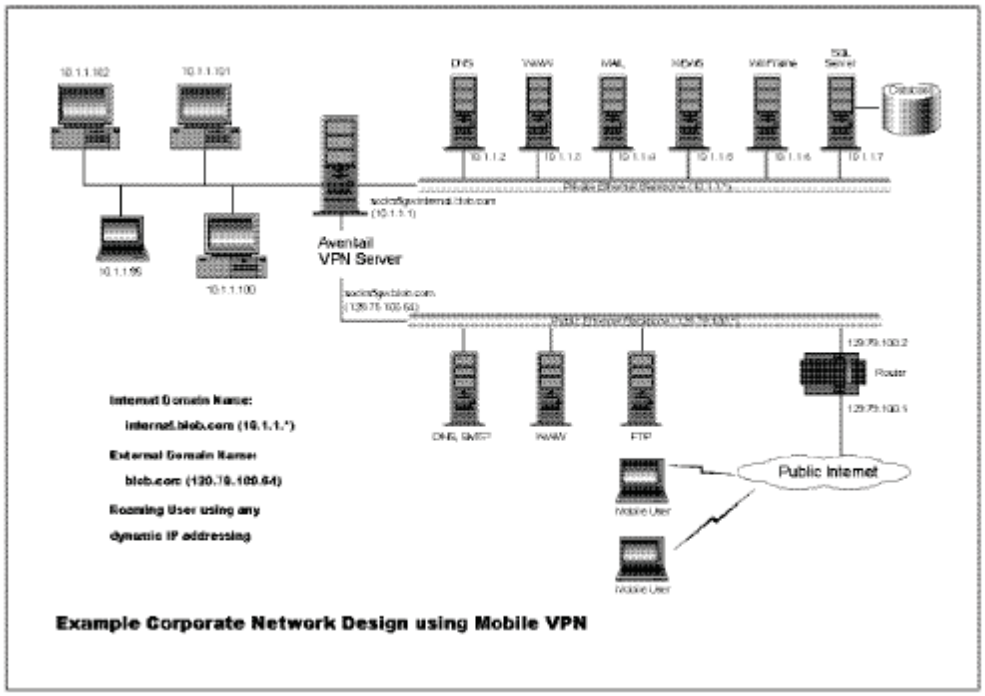
- If the request contains a false DNS entry (from step 1), it will be proxied.
- If the request contains a routable IP address, and the rules in the configuration file say it must be proxied, Aventail Connect will call WinSock to begin the TCP handshake with the server designated in the configuration file.
- If the request contains a real IP address and the configuration file rule says that it does not need to be proxied, the request will be passed to WinSock and processing jumps to step 3 as if Aventail Connect were not running.

97. Pages 11 to 13 of Exhibit J explain that if Aventail Connect v3.1/2.6 determined that a DNS request specified a destination that required authentication and/or encrypted communications (e.g., a secure website on a private network), it would cause the client computer to communicate with the "proxy" server (i.e., a computer running the Aventail Extranet Server v3.1). The client computer and the server computer would then do the following as explained on pages 12-13 of Exhibit J:

- It [i.e., the client computer running Aventail Connect v3.1/2.6] sends the list of authentication methods enabled in the configuration file.
- Once the server selects an authentication method, Aventail Connect executes the specified authentication processing.
- It then sends the proxy request to the extranet (SOCKS) server. This includes either the IP address provided by the application or the DNS entry (hostname) provided in step 1.

98. Under this sequence of steps, the client computer running Aventail Connect v3.1/2.6 would not send the original hostname in the DNS request to the DNS proxy server (Aventail Extranet Server) for resolution before the client had been successfully authenticated.

99. Pages 76 to 79 of Exhibit J describe a VPN implemented using Aventail Connect v3.1/2.6 software running on client computers (called "mobile users") and the Aventail Extranet Server v3.1 software running on a separate computer that sits between the private network and the public Internet and regulates access to the private network (a "gateway" computer). A figure describing this VPN is shown on page 77 of Exhibit J, and is reproduced below:



100. Pages 77 to 78 of Exhibit J describe implementation of a VPN using the Aventail Connect v3.1/2.6 software running on client computers in conjunction with an Aventail VPN Server (i.e., Aventail Extranet Server):

> The design used in the example above consists of two individual Ethernet segments, one public and one private. The public segment is used to host anonymous services available to the general public. The public access is provided through a router that is connected to the public Internet. The private segment is used to house all of the corporation's private network resources and data to be used only by internal company employees. The Aventail ExtraNet Server depicted in this example is used to provide secure and monitored access to the private LAN for mobile employees and partners. For security reasons the Aventail ExtraNet Server is configured such that operating system routing is disabled. Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the Aventail ExtraNet Server.
>
> The mobile user workstations connected to the public Internet are the client workstations, onto which, Aventail Connect will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail ExtraNet Server unless they are running Aventail Connect. Depending on the security policy and the Aventail ExtraNet Server configuration, Aventail Connect will automatically proxy their allowed application traffic into the private network. In this situation, Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then,

> based on the security policy, the Aventail ExtraNet Server will proxy mobile user traffic into the private network but only to those resources allowed. The client workstations we focus on in this section are Microsoft Windows based PCs.
>
> User authentication and encryption on the Aventail ExtraNet Server require all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s). For this example, the Aventail ExtraNet Server encrypts all sessions with SSL.
>
> Installing and using Aventail Connect for remote access purposes differs a bit from its installation and use within a corporate network. First, configuration files need to be kept locally on the workstation or laptop. This is due to the inability to share a file server that allows direct access outside the perimeter of the private network. Second, not all traffic passes through to the Aventail ExtraNet Server. Only traffic destined for the internal network is authenticated and encrypted; all other traffic passes through Aventail Connect unchanged. For instance, browsing the Internet from the mobile user workstation occurs as if Aventail Connect is not even running in the background. Large sites with many mobile users will want to set up an internal file server for a network installation for all mobile users to easily install and configure Aventail Connect. For more information, refer to "Network Installation." (emphasis added)

101. As explained above, “extranets” are functionally identical to VPNs – an “extranet” is simply a VPN that has been established between a non-employee’s client computer and the private network. The same procedures and steps are followed regardless of whether a client connection being managed by Aventail Connect was connecting to a “VPN” or an “extranet.”

102. Client computers running Aventail Connect v3.1/2.6 could be configured to use a number of different authentication techniques. See Exhibit J, at pages 29 to 30; 33 to 34; 42 to 61. In any of these schemes, authentication must be succeed before Aventail Connect v3.1/2.6 and the Aventail Extranet Server v3.1 would permit establishment of a secure connection and transmission of data. See, e.g., page 46 of Exhibit G, which explains:

> SOCKS v5 servers often require user authentication before allowing access. Aventail Connect authentication modules display dialog boxes that prompt users to enter username and password information as well as other authentication credentials.

103. If authentication failed, an error would be returned to the client computer running Aventail Connect v3.1/2.6, and depending on the configuration of the client, an error notification would be provided to the user. For example, an Aventail Connect client computer configured to use SSL for authentication and encryption would display to the user a server certificate it determined to be suspect, or would both display the certificate

and reject a connection based on this authentication failure. See, e.g., Exhibit J at page 53.

104. Communications between a client computer running Aventail Connect v3.1/2.6 and the server computer (e.g., the gateway computer running the Aventail Extranet Server v3.1) could be automatically encrypted. As explained on page 13 of Exhibit J:

> If an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server on behalf of the application. If data is being returned, Aventail Connect decrypts it so that the application sees cleartext data.

105. On pages 68 to 75, Exhibit J explains that client computers running Aventail Connect v3.1/2.6 could be configured to route TCP/IP communications between the client and server computers through intermediary destinations according to different routing schemes.

106. One routing scheme described in Exhibit J was called "Aventail Multiproxy." This technique is generally described on page 68 of Exhibit J as follows:

> The Aventail MultiProxy feature allows Aventail Connect to traverse multiple firewalls by making connections through successive proxy servers. Aventail Connect makes a connection with each proxy server individually. Each proxy server forms a link in a chain that connects Aventail Connect to the final destination. Any or all of the proxy servers can apply authentication and access control rules. Proxies can be Aventail ExtraNet Servers, other SOCKS 5 servers, SOCKS 4 servers, or HTTP proxies.
>
> Using an HTTP proxy server to control outbound traffic eliminates the need to install a separate SOCKS server. This HTTP proxy can filter outbound connection requests and route those requests to the specified servers. MultiProxy supports RFC 2068 HTTP Basic (username/password) authentication. If your proxy uses HTTP Basic (username/password) authentication, Aventail Connect will store the username and password information in the credential cache, as it does with SOCKS servers.

107. As explained on page 69 of Exhibit J, the client computer running Aventail Connect manages the routing of these communications, and handles authentication, encryption and access parameters to each of the intermediary proxy servers, which could be SOCKS servers or HTTP proxy servers. In particular, Exhibit J explains:

> The steps for making a connection using MultiProxy are:
>
> 1. The client application requests access to the destination server.
>
> 2. Aventail Connect establishes a connection with the outbound server (SOCKS server or HTTP proxy). Aventail Connect then sends the

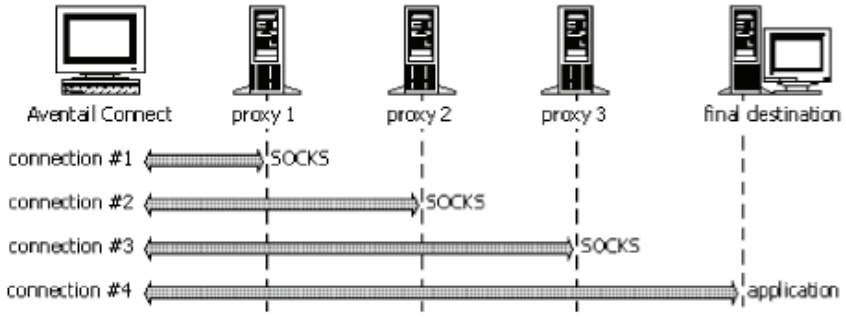
access request to the outbound server, specifying the Aventail ExtraNet Server as the destination. The user authenticates with the outbound server, if necessary.

3. Aventail Connect instructs the outbound server to establish a connection with the Aventail ExtraNet Server on the specified port. The user authenticates with the Aventail ExtraNet Server, if necessary.

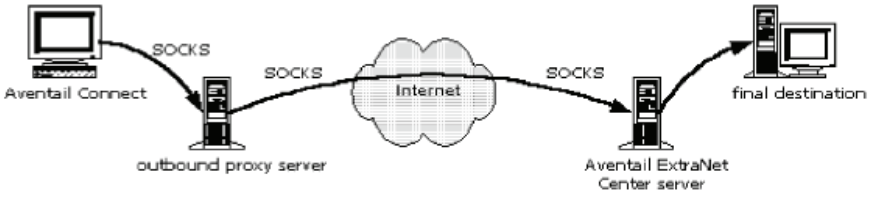
4. Aventail Connect instructs the Aventail ExtraNet Server to proxy its connection to the final destination.

5. Once the connection between the client and the Aventail ExtraNet Server is established, the outbound server simply relays the data.

The following example illustrates the connections made during a MultiProxy connection through three proxy servers



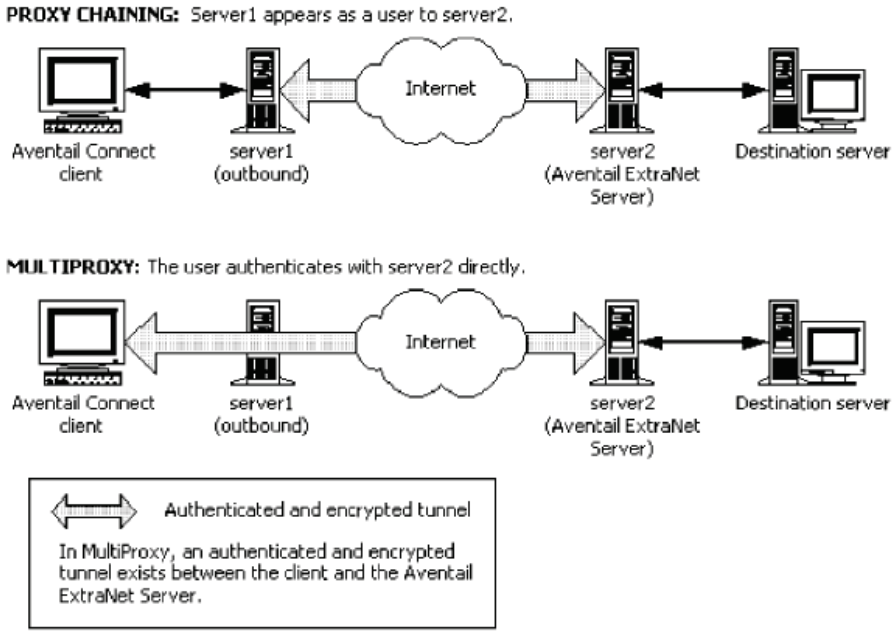
108. Also, as explained on page 69 of Exhibit J, the proxy server computer (i.e., the Aventail Extranet Server v3.1) "acts both as a destination and a server. It is a destination because a proxy server routes traffic to it. It is a server because it routes traffic to the final destination."



109. As explained on page 70 of Exhibit J, Aventail Connect v3.1/2.6 could be configured to specify one or more intermediary proxies through which communications could be routed. After defining the final destination and the extranet server locations, one or more intermediary proxy servers could be added (a "destination"). All or some of the traffic from the client could be routed through each destination, based on how the client computer running Aventail Connect was configured.

110. As explained on pages 72 to 73 of Exhibit J, another routing scheme used by client computers running Aventail Connect v3.1/2.6 was called "proxy chaining." In this scheme, the client computer running Aventail Connect would be configured to send traffic to a specified intermediary proxy server. That server would then forward the traffic on to one or more subsequent proxy servers. The Aventail Connect client computer would authenticate the connection to the first intermediary. After that, the intermediary servers would authenticate to the next proxy server, and so on.

111. On pages 72 and 73 of Exhibit J, a comparison is provided of the authentication and access rule variables for the "MultiProxy" and "Proxy Chaining" routing schemes:



112. Client computers running Aventail Connect v3. 1/2.6 could dynamically browse and access resources within a private network once a secure connection had been established. As explained on pages 95 to 106 of Exhibit J, Aventail Connect v3.1/2.6 included a feature called "Secure Extranet Explorer (SEE)," which was described as follows:

> Secure Extranet Explorer (SEE) allows you to view your Extranet Neighborhood, which is accessed through the Extranet Neighborhood icon on your desktop. The Extranet Neighborhood user interface resembles that of Network Neighborhood. However, while Network Neighborhood displays all computers on your local network, Extranet Neighborhood allows you to browse, copy, move, and delete files from remote computers via the Aventail Connect extranet connection. With Extranet Neighborhood, all interaction with the remote server can be
>
> secured. Network administrators determine which local and remote computers are available to users.

113. This functionality in Aventail Connect v3.1/2.6 allowed a remote user who had successfully established a VPN to a private network to see and access all of the network resources that user was authorized to access. The remote user would be equivalent to a local user in that user's ability to see and/or access network resource, such as a secure website on the private network.

114. The SEE functionality was implemented in Aventail Connect v3.1/2.6 via a Windows Explorer shell extension ("Extranet Neighborhood") that enabled a Windows client to visually navigate resources on a private network to which the client has established a VPN connection. As explained on page 95 of Exhibit J:

> Extranet Neighborhood, a Windows Explorer shell extension, is a collection of Windows file servers and Windows NT domains. Network Neighborhood displays only those remote computers that the network administrator has specified. SEE requires a hosts file (SEEHosts) that determines which Windows file servers and NT domains are available. You can include a SEEHosts file with the Aventail Customizer tool. If users install a custom package that does not include a SEEHosts file, then the first time they open Extranet Neighborhood, SEE will create a SEEHosts file. For more information, see the "Customizer" section in the Administrator's Guide.
>
> Extranet Neighborhood offers Aventail Connect users a secure alternative to traditional file-browsing methods. Users can securely access computers from the desktop through Extranet Neighborhood (see icon below), or through Windows Explorer.
>
> Generally, you will use Extranet Neighborhood to connect to a remote network through Aventail Connect. For example, you will use Extranet Neighborhood when:
>
> - you are inside the office, on the corporate network, and you connect through an Aventail ExtraNet Server to your company's remote site, or to another company's network.
> - you are outside the office, and you connect your laptop through an Aventail ExtraNet Server to your internal company network, or to another company's network.

115. As explained on page 96 of Exhibit J, the Extranet Neighborhood functionality in Aventail Connect was implemented by redirecting Windows communications in NetBIOS (NBT) to WinSock. This allowed Aventail Connect to handle those communications in the same way it handles other DNS requests and TCP/IP traffic. See Exhibit J at page 91 ("To deliver a secured version of standard Windows browsing, Aventail Connect redirects NBT calls to WinSock.") This also enabled a client computer running Aventail Connect to view and access a "dynamic list of available Windows hosts" (e.g., secure sites within the private network). Thus, a client computer running

Aventail Connect via the Extranet Neighborhood feature would be able to see the same resources (subject to administrator defined limitations) that other users on the private network would see.

F. Comparison of Claims 1 to 12 of the '135 Patent to Aventail AutoSOCKS, Aventail Connect v3.01/2.51 and Aventail Connect v3.1/2.6

1. Requirements of the Claims of the '135 Patent

116. In the 1998 to 2000 time frame, the phrase "virtual private network" or "VPN" did not have a single or uniform definition. Instead, people used "VPN" to refer to a group of networking protocols and techniques that enabled a remote user to securely gain access to one or more resources available on a private network via a public network, such as the Internet.

117. I understand that a court has already interpreted several of the phrases used in the '135 patent claims. See Exhibit K (Claim Construction Ruling). These include:

- "Virtual Private Network (VPN)" was interpreted to mean "a network of computers which privately communicate with each other by encrypting traffic on insecure communication paths between the computers." See Exhibit K, pages 4 to 10

- "Domain Name Service (DNS)" was interpreted to mean "a lookup service that returns an IP address for a requested domain name." See Exhibit K, pages 11-12.

- "Domain Name" was interpreted to mean "a name corresponding to an IP address." See Exhibit K at pages 12-15.

- "web site" was interpreted to mean "one or more related web pages at a location on the World Wide Web." See Exhibit K at pages 15-17.

- "secure web site" was interpreted to mean "a web site that requires authorization for access and that can communicate in a VPN." See Exhibit K at pages 18-19.

- "automatically initiating the VPN" was interpreted to mean "initiating the VPN without involvement of a user." See Exhibit K at pages 21-22.

- "DNS proxy server" was interpreted to mean "a computer or program that responds to a domain name inquiry in place of a DNS." See Exhibit K at pages 22-24.

118. The Court's interpretations are generally consistent with my understanding of the meaning of these phrases during the period 1998 to 2000.

119. In my review of claims 1-12 of the '135 patent, I saw no requirement that users be provided unrestricted access to all the resources on the private network that are made available to local users. Similarly, none of these claims require that a remote user be

able to communicate directly with a local user (i.e., not via an intermediary computer). Neither would have been surprising in the 1997-2000 time frame to someone working in the field of network security. For example, a basic principle of network design followed then and now is that only specific, defined network resources should be made available to users, regardless of whether the user was remote or local.

120. In addition, in the 1997 to 2000 time frame, most network services were provided by and thus regulated by servers (e.g., mail servers, printer servers, file servers). There were relatively few network services at that time which allowed communications to occur directly between users within a network.

121. The Court's definition of a VPN does not add any additional requirements to those in the claims because the Court's definition uses a general and simple definition of a VPN. Under that definition, a remote user's computer establishes a VPN simply by communicating with one other computer on the private network using encrypted communications over an insecure communication path (e.g., the Internet) between the computers.

2. Comparison of the Claims of the '135 Patent to Aventail AutoSOCKS, Aventail Connect v3.01/2.51 and Aventail v3.1/2.6

122. I reviewed the '135 patent. I also evaluated the claims of the '135 patent and compared the requirements of the claims to the Aventail VPN solutions that are described in Exhibits B, G and J.

123. As I explained above, the Aventail VPN solutions include client and server components. In my comments below, when I refer to "Aventail clients" I am referring to Aventail AutoSOCKS v2.1 (described in Exhibit B), Aventail Connect v3.01/2.51(described in Exhibit G) and Aventail Connect v3.1/2.6 (described in Exhibit J). When I refer to "Aventail VPN Servers" I am referring to the Aventail VPN Servers that work with the three corresponding Aventail clients; namely, Aventail MobileVPN and PartnerVPN are paired with Aventail AutoSOCKS v2.1 (Exhibit B), Aventail Extranet Server v3.0 is paired with Aventail Connect v3.01/2.51 (Exhibit G), and Aventail Extranet Server v3.1 is paired with Aventail Connect v3.1/2.6 (Exhibit J). Also, when I refer to an Aventail VPN solution, I am referring to one of the paired Aventail client and VPN server products.

124. Claim 1 of the '135 patent specifies a method for transparently creating a virtual private network (VPN) between a client computer and a target computer. The process has three identified steps.

125. Exhibits B, G and J show that all three Aventail VPN solutions operated transparently in setting up VPNs. See ¶¶ 24-25 (Aventail AutoSOCKS); ¶¶ 55-56, 62 (Aventail Connect v3.01); ¶¶ 87-88, 93 (Aventail Connect v3.1).

126. The first step of claim 1 is generating from the client computer a DNS request that requests an IP address corresponding to a domain name associated with the target

computer. This is done, for example, by a user entering in the domain name of a target computer, or the IP (Internet Protocol) address of that target computer, in a web browser.

127. Exhibits B, G and J show that all three Aventail clients evaluate and act upon DNS requests made by applications running on client computers on which the Aventail client software has been installed. See ¶¶ 24-30 (Aventail AutoSOCKS); ¶¶ 57, 59-62 (Aventail Connect v3.01); ¶¶ 89, 91-93 (Aventail Connect v3.1).

128. The second step of claim 1 is "determining whether the DNS request transmitted in step (1) is requesting access to a secure web site." This determination can be made either at the client computer or by another computer, such as a proxy server. See Exhibit K at pages 19-21.

129. Exhibits B, G and J show that client computers running Aventail clients determine whether a DNS request made by an application on the client computer "is requesting access to a secure web site." The Aventail products do this by evaluating the hostname or IP address in the connection request to see if matches a redirection rule, which indicates that the destination requires a VPN. A hostname or IP address that matches a redirection rule in the Aventail VPN solution is a "secure web site" because these destinations will cause the Aventail VPN server to establish a VPN between that destination and the client computer making the connection request. See ¶¶ 34-39, 42-47 (Aventail AutoSOCKS); ¶¶ 62-66, 69-72 (Aventail Connect v3.01); ¶¶ 88, 94-97, 99-102 (Aventail Connect v3.1).

130. The third step of claim 1 is, in response to a determination that the DNS request is requesting access to a secure target web site, to automatically initiate the VPN between the client computer and the target computer. None of the claims require that this step be performed by the client computer or by another computer, so it can be done on either.

131. Each of Exhibits B, G and J show a VPN being automatically established between a client computer running an Aventail client and a secure destination computer. In particular, the client computer running the Aventail client automatically performs the authentication of the client with the VPN Server. If that authentication is successful, the Aventail VPN Server then establishes the VPN automatically with the destination specified in the DNS request. The VPN transports encrypted network traffic between the client and destination over the Internet. The Aventail client automatically encrypts outgoing traffic and decrypts incoming traffic from the secure destination. See ¶¶ 30-32, 43-44, 47-49, 51-52, 54-55 (Aventail AutoSOCKS): ¶¶ 56-57, 65-67, 69-72, 74-75 (Aventail Connect v3.01); ¶¶ 88-89, 94, 97-98, 100-105 (Aventail Connect v3.1).

132. Claim 2 indicates that steps (2) and (3) of claim 1 are performed at a DNS server separate from the client computer.

133. Exhibits B, G and J each show that client computers running Aventail clients can be configured to proxy all non-local DNS requests containing hostnames to the VPN server for resolution. This information from the VPN server is then used to determine if the destination requires a VPN (e.g., because it matches a redirection rule). Exhibits B, G

and J also show that for each Aventail VPN solution, the server component establishes the VPN between the client and the destination after it authenticates the user. See ¶¶ 30-32, 39-40, 43-44, 47-49 (Aventail AutoSOCKS); ¶¶ 56-57, 64-71 (Aventail Connect v3.01); ¶¶ 88-89, 95-96, 98-102 (Aventail Connect v3.1).

134. Claim 3 adds an additional step; namely, that if the DNS request in step (2) is determined to not be requesting access to a secure target web site, the IP address for the domain name is resolved and returned to the client computer.

135. Exhibits B, G and J show that each Aventail client will pass a DNS request containing a hostname or IP address that does not match a redirection rule (e.g., a website that is not a secure target website) to the operating system of the client computer for DNS resolution. See ¶¶ 33-34, 42, 46 (Aventail AutoSOCKS); ¶¶ 62, 64, 66 (Aventail Connect v3.01); ¶¶ 92-93, 95, 97 (Aventail Connect v3.1).

136. Claim 4 says that the process will, prior to automatically initiating the VPN between the client and target computers, determine if the client computer is authorized to establish a VPN with the target computer, and if not, an error should be returned from the DNS request. The claim does not indicate what type of error must be provided (e.g., whether it must be a DNS error or some other type of error).

137. Exhibits B, G and J show that each Aventail client will attempt to authenticate a user with an Aventail VPN server if the client computer receives a DNS request specifying a destination matching a redirection rule (i.e., a destination requiring a VPN). In step 2.b of the process each of these Aventail clients follows, the client establishes a connection to a pre-designated Aventail VPN server and begins a SOCKS negotiation. See ¶¶ 34-39, 42-47 (Aventail AutoSOCKS); ¶¶ 62-66, 69-72 (Aventail Connect v3.01); ¶¶ 88, 94-97, 99-102 (Aventail Connect v3.1).

138. When the Aventail clients and VPN servers make a connection and perform authentication, they follow the sequence of steps dictated by the SOCKS v5 protocol:

> When a TCP-based client wishes to establish a connection to an object that is reachable only via a firewall (such determination is left up to the implementation), it must open a TCP connection to the appropriate SOCKS port on the SOCKS server system. The SOCKS service is conventionally located on TCP port 1080. If the connection request succeeds, the client enters a negotiation for the authentication method to be used, authenticates with the chosen method, then sends a relay request. The SOCKS server evaluates the request, and either establishes the appropriate connection or denies it.
>
> See Exhibit L (RFC 1928, Socks Protocol Version 5, March 1996, available at http://tools.ietf.org/html/rfc1928)):

139. The Aventail client performing its SOCKS v5-compliant transactions will be returned an error from the Aventail VPN server if the authentication step is unsuccessful. The form of the response to the client from the server is governed by § 6 of RFC 1928:

> The SOCKS request information is sent by the client as soon as it has established a connection to the SOCKS server, and completed the authentication negotiations. The server evaluates the request, and returns a reply formed as follows:
>
> ```
> +----+-----+-------+------+----------+----------+
> |VER | REP |  RSV  | ATYP | BND.ADDR | BND.PORT |
> +----+-----+-------+------+----------+----------+
> | 1  |  1  | X'00' |  1   | Variable |    2     |
> +----+-----+-------+------+----------+----------+
> ```

140. In the Socks v5 protocol, the value of the REP field informs the SOCKS client (i.e., the Aventail client) whether authentication was successful. If not, an error value is returned in this field to the Aventail client (e.g., "X'02' connection not allowed by ruleset" or "X'05' Connection refused"). See § 6 of RFC 1928.

141. In addition, I note that all of the Aventail VPN solutions are implemented in TCP/IP communications. As such, these solutions would inherently know how to handle errors returned according to the relevant DNS and TCP/IP communication protocols.

142. One such protocol is the DNS Standard IETF RFC 1035, which was ratified as Standard 13 in November, 1987. See Exhibit M (RFC 1035, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION," Section 4.1.1. Header section format (November 1987)(available at http://www.rfc-archive.org/getrfc.php?rfc=1035&tag=Domain-Names---Implementation-and-Specification). RFC 1035 describes DNS query formats and response codes, called RCODE. There are two RCODEs that are of interest. Code 3 is used to indicate that the requested host name does not exist. This is typically referred to as "host not found." The other response code, Code 5, occurs when a DNS server refuses to provide a response due to a policy restriction. For example, a policy restriction would be that clients outside the internal network should not be able to resolve certain names.

143. Claim 4 does not indicate if a DNS response code must be returned or what part of the system (e.g., client, server, destination) must return the response. If claim 4 requires a DNS response code to be returned, that type of response, and how it would be returned, would have been dictated by the DNS standards that govern transmission of policy decisions within DNS.

144. I note, for example, under the procedures outlined in RFC 1035, a successful DNS name resolution results in an address record that contains the IP address of the requested hostname. A DNS response that contains an error does not contain an associated IP address record. So, if a client computer required an IP address to come back from the DNS query to take additional steps, it would not be able take those steps if a DNS query

returned an error. This is because an IP address record (i.e., a value in the DNS record specifying an IP address) is returned only if there is a successful resolution of the address.

145. A person of ordinary skill in the field of VPN technologies in 1999 would have considered it to be an obvious design choice when implementing a VPN using the Aventail VPN products to employ standard DNS error reporting codes to notify a client computer requesting access to a secure destination that it was not authorized to access that destination. This is because the DNS response codes serve the role of informing a requesting computer that a policy required to maintain the communication had not been satisfied by the requesting client. In addition, Code 5 was specifically designed to communicate to a requesting computer that it was not authorized to communicate with the destination computer.

146. Claim 5 says that the process will, prior to automatically initiating the VPN between the client and target computers, determine if the client computer is authorized to resolve addresses of non-secure target computers. If it is not, then the process returns an error from the DNS request.

147. Exhibits B, G and J each show that the Aventail clients must successfully authenticate a client computer with the Aventail VPN server before the server may resolve a hostname forwarded by the client computer for resolution. This occurs in step 2.b. of the process each of these clients follow in handling the connection request. If authentication is unsuccessful, an error is returned to the Aventail client by the Aventail VPN server in each of the VPNs described in Exhibits B, G and J. See ¶¶ 34-39, 42-47 (Aventail AutoSOCKS); ¶¶ 62-66, 69-72 (Aventail Connect v3.01); ¶¶ 88, 94-97, 99-102 (Aventail Connect v3.1).

148. Claim 6 says that the process will establish a VPN by creating an "IP address hopping" scheme between the client computer and the target computer. The claim does not identify any requirements of the IP hopping scheme, so any type of IP routing protocol used by the client and server computers would meet this requirement.

149. The TCP/IP protocol inherently creates an "IP address hopping" scheme to route IP traffic from a client to a destination computer. This is because the protocol is designed to forward IP packets to a succession of intermediate destinations before it reaches it final destination.

150. In addition, Exhibits G and J show that the Aventail Connect clients describe two different IP address hopping schemes that routes IP traffic between a client computer and a destination computer through intermediate destinations; namely, Proxy Chaining and the Multi-Proxy scheme. See ¶¶ 76-83 and ¶¶ 106-112, respectively.

151. Under the Aventail Connect Proxy Chaining scheme, each Aventail Connect Server in the chain is configured with destination rules that define the destination server in the chain. A new connection is made at each hop of the Proxy chain, effectively hiding the source and destination IP address of the client connection. The Aventail Proxy Chaining scheme is transparent to the client computer. The users passwords are cached on the first

proxy and re-used on subsequent authentication attempts. See ¶¶ 82-83 (Aventail Connect v3.01); ¶¶ 111-112 (Aventail Connect v3.1).

152. Under the Aventail Connect MultiProxy scheme, a client connection may be forwarded to different types of proxies (HTTP, SOCKS 4, SOCKS 5, or an Aventail Extranet Server). At each proxy server, the source and destination IP addresses are changed, effectively hiding the true source and destination IP addresses. Multi-Proxy is transparent to the client computer. The users passwords are cached on the first proxy and re-used on subsequent authentication attempts. See ¶¶ 77-81 (Aventail Connect v3.01); ¶¶ 107-110 (Aventail Connect v3.1).

153. Claim 7 says that the third step of the process is to be implemented using a gatekeeper computer that allocates VPN resources for communicating between the client computer and the target computer. A VPN server, for example, that is separate from the client computer and which regulates or manages VPNs established between client computers and target computers, access to which it is controlling, would meet this requirement.

154. Exhibits B, G and J each show VPNs in which an Aventail VPN Server computer allocates VPN resources for communicating between the client computer and the target computer. In these examples, the Aventail VPN server authenticates users, and handles the establishment and maintenance of the VPN between the client computer and the destination computer. See ¶¶ 30-32, 47-49 (Aventail AutoSOCKS); ¶¶ 56-57, 65-68, 70-72 (Aventail Connect v3.01); ¶¶ 88-89, 96-98, 100-103 (Aventail Connect v3.1). The role played by the Aventail VPN Server in performing these functions is to allocate VPN resources according to claim 7.

155. Claim 8 says that the step of determining if a DNS request is requesting access to a secure web site is to be performed in a DNS proxy server that passes through the request to a DNS server if access to a secure target web site is not being requested. So, for example, a program running on the client computer that determines that a particular DNS request is specifying a non-secure target computer and then simply passes the resolved IP address to the normal TCP/IP handling routines on that computer would meet this requirement. I note that there is no requirement in this claim that the DNS proxy server be running on a different computer than the client computer.

156. Exhibits B, G and J show that each of the Aventail clients is a proxy server running on the client computer. Each Aventail clients captures DNS requests to Internet hosts, and if there is no redirection rule directing the client to proxy the traffic to a VPN via an Aventail VPN Server (e.g., MobileVPN/PartnerVPN or Extranet Server), the DNS request is passed to the underlying operating system of the client computer (i.e., Winsock and the TCP/IP stack in the Windows OS) for resolution. See ¶¶ 33-34, 42, 46 (Aventail AutoSOCKS); ¶¶ 62, 64, 66 (Aventail Connect v3.01); ¶¶ 92-93, 95, 97 (Aventail Connect v3.1).

157. Claim 9 says that the third step of claim 1 is to be performed by "transmitting a message to the client computer to determine whether the client computer is authorized to establish the VPN target computer." A process that sends an authentication prompt or other type

of authentication verification message to the client computer during the process of establishing a VPN would satisfy this requirement.

158. Exhibits B, G and J show that each of the Aventail clients will be sent messages by the Aventail VPN server during the process of authenticating the client computer, and that the Aventail clients and VPN servers could be configured to use one of several different authentication methods. See ¶¶ 47-48; (Aventail AutoSOCKS); ¶¶ 67-68 (Aventail Connect v3.01); ¶¶ 98-99 (Aventail Connect v3.1).

159. Exhibits B, G and J also include example sequences where the Aventail client authenticates with the Aventail VPN server. In one example, the client is prompted to enter credentials during the authentication process. ¶¶ 51-52 (Aventail AutoSOCKS); ¶ 72 (Aventail Connect v3.01); ¶ 103 (Aventail Connect v3.1). In another, the client computer is shown information about suspect server certificates when authentication is being done using certificates. ¶¶ 53 (Aventail AutoSOCKS); ¶ 73 (Aventail Connect v3.01); ¶ 104 (Aventail Connect v3.1). Each of these examples shows that the Aventail VPN server would transmit a message to the client computer to determine whether the client computer would be authorized to establish a VPN with a target computer.

160. I also note that the DNS system, which includes the DNS server and the DNS client on the host computer, already inherently supports notification of policy violations through the RCODE mechanism. On a system where there may not be specific client software running on the client computer, using DNS response codes would be an obvious choice to send notifications because the client DNS program would already know what to with a RCODE 5 message and would be able to notify the application of the response code. A client computer running Aventail client software connecting to an Aventail VPN Server, regardless of whether the client or server was performing DNS resolution, would be inherently capable of processing an RCODE 5 message and ending further attempts at connecting to the target server.

161. Claim 10 defines a system that transparently creates a VPN between a client computer and a secure target computer. One part of this system is a DNS proxy server that receives a request from the client computer to look up an IP address for a domain name. If it is determined that access is being requested to a non-secure website, the DNS server returns the IP address for the requested domain. If it is determined that the access is being requested to a secure website, then the DNS proxy server returns a request to create the VPN between the client and target computers. The second part of the system identified in the claim is a gatekeeper computer that allocates resources for the VPN between the client and the secure web computer in response to the request by the DNS proxy server. Claim 10 allows the DNS proxy server and the gateway computer to be the same or different computers, and places no restrictions on how the DNS proxy server and gateway computer interact to allocate resources for the VPN between the client and target computers.

162. Exhibits B, G and J show systems that transparently create a VPN between a client and a secure target computer. The systems use Aventail clients, which can function as proxy servers that resolve DNS requests containing hostnames. The Aventail clients determine

if access is being requested to a non-secure website, in which case the Aventail client will pass the DNS request to the operating system of the client computer for normal handling under the TCP/IP procedures of the operating system (e.g., Winsock on Windows computers). The Aventail clients also each are able to generate a request to the Aventail VPN server to establish a VPN if the client determines that that the DNS request specifies a destination requiring a VPN (e.g., because the hostname or IP address specified in the DNS request matches a redirection rule). In that case, the client computer running the Aventail client will establish a connection to a designated Aventail VPN server and attempt to authenticate the client computer. If authentication is successful, the Aventail VPN server establishes the VPN. See ¶¶ 24, 30-35, 41-49, 51-53 (Aventail AutoSOCKS); ¶¶ 56-57, 61-75 (Aventail Connect v3.01); ¶¶ 88-89, 93-105 (Aventail Connect v3.1).

163. In each of the VPNs described in Exhibits B, G and J, network traffic between the client computer and destination is encrypted, and that the Aventail client is able to automatically encrypt/decrypt the network traffic. See ¶¶ 51, 54-55 (Aventail AutoSOCKS); ¶¶ 70, 74 (Aventail Connect v3.01); ¶¶ 101, 105 (Aventail Connect v3.1).

164. The VPN systems described in Exhibits B, G and J also include an Aventail VPN server which runs on a different computer than the client computer on which the Aventail client is running. See ¶ 37 (Aventail AutoSOCKS); ¶ 69 (Aventail Connect v3.01); ¶ 100 (Aventail Connect v3.1).

165. When the Aventail Extranet Server receives a DNS response that resolves the server name with an IP address, it establishes a TCP connection with the next hop in the connection—either another proxy or the destination server—and allocates server resources such as memory, disk, network connectivity to be used for the connection. This is how network servers operate normally and is not specific to VPN gateways.

166. Claim 11 says that the gateway computer element of the system in claim 10 creates the VPN by establishing an IP hopping regime that is used to pseudorandomly change IP addresses in packets transmitted between the client computer and the secure target computer.

167. IP hopping that using pseudorandom IP addresses was well known as far back as 1996. See Exhibit N (Reed et al., "Proxies for Anonymous Routing," presented at the 12th Annual Computer Security Applications Conference, Dec 9-13, 1996 (available at http://www.onion-router.net/Publications/ACSAC-1996.pdf); Exhibit O (Goldschlag et al., "Hiding Routing Information," Information Hiding, R. Anderson (editor), Springer-Verlag LLNCS 1174, 1996 (available at http://www.onion-router.net/Publications/IH-1996.pdf).

168. Exhibit N identifies the problem of routing secure traffic over a public network; namely, that "using traffic analysis, it is possible to infer who is talking to whom over a public network." See Exhibit N at page 1. Exhibit N then explain a scheme called "onion routing" which was devised to solve this problem.

169. Exhibit N explains that onion routing is one type of pseudorandom IP hopping method specified in Claim 11. For example, in section 1.3, Overview of the Solution, Exhibit N states:

> An application, instead of making a (socket) connection directly to a destination machine, makes a connection to an onion routing proxy on some remote machine. That onion routing proxy builds an anonymous connection through several other onion routers to the destination.

170. Exhibit N also explains that onion-routing "is designed to interface with a wide variety of unmodified Internet services by means of proxies" and that onion-routing has been implemented in a wide variety of systems that use proxy servers, including those for World Wide Web browsing (HTTP), remote logins (RLOGIN), e-mail (SMTP), and file transfers (FTP). The Aventail VPN solution, as explained above, is implemented using VPN servers that proxy HTTP traffic onto private networks.

171. Similarly, in Exhibit O at section 3.2 (titled "Loose Routing"), a method for pseudorandomly selecting onion routers is described. As explained on page 2:

> It is not necessary that the entire route be pre-specified by the initiator's proxy. He can instruct various nodes along the route to choose their own route to the next pre-specified node. This can be useful for security, adding more hops to the chain."

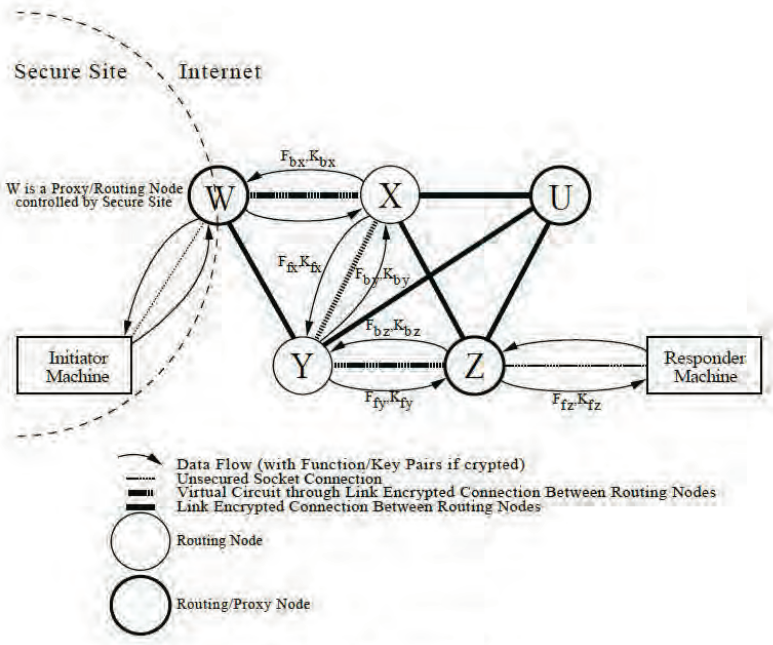


Fig. 3. A Virtual Circuit.

172. The figure above from Exhibit O shows an onion network containing 5 onion routers, W, X, Y, Z, and U. The path through the onion network is selected by each onion router

from a selection of available neighbors. Since the number of onion routers is finite, the path through the network is pseudorandom chosen at each hop by that particular onion router.

173. Claim 11 does not require that the client gateway, or any other router perform the pseudorandom route selection, so a scenario where the onion router selects the next hop satisfies that claim requirement.

174. I believe a person experienced with VPN design in the 1997-2000 time frame and wishing to ensure a higher degree of security for a VPN would have used a VPN design that would make it harder to determine the path used to carry VPN traffic between the client and destination computers. I note that each of Exhibits B, G and J points out that "monitoring of network usage" (i.e., network traffic analysis or monitoring) is an important concern to address in network security design. The authors of Exhibits N and O point to the same problem. The obvious solution that a person would have seen at that time for preventing traffic analysis would have been to implement an onion-routing scheme in the VPN. The onion-routing scheme was developed for this precise purpose. In addition, Exhibit N and Exhibit O show that the onion-routing scheme, by 1997, had been successfully implemented within proxy-based HTTP solutions. All of the Aventail VPN solutions were implemented through proxy servers. Given the known suitability of the onion-routing solution to the Aventail VPN solutions, I believe a person of ordinary skill in the art would have considered it obvious to modify the VPN designs reflected in Exhibits B, G and J by incorporating an onion-routing scheme that is being described in Exhibits N or O. That combination would create a system that meets the requirements of claim 11.

175. Claim 12 says that the gatekeeper computer element determines if the client computer has sufficient security privileges to create the VPN, and, if it lacks sufficient security privileges, it rejects the request to create the VPN.

176. Each of Exhibits B, G and J show that the Aventail VPN servers perform authentication of the client computer seeking to establish a VPN with a destination computer requiring a VPN. This is done in step 2.b of the process. The VPN server in each case will evaluate the credentials presented by the client computer according to the authentication method specified by the Aventail VPN Server. If the VPN server determines that the client computer making the request does not have sufficient security privileges, it will reject the request the request to create the VPN. See ¶¶ 30-32, 47-53, 55 (Aventail AutoSOCKS); ¶¶, 67-68, 72-73 (Aventail Connect v3.01); ¶¶ 94, 98, 100, 103-105 (Aventail Connect v3.1).

177. In the Aventail VPN solutions, administrators could specify different services and privileges for different users. Those policies would be used during the authentication process to determine whether to establish the VPN in response to the connection request made by a client computer running an Aventail client. See ¶¶ 30-32 (Aventail AutoSOCKS); ¶¶, 67-68 (Aventail Connect v3.01); ¶¶ 100-101 (Aventail Connect v3.1).

178. Claim 13 of the '135 patent defines a method for establishing communications between one of a several client computers and a central computer that maintains authentication tables that correspond to one of the client computers. The process has four identified steps.

179. Exhibits B, G and J show that all three Aventail VPN solutions establish communication between client computers and a central computer that maintains authentication tables that correspond to each client computer. See ¶¶ 49-54 (Aventail AutoSOCKS); ¶¶ 62-66 (Aventail Connect v3.01); ¶¶ 94-102 (Aventail Connect v3.1).

180. The first step of claim 13 says that the central computer receives from one of the client computers a request to establish a connection. This is done, for example, by a user entering in the domain name of a target computer, or the IP (Internet Protocol) address of that target computer, in a web browser.

181. Exhibits B, G and J each show that the Aventail clients evaluate and act upon DNS requests made by applications running on client computers on which the Aventail client software has been installed. See ¶¶ 24-30 (Aventail AutoSOCKS); ¶¶ 57, 59-62 (Aventail Connect v3.01); ¶¶ 89, 91-93 (Aventail Connect v3.1).

182. The second step of claim 13 says that the request is authenticated as being from an authorized client by referring to one of the authentication tables maintained by the central computer.

183. Exhibits B, G and J show that all of the Aventail clients must successfully authenticate a client computer with the Aventail VPN server before the requested communication may proceed. The Aventail VPN Server maintains authentication modules, including username and password and Secure Socket Layer credentials, among others, in order to facilitate the communications process. See ¶¶ 49-54 (Aventail AutoSOCKS); ¶¶ 62-66 (Aventail Connect v3.01); ¶¶ 94-102 (Aventail Connect v3.1).

184. The third step of claim 13 says that resources are allocated to establish a virtual private link between the client and a second computer in response to a determination that the request is from an authorized client.

185. Each of Exhibits B, G and J show a VPN being automatically established between a client computer running an Aventail client and a secure destination computer. In particular, after the VPN Server authenticates the client computer, it establishes the VPN automatically with the destination specified in the DNS request. The VPN transports encrypted network traffic between the client and destination over the Internet. The Aventail client automatically encrypts outgoing traffic and decrypts incoming traffic from the secure destination. See ¶¶ 30-32, 43-44, 47-49, 51-52, 54-55 (Aventail AutoSOCKS): ¶¶ 56-57, 65-67, 69-72, 74-75 (Aventail Connect v3.01); ¶¶ 88-89, 94, 97-98, 100-105 (Aventail Connect v3.1).

186. The fourth step of claim 13 says that the communications then occur between the authorized client and the second computer using the virtual private link.

187. Exhibits B, G and J each describe that the VPN transports encrypted network traffic between the client and a second computer over the Internet. The Aventail client automatically encrypts outgoing traffic and decrypts incoming traffic from the secure destination. See ¶¶ 30-32, 43-44, 47-49, 51-52, 54-55 (Aventail AutoSOCKS): ¶¶ 56-57, 65-67, 69-72, 74-75 (Aventail Connect v3.01); ¶¶ 88-89, 94, 97-98, 100-105 (Aventail Connect v3.1).

188. Claim 14 of the '135 patent says that step 4 of claim 13 includes the step of communicating according to a scheme by which at least one field in a series of data packets is periodically changed according to a known sequence.

189. As explained above in the context of claim 11, the concept of IP hopping using pseudorandom IP addresses was well known as far back as 1996. See Exhibit N (Reed et al., "Proxies for Anonymous Routing," presented at the 12th Annual Computer Security Applications Conference, Dec 9-13, 1996 (available at http://www.onion-router.net/Publications/ACSAC-1996.pdf); Exhibit O (Goldschlag et al., "Hiding Routing Information," Information Hiding, R. Anderson (editor), Springer-Verlag LLNCS 1174, 1996 (available at http://www.onion-router.net/Publications/IH-1996.pdf).

190. The onion routing schemes described in Exhibits N and O address the problem of routing secure traffic over a public network; namely, that "using traffic analysis, it is possible to infer who is talking to whom over a public network." *See, e.g.,* Exhibit M at page 1.

191. Exhibit N shows that onion routing is a type of pseudorandom IP hopping scheme and is implemented by periodically changing a data field in a series of data packets in the way that is specified in Claim 14. For example, in section 1.3, Overview of the Solution, Exhibit N states:

> An application, instead of making a (socket) connection directly to a destination machine, makes a connection to an *onion routing proxy* on some remote machine. That onion routing proxy builds an anonymous connection through several other *onion routers* to the destination. . . . Data passed along the anonymous connection appears different *at* and *to* each onion router, so data cannot be tracked en route and compromised onion routers cannot cooperate.

192. Exhibit N also explains that onion-routing "is designed to interface with a wide variety of unmodified Internet services by means of proxies" and that onion-routing has been implemented in a wide variety of systems that use proxy servers, including those for World Wide Web browsing (HTTP), remote logins (RLOGIN), e-mail (SMTP), and file transfers (FTP). The Aventail VPN solution, as explained several times above, is implemented using VPN servers that proxy HTTP traffic onto private networks. See Exhibit N at 1 (abstract).

193. Claim 15 of the '135 patent says that step 4 of claim 13 includes the step of comparing an IP address in a header of each data packet to a table of valid IP addresses maintained in a table in the second computer.

194. As described above, the Aventail VPN solutions would inherently be able to handle standard TCP/IP protocols for handshaking, routing and transmission that were defined in TCP/IP standards. A second computer in the Aventail VPN solution would therefore routinely verify that the IP address assigned to the original computer does not change when compared to the known valid IP address as maintained by the second computer's TCP/IP stack and coincides with the duration of the persistent communication.

195. Claim 17 of the '135 patent indicates that step 2 of the claim 13 comprises the step of using a checkpoint data structure that maintains synchronization of a periodically changing parameter known by the central computer and the client computer to authenticate the client.

196. The technique of maintaining synchronization of a periodically changing parameter known by the central computer and the client computer would have been inherent in all three Aventail VPN solutions in view of the security and cryptographic methods known at the time, including systems that electronically generated variable, non-predictable time-based codes that were synchronized on separate devices, and the validation and comparison of such codes for the purpose of authenticating a user of a system involving such separate devices. See Exhibit P (U.S. Patent No. 4,885,778 to Weiss, entitled "Method and Apparatus for Synchronizing Generation of Separate, Free Running, Time Dependent Equipment.")

197. Claim 18 of the '135 patent describes a method of transparently creating a virtual private network between a client computer and target computer.

198. Exhibits B, G and J show that all three Aventail VPN solutions operated transparently in setting up VPNs. See ¶¶ 24-25 (Aventail AutoSOCKS); ¶¶ 55-56, 62 (Aventail Connect v3.01); ¶¶ 87-88, 93 (Aventail Connect v3.1).

199. The first step of claim 18 says that a DNS request that requests an IP address corresponding to a domain name associated with the target computer is generated on the client computer. This is done, for example, by a user entering in the domain name of a target computer, or the IP (Internet Protocol) address of that target computer, in a web browser.

200. Exhibits B, G and J show that all three Aventail clients evaluate and act upon DNS requests made by applications running on client computers on which the Aventail client software has been installed. See ¶¶ 24-30 (Aventail AutoSOCKS); ¶¶ 57, 59-62 (Aventail Connect v3.01); ¶¶ 89, 91-93 (Aventail Connect v3.1).

201. The second step of claim 18 is "determining whether the DNS request transmitted in step (1) is requesting access to a secure web site." This determination can be made either at the client computer or by another computer, such as a proxy server. See Exhibit K at pages 19-21.

202. Exhibits B, G and J show that client computers running Aventail clients determine whether a DNS request made by an application on the client computer "is requesting access to a secure web site." The Aventail products do this by evaluating the hostname

or IP address in the connection request to see if matches a redirection rule, which indicates that the destination requires a VPN. A hostname or IP address that matches a redirection rule in the Aventail VPN solution is a "secure web site" because these destinations will cause the Aventail VPN server to establish a VPN between that destination and the client computer making the connection request. See ¶¶ 34-39, 42-47 (Aventail AutoSOCKS); ¶¶ 62-66, 69-72 (Aventail Connect v3.01); ¶¶ 88, 94-97, 99-102 (Aventail Connect v3.1).

203. The third step of claim 18 says that, in response to a determination that the DNS request is requesting access to a secure target web site, a VPN is automatically initiated between the client computer and the target computer. None of the claims require that this step be performed by the client computer or by another computer, so it can be done on either.

204. Each of Exhibits B, G and J show a VPN being automatically established between a client computer running an Aventail client and a secure destination computer. In particular, the client computer running the Aventail client automatically performs the authentication of the client with the VPN Server. If that authentication is successful, the Aventail VPN Server then establishes the VPN automatically with the destination specified in the DNS request. The VPN transports encrypted network traffic between the client and destination over the Internet. The Aventail client automatically encrypts outgoing traffic and decrypts incoming traffic from the secure destination. See ¶¶ 30-32, 43-44, 47-49, 51-52, 54-55 (Aventail AutoSOCKS): ¶¶ 56-57, 65-67, 69-72, 74-75 (Aventail Connect v3.01); ¶¶ 88-89, 94, 97-98, 100-105 (Aventail Connect v3.1).

205. The third step of claim 18 includes two additional requirements. The first additional requirement is that that steps (2) and (3) of claim 18 are performed at a DNS server separate from the client computer.

206. Exhibits B, G and J each show that client computers running Aventail clients can be configured to proxy all non-local DNS requests containing hostnames to the VPN server for resolution. This information from the VPN server is then used to determine if the destination requires a VPN (e.g., because it matches a redirection rule). Exhibits B, G and J also show that for each Aventail VPN solution, the server component establishes the VPN between the client and the destination after it authenticates the user. See ¶¶ 30-32, 39-40, 43-44, 47-49 (Aventail AutoSOCKS); ¶¶ 56-57, 64-71 (Aventail Connect v3.01); ¶¶ 88-89, 95-96, 98-102 (Aventail Connect v3.1).

207. The second additional requirement for the third step of claim 18 is that, prior to automatically initiating the VPN between the client and target computers, determining if the client computer is authorized to resolve addresses of non-secure target computers. If it is not, then the process returns an error from the DNS request.

208. Exhibits B, G and J each show that the Aventail clients must successfully authenticate a client computer with the Aventail VPN server before the server may resolve a hostname

forwarded by the client computer for resolution when the Aventail Connect client is configured to proxy all non-local DNS requests. This occurs in step 2.b. of the process each of the Aventail clients follow in handling the connection request. If authentication is unsuccessful, an error is returned to the Aventail client by the Aventail VPN server in each of the VPNs described in Exhibits B, G and J. See ¶¶ 34-39, 42-47 (Aventail AutoSOCKS); ¶¶ 62-66, 69-72 (Aventail Connect v3.01); ¶¶ 88, 94-97, 99-102 (Aventail Connect v3.1).

209. In view of these observations, I believe that the processes and systems described in Exhibits B, G and J meet every requirement of claims 1 to 12 of the '135 patent, alone or in conjunction with the additional publications that I refer to above.

* * * * * * * * * * *

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the patent subject to this reexamination proceeding.

Michael A. Fratto

7/7/11

Date

List of Exhibits

Exhibit	*Description*
A	Curriculum Vitae of Michael Allyn Fratto
B	Aventail AutoSOCKS v2.1 Administration & User's Guide
C	PR Newswire, "Aventail Ships the First Standards-Based Virtual Private Network Software Solution" (May 2, 1997)
D	Infoworld Review of AutoSOCKS, Vol. 19, Issue 25 (June 23, 1997) at page 70
E	Fratto, "Aventail VPN 2.5: Not Your Father's Socks," Network Computing, Vol. 8, No. 18 (October 1, 1997)
F	Fratto, "Footlose and Fancy Free with Three SOCKS 5-based proxy servers," Network Computing, Vol. 9, Issue 11 (June 15, 1998)
G	Aventail Connect v3.01/2.51 Administrator's Guide
H	PR Newswire, "Aventail Ships Directory-enabled Extranet Solution; Aventail Extranet Center V3.1 Available At www.aventail.com." (August 9, 1999)
I	"Intranet Applications: Briefs," Network World, at page 55 (October 19, 1998)
J	Aventail Connect v3.1/v2.6 Administrator's Guide
K	Aventail Ships Directory-Enabled Extranet Solution; Aventail Extranet Center V3.1 Available at www.aventail.com (Business Wire, August 9, 1999)
L	RFC 1928, Socks Protocol Version 5, March 1996, available at http://tools.ietf.org/html/rfc1928
M	RFC 1035, Domain Names-Implementation and Specification
N	Reed et al., "Proxies for Anonymous Routing," presented at the 12th Annual Computer Security Applications Conference, Dec 9-13, 1996 (available at http://www.onion-router.net/Publications/ACSAC-1996.pdf);
O	Goldschlag et al., "Hiding Routing Information," Information Hiding, R. Anderson (editor), Springer-Verlag LLNCS 1174, 1996 (available at http://www.onion-router.net/Publications/IH-1996.pdf
P	U.S. Patent No. 4,885,778 to Weiss

EXHIBIT A

TO MICHAEL FRATTO'S DECLARATION

CURRICULUM VITAE

Mike Fratto
105 Marion Ave
Syracuse, NY 13219
315-567-9866
mfratto@gmail.com

Employment History

August 2009 to present, Editor, Network Computing
I manage the daily editorial budget for the Network Computing site and digital editions. I also research and write about networking, cloud computing, and data center trends and products. I also guide freelancers in coverage of more technical content.

January 1st, 2008 to August, 2009, Managing Editor, Labs, InformationWeek.
In addition to researching and writing articles and industry analysis, I coordinate product testing among staff and freelance editors. I manage the lab in CENT and provide guidance to student lab assistants who work in CENT and assist in lab management.

April, 2006 to December, 2008, Senior Technology Editor and Lab Manager, Network Computing
I managed coverage for network security, network infrastructure, and WAN optimization beat areas. That included indentifying new trends, developing stories, and managing freelancers. I also managed a $250,000 budget for lab equipment.

July 2004 to April, 2006 Editor, Secure Enterprise Magazine
Secure Enterprise was a new magazine spun off from Network Computing. I identified new trends, developed the editorial calendar, managed the content creation, and collaborated with other CMP publications on content creation. I also managed a staff reporter located in Syracuse as well as freelancer writers.

June 1997 to June 2004, Senior Technology Editor, Network Computing
I started out the as an Associate Technology Editor and rose to the Senior Technology Editor position in two years. I covered network security, then an emerging field within IT.

April, 1994 to June, 1997, Freelance editor with Network Computing
I primarily covered remote access and telecommunications while attending Syracuse University.

1987 to 1992, Consultant
I was an independent consultant focused on remote office automation. I wrote programs that would connect to remote offices, gather data from DOS programs, and then prepared that data for input into other computer programs.

Education

Bachelor of Science, Syracuse University, 2001
Introduction to Cisco Routers, 1997

Courses

IST 634: Security in Networked Environments

Through directed projects, lectures, research, and some lab work, students learn practical aspects of network security and how to evaluate security technology based on fundamental principles. I have adapted the course over the years by adjusting the technical detail to meet the students at their level and allowing students to pursue projects relevant to their studies.

IST 500: Security+

In the study group I mentor students through the material with the goal of achieving the Security+ certification. I designed the course to create a group where students teach and learn from each other by presenting presentations, developing study resources, and sample questions. Where needed, I provide guidance for further study or insights into the topic.

IST 423: Introduction to Information Security

Through lectures and readings, students were introduced to concepts in information security starting from policies to more technical content. The lectures tied together topics students should have encountered such as networking fundamentals, organizational structure, and applications.

IST 623: Introduction to Information Security

The one semester I taught this course, I focused on the theory supporting information security. By focusing on models and processes rather than technology, students learned to think about processes and entities rather than technologies as solutions.

Web Security Certification course, CBIT

I taught this course for three classes for CBIT as a practical, lab-driven course covering server and web application security. Through lectures provided by the certification body and labs I developed for the course, students received the necessary knowledge to successfully test for the certification.

Books

Vacca, John, R, *Public Key Infrastructure: Building Trusted Applications and Web*, 2004, Chapter 3 “In Pki We Trust?”

Citations

Horak, Ray, *Telecommunications and Data Communications Handbook*, 2007

Cho, Kenjiro, *Technologies for Advanced Heterogeneous Networks: First Asian Internet Engineering Conference, AINTEC 2005, Bangkok, Thailand, December 13-15, 2005, Proceedings (Lecture Notes in Computer Science)*, 2006

Singh, Munindar, *The Practical Handbook of Internet Computing*, 2004

EXHIBIT B

TO MICHAEL FRATTO'S DECLARATION

AVENTAIL AUTOSOCKS V2.1
ADMINISTRATION & USER'S GUIDE

2.1 Aventail AutoSOCKS

ADMINISTRATION & USER'S GUIDE

AVENTAIL

Aventail AutoSOCKS v2.1 *Administration and User's Guide*

117 South Main Street
4th Floor
Seattle, WA 98104-2540
USA

Printed in the United States of America.

Trademarks and Copyrights

Table of Contents

Introduction

Welcome to the AutoSOCKS™ v2.1 secure Windows client for 16- and 32-bit Windows applications. AutoSOCKS v2.1 is the first commercial application to incorporate the SOCKS v5 security protocol standard, simplifying SOCKS deployment for end users and network managers.

AutoSOCKS transparently intercepts WinSock communication requests issued by TCP/IP applications and processes them based upon a set of routing directives (rules) assigned when AutoSOCKS is configured. (For more information about WinSock, TCP/IP, and general network communications, see "Getting Started.")

On larger networks, AutoSOCKS can address multiple SOCKS v5 servers based on end destination and type of service. This feature enables network administrators to effectively monitor and direct network traffic.

Features of AutoSOCKS v2.1:

- Supports both SOCKS v4 and SOCKS v5 standards
- Supports RFC1928 and RFC1929 SOCKS v5 standards
- Network-based setup provides a single configuration point with a simple user interface
- Transparently route connections from Windows applications to external networks through any SOCKS-based firewall system
- Logging utility to troubleshoot problems with network connections
- Enables internal network connections to pass through without interference
- Enables network redirection through multiple SOCKS servers
- Supports multiple authentication methods including SOCKS v4 Identification, username/password, CHAP, and SSL 3.0. Other authentication modules can be added
- Supports 16-bit WinSock 1.1 applications under Windows 3.1 and Windows for Workgroups 3.11
- Supports both 16- and 32-bit applications under Windows 95, Windows NT 3.51, and Windows NT 4.0
- Provides automated installation and uninstallation
- WinSock interoperability tested at Stardust WinSock Labs

About This Document

The AutoSOCKS v2.1 *Administration and User's Guide* provides basic information about AutoSOCKS v2.1. It is designed to include entry-level data for non-technical users as well as more advanced installation, setup, and configuration information for network administrators.

This information is also available via online AutoSOCKS Help and the Aventail web site at **http://www.aventail.com/**.

Document Organization

This document is divided into two primary sections: the Administrator's Guide and the AutoSOCKS *Utilities Reference Guide*. The Administrator's Guide describes procedures for setting up, installing, and configuring AutoSOCKS for individual and multiple networked workstations.

The AutoSOCKS *Utilities Reference Guide* describes the AutoSOCKS system menu commands and utility programs. It contains detailed information about using Ping and Traceroute utilities and documents the authentication/encryption modules and settings.

In addition to the AutoSOCKS v2.1 *Administration and User's Guide* and the AutoSOCKS *Utilities and Reference Guide*, this document includes a removable AutoSOCKS User's Supplement which describes screen displays and features that end-users may encounter while running AutoSOCKS in their client workstations. The document concludes with Appendix 1: Troubleshooting and a Glossary.

Check the Quick Start Card, a short document designed to help you install AutoSOCKS to an individual workstation.

Document Conventions

The following typographic conventions are used in this document. Exceptions may be made for online material; for instance, italics may be difficult to read online.

Convention	Usage
ALL CAPITALS	Filenames and extensions, directory names, keynames, and pathnames.
Bold	Anything the user types, including command-line commands, addresses or URLs, options, and portions of syntax that must be typed exactly as shown. Dialog box controls (**Destination** field), e-mail addresses (**support@aventail.com**), URLs (**http://www.aventail.com/**), and IP addresses (**165.121.6.26**) are also bold.
Italic	Placeholders that represent information the user must insert.
"To Do" Procedures	Underlined *To Do* headings indicate procedures and step-by-step directions. Multi-step procedures are numbered; single-step procedures are bulleted.

Technical Support

If you experience problems installing, configuring, or running AutoSOCKS refer to any of the following:

- The Aventail web site, **http://www.aventail.com/**, for the latest list of known problems.
- The README.TXT documentation for additional information not contained in the manual.

If necessary, report problems to Aventail using the Bug Report form at the Aventail web site.

Aventail Technical Support:

Web site: **http://www.aventail.com/**

E-mail: **support@aventail.com**

Phone: 206.777.5640

Fax: 206.777.5656

About Aventail Corporation

Aventail Corporation is the leading vendor of next-generation Internet security systems. Its software allows organizations to secure their networks, manage their employees' access to the Internet and build Virtual Private Networks (VPNs). Creating a VPN gives organizations the ability to dynamically create a private communication or data channel over the Internet. Aventail's adherence to open security standards simplifies VPN deployment, enables interoperability, and leverages corporations' existing network investments. Its VPN solutions allow companies to extend the reach of their corporate Intranets to customers, partners, remote offices, and worldwide employees.

Aventail Corporation

117 South Main Street

4th Floor

Seattle, WA 98104-2540

Phone: 206.777.5600

Fax: 206.777.5656

http://www.aventail.com/

info@aventail.com

AutoSOCKS v2.1 *Administration and User's Guide*

This section includes procedural and background information on installing AutoSOCKS to both single and networked workstations. It includes:

- Getting Started with brief explanations of network security and communications
- Definitions of SOCKS and AutoSOCKS
- AutoSOCKS platform and installation requirements
- Installing AutoSOCKS, including network diagrams of Aventail VPN, Aventail Internet Policy Manager, and SOCKS v4-based server configurations
- Creating and editing configuration files

Note: Aventail understands the importance of a flexible, easy-to-use installation process. If you have feedback regarding the AutoSOCKS installation procedures, or if there are additional features you wish to see implemented, please e-mail comments to **support@aventail.com**. Your input is appreciated.

Getting Started

If you're new to AutoSOCKS technology, the following section will help you understand what AutoSOCKS is and does, as well as its relationship to network security in general.

Network Security in a Nutshell

Escalating threats of computer viruses and increased potential for unwelcome hackers are forcing companies to seek ways to safeguard their corporate networks and the information they exchange. The first response to these concerns has been the development of security firewalls—software barriers that control the flow of information. But firewalls can't easily be configured to handle complex security issues such as monitoring network usage, providing private communication over public networks, and enabling remote users to gain secure access to internal network resources.

Enter SOCKS v5, an Internet Engineering Task Force (IETF)-approved security protocol targeted at securely traversing corporate firewalls. It was originally developed in 1990, and is now maintained by NEC. SOCKS acts as a circuit-level proxy mechanism that manages the flow and security of data traffic to and from your local area network or intranet. A workstation whose traffic is proxied by SOCKS is considered "socksified." SOCKS is more than a standard security firewall. It also features:

- Client Authentication: (SOCKS v5 only) Authentication allows network managers to provide selected access to internal and external areas of a network.
- Traffic Encryption: (SOCKS v5 only) Encryption ensures that network traffic is private and secure.
- UDP Support: (SOCKS v5 only) User Datagram Protocol (UDP) has traditionally been difficult to proxy with the exception of SOCKS v5.
- Cross-Platform Support: Unlike most UNIX security solutions, SOCKS code can easily be ported to platforms such as Windows NT, Windows 95, and Macintosh systems.

What is AutoSOCKS?

AutoSOCKS automates the "socksification" of client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol. When you run AutoSOCKS on your system, it automatically routes appropriate network traffic from a WinSock application to the SOCKS server. (WinSock is a Windows component that connects a Windows PC to the Internet using Transmission Control Protocol/Internet Protocol—TCP/IP.) The SOCKS server then sends the traffic to the Internet or the external network. Your network administrator defines sets of rules by which this traffic is to be routed.

AutoSOCKS is designed to run transparently on each workstation. In most cases, you'll interact with AutoSOCKS only when it prompts you to enter authentication information for a connection to a secure SOCKS server. You may also occasionally need to start and exit AutoSOCKS, although network administrators often configure it to run automatically at startup.

To understand AutoSOCKS, you first need to understand a few basics of TCP/IP communications.

TCP/IP Communications

Windows TCP/IP networking applications such as e-mail or ftp use WinSock to gain access to the network or the Internet. WinSock (Windows Sockets) is the core component of TCP/IP under Windows. (TCP/IP is a suite of protocols that the Internet uses to provide for services such as e-mail, ftp, and telnet.)

WinSock Connection to A Remote Host

Via WinSock, an application goes through the following steps to connect to a remote host on the Internet or corporate intranet:

1. The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address. If the application already knows the IP address, this step is skipped.
2. The application requests a connection to the specified remote host. This causes the underlying stack to begin the TCP handshake. (The TCP handshake is the process by which two computers initiate communication with each other.) When the handshake is

complete, the application is notified that the connection is established, and that data may now be transmitted and received.

3. The application sends and receives data.

What Does AutoSOCKS Do?

AutoSOCKS slips in between the Windows TCP/IP application and the single access point—WinSock. In simple terms, AutoSOCKS redirects WinSock calls (both parameters and data) and reroutes them through a SOCKS-based server when required. The routing is determined by the rules described in the configuration file created when AutoSOCKS is installed. (See "Configuring AutoSOCKS.")

Because AutoSOCKS intercepts calls to WinSock, AutoSOCKS must duplicate WinSock functionality. Since AutoSOCKS also makes calls directly into WinSock, it must behave as a typical WinSock application as well. (See Figure 1.)

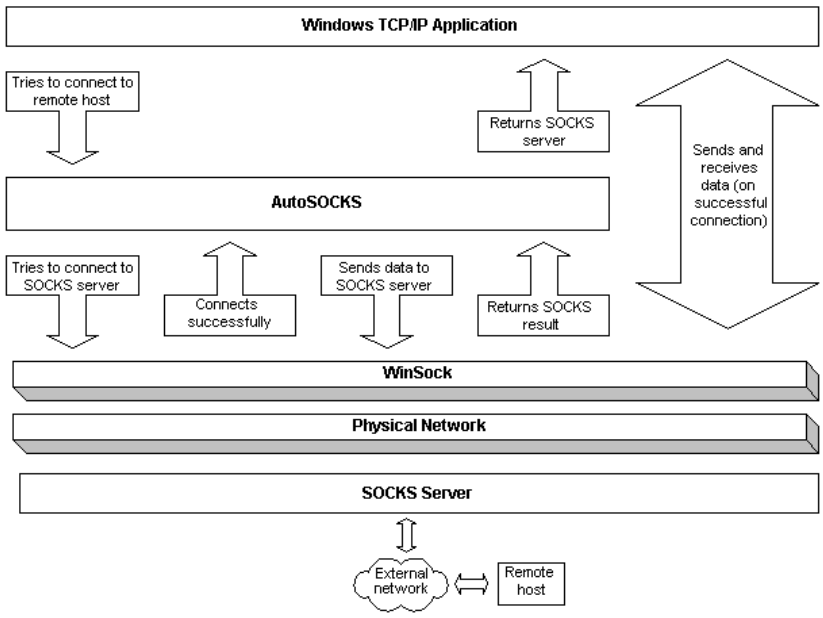


Figure 1. Network application calls intercepted by AutoSOCKS

With AutoSOCKS running, an application executes additional steps in order to connect to a remote host through WinSock. These steps must be transparent to the application so that it cannot differentiate between when AutoSOCKS is running and when it is not. The following three steps are identical to standard WinSock communications steps described above; however, nested inside them are additional actions and options introduced by AutoSOCKS.

1. The application does a DNS lookup to convert the hostname to an IP address. However, if the application already knows the IP address, this entire step is skipped. Otherwise, AutoSOCKS does the following:

    - If the hostname matches a local domain string or does not match a redirection rule, AutoSOCKS passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack then performs the lookup as if AutoSOCKS is not running.
    - If the DNS proxy option is disabled, AutoSOCKS allows the request to go through unchanged.
    - If the destination hostname matches a redirection rule domain name (i.e. the host is part of a domain we are proxying traffic to) then AutoSOCKS creates a false DNS entry (HOSTENT) that it can recognize during the connection request. AutoSOCKS will forward the hostname to the SOCKS server in step 2 and the SOCKS server performs the hostname resolution.
    - If the DNS proxy option is enabled and the domain cannot be looked up directly, AutoSOCKS creates a fake DNS entry that it can recognize later, and returns this to the calling application. The false entry tells AutoSOCKS that the DNS lookup should be proxied, and that it should send the fully qualified hostname to the SOCKS server with the SOCKS connection request.

2. The application requests a connection to the remote host. This causes the underlying stack to begin the TCP handshake. When the handshake is complete, the application is notified that the connection is established and that data may now be transmitted and received. AutoSOCKS does the following:

    a. AutoSOCKS checks the connection request.

        - If the request contains a false DNS entry (from step 1) it will be proxied.
        - If the request contains a real IP address and the rules in the configuration file say it should be proxied, AutoSOCKS calls WinSock to begin the TCP handshake with the server designated in the config file.
        - If the request contains a real IP address and the configuration file rules says that it should *not* be proxied, the request is passed to WinSock and processing jumps to step 3 as if AutoSOCKS is not running.

    b. When the connection is completed, AutoSOCKS begins the SOCKS negotiation.

        - It sends the list of authentication methods enabled in the configuration file.
        - Once the server chooses an authentication method, AutoSOCKS executes the specified authentication processing.
        - It then sends the proxy request to the SOCKS server. This includes either the IP address provided by the application or the DNS entry (hostname) provided in step 1.

c. When the SOCKS negotiation is completed, AutoSOCKS notifies the application. From the application's point of view, the entire SOCKS negotiation including the authentication negotiation, is merely the TCP handshaking.

3. The application transmits and receives data.

If an encryption module is enabled and selected by the SOCKS server, AutoSOCKS encrypts the data on its way to the server on behalf of the application. If data is being returned, AutoSOCKS decrypts it so that the application sees clear text data.

AutoSOCKS Platform Requirements

AutoSOCKS runs under Windows 3.1, Windows for Workgroups 3.11, Windows 95, and Windows NT 3.51 and 4.0. These five platforms can be divided into two groups. Operating requirements and interface features unique to each group are described below.

Windows 95 and Windows NT 4.0

Windows 95 and Windows NT 4.0 have virtually identical interfaces. AutoSOCKS commands are accessed in the Programs list located on the Start menu and from the minimized AutoSOCKS icon on Taskbar tray.

System Requirements

AutoSOCKS system requirements for Windows 95 and Windows NT 4.0 include the following:

- Pentium-based personal computer
- Windows 95 or Windows NT 4.0
- 16 MB application RAM (8 MB on Windows 95)
- 3.5 MB hard disk space
- 16- or 32-bit WinSock-based TCP/IP application(s)
- Network-accessible SOCKS v4 or SOCKS v5 compliant server
- A WinSock compatible TCP/IP stack needs to be installed and configured prior to running AutoSOCKS. This can be the Microsoft-provided TCP/IP stack or a third-party TCP/IP stack.

Interface Features

- The AutoSOCKS program icon can be accessed via the Start menu, Programs option, and Aventail AutoSOCKS menu command.
- When AutoSOCKS is running in the background, the AutoSOCKS icon is visible in the system tray (unless the Hide Icon command is enabled).
- The AutoSOCKS system menu can be displayed by right-clicking the AutoSOCKS icon located in the Taskbar tray.
- AutoSOCKS can be uninstalled via the Start menu by using the **Add/Remove Programs** icon in the Control Panel folder.

Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51

Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51 have similar interfaces. AutoSOCKS commands are accessible from the Aventail AutoSOCKS program group and from the minimized icon's System menu when AutoSOCKS is running.

System Requirements

AutoSOCKS system requirements for Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51 include the following:

- 486-based personal computer
- 4 MB application RAM for Windows 3.1 and Windows for Workgroups 3.11; 16 MB for Windows NT
- 3.5 MB hard disk space
- 16- or 32-bit WinSock-based TCP/IP application(s)
- Network-accessible SOCKS v4 or SOCKS v5 compliant server
- A WinSock compatible TCP/IP stack needs to be installed and configured prior to running AutoSOCKS. This can be the Microsoft-provided TCP/IP stack or a third-party TCP/IP stack.

Interface Features

- The AutoSOCKS program icon is accessed via the AutoSOCKS program group window in Program Manager.
- The AutoSOCKS system menu is displayed by clicking the AutoSOCKS icon located in the AutoSOCKS program group.
- AutoSOCKS can be uninstalled using the Uninstall icon in the AutoSOCKS program group window.
- When AutoSOCKS is running in the background, the AutoSOCKS icon is minimized on the desktop (unless the Hide Icon command is enabled)

Installation Source Media

Regardless of platform, AutoSOCKS can be delivered on CD; in a network-delivered, self-extracting archive file; or on diskette.

This runs SETUP.EXE and installs AutoSOCKS. You can specify an installation directory, or AutoSOCKS will install in the default AutoSOCKS directory.

- **CD**: The CD contains the AutoSOCKS installation program, SETUP.EXE. It also contains in the \DOCS directory the AutoSOCKS v2.1 *Administration and User's Guide* formatted for Acrobat Reader.

- **Network Delivered Source Media**: The network-delivered source media is a self-extracting archive containing the required disk/directory structure within the archive file. The archive filename will be similar to AS21ED.EXE.
- **Diskette Based Source Media.** The diskette based source media is composed of two separate disks (labeled Disk 1 and Disk 2) that contain all of the AutoSOCKS installation files.

Installing AutoSOCKS

AutoSOCKS can be installed to a single workstation or to multiple networked workstations. In either case, you must perform an initial installation of the software and create one or more configuration files. This procedure is described under "Individual Installation." Once the initial installation is complete, you can then install to a series of networked computers using the instructions and information described under “Network Installation.”

Note: Check the Quick Start Card for an easy-to-follow guide to individual workstation installation.

Configuration Files

Integral to the initial installation of AutoSOCKS is deciding how SOCKS traffic should be redirected through the network. Network redirection rules (used to determine if and how SOCKS redirection should occur) are defined in the AutoSOCKS configuration file. Configuration files are initially created at the end of the installation process; however; they can be added, edited, and removed at any time using the Config Tool (in Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51 via the System menu in the Aventail Program Group; in Windows 95 or Windows NT 4.0 via the Aventail icon in the Taskbar tray). The process of creating one or more configuration files is described under “Configuring AutoSOCKS.”

If you are installing AutoSOCKS on multiple networked workstations, refer to “Network Installation” to determine the best method for maintaining and distributing configuration files. You can then proceed through the initial installation. An Installation Wizard will guide you through the steps, culminating with the option to create a configuration file.

Individual Installation

To install AutoSOCKS

Before running Setup, it is advisable to close all open Windows applications.

1. Installation procedures vary slightly, depending on which media source you use:
   - If you are installing directly from CD-ROM, run SETUP.EXE from the AutoSOCKS directory (\AS_v21).
   - If you are installing directly from diskette, run SETUP.EXE on disk 1.

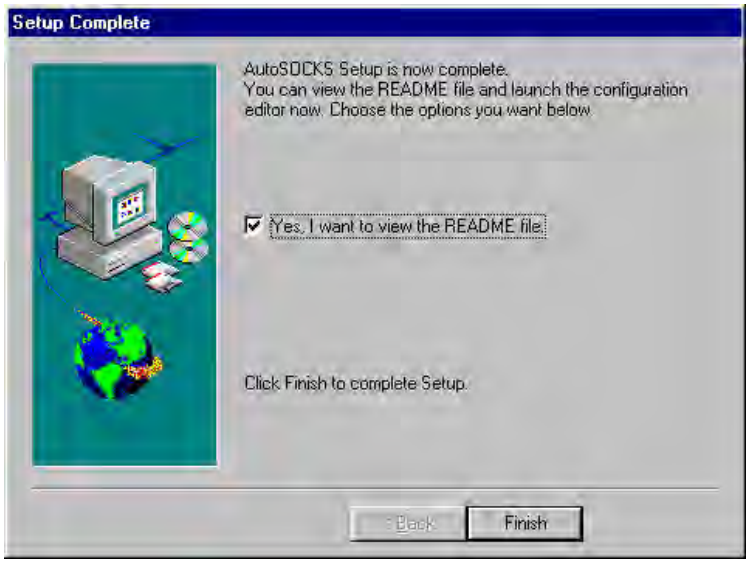
- If you are installing from a network-delivered self-extracting archive, simply run the archive file. This will extract the installation files and automatically launch the setup program.

The AutoSOCKS Installation Wizard then guides you through the process of installing the AutoSOCKS application.

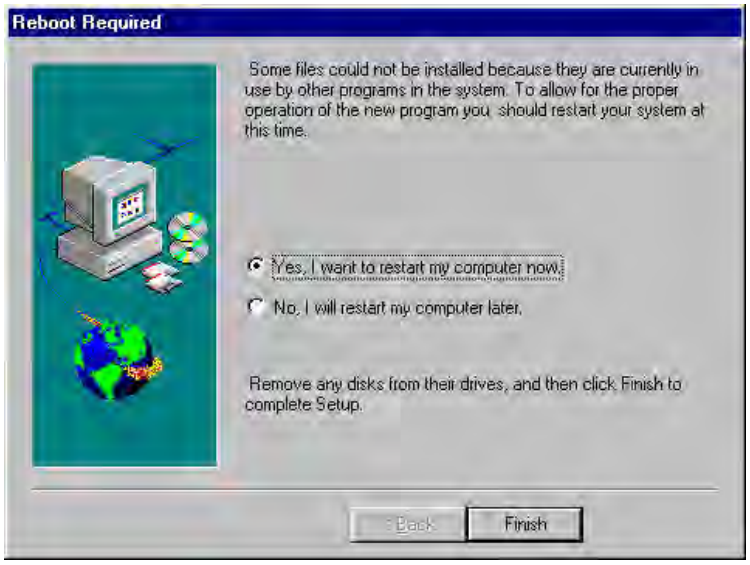
2. At the end of the Setup Program you can click the **Yes, I want to view the README file** box in the Setup Complete dialog box. This opens the README file for the latest information on AutoSOCKS.

   -OR-

   Simply click the **Finish** button to complete the Setup Program.



3. The setup program will then ask you if you want to restart now or later.



4. After restarting your PC, start AutoSOCKS for the first time.
5. AutoSOCKS will ask you if you want to run the Configuration Wizard.

   If you select **Yes,** then the Configuration Wizard will launch to help you create a new configuration file.

   If you select **No**, then AutoSOCKS will ask you to select a configuration file to use.
6. After creating or selecting a configuration file, AutoSOCKS will now be finished installing.

To uninstall AutoSOCKS

The procedure to uninstall (remove) AutoSOCKS varies depending on whether you are running a 16- or 32-bit Windows operating system.

- To uninstall AutoSOCKS from Windows 95 and Windows NT 4.0, double-click **Add/Remove Programs** in the Control Panel window, select AutoSOCKS from the list of programs on the Install/Uninstall tab, and click the **Add/Remove** button.
- To uninstall AutoSOCKS on Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51, use the Uninstall icon in the AutoSOCKS program group.

Network Installation

In general, the process of installing AutoSOCKS to multiple networked workstations involves selection of a file server to use, creation of a staging area for the AutoSOCKS software, and copying the AutoSOCKS files to a shared network directory from the source media. Additional

options include adding a default configuration file, and creating a universal batch/script file that specifies required default command line options when executed by the end user or installation personnel. AutoSOCKS files should be placed in a network drive which can be accessed as a mapped drive or, for Microsoft networks, via a UNC path name (**\\computer_name\share_name\AutoSOCKS**).

Networked Configuration File Setup

There are a number of ways to set up networked client configuration files. These are the most common:

- Client configuration file distributed via a mapped network drive (Novell or Microsoft)
- Client configuration file distributed via a Microsoft UNC path and filename
- Local client configuration file common for all users, but distributed via the standard AutoSOCKS installation and upgrade program

Administrator-Maintained Shared Configuration Files

This is the most desirable configuration method—multiple workstations sharing one or more administrator-maintained configuration files located in a common directory. It is an easily managed configuration because the configuration file is maintained by the network administrator and changes to network topology can be reflected quickly via network distribution. For example:

- A single-networked (usually read-only) configuration file is shared by more than one client workstation. This method is appropriate when workstations share identical message traffic routing rules.
- Multiple configuration files are shared by multiple workstations. This option is useful when you have workstations organized into functional groups (engineering, marketing, accounting, etc.) with group-specific message traffic routing rules.

Shared Configuration File Distribution

Shared configuration files can be easily distributed and, if necessary, updated via the network. All configuration files should be tested first before being distributed.

To distribute a shared configuration file

There are three methods for distributing shared configuration files.

- Copy the file to a Microsoft or Novell network mapped drive accessible by all users. Make sure that end users configure their AutoSOCKS clients to load the configuration file located on the mapped drive.

  -OR-

- Copy the file to a Microsoft Windows workstation supporting UNC-sharing for file resources. (Both the 16- and 32-bit AutoSOCKS clients support specification of the configuration file using the Microsoft UNCs.)

This distribution method has all the benefits of placing the file on a network mapped drive with the added bonus of convenience—end users don't have to actually map the network drive.

-OR-

- Create a shared configuration file, AUTOSOCK.CFG, to be installed on workstations during the standard AutoSOCKS installation/upgrade process. (Place the shared configuration file into the DISK1 directory.) Whenever the AutoSOCKS client is installed or updated, it will to automatically copy AUTOSOCK.CFG to the end user's workstation and set AutoSOCKS to use it.

  Note: If a configuration file is specified as a command line option in the Setup program, installation of the AUTOSOCK.CFG configuration file will be overridden.

Setup Command Line Options

The AutoSOCKS setup program accepts several command line options which allow you to customize the installation process. By using options on the command line, installation can either run entirely unattended, or it can be used to specify a network-based AutoSOCKS configuration file. Each of the command line options are listed in the following table along with a brief explanation. Specifying any of the options that support unattended mode will cause the setup program to perform an automatic installation using default values for any options not explicitly specified.

Option	Explanation	Default Value	Unattended
config=*path*	Specifies the location of the AutoSOCKS configuration file. The destination can be either a local file, or can be specified with a UNC filename or common mapped drive.	Nothing	No
dir=*path*	Specifies the directory containing AutoSOCKS installation files.	**C:\Program Files\Aventail\ AutoSOCKS**	Yes
autostart	If specified, moves the AutoSOCKS application into the Startup group; otherwise AutoSOCKS must be started manually.	Don't put in startup	Yes
nocfg	Specifies that none of the configuration tools should be installed. This option will keep the Config Tool and Configuration Wizard from being installed.	Configuration tools are installed	No
nt=16|32|both	Selects the type of WinSock applications supported by AutoSOCKS: 16-bit, 32-bit or both. This option is only valid for Windows NT	Both	Yes

Configuring AutoSOCKS

Configuration files are created using the Config Tool application. This application can be launched during AutoSOCKS installation or any time you wish to add, modify, or remove a configuration file.

The steps for creating a new configuration file are:

1. Define the SOCKS servers
2. Define the destinations (networks and hosts)
3. Specify redirection rules
4. Enter Local Name Resolution (optional)
5. Manage authentication modules

These procedures are described in the text below.

To launch the Config Tool

The Config Tool opens with the Open AutoSOCKS Configuration File dialog box. After a configuration file is selected or a new file name is entered, the main window of the Config Tool appears.

1. Click the **Yes, I want to configure AutoSOCKS** box in the Setup Complete dialog box (during installation).

   -OR-

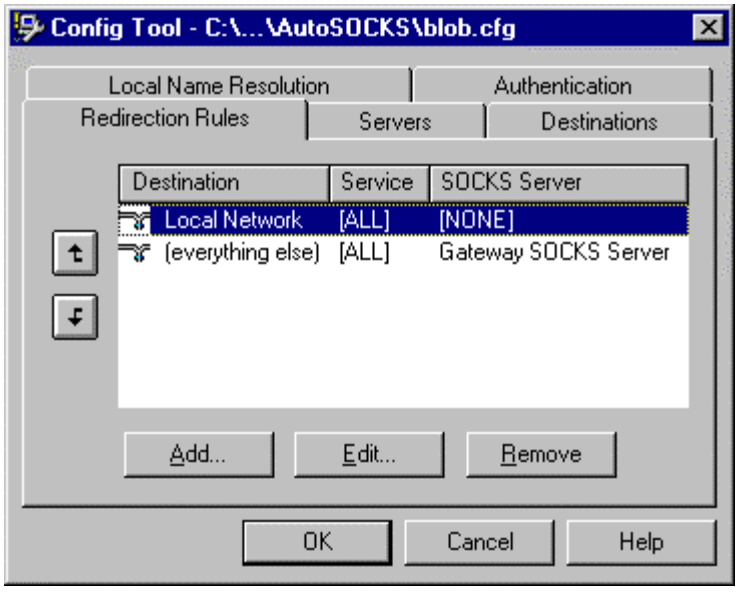
   Select Config Tool from the Aventail AutoSOCKS program group (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51) or the Aventail AutoSOCKS menu (Windows 95 or Windows NT 4.0 Programs menu option).

2. If you are creating a new configuration file, enter a name for the configuration file. (AutoSOCKS defaults to AUTOSOCK.CFG).

   -OR-

   Select the configuration file you wish to open.

   This displays the main window of the Config Tool.



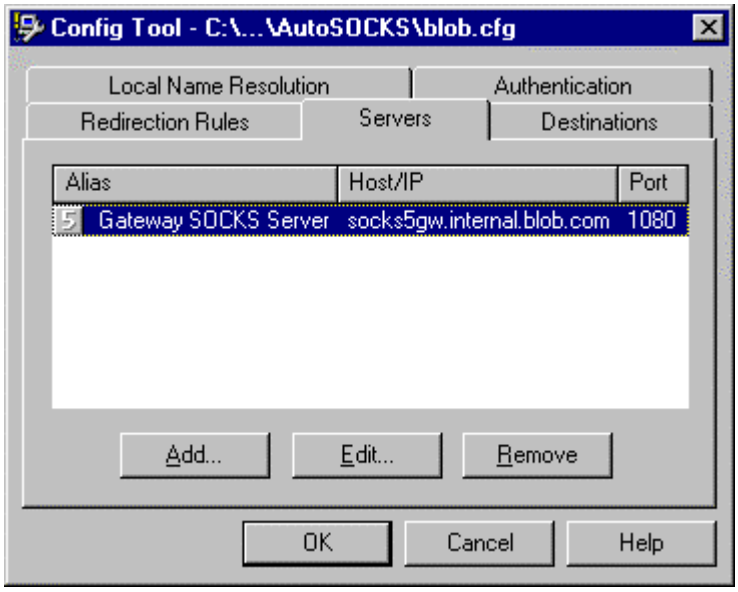
   The Config Tool window contains five tabs. The properties defined on each tab can be edited at any time.

Tab	Function
Redirection Rules	Specifies how network requests are routed to the SOCKS servers.
Servers	Defines the SOCKS servers.
Destinations	Specifies the network and host addresses that should be routed through SOCKS servers.
Local Name Resolution	(Optional) Specifies hostnames that will be resolved by the local workstation.
Authentication	Enables, disables, and sets properties for the authentication modules.

You can change the width of any of the fields on the tabs by moving the cursor to the dividing line between the fields on the field bar. When the cursor changes to a double-headed arrow, click and drag to resize the field.

Define a SOCKS Server

SOCKS servers are defined on the Servers tab in the Config Tool.

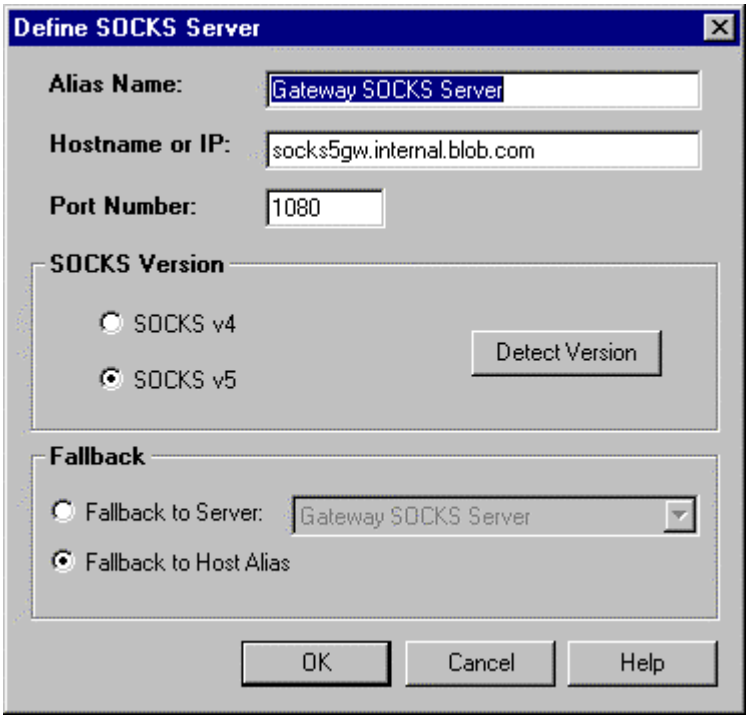


Field	Definition
Alias	The descriptive name you assign to the server. (The number is the SOCKS version.)
Host/IP	The hostname and/or IP address of the server.
Port	The port on which the server is listening.

To add a SOCKS server

1. On the Server tab, click the **Add** button.

   The Define SOCKS Server dialog box appears.



Field	Definition	
Alias Name	User-friendly alias for SOCKS server.	
Hostname or IP	Actual hostname or full numeric IP address for SOCKS server.	
Port Number	SOCKS server port. Default value is 1080.	
SOCKS Version	SOCKS v4:	SOCKS Version 4.0
	SOCKS v5:	SOCKS Version 5.0
	Detect Version:	Detect SOCKS version number.
Fallback	Fallback to Server:	SOCKS server alias for redundant server
	Fallback to Host Alias:	Use DNS records for redundancy

2. In the Alias Name box, type a user-friendly alias for the SOCKS server.
3. In the Hostname or IP box, type the actual hostname of the SOCKS server or its IP address.
4. In the Port Number box, type the SOCKS server's port number. If you don't enter a value, it defaults to the standard SOCKS port 1080.
5. Under SOCKS Version, select the version of SOCKS supported by the server. If you're unsure of the version, click the **Detect** button.

   Note: Typically you should select SOCKS v5 unless the server can only support SOCKS v4.

6. Under Fallback, directly specify a SOCKS server for redundancy or use the Host Alias to specify a SOCKS server.

To edit SOCKS server properties

- Select the SOCKS server you want to edit and click the **Edit** button.

  The Define SOCKS Server dialog box appears with the selected server data filled in. Edit any of the information.

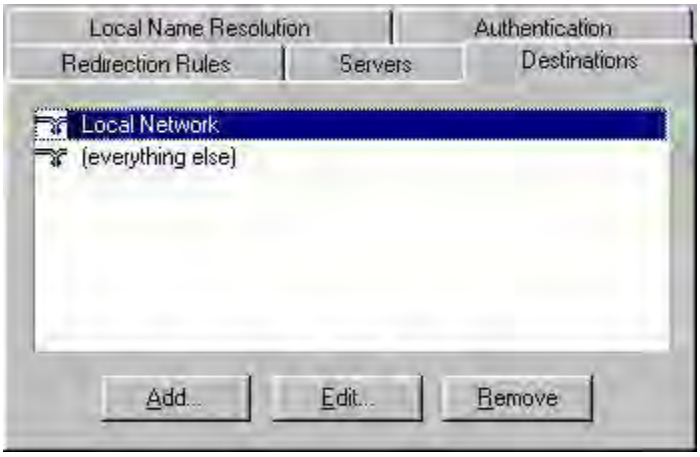
To remove a SOCKS server definition

- Select the SOCKS server you want to remove and click the **Remove** button.

  The server is deleted from the list. Corresponding redirection rules will also be deleted.

Define a Destination

Destinations are defined on the Destinations tab in the Config Tool.



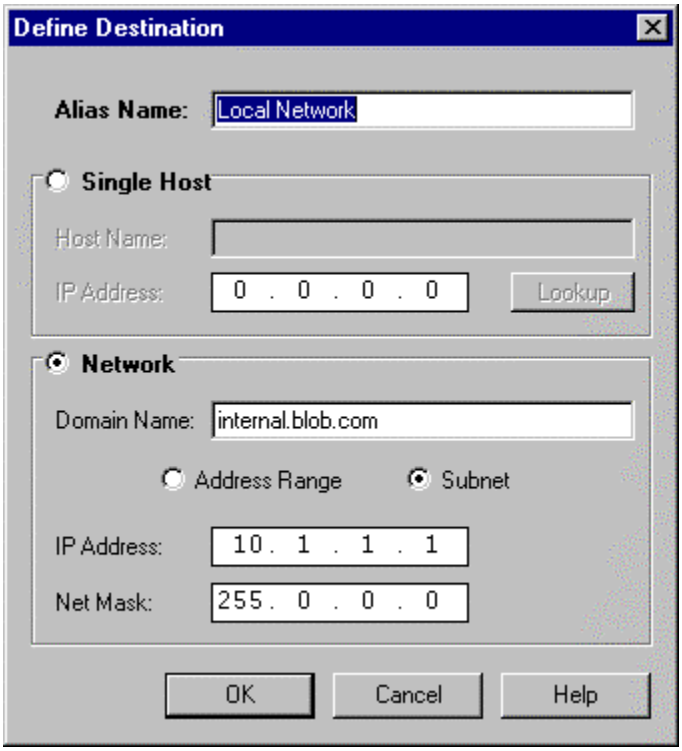
After one or more SOCKS servers are defined, destinations to be routed through them should be added.

Note: The **(everything else)** destination refers to all network and host addresses not otherwise defined.

To add a destination

1. On the Destinations tab, click the **Add** button.

   The Define Destination dialog box appears.



Field	Definition	
Alias Name	User-friendly alias for destination network or host	
Single Host	A specific destination computer	
	Hostname:	Actual name of destination network or host
	IP Address:	Full numeric IP address
	Lookup:	Look up IP address
Network	One or more computers in a network	
	Domain Name:	Domain of the network
	Address Range:	Beginning and ending IP addresses
	Subnet:	IP address and netmask
	From:	Address Range: Starting IP address. Subnet: IP address
	To:	Address Range: Ending IP address. Subnet: Net mask

2. In the Alias Name box, type a user-friendly alias to use for the destination network or host.
3. Choose either the Single Host or Network option:

   Under Single host, type the actual name of the host system and/or its full, numeric IP address. If you don't know the Host's IP address, the **Lookup** button will help you locate it.

   -OR-

   Under Network, type the domain of the network and choose either the Address Range or Subnet options:

Use	To
Address Range	Enter a starting and ending IP address. All addresses between the two will be included as part of the destination. For example, a starting IP address of 192.168.1.0 and an ending IP address of 192.168.1.255 would include all hosts on the 192.168.1 subnet.
Subnet	Enter an IP address and a net mask. This is another way to specify a group of destinations. For example, an IP address of 192.168.1.0 and a net mask of 255.255.255.0 defines the same address range as shown above.

To edit a destination

- Select the destination you want to edit and click the **Edit** button.

  The Define Destination dialog box appears with the selected destination data filled in. Edit the data as necessary.

To remove a destination

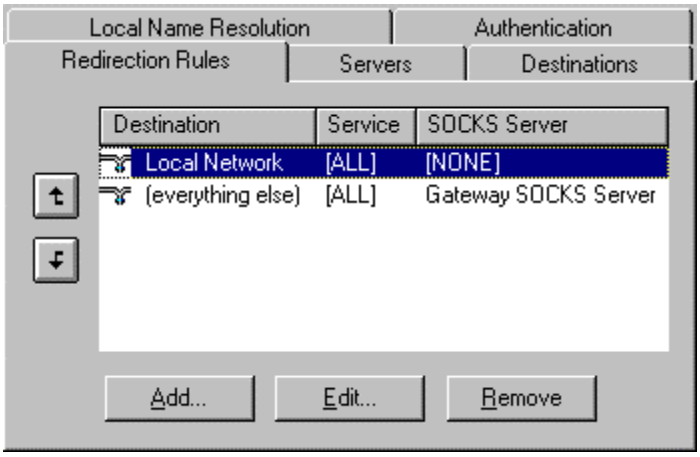
- Select the destination you want to remove and click the **Remove** button.

  The destination is deleted from the list. The corresponding redirection rule will also be deleted.

Enter Redirection Rules

Once servers and destinations are defined, you can then specify how you want AutoSOCKS to redirect (or deny) access to various hosts and services such as e-mail, FTP, and HTTP.

Redirection rules are specified on the Redirection Rules tab in the Config Tool.



In the above example, the redirection rules specify that network traffic on the Local Network will not be redirected through a SOCKS server. All traffic not directed to the Local Network will be proxied through the Gateway SOCKS Server.

Field	Definition
Destination	Destinations defined on the Destination tab
Service	Type of Internet traffic
SOCKS Server	Servers defined on the Server tab

You can change the width of any of the three fields by moving the cursor to the dividing line between the fields on the field bar. When the cursor changes to a double-headed arrow, click and drag to resize the field.

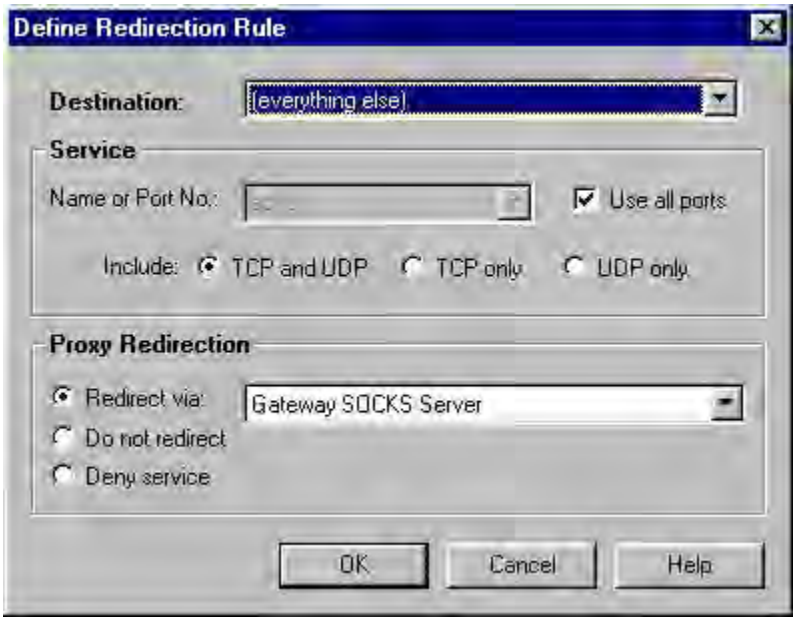
To add a redirection rule

As you add destinations, use the arrow buttons to prioritize them. List the most specific rules first and the general rules last.

Note: AutoSOCKS scans the list from the top down and uses the first matching rule it finds, so it is important to list the most specific rules first.

1. On the Redirection Rules tab, click the **Add** button.

The Define Redirection Rule dialog box appears.



Field	Definition	
Destination	Host or server destination for message traffic.	
Service	Type of Internet traffic.	
	Name or Port No.:	Select from a list of common service ports or enter a new port.
	Use all ports:	Apply the rule to all services.
	TCP and UDP:	Apply the defined rule to both TCP and UDP traffic.
	TCP only:	Apply the defined rule to TCP traffic only.
	UDP only:	Apply the defined rule to UDP traffic only.
Proxy Redirection	Specify how to redirect traffic.	
	Redirect via:	Redirect all traffic through the SOCKS server selected from the list.
	Do not redirect:	Route traffic directly to the specified destination without being redirected through SOCKS.
	Deny service:	Deny access to the specified destination. The network connection is blocked locally instead of at the server level.

2. Select a destination from the Destination list.
3. Under Service, check the **Use all ports** box to apply the rule to all services. Otherwise, select an individual service from the **Name or Port No**. list.
4. Under Proxy Redirection, select one of three redirection options:

   Note: If you select Deny Service and the user has edit control of the Config file, the option can be circumvented by quitting AutoSOCKS or by changing the option in the dialog box.

To edit a redirection rule

- Select the redirection rule you want to edit and click the **Edit** button.

  The Define Redirection Rule dialog box appears with the selected data filled in. Edit any of the information.

To remove a redirection rule

- Select the redirection rule you want to remove and click the **Remove** button.

  The redirection rule is deleted from the dialog box.

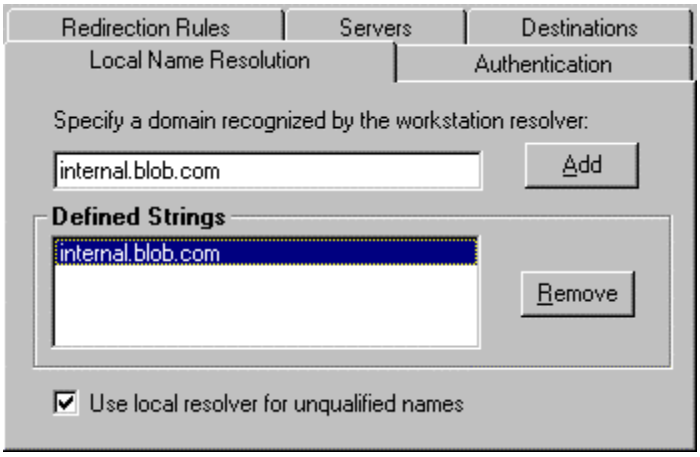
Define Local Name Resolution

Local Name Resolution instructs AutoSOCKS to resolve hostnames locally without needing to venture on to the Internet. This optional feature offers you another level of control over how AutoSOCKS performs name resolution.

The local workstation resolver is the name resolution component of the local TCP/IP stack. This feature acts as a shortcut; hostnames matching the strings defined in the Local Name Resolution dialog box are passed to the local resolver for name resolution instead of being proxied through the SOCKS v5 server.

For example, if **internal.blob.com** is added to the Defined Strings list, then a workstation attempting to connect to **www.internal.blob.com** would perform hostname resolution using the local TCP/IP stack.

Local Name Resolution is specified on the Local Name Resolution tab in the Config Tool.



Field	Definition
Specify Domain	New domain name
Defined Strings	List of domain names that can be resolved locally
Use local resolver	Pass through unqualified hostnames to the local resolver

To add a local domain name

- On the Local Name Resolution tab, type the new name in the Specify Domain text box and click the **Add** button.

  The new name is moved into the Defined Strings text box. It is now active.

To remove a local name

- Select the domain name you want to remove from the Defined Strings text box and click the **Remove** button.

  The domain name is removed from the list.

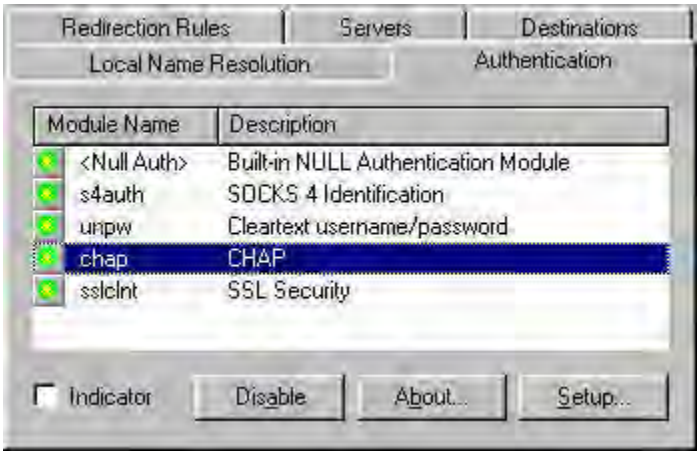
Managing Authentication Modules

SOCKS v5 servers often require user authentication before allowing access. AutoSOCKS authentication modules facilitate this process by displaying dialog boxes which ask for username and password information as well as other authentication credentials.

The current AutoSOCKS authentication modules (SOCKS v4 Identification, Username/Password, Challenge Handshake Authentication Protocol, and Secure Socket Layer) support an AutoSOCKS feature known as credential caching. Credential caching is the process of retaining your authentication credentials once they've been accepted by the SOCKS server. Using credential caching, you can enter your credentials for a SOCKS server once per AutoSOCKS session, rather than once for each individual connection (a tedious task for applications such as WWW browsers).

AutoSOCKS can cache authentication credentials in memory, based on the option you select in the Authentication dialog box. Memory caching stores the credentials for the current session only. When you restart AutoSOCKS or Windows, the memory cache is flushed and you must renter your credentials as prompted.

Authentication modules are managed and configured on the Authentication tab in the Config Tool.



Field	Definition
Module Name	The name of the authentication module on disk;. <Null Auth> indicates that no authentication module will be used.
Description	The description of the authentication method.
Indicator	Check this option to display a visual indication of the authentication/encryption being used as network traffic is generated.

Each authentication module includes its own module-specific configuration. To view or edit a module's configuration dialog box, select the module from list on the Authentication tab and then click the **Setup** button.

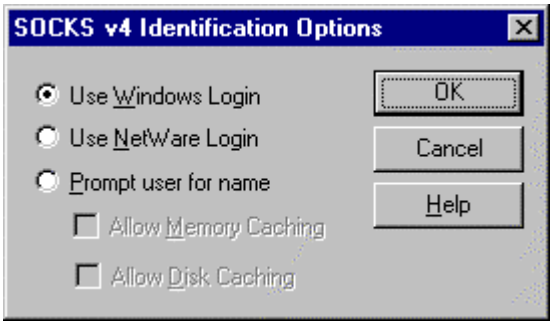
Authentication modules can be selectively enabled and disabled using the Disable/Enable button. By default, the modules are all enabled. This is indicated by the green button next to the module name. When a module is disabled, the button is red.

To configure the SOCKS v4 Identification module

AutoSOCKS includes backward compatibility for the SOCKS v4 protocol. SOCKS v4 does not support password authentication; only your username is sent unencrypted to the SOCKS server along with your connection request. Your username is determined by entries in the SOCKS v4 Identification Module configuration dialog box.

1. On the Authentication tab in the Config Tool, select **sv4auth** (SOCKS v4 Authentication) and click the **Setup** button.

   The SOCKS v4 Identification dialog box appears.



Field	Description	
Use Windows Login	Identify users by their Windows Login names.	
Use NetWare Login	Identify users by their Novell NetWare login names.	
Prompt user for name	Identify users by the names they enter for this specific purpose.	
	Allow Memory Caching	Stores credentials in memory for this session only. Cache is flushed upon restart; credentials must be reentered as prompted.
	Allow Disk Caching	This option is currently unavailable. (Stores credentials on disk for future sessions.)

2. When the option **Prompt user for name** is selected, choose the desired caching option. (Currently only Memory Caching is available.)
3. After making appropriate selections, click **OK**.

   The dialog box closes and the Config Tool is displayed.

To configure the Username/Password authentication module

AutoSOCKS supports the RFC 1928 (Internet standards document) username and password authentication protocol. This authentication method sends your username and password *in clear text* across the network to the destination server. The Username/Password authentication module dialog box contains only credential caching options.

1. On the Authentication tab in the Config Tool, select **unpw** (Clear text username/password) and click the **Setup** button.

   The Username/Password dialog box appears.

Field	Description
Allow Memory Caching	Stores credentials in memory for this session only. Cache is flushed upon restart, credentials must be reentered as prompted.
Allow Disk Caching	This option is currently unavailable. (Stores encrypted credentials on disk for future sessions.)

2. The selection defaults to **Allow Memory Caching**. Click **OK**.

   The dialog box closes and the Config Tool is displayed.

To configure the CHAP Authentication module

AutoSOCKS supports the Challenge Handshake Authentication Protocol (CHAP). This authentication method sends your username and password *encrypted* across the network to the destination server. The CHAP authentication module dialog box contains only credential caching options.

1. On the Authentication tab in the Config Tool, select **chap** (CHAP) and click the **Setup** button.

   The CHAP Options dialog box appears.

Field	Description
Allow Memory Caching	Stores credentials in memory for this session only. Cache is flushed upon restart, credentials must be reentered as prompted.
Allow Disk Caching	Currently Unavailable. (Stores encrypted credentials on disk for future sessions.)

2. The selection defaults to **Allow Memory Caching**. Click **OK**

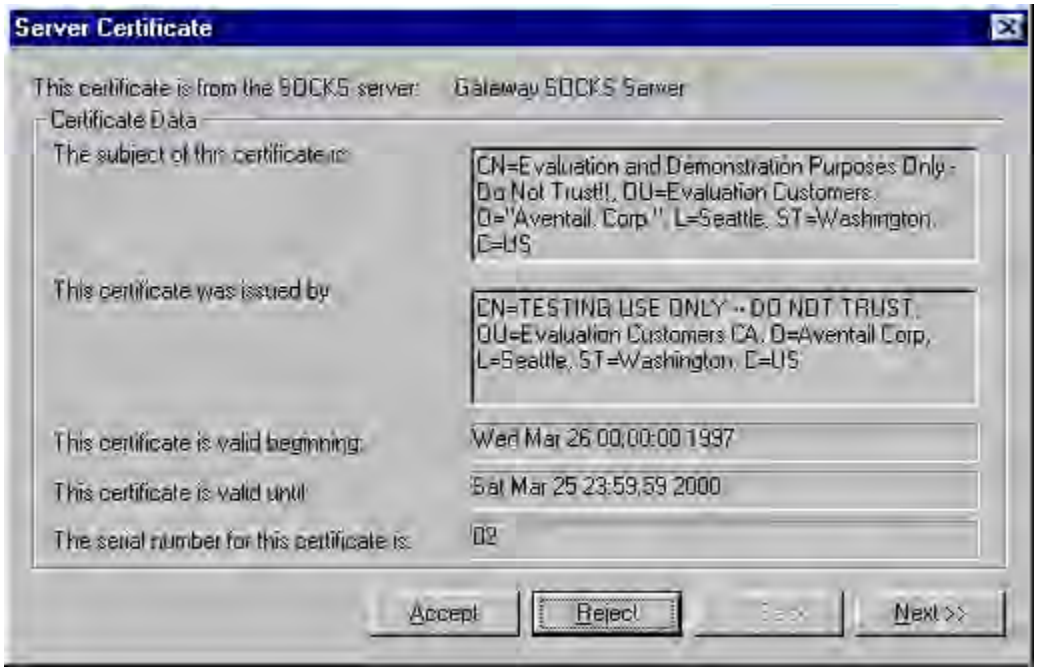
The dialog box closes and the Config Tool is displayed.

To configure the SSL security module

AutoSOCKS supports Secure Socket Layer (SSL) v3.0, a session-layer protocol for securing connections in a general, protocol-independent fashion. At this time, SSL is a TCP-only enhancement; when using SSL with UDP associations, the bulk data is passed without protection.

Normally SSL servers are required to have an RSA key pair and a certificate. RSA is a public/private-key cryptographic system; it creates a key pair: a private key (which, as the name suggests, is kept absolutely private and never shared) and a public key (which is widely published.)

However, you normally must then establish some kind of relationship between your RSA public key and the identity of the server, so that somebody else cannot create their own RSA key information and use it to impersonate your server. *Certificates* establish this relationship. A certificate is essentially an electronic "statement" which verifies that a certain RSA public key is associated with a particular name.



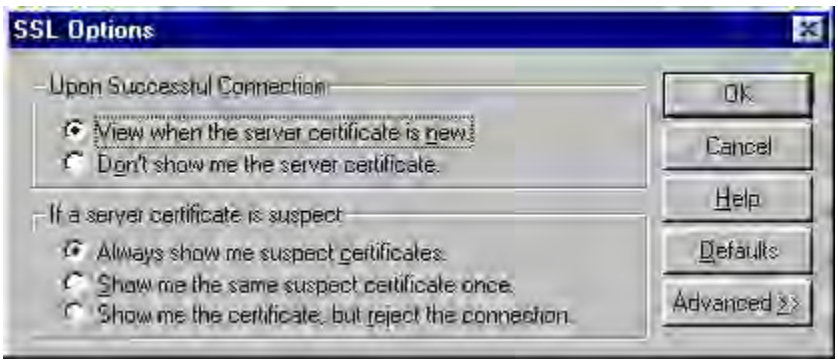
Certificates are issued by a Certification Authority (CA), and are linked together to form a construct called a certificate *chain of authorities*, each one having a previous entity vouching for its identity. In practice, chains generally include two certificates: one confirming the identity of the server, and the other—a "root" certificate—containing the identity and public key of the CA.

Certificates contain special integrity checks and electronic signatures which verify that the certificate is genuine, was issued by some certification authority, and was not tampered with. Anybody can issue a certificate that says anything; the client must know who issued the certificate, and have some trust relationship in order to believe that it is in fact true. The client has a list of trusted CAs. A set of certificate chains can be structured as a tree, with new certificates stemming from old ones. A base CA is sometimes called the “root” or "trusted root" of this tree.

The SSL module dialog box contains an initial set of options regarding the viewing of certificates. It expands into more detail when the **Advanced** button is clicked.

1. On the Authentication tab in the Config Tool, select **sslclnt** (SSL Security) and click the **Setup** button.

   The SSL Options dialog box appears.



Field		Description
Upon Successful Connection:		The certificate is valid.
	View when the server certificate is new.	Upon successful connection, display the server certificate if it hasn't been displayed during the current session.
	Don't show me the certificate.	Never display the server's certificate if it is deemed valid.
If a server certificate is suspect:		The certificate may not be valid.
	Always show me suspect certificates.	Each time a certificate is deemed suspect by AutoSOCKS, display it.
	Show me the same suspect certificate once.	Once a suspect certificate has been accepted by the user, don't display it again.
	Show me the certificate, but reject the connection.	Reject the connection, but display the suspect certificate.

2. Select an action that AutoSOCKS should take once it deems the server certificate acceptable. (Under normal circumstances, the server will provide AutoSOCKS with a certificate to match with one of AutoSOCKS' trusted roots, if any exist):

   - **View when the server certificate is new:** AutoSOCKS displays the certificate the first time it's seen. Subsequent connections to the same SOCKS server will not cause the certificate to be redisplayed.
   - **Don't show me the server certificate:** AutoSOCKS will never display a valid certificate.

3. Select an action that AutoSOCKS should take if it receives a server certificate that is suspect:

   - **Always show me suspect certificates**: AutoSOCKS will display suspect certificates each time they are received. The certificate dialog box will appear for each new connection to the server(s) sending a suspect certificate. (This option allows you to continue the connection despite the fact that the certificate is questionable.) The SSL module authenticates the server's certificate based on the following questions:

Is the certificate valid?

Did a trusted certificate authority (CA) issue the certificate?

Is the name established by the certificate the same as the name of the server for this connection?

If a certificate does not pass all three tests, it is considered a suspect certificate.

- **Show me the same certificate once:** AutoSOCKS will display a suspect certificate the first time that it is received. If you choose to maintain the connection, the questionable certificate will not be displayed again during the current session.
- **Show me the certificate, but reject the connection:** AutoSOCKS will reject a connection if the certificate is suspect. It will display the certificate to allow you to view it.

4. Clicking the **Advanced** button in the dialog box to expand the dialog box into acceptable cipher (a cryptographic algorithm used to encrypt the data stream) options.

Field	Description	
Allow RC4	Offer the RC4 cipher to the server.	
Allow DES	Offer the DES cipher to the server.	
Allow NULL Encryption	Do no encryption. SSL will be used to authenticate, not encrypt.	
Allow Diffie-Hellman Anonymous	Don't authenticate the server; only do encryption.	
Trusted roots	Choose a file with a certificate that specifies certificate chain roots that are to be trusted.	
	Add	Add a new trusted root.
	Import	Import a trusted root.
	Delete	Delete a trusted root.
	View	View a trusted root certificate file.

During the initial SSL connection negotiation, the client and the server negotiate which cipher to use. Checking a particular cipher in the dialog box doesn't mean that it will be used. Instead, each checked cipher is *offered* to the server, but the server must make the final determination. If the server requires a cipher that isn't selected in this dialog box, the authentication will fail.

Any or all of the acceptable cipher options can be selected:

- **Allow RC4:** AutoSOCKS encrypts the information using the RC4 cipher.
- **Allow DES:** AutoSOCKS encrypts the information using the DES cipher.
- **Allow Null Encryption:** AutoSOCKS allows the server to choose *no* encryption. Message integrity is still assured, but the data will be sent in the clear.
- **Allow Diffie-Hellman Anonymous:** AutoSOCKS will be able to communicate with the SOCKS server without requiring a server certificate. The client and server will not exchange certificates, so there will be no authentication. The encryption will still be negotiated, and the data stream will still be encrypted (unless NULL encryption is chosen by the server).

5. If necessary, add a trusted root to the list of trusted roots by pressing the **Add** button, and selecting a file that contains a trusted root certificate.

   When AutoSOCKS receives a certificate from a server, it looks at the root of the certificate chain and matches it against AutoSOCKS' list of trusted root certificates.

6. After making appropriate selections, click OK.

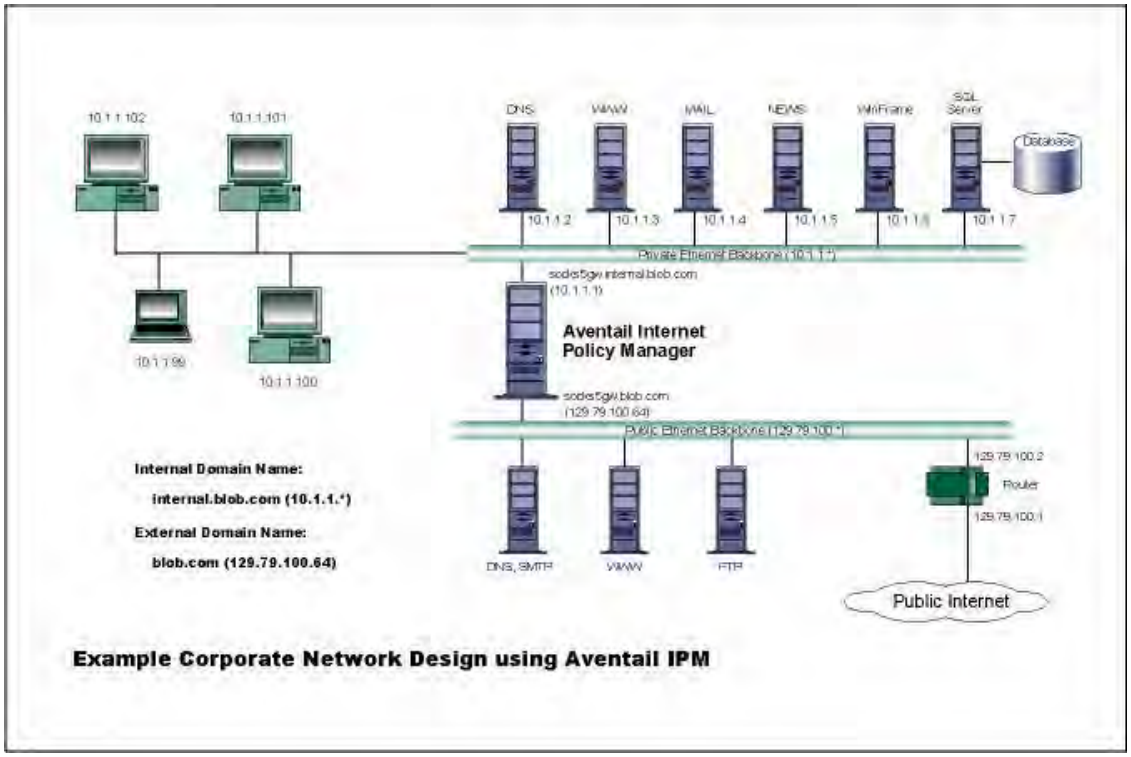
   The dialog box closes and the Config Tool is displayed.

Example Network Configurations

The following sections describe the setup of AutoSOCKS in an example network configuration using the Aventail Internet Policy Manager (IPM) and the Aventail VPN Server.

Configuration Using Aventail Internet Policy Manager

To better describe how to get started configuring AutoSOCKS for use with the Internet Policy Manager, we have created an example network configuration that will be used in all examples throughout this section. Below is an example network topology architecture that emphasizes simplicity to facilitate easy adaptation to real world network designs.



AutoSOCKS in an Aventail Internet Policy Manager Environment

The design used in the example above consists of two individual Ethernet segments, one public and one private. The public segment is used to host anonymous services available to the general public. The public access is provided through a router that is connected to the public Internet. The private segment is used to house all of the corporation's private network resources and data to be used only by internal company employees. To provide protection of the private LAN from unwanted external access, the Aventail IPM is configured such that operating system routing is disabled. Therefore, no direct network connections between the public LAN and the private LAN can be created without being proxied through the Aventail IPM.

The end user workstations (10.1.1.99 through 10.1.1.102) illustrate client workstations, onto which, AutoSOCKS will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail IPM unless they are running AutoSOCKS, which will automatically proxy their application traffic. In this situation, AutoSOCKS will forward traffic destined for the Internet to the Aventail IPM. Then, based on the administrative configuration, the Aventail IPM will proxy end user traffic out beyond the boundary on which the Aventail IPM is located. The client workstations used in this example are Microsoft Windows based PC's.

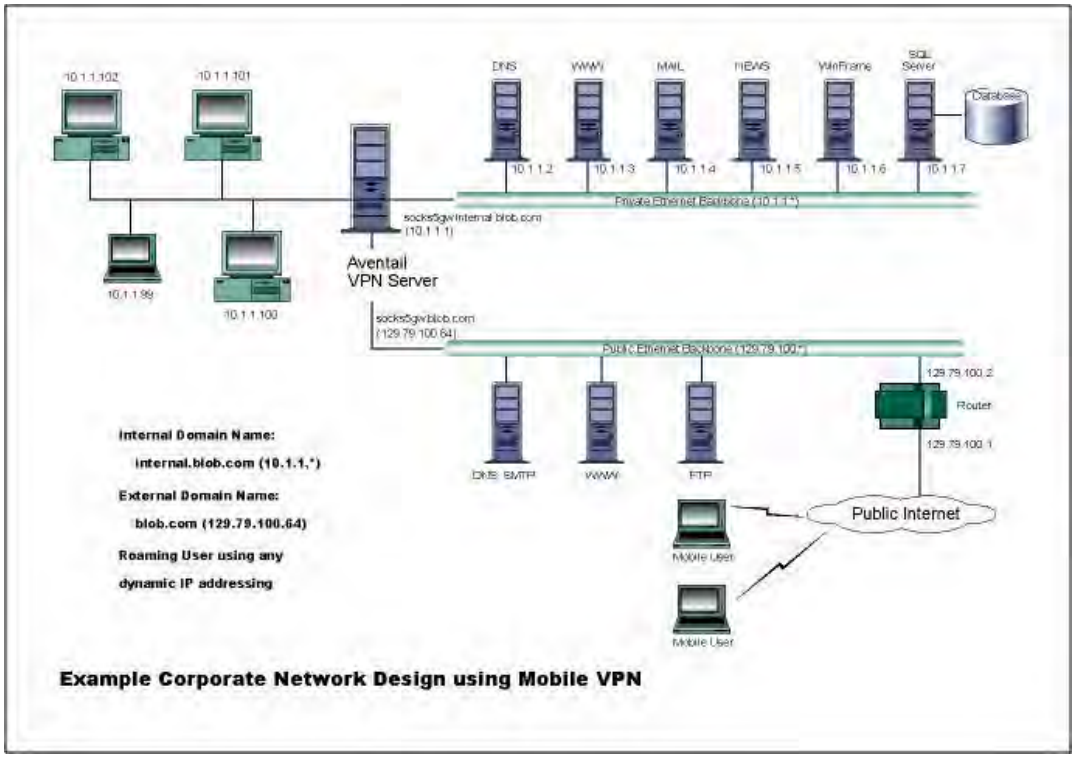
The other servers on the private segment are "internal" or private servers that contain information and tools that are not intended for public use or consumption. If these individual hosts require access beyond the Aventail IPM they can also be configured to use AutoSOCKS. As in the client workstation case, AutoSOCKS will allow applications running on these hosts to traverse the Aventail IPM public/private boundary. In most situations, for more stringent security, these hosts don't have access to the public network at all.

The Aventail IPM in our example, has two network adapters configured to use the internal IP address of 10.1.1.1 and an external address of 129.79.100.64. Since the internal network address space is part of the IANA reserved address space (per BCP RFC 1918) routing MUST be disabled on this host and routing advertisements for this internal network MUST NOT be propagated to the outside world. End user authentication has been enabled on the Aventail IPM server, which will require that users present their credentials before being allowed to have any connectivity to the external public network(s). For this example, Aventail IPM is configured to use RFC1929 Username/Password for authenticating connections AutoSOCKS forwards to it. For additional information on how to configure the Aventail IPM product, consult the Aventail IPM *Administration Guide*.

Subsequently, in most Aventail IPM environments there are large numbers of clients that require installation and configuration. For completeness we will illustrate how to install and configure AutoSOCKS on a large number of client workstations. The easiest and best mechanism for installation of AutoSOCKS to many client workstations is to follow the AutoSOCKS network installation procedures. For our example, we will be installing the base AutoSOCKS client distribution to a network file server that will be used to pull the AutoSOCKS software and client configuration to the desktops. It is often the case that MIS personnel install single copies of AutoSOCKS for testing and evaluating prior to mass deployment. The configuration file that is created through the testing phases will then be copied to a shared file server for group access. This way each client workstation maintains the exact same configuration as determined by the network security policy.

Configuration Using Aventail VPN Server

The following example network configurations show the Aventail VPN Server configured for a Mobile VPN environment and a Partner VPN environment. This example emphasizes simplicity to facilitate easy adaptation to real world network designs.



AutoSOCKS in an Aventail Mobile VPN Environment

The design used in the example above consists of two individual Ethernet segments, one public and one private. The public segment is used to host anonymous services available to the general public. The public access is provided through a router that is connected to the public Internet. The private segment is used to house all of the corporation's private network resources and data to be used only by internal company employees. The Aventail VPN Server depicted in this example is used to provide secure and monitored access to the private LAN for mobile employees and partners. For security reasons the Aventail VPN Server is configured such that operating system routing is disabled. Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the VPN server.

The mobile user workstations connected to the public Internet are the client workstations, onto which, AutoSOCKS will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail VPN Server unless they are running AutoSOCKS. Depending on the security policy and the Aventail VPN Server configuration, AutoSOCKS will automatically proxy their allowed application traffic into the private network. In this is situation, AutoSOCKS will forward traffic destined for the private internal network to the Aventail VPN Server. Then, based on the security policy, the Aventail VPN

Server will proxy mobile end user traffic into the private network but only to those resources allowed. The client workstations we focus on in this section are Microsoft Windows based PC's.

The Aventail VPN Server in our example, has two network adapters configured to use the internal IP address of 10.1.1.1 and an external address of 129.79.100.64. Since the internal network address space is part of the IANA reserved address space (per BCP RFC 1918) routing MUST be disabled on this host and routing advertisements for this internal network MUST NOT be propagated to the outside world. End user authentication and encryption has been enabled on the Aventail VPN Server, which will require all end users to use AutoSOCKS to enable authentication and encryption of their sessions before being allowed to have any connectivity to the internal private network(s). For this example, the Aventail VPN Server is configured to use SSL for encryption of all sessions. For additional information on how to configure the Aventail VPN Server product, consult the Aventail VPN Server *Administration Guide*.

Installation and use of AutoSOCKS for remote access purposes differs a bit from its installation and use with the Aventail IPM product. First, configuration files need to be kept locally on the end user workstation or laptop. This is due to the inability to have a shared file server that allows direct access outside the perimeter of the private network. Second, not all traffic is passed through to the Aventail VPN Server. Only traffic that is destined for the internal network is authenticated and encrypted, all other traffic passes through AutoSOCKS unchanged. For instance, browsing the Internet from the mobile user workstation occurs as if AutoSOCKS was not even running in the background. Large sites with many mobile users will want to setup an internal file server and perform a network installation for use by all of the mobile users to install and configure AutoSOCKS easily. For more information, consult the "Network Installation."

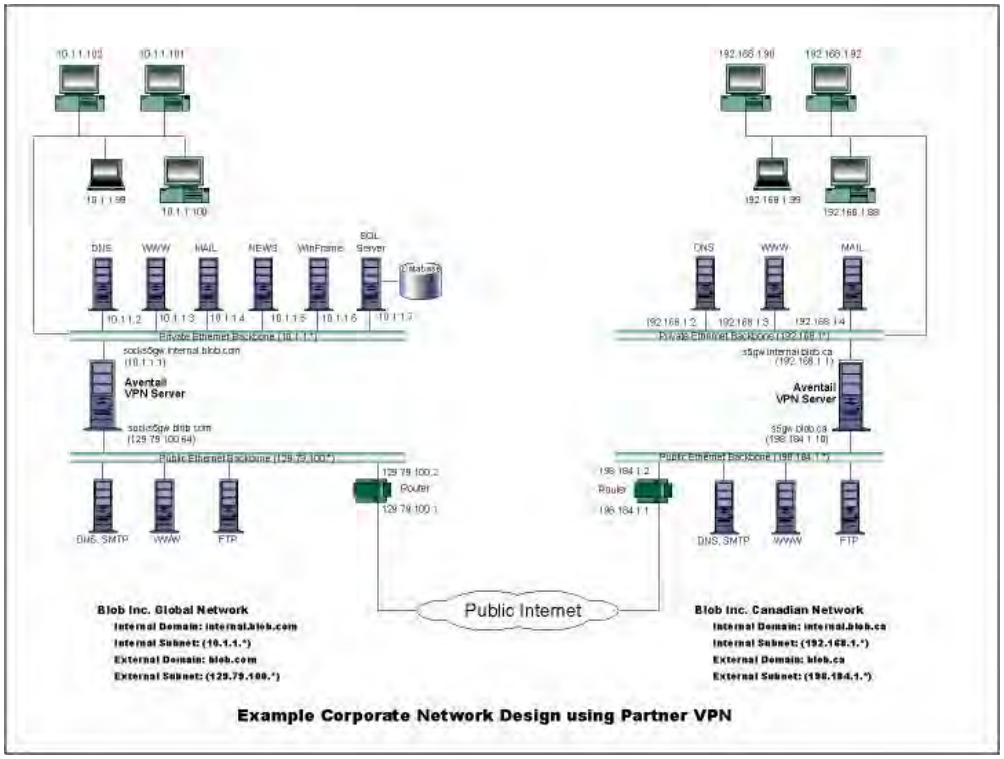


Figure 4. AutoSOCKS in a Partner VPN Environment

AutoSOCKS *Utilities Reference Guide*

Section II, the AutoSOCKS *Utilities Reference Guide*, covers the utilities available from the AutoSOCKS system menu. This section explains:

- Using commands in the System menu including Close, Hide Icon, Help, About, Credentials, Configuration File, Config Tool
- Using the Logging Tool to track AutoSOCKS activity and S5 Ping to check network connectivity

System Menu Commands

Even though AutoSOCKS requires little to no interaction with the end user, there are functions available by way of the AutoSOCKS System menu. To display the System menu, right-click the minimized AutoSOCKS icon (Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.1) or click the AutoSOCKS icon in the Taskbar tray (Windows 95 and Windows NT 4.0).

AutoSOCKS System Menu Commands

Menu Command	Function
Close	Closes AutoSOCKS.
Hide Icon	Hides the AutoSOCKS icon from view.
Help	Accesses online Help.
About	Displays Aventail AutoSOCKS About box.
Credentials	Displays authentication credentials.
Configuration File	Selects a new configuration file.
Config Tool	Runs the Config Tool.
Logging Tool	Runs the Logging Tool.
S5 Ping	Runs the ping and traceroute utilities.

Each of the commands are discussed in the paragraphs below.

Note: The Config Tool, Logging Tool, and S5 Ping commands are optional components and will only appear when they have been installed by the

network administrator. They are discussed in the sections "Logging Tool" and "S5 Ping" below.

Close

This command closes AutoSOCKS. Exiting AutoSOCKS may limit access to certain remote hosts or prevent you from using certain WinSock applications.

Hide Icon

This command hides the AutoSOCKS icon from view. AutoSOCKS will be running the background; however, the icon won't be visible in the system tray (Windows 95, Windows NT 4.0) or minimized on the desktop (for Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51).

Help

This command accesses AutoSOCKS online Help menu.

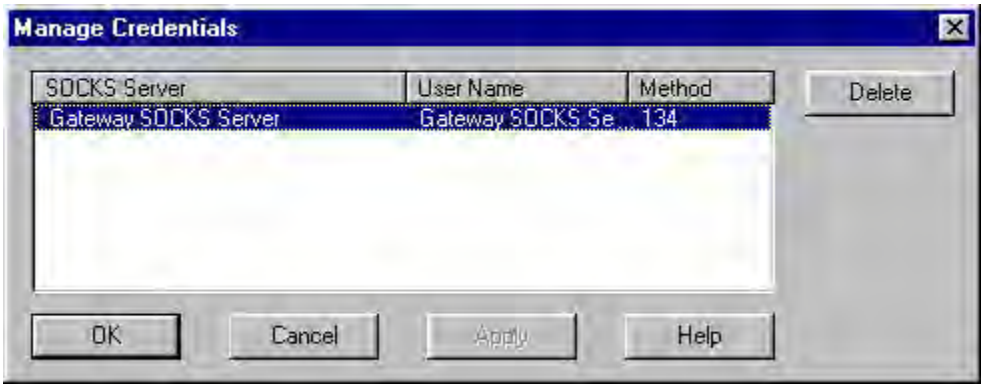
About

This command displays the Aventail AutoSOCKS About box which includes AutoSOCKS software copyright notification, version information, and so on. The **More** button displays a list of files used by the current version of AutoSOCKS.

Credentials

This command displays the Manage Credentials dialog box. Credentials include the information (such as username/password) that you enter when establishing a connection to a SOCKS server requiring user authentication. (AutoSOCKS prompts you with an authentication dialog box.) As long as your credentials are in memory, you can establish connections to associated SOCKS servers without needing to re-enter the authentication information.

Currently, there is no way to edit credential data fields; you can only delete the entire credential entry or clear the password portion of it. In either case, AutoSOCKS will prompt you to enter updated authentication information when you re-establish a connection to the associated SOCKS server.



Field	Definition
SOCKS Server	SOCKS server name
User Name	User name for the SOCKS server
Method	Numeric identifier of authentication method (2=username/password, 3=CHAP, 134=SSL)

To delete a credential entry

Delete authentication credentials when they are no longer correct. After the credentials are deleted, you'll be prompted to reenter them the next time you connect to the associated SOCKS server.

- Select the credential entry you wish to delete and click the **Delete** button.

  This deletes the credential information.

To exit the Manage Credentials dialog box

- Click the **OK** button to accept changes to the credentials and close the dialog box.

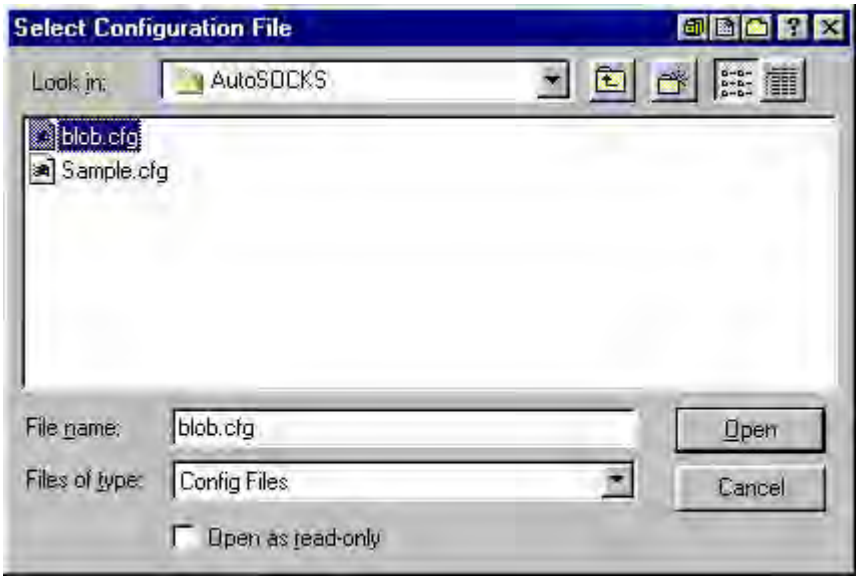
  -OR-

  Click the **Cancel** button to close the dialog box without accepting any changes you might have entered.

  Note: The **Apply** button makes changes permanent but keeps the dialog box open so you can keep working.

Configuration File

This command lets you load a different configuration file from the Select Configuration dialog box. AutoSOCKS defaults to AUTOSOCKS.CFG.



For more information about the configuration file, refer to "Creating Configuration Files."

To load a configuration file

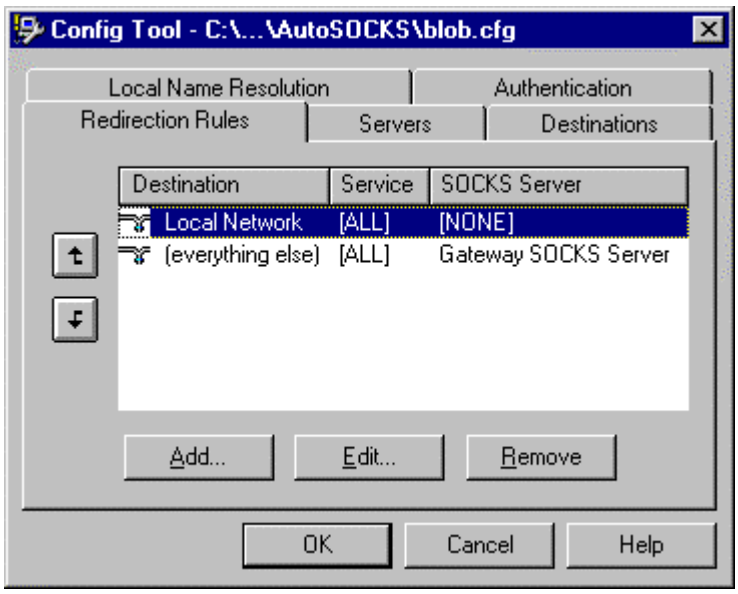
Check with the network administrator before making any changes to the configuration.

- Select the configuration file you wish to load and click the **Open** button.

  The new configuration file is transparently loaded into AutoSOCKS. AutoSOCKS must be restarted for the new configuration parameters to take effect.

Config Tool

The AutoSOCKS Config Tool creates configuration files used to determine how network requests should be routed and which authentication protocols should be enable. (This option may not be available to all users.)



Configuration files should be set up by a network administrator. They are usually created during AutoSOCKS installation but they can also be added, removed, or modified at any time. If necessary, several configuration files can be created for different users or user groups. Some configuration files may reside on a networked drive, accessible by multiple users; other configuration files may be tailored to a specific user on an individual workstation. The Config Tool dialog box is discussed in detail under "Creating Configuration Files."

Logging Tool

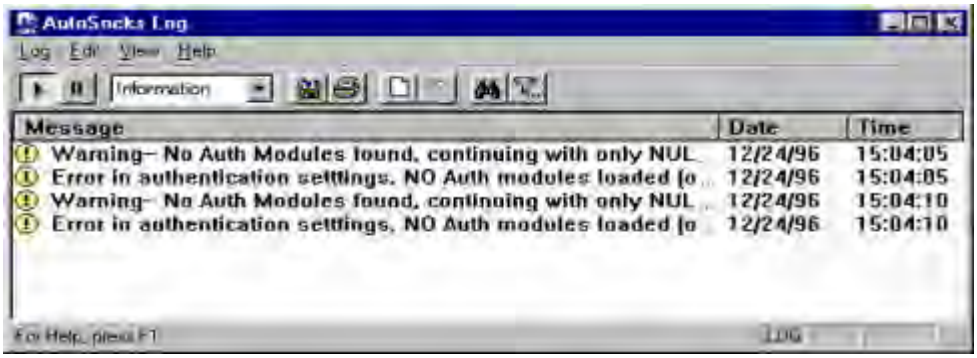
The Logging Tool is a diagnostic utility used to trace AutoSOCKS activity. (This option may not be available to all users.) When running a trace, the Logging Tool displays errors, warnings, and information messages as AutoSOCKS generates them. If desired, the message list can be saved to a log file for later study. Log files can be used to troubleshoot technical problems. They are also useful when running AutoSOCKS for the first time to ensure that network traffic is being routed appropriately.

To trace AutoSOCKS activity

1. Windows 95 or Windows NT 4.0: From the Programs command in the Start menu, point to Aventail AutoSOCKS and click Logging Tool.

   -OR-

   for Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51: From Aventail AutoSOCKS program group, double-click the Logging Tool program icon.



2. In the Log menu, select **Level** and then click one of the three levels of information you wish to trace.

   -OR-

   Select one of the three levels from the list on the toolbar.

Choose	To Log
Errors	Errors only
Warnings	Errors and warnings only
Information	Errors, warnings, and information

3. In the Log menu, click **Trace.**

   -OR-

   Click the **Trace On** button on the toolbar.

   The log window will now record and display trace information as it is generated by AutoSOCKS. You can tell when the trace function is active because messages are scrolling down the screen and the **Trace On** button is depressed.

5. When you're ready to stop the Trace function, click **Trace** in the Log menu

   -OR-

   Click the **Trace Off** button on the toolbar.

   The Trace function is stopped. You can now scroll through the results, print them, and/or save them to a file.

To save a log file

The Logging Tool allows you to append each new message to the end of a .LOG file as the trace is executed, or save the contents of the log window at any time. If you save as the trace is being executed, AutoSOCKS will append messages to the log file until you stop the log function. Data in the log window will not be retained unless it is saved.

There is no way to open a log file from within the log window. You must open a log file using a text editor such as Notepad.

- To save a log file as the data is being generated, click **Log to File** in the log menu. Enter the filename in the Select Log File dialog box.

  -OR-

  Click the **File Logging** button on the toolbar. Enter the filename in the Select Log File dialog box.

- To save the contents of the log window at any time, click **Save As** in the log menu and enter the filename.

To filter messages in the log window

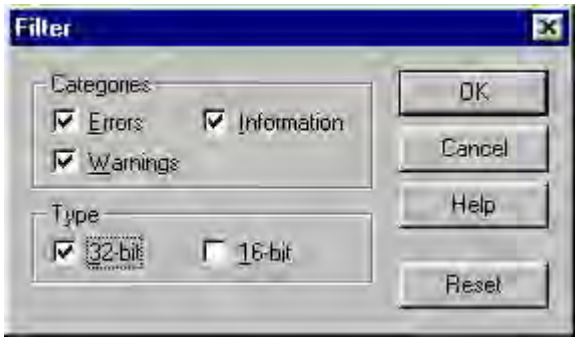
The contents of a log window can be filtered by selecting the types of messages you wish to view. Selecting a specific type of message can make it easier to scan the information onscreen. If the data has been saved to a log file, a view filter will not affect the file contents; it merely adjusts the screen display of those contents.

1. In the View menu, click **Filter Messages** to display the Filter dialog box

   -OR-

   Click the **Filter** button on the toolbar.

   Note: The Filter option is an on/off toggle. If the filter is enabled, click **Filter Messages** to turn it off, then select it again to display the Filter dialog box.



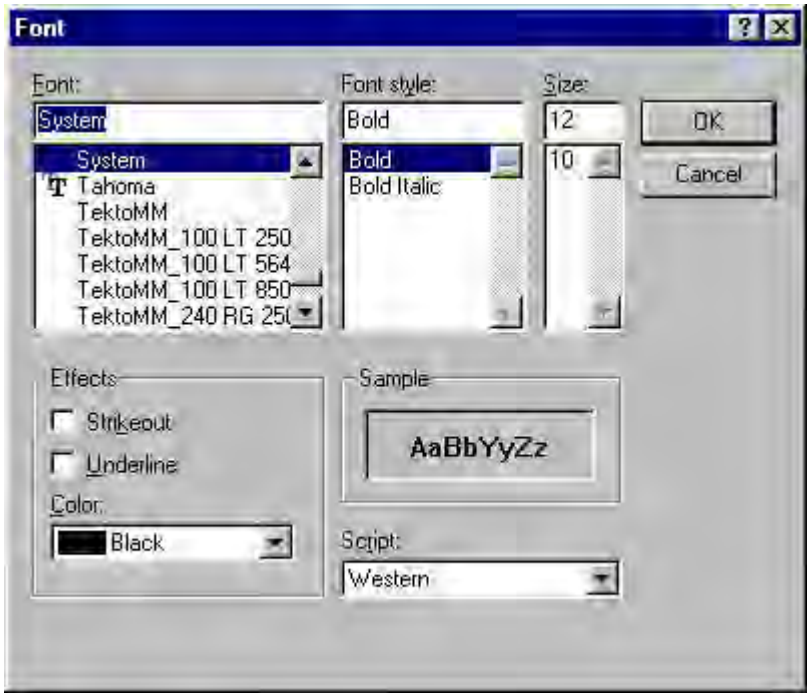
Field	Definition	
Categories	Select any of the three filters to display errors, warnings, and/or information in the log window.	
Type*	32-bit:	Show messages from 32-bit applications.
	16-bit:	Show messages from 16-bit applications.
	*These options are disabled if you're running 16-bit Windows.	

2. Under Categories, select one or more the three filter check boxes. The Log window will adjust the display based on your selection(s).
3. Under Type, select one or both of the check boxes.

To change the view parameters

The display font and window options can be customized as follows:

- In the View menu, click **Font**. Enter your font preferences into the standard Windows Font dialog box.



- To display and hide the toolbar and status bar, click **Toolbar** and/or **Status Bar** in the View menu.

To copy the log window

The log window contents can be copied to the Windows Clipboard.

- To copy all of the window contents to the Windows Clipboard, click **Select All** in the View menu. Then click **Copy** in the Edit menu or click the **Copy** button on the toolbar.
- To copy selected messages to the Windows Clipboard, drag the mouse over the messages to highlight them. Then click **Copy** in the Edit menu or click the **Copy** button on the toolbar.

To print the log window

The contents of the log window can only be printed in its entirety.

- To print the log window contents, click **Print** in the log menu.

  -OR-

  Click the **Print** button on the toolbar.

  The entire contents of the window will be printed, regardless of whether you have specific messages selected. If the display has been filtered, only the filtered messages will be printed.

To find a specific message

The Find function will only work with data displayed in the window. If the display has been filtered, only the filtered messages will be searched. The Find dialog box remains active until you close it.

- In the Edit menu, select **Find.**

  -OR-

  Click the **Find** button on the toolbar.

  Then enter your search parameters into the Find dialog box.

To clear the log window

Log window contents should be cleared when you're ready to execute a new trace, and you no longer need to see the old data.

- In the Edit menu, select **Clear All**.

  -OR-

  Click the **Clear All** button on the toolbar.

To close the log window

When you're ready to close the Log window, make sure you've saved the contents of the trace for later reference if necessary. All settings are saved when you exit.

- In the File menu, select **Exit.**

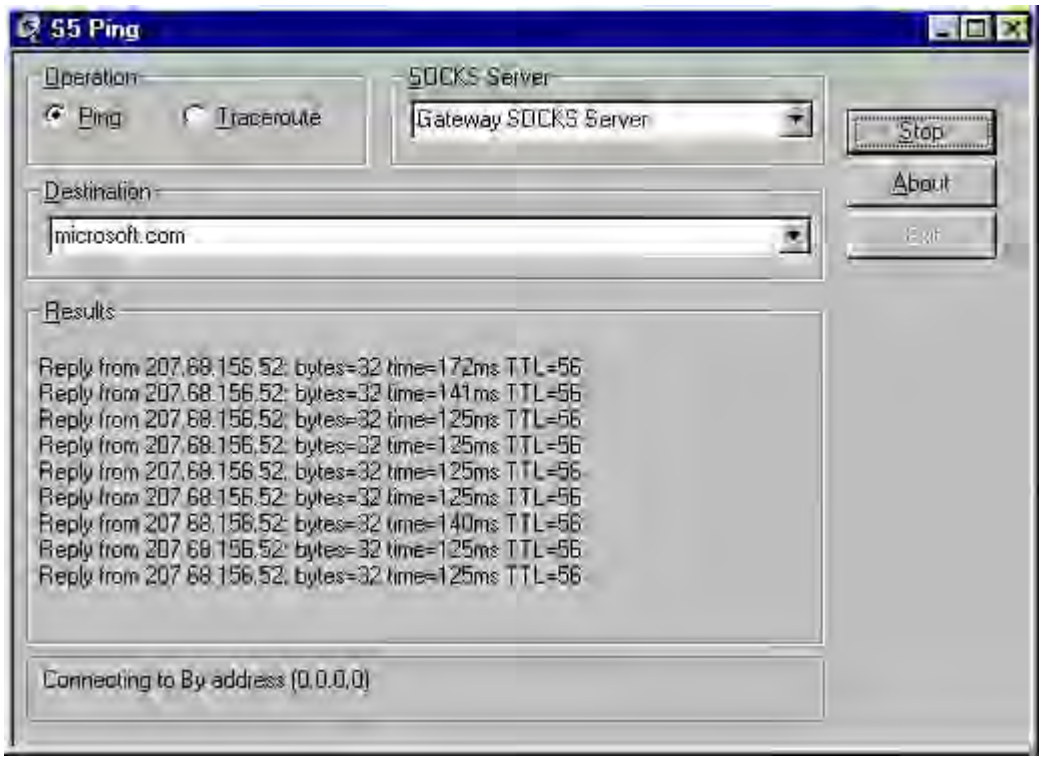
S5 Ping

Two of the most useful diagnostic tools in an administrator's arsenal are ping and traceroute.

- The ping utility checks for network connectivity between two hosts and returns information about the quality of the connection.
- The Traceroute utility checks for network connectivity by displaying information about routers between two hosts. It displays information for each hop.

Ping and traceroute both use Internet Control Message Protocol (ICMP). SOCKS v5 is designed to handle TCP and UDP protocols; however, ICMP is not supported. Because ping and traceroute are based on ICMP, there's no way to directly proxy a ping or traceroute request. To circumvent this problem, AutoSOCKS provides a utility called S5 Ping.

S5 Ping will ping (or traceroute to) a host outside of a SOCKS server by having the client request the SOCKS v5 server to ping the host in question. When a response from the host is returned, the SOCKS server relays the data back to the client and displays it in the S5 Ping window.



Field	Definition
Operation	Select the program you wish to run.
SOCKS Server	The SOCKS server which will execute the operation. If AutoSOCKS is already configured, this list will be preloaded with SOCKS servers from the configuration file.
Destination	The SOCKS server you wish to ping (or traceroute). If AutoSOCKS is already configured, this list will be preloaded with single host destinations defined in the configuration file. (See "Configuring AutoSOCKS.")
Results	The results of the operation once the connection succeeds. The format of the results will vary based upon the SOCKS server platform.

S5 Ping can be used whether or not AutoSOCKS is running. However, if the server that you're connecting through requires authentication, AutoSOCKS must be loaded. The availability of S5 Ping is determined by the network administrator when AutoSOCKS is first installed. In some cases, the S5 Ping command won't appear on the AutoSOCKS System menu or in the program group.

To run ping or traceroute using S5 Ping:

1. Launch S5 Ping.
2. Select the network operation to use (ping or traceroute).
3. Choose which SOCKS server will carry out the ping or traceroute operation.
4. Select the host to ping or traceroute.
5. Click the **Start** button to start the operation.

These procedures are described in the text below.

To launch S5 Ping

S5 Ping can be used whether or not AutoSOCKS is running.

1. Windows 95 or Windows NT 4.0: From the Programs command in the Start menu, point to Aventail AutoSOCKS and click **S5 Ping**.

   -OR-

   Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51: From Aventail AutoSOCKS program group, double-click the S5 Ping program icon.

   -OR-

   If AutoSOCKS is already running, choose the S5 Ping menu item from the AutoSOCKS tray icon menu (Windows 95, Windows NT 4.0) or from the minimized AutoSOCKS

icon System menu (Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51).

The S5 Ping window appears.

Note: S5 Ping will function without a properly configured AutoSOCKS; however, the user will be required to type the information about the target SOCKS server and target host into the SOCKS Server and Destination text boxes.

Once the S5 Ping window opens, you can execute a ping or traceroute network operation.

To run ping or traceroute using S5 Ping

S5 Ping has two modes of operation: ping and traceroute.

1. Under Operation, select one of the two options, Ping or Traceroute.
2. Under SOCKS Server, select a SOCKS server to carry out the operation. If no servers are listed (because S5 Ping did not locate an AutoSOCKS configuration file), type the SOCKS server's hostname or IP address.
3. Under Destination, select a single host destination to ping or traceroute. If no hosts are listed (because S5 Ping did not locate an AutoSOCKS configuration file), type the hostname or IP address of the host you wish to ping or traceroute.
4. Click the **Start** button to execute the operation. The **Start** button then changes to **Stop**. Results from any previous operation are cleared from the window.
5. If the SOCKS server requires authentication, you may be prompted with a server certificate or required to enter a username and password. (For more information about server certificates and username/password authentication, see "Managing Authentication Modules" in the AutoSOCKS v2.1 *Administration and User's Guide*.)
6. Once the connection to the host has been made, the information returned from the server will be displayed in the Results window.

To stop ping or traceroute

- Click the **Stop** button.

  This stops the operation and changes the **Stop** button back to **Start**. The results of the operation remain displayed in the S5 Ping window.

To exit S5 Ping

- Click the **Exit** button.

  This clears the results and closes the S5 Ping window.

AutoSOCKS *User Supplement*

AutoSOCKS automatically routes appropriate network traffic from a WinSock-compatible TCP/IP application such as an e-mail program or a web browser to a SOCKS-based server. (WinSock is a Windows TCP/IP interface that connects a Windows PC to the Internet.) The SOCKS server then sends the traffic to the Internet or the network. Your network administrator defines sets of rules by which this message traffic is to be routed.

This AutoSOCKS *User Supplement* is designed to familiarize you with aspects of the AutoSOCKS interface. Because AutoSOCKS is designed to run transparently, in most cases you'll interact with AutoSOCKS only when it prompts you to enter authentication information for a connection to a secure SOCKS server on the Internet or corporate intranet. You may also occasionally need to start and exit AutoSOCKS although network administrators often configure it to run automatically at startup.

If you have questions about how AutoSOCKS is running on your system, contact your network administrator. Details about other AutoSOCKS commands and utilities are described in the AutoSOCKS v2.1 *Administration and User's Guide*. You might find the section, "Getting Started" to be helpful.

How to Start and Close AutoSOCKS

Because network administrators often set up AutoSOCKS to run minimized at startup, you may never need to actually launch the AutoSOCKS application. When AutoSOCKS is started, it loads a default configuration file, AUTOSOCKS.CFG. This file contains the rules AutoSOCKS uses to properly route network traffic to and from your individual workstation. Your network administrator will inform you if the configuration file name should be different.

Closing AutoSOCKS may limit access to certain remote hosts or prevent you from using certain WinSock applications. Before closing AutoSOCKS it's a good idea to check with your network administrator.

To start AutoSOCKS

- Windows 95 and Windows NT 4.0: From the Programs command in the Start menu, point to Aventail AutoSOCKS and click AutoSOCKS v2.1.

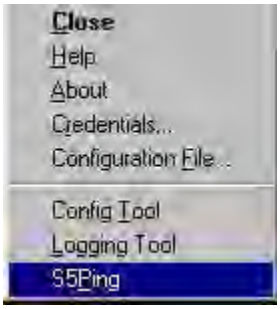
  -OR-

  Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51: In the Aventail AutoSOCKS program group, double-click the AutoSOCKS v2.1 program icon.

  You'll see a minimized AutoSOCKS icon indicating that AutoSOCKS is running in the background. In Windows 95 and Windows NT 4.0, this icon is located in the system tray on the Task bar.

To close AutoSOCKS

- Windows 95 and Windows NT 4.0: In the system tray, right-click the minimized AutoSOCKS icon to display the Aventail System menu, and click **Close**.



-OR-

Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51: Click the minimized AutoSOCKS icon to display the Windows System menu, and click **Close**.

Note: The Config Tool, Logging Tool, and S5Ping may not appear on the Aventail System menu in the program group. This is a configuration option determined when AutoSOCKS is first installed.

How to Enter Authentication Credentials

Some SOCKS servers ask you to authenticate yourself before you are allowed to access them. If you try to connect to a secure SOCKS server, AutoSOCKS may display a dialog box asking you to enter authentication credentials. (For some types of authentication methods, your input isn't required.) Credentials can be as simple as your username or password, or they can be more complicated information. Credentials are assigned to you by your network administrator.

Note: Never talk about credentials over cellular or cordless phones. These lines are not secure and you could be compromising system integrity. If you've mistakenly done so, be sure to let your network administrator know so that you can be assigned a new password.

Currently, AutoSOCKS supports four kinds of user authentication protocols: Username/Password, Challenge Handshake Authentication Protocol (CHAP), Secure Socket Layer (SSL), and SOCKS v4 Identification. To read more about these protocols, see "Managing Authentication Modules" in the AutoSOCKS v2.1 *Administration and User's Guide*.

Once you enter your credentials, AutoSOCKS will save them in memory. This is known as memory caching. Memory caching stores the credentials for the current session only. When

you restart AutoSOCKS or Windows, the memory cache is flushed. If you reconnect to the secure SOCKS server, you must again enter your credentials as prompted.

The following discussion includes Username/Password, CHAP, and SSL authentication. SOCKS v4 authentication does not require user interaction and therefore is not covered in this supplement.

Username/Password and CHAP Authentication

Username/Password and CHAP authentication use basically the same dialog boxes.

To enter authentication credentials

If the secure SOCKS server to which you're connecting uses Username/Password or CHAP authentication, you'll see a dialog box similar to the following:

Note: If you don't know what to enter into the dialog box fields, check with your network administrator.

1. In the Username text box, type your user name.

   Press TAB to move to the next field, or click the Password text box to place the insertion point. Be sure to type your username and password accurately.

2. In the Password text box, type your password.

   Your password is concealed as you type it; it displays on screen as a series of asterisk (*) characters.

3. Under Credential Caching, use the default option **Cache** for this session. Click **OK**.

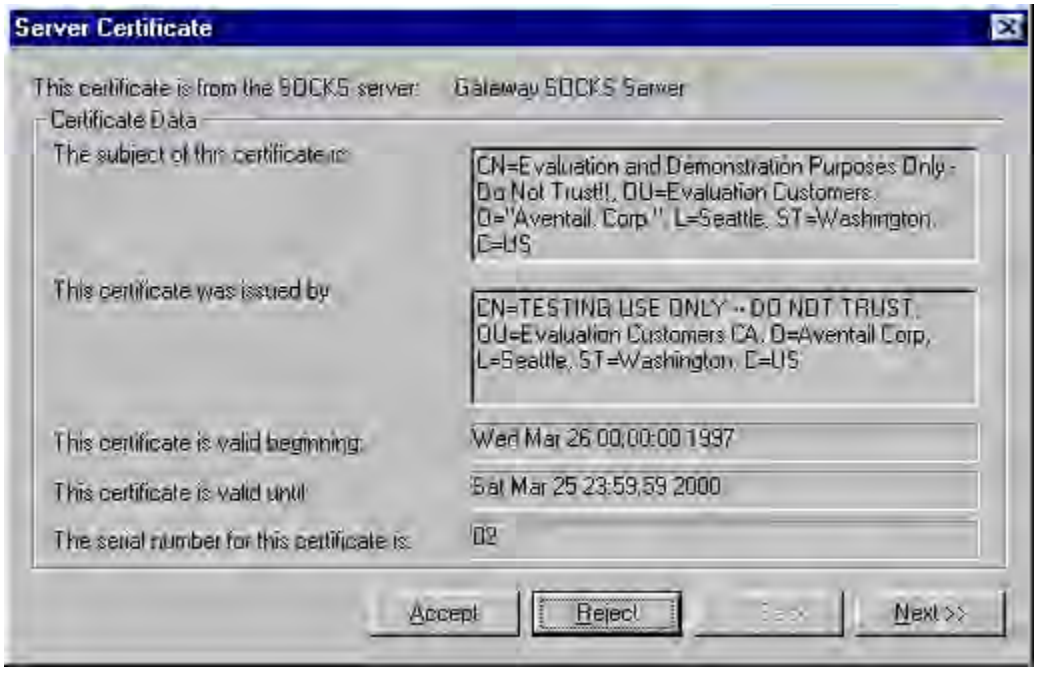
   When you click OK, your credentials are sent to the secure SOCKS server and if they are accepted, you'll continue your processing without hindrance.

If your credentials are refused by the server, the application will display an alert stating that the message traffic didn't go through. Try the transaction again, reentering your username/password. If problems persist, contact your network administrator.

SSL Authentication

SSL authentication, originally developed by Netscape for secure Web communications, uses *authentication certificates* to identify authorized users. A certificate is essentially an electronic "statement" which verifies the integrity of a connection. When you attempt to connect to an SSL server, AutoSOCKS may display the SSL certificate sent by the server. This may not always be the case, depending on how your network administrator has configured the system.

Note: It isn't the mission of this supplement to explain the intricacies of authentication or the components of SSL certificates. If you're interested in learning more about them, talk to your system administrator or read about them in the AutoSOCKS v2.1 *Administration and User's Guide* under "Managing Authentication Modules."



To accept an SSL certificate

Because anyone can issue a certificate that says anything, you should accept certificates only from trusted sources. Otherwise, the information you receive may be invalidated. If you have any concerns about whether or not to accept a certificate, talk with your network administrator.

1. When you see a trusted certificate display on screen, click **Accept**.

   If you click **Reject**, your connection won't be established. If you click **Next**, you see a second "page" of the certificate data with the same Accept and Reject buttons.

   If you click **Accept**, the certificate is accepted as valid and AutoSOCKS *may* display a Username/Password dialog box for you to fill in. The Username/Password dialog will only display if sub-authentication is being negotiated. With SSL authentication, the network administrator has the additional option of requiring you to perform a second (sub) level of authentication.

2. In the **Username** text box, type your user name.

   Press **TAB** to move to the next field, or click the Password text box to place the insertion point. Be sure to type your username and password accurately.

3. In the **Password** text box, type your password.

   Your password is concealed as you type it; it displays on screen as a series of asterisk (*) characters.

4. Under Credential Caching, use the default option **Cache** for this session. Click **OK**.

   When you click OK, your credentials are sent to the secure SOCKS server and if they are accepted, you'll continue your processing without hindrance.

Appendix 1:
Troubleshooting

AutoSOCKS-related problems tend to fall into four categories: Installation, Network Connectivity, Configuration, and Application and TCP/IP Stack Interoperability.

AutoSOCKS Installation Problems

When the instructions in Installing AutoSOCKS in the AutoSOCKS v2.1 *Administration and User's Guide* are followed, problems installing AutoSOCKS are rare. When they occur, they are often the result of:

Toolbars, virus-checking utilities, or other Windows applications running during the installation

If any of these are found to have been running during a failed installation, close them, uninstall AutoSOCKS, reboot, and then re-install AutoSOCKS, taking care to ensure that the toolbars, virus-checking utilities, or applications were not automatically restarted when the system was rebooted.

Insufficient RAM or free space on the volume to which AutoSOCKS is being installed

If either of these is suspected as the cause of a failed installation, increase the available resources according to the System Requirements of the AutoSOCKS v2.1 *Administration and User's Guide* and retry the installation.

Corrupted AutoSOCKS installation media or corrupted or incomplete FTP of AutoSOCKS self-extracting, executable installation file

If corrupted AutoSOCKS installation diskettes are suspected causes of a failed installation, contact Aventail Technical Support for assistance in determining whether the files on the diskettes may have been corrupted and whether replacement diskettes must be obtained from Aventail or your vendor.

If corrupted or incomplete FTP transfer of AutoSOCKS installation files obtained over the Internet is suspected, retry the transfer, taking care to ensure that the FTP client is in binary mode and confirm that the transfer completes normally. Contact Aventail Technical Support to confirm that the byte size of the transferred installation file is correct.

Installation to a workstation on which AutoSOCKS was running or from which a previous version of AutoSOCKS was not completely uninstalled

If either of these circumstances is suspected causes of a failed installation, contact Aventail Technical Support.

Installation script errors

AutoSOCKS is installed with InstallShield. If InstallShield reports errors during a failed installation, note the text of the error messages and the specific circumstances in which they occurred and contact Aventail Technical Support.

Network Connectivity Problems

Before AutoSOCKS can be used to successfully redirect WinSock application connections:

1. The workstation on which AutoSOCKS is installed must also have a properly installed, Winsock-compatible, TCP/IP stack running on it.

This installation can be confirmed by successfully pinging the IP address of the workstation, from the workstation itself, using a WinSock ping application. If this test fails, the failure must be corrected before AutoSOCKS can be tested and before Aventail Technical Support can provide assistance.

2. Basic TCP/IP network connectivity must exist between the client workstation on which AutoSOCKS is installed and the SOCKS server(s) to which it is configured to redirect connections.

This connectivity can be confirmed by successfully pinging the SOCKS server(s) by IP address, from the client workstation. If this test fails, the failure must be corrected before AutoSOCKS can be tested and before Aventail Technical Support can provide assistance.

3. Basic TCP/IP network connectivity must also exist between the SOCKS server(s) and the network host(s) to which the SOCKS server(s) are expected to proxy connections.

This connectivity can be confirmed by successfully pinging the network host(s), by IP address, from the SOCKS server(s). If this test fails, the failure must be corrected before AutoSOCKS can be tested and before Aventail Technical Support can provide assistance.

AutoSOCKS Configuration Problems

This section addresses troubleshooting of simple AutoSOCKS configuration problems. Troubleshooting of complex AutoSOCKS configuration problems is beyond the scope of this section.

It is easiest to troubleshoot AutoSOCKS configuration problems by creating and testing simple AutoSOCKS configuration files, such as those that may be created with the AutoSOCKS Configuration Wizard. However, all references to host and domain names should be removed from configuration files created with the wizard, before testing, to defer possible name resolution complications until the files can be demonstrated to work with IP addresses, alone.

Note: The IP address and SOCKS port number of the SOCKS server(s) to which AutoSOCKS must connect must be known, before troubleshooting AutoSOCKS configuration problems. Neither AutoSOCKS, nor Aventail

Technical Support, can discover the IP address or port number of the SOCKS server(s).

When troubleshooting AutoSOCKS configuration problems, confirm that the AutoSOCKS configuration file that is currently selected in the Configuration File… dialog is the one intended for testing.

After selecting a configuration file to test, open the AutoSOCKS Config Tool and:

1. Confirm that the SOCKS server has been correctly identified by IP address.

Click on the Servers tab, click on the server alias, and then click on the **Edit** button. Compare the IP address in the Hostname or IP: field with that of the SOCKS server.

If the SOCKS server is a SOCKS v5 server, click on the SOCKS v4 radio button in the SOCKS Version section of the Servers tab. Then click on the **Detect Version** button. The selection should revert to the SOCKS v5 radio button, indicating that AutoSOCKS detected a SOCKS v5 server running at the IP address specified in the Hostname or IP: field.

If, on the other hand, the SOCKS server is a SOCKS v4 server, click on the SOCKS v5 radio button in the SOCKS Version panel. Then click on the **Detect Version** button. The selection should revert to the SOCKS v4 radio button, indicating that AutoSOCKS detected a SOCKS v4 server running at the IP address specified in the Hostname or IP: field.

If **Detect Version** fails to detect a SOCKS server of either version, it is possible that no SOCKS server is running on the host identified in the Hostname or IP: field. Contact your SOCKS server administrator to confirm that the SOCKS server is running at the address specified.

2. Confirm that all AutoSOCKS Authentication Modules are enabled.

Click on the Authentication tab and confirm that the "traffic light" icons for all of the Authentication Modules are green, indicating that the modules are enabled. Enabling all the modules configures AutoSOCKS to attempt any form of authentication demanded by the SOCKS server or null (no) authentication. Note the form of authentication demanded by the SOCKS server and, if necessary, obtain the proper authentication credentials, such as a SOCKS server username and password, from the SOCKS server administrator.

3. Confirm that the network hosts to which the SOCKS server is expected to proxy connections are within a redirected destination.

Click on the Destinations tab, click on the Destination which includes the network host to which the SOCKS server is expected to proxy connections, and then click on the Edit button. Confirm that the definition of the Destination includes the network host.

Next, click on the Redirection Rules tab. Confirm that connections to the Destination are configured to be redirected by the SOCKS server.

After making any necessary changes to the AutoSOCKS configuration, restart AutoSOCKS and then restart any WinSock applications, before testing the new configuration.

Application and TCP/IP Stack Interoperability Problems

AutoSOCKS is intended to "automatically socksify" all "well-behaved" Winsock applications. Occasionally, Winsock applications are found which AutoSOCKS does not socksify, due to interoperability problems with the application.

AutoSOCKS is also intended to run on all WinSock-compliant Microsoft Windows TCP/IP stacks. Occasionally, WinSock stacks are found on which AutoSOCKS does not run as expected, due to interoperability problems with the stack.

If an application or stack inter-operability problem is suspected, report it to Aventail Technical Support. Aventail will make every effort to resolve interoperability problems.

AutoSOCKS Trace Logging

AutoSOCKS includes a Logging Tool for doing traces of AutoSOCKS and Winsock activity. AutoSOCKS traces are often useful in troubleshooting AutoSOCKS network, SOCKS server, and Winsock application interoperability problems. Aventail Technical Support engineers may request that you perform a debug-level trace, log it to file, and e-mail it to them.

Before Starting an AutoSOCKS Trace:

1. Close any WinSock applications that are running on the workstation.

2. Close AutoSOCKS, if it is running.

3. Start an AutoSOCKS Trace.

4. Click on the Windows Start | Programs | Aventail AutoSOCKS | Logging Tool menu bar item. The AutoSOCKS Logging Tool window should open, as illustrated in Figure 1, below.

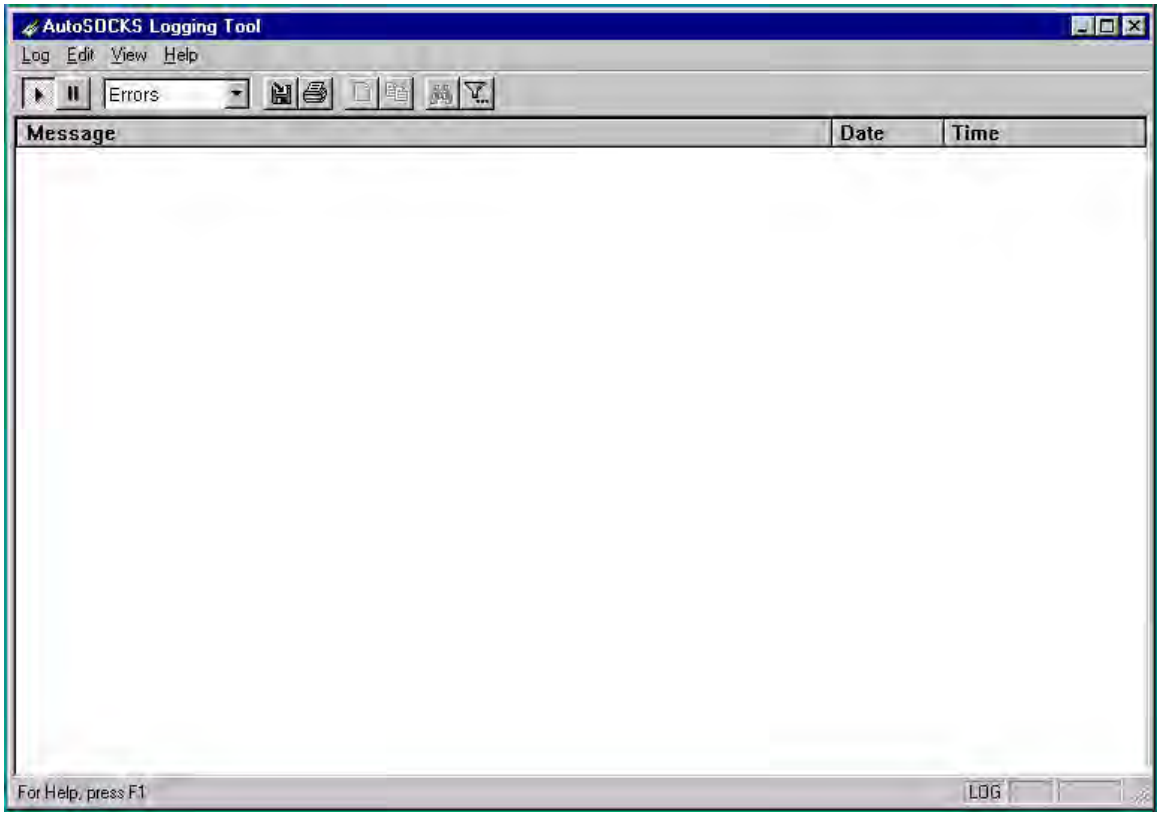


Figure 1

5. In the Logging Tool window Log menu, confirm that the Trace option is checked. If it is not, click on the Trace option, to check it.

Saving an AutoSOCKS Trace to a File:

1. In the AutoSOCKS Logging Tool window Log menu, confirm that the Log To File... option is checked. If it is not, click on the Log To File... option, to check it. The AutoSOCKS Logging Tool window Log menu should appear as illustrated in Figure 2, below.

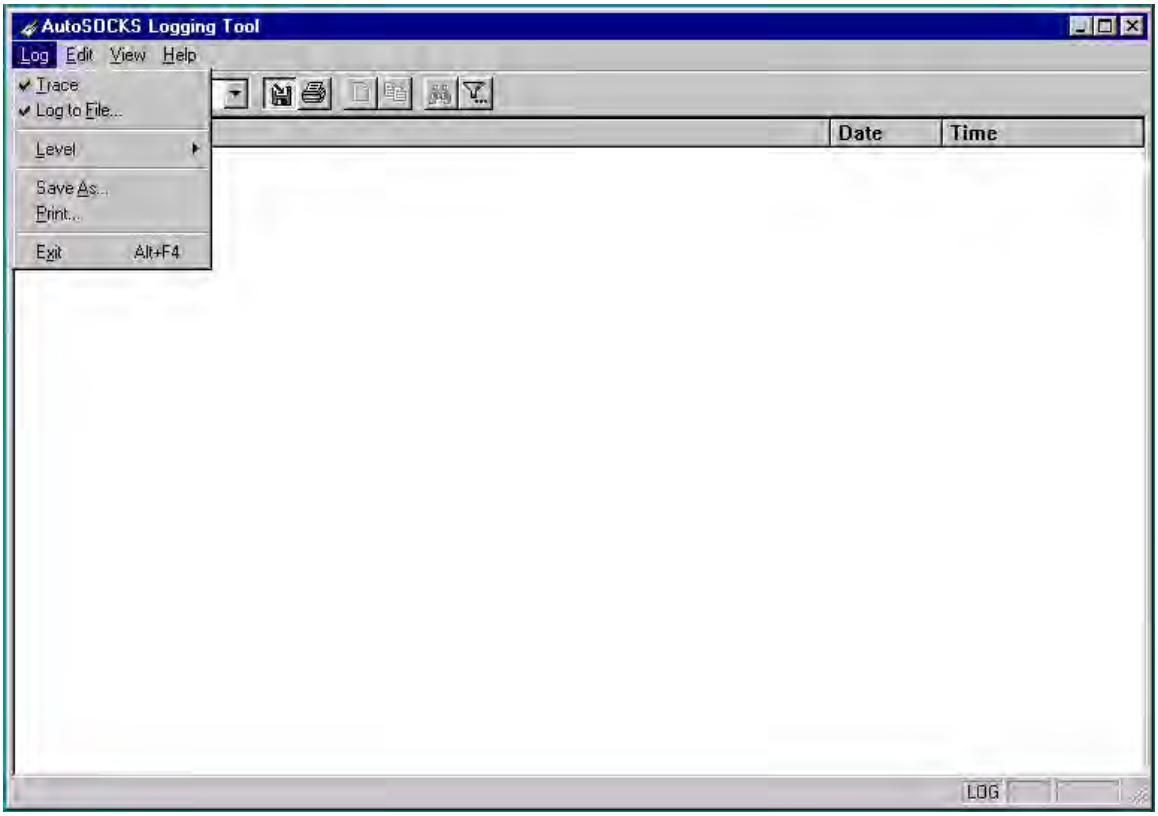


Figure 2

2. A Select Log File dialog box should appear, as illustrated in Figure 3, below. Enter a file name appropriate to later identify the file and click Save.

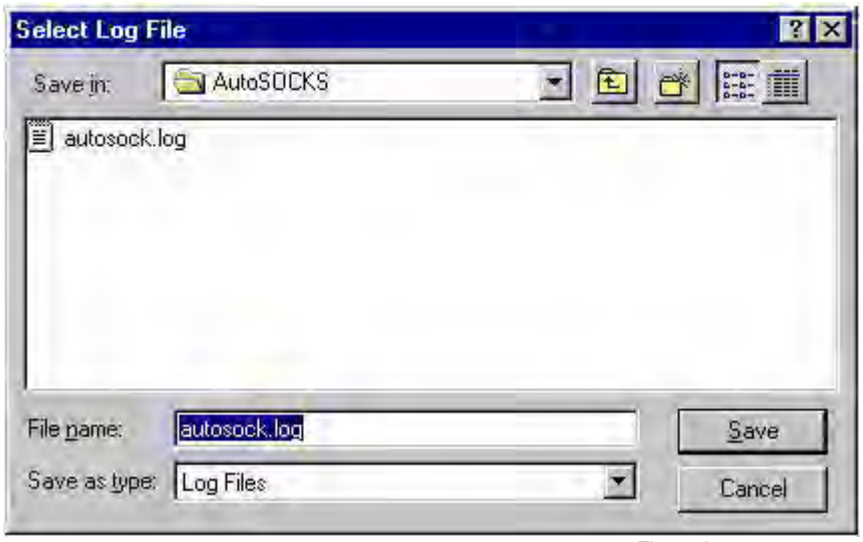


Figure 3

Setting the AutoSOCKS Trace Level to Debug:

1. Click on the AutoSOCKS Logging Tool window and then press <Ctrl><4>."Debug" should appear in the drop-down text box in the AutoSOCKS Logging Tool toolbar, as illustrated in Figure 4, below.

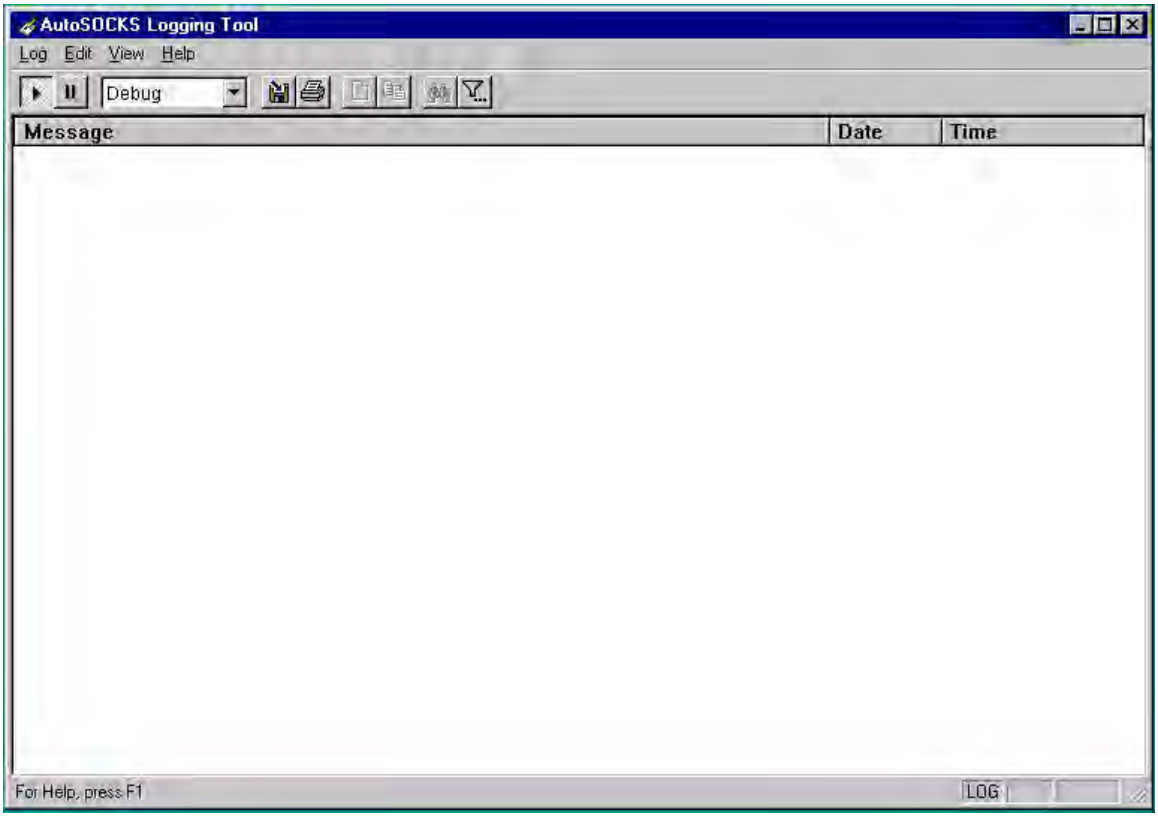


Figure 4

Note that, when tracing in Debug mode, not all messages that are displayed are indicative of error.

Logging Trace Data:

1. Start AutoSOCKS.

2. Start the Winsock application.

3. Reproduce the problem and only the problem.

4. Close the trace log file and confirm that it was saved.

Reporting AutoSOCKS Problems

Report AutoSOCKS problems to Aventail Technical Support, ideally by completing and submitting an AutoSOCKS Problem Report on the Support page of the Aventail website.

Glossary

alias
User-friendly name for destination network or host computer.

authentication
A method for identifying a user in order to establish access to a system resource or network. Authentication information such as username/password is entered via prompts.

certificate
A certificate is essentially an electronic "statement" which verifies that a certain RSA public key is associated with a particular name. Certificates are issued by a Certification Authority (CA).

client
A program or Internet service that sends commands to and receive information from a corresponding program known as a server. Most Internet services run as client/server programs.

configuration file
A file of information containing traffic redirection rules used to determine if and how SOCKS redirection should occur.

credentials
Credentials include the information (such as username/password) that you enter when establishing a connection to a SOCKS server requiring user authentication.

domain
Internet name for a network or computer system.

encryption
A security procedure that converts data into a format which can be read only by the intended recipient computer.

firewall
Software or hardware barriers that control the flow of information to Private networks.

host
A server connected to the Internet.

Internet Protocol (IP)
The basic data transfer protocol used for the Internet. Information such as the address of the sender and the recipient is inserted into an electronic “packet” which is then transmitted.

intranet
A network that is internal to a company or organization.

log window
The window of the Logging Tool which shows alerts, messages, and warnings generated by AutoSOCKS.

ping
A utility that determines if a remote host computer is up. ping sends data packets to the host. If the packets are not returned, the host is down.

protocol
Rules and procedures used to exchange information between networks and computer systems.

redirection rule
Rules defined in the configuration file which specify how network requests are routed to SOCKS servers.

server
A networked computer that shares resources with other computers. Servers "serve up" information to clients.

SOCKS
SOCKS is a security protocol. It acts as a proxy mechanism that manages the flow and security of data traffic to and from your local area network or intranet.

SSL
Security Sockets Layer, an authentication protocol.

Transmission Control Protocol (TCP)
A means of sending data over the Internet with guaranteed delivery.

Transmission Control Protocol/Internet Protocol (TCP/IP)
A suite of protocols the Internet uses to provide for services such as e-mail, ftp, and telnet.

traceroute
A utility that traces the routing of data over the Internet to a specific computer. Traceroute sends a data packet and then lists the intermediate host computers that it traverses on it's way to the destination machine.

User Datagram Protocol (UDP)
A means of sending data over the Internet without guaranteed delivery. Also known as "connectionless" protocol, it is used for data such as RealAudio®.

Universal Naming Convention (UNC)
A way of accessing a file or directory on another computer. For example:
//host/share/directory/file ("**share**" refers to the alias used to make the resource available.)

WinSock
(Windows Socket) A Windows component that connects a Windows PC to the Internet using TCP/IP.

workstation
Any computer connected to a network.

Index

EXHIBIT C

TO MICHAEL FRATTO'S DECLARATION

PR NEWSWIRE, "AVENTAIL SHIPS THE FIRST STANDARDS-BASED VIRTUAL PRIVATE NETWORK SOFTWARE SOLUTION" (MAY 2, 1997)

Aventail Ships the First Standards-Based Virtual Private Network Software Solution

Publication: **PR Newswire** Publish date: **May 2, 1997**

Aventail MobileVPN and PartnerVPN Include Granular Access Controls and Support

For Multiple Authentication and Encryption Methods

SEATTLE, May 2 /PRNewswire/ -- Aventail Corporation announced today the availability of the industry's only standards-based Virtual Private Network (VPN) software solutions. Aventail MobileVPN and Aventail PartnerVPN for Windows NT will begin shipping today and pricing starts at $4,995. UNIX versions will be available at the end of this month.

Aventail MobileVPN and Aventail PartnerVPN enable organizations to securely communicate over the Internet, allowing companies to extend the reach of their corporate intranet to customers, partners, remote offices, and mobile employees. Aventail's adherence to standards simplifies VPN deployment, enables interoperability, and leverages corporations' existing network investments.

"Aventail has moved the concept of a VPN to the next level. They are the only company providing a highly secure circuit-level solution that is deployable over existing network infrastructure and has the ability to work with a variety of authentication and encryption technologies," says Ira Machefsky, vice president at Giga Information Group.

The Only Standards-Based VPN Product

Aventail MobileVPN and Aventail PartnerVPN are the first VPN solutions based on SOCKS v5, an open Internet Engineering Task Force (IETF) standard. SOCKS is a distributed network security standard that represents the next-generation of Internet security. The SOCKS protocol has received widespread support from leading Internet vendors, including Netscape (Nasdaq: NSCP), Microsoft (Nasdaq: MSFT), IBM (NYSE: IBM), Sterling Software (NYSE: SSW), NetManage (Nasdaq: NETM), FTP Software (Nasdaq: FTPS), and Pointcast. NEC USA, Incorporated has been the driving force behind SOCKS with the vision that it would be the most important communication technology for the Internet.

Powerful Security and Management Tools

Aventail MobileVPN and Aventail PartnerVPN are the only products to support all of the popular authentication and encryption methods, such as SSL, DES, TripleDES, CHAP, RC4, MD4, MD5, and RADIUS. Other features include:

* Access Control Tool allows the IS administrator to specify access based on destination, source, application usage, type of encryption and/or authentication, and specific filtering profiles.

* Protocol Filtering blocks specific JAVA, ActiveX or any other application that could demand too much bandwidth or infect the network with a virus.

* Content Filtering blocks out objectionable content that may interfere with employee productivity.

* Traffic Monitor shows real-time inbound and outbound traffic through a graphical interface.

* Reporting and Logging Tool monitors and logs server activity so that reports can easily be produced from any SQL supported database.

* Administration Tool enables IS managers to easily configure the server and add or modify security or management modules.

Product Demonstrations at Networld+Interop

Aventail will be conducting product demonstrations in Booth 1710 at Networld+Interop in Las Vegas from May 6th to 8th.

About Aventail

Aventail Corporation is the leading developer of Virtual Private Network (VPN) solutions. Aventail software allows organizations to build session- layer VPNs so corporations can privately communicate with mobile employees, remote offices, and business partners. Aventail's standards-based products represent the next-level of secure communication by providing strong authentication and encryption, customizable access controls, comprehensive monitoring, logging and reporting capabilities.

Aventail offers four security solutions: Aventail MobileVPN, Aventail PartnerVPN, Aventail Internet

Policy Manager (IPM), and Aventail AutoSOCKS. Aventail MobileVPN enables mobile or remote employees to have secure and managed access into the corporate network. Aventail PartnerVPN allows a company to extend their network to customers, suppliers, remote offices or corporate partners. Aventail IPM allows corporations to control and implement their Internet security policies. Aventail AutoSOCKS enables client TCP/IP applications to securely traverse existing SOCKS-based firewalls and servers.

The company has offices in Seattle, Washington and can be contacted by phone: 888-SOCKSV5 (762-5785), fax: 206-777-5656, or email: info@aventail.com. Aventail's Web address is www.aventail.com.

NOTE: Aventail, MobileVPN, and PartnerVPN are trademarks of Aventail Corporation. All other brands, products, and service names mentioned are trademarks or registered service marks of their respective owners.

SOURCE Aventail Corporation

-0- 05/02/97

/CONTACT: Deanna Leung of Aventail Corporation, 206-777-5617, or deanna@aventail.com; or Jessica Maco of Reed, Revell-Pechar, Inc., 206-462-4777, or jmaco@rrp.com/

CO: Aventail Corporation ST: Washington IN: CPR MLM SU: PDT

DC-KW -- SFF006 -- 9928 05/02/97 08:01 EDT http://www.prnewswire.com

EXHIBIT D

TO MICHAEL FRATTO'S DECLARATION

INFOWORLD, REVIEW OF AUTOSOCKS V2.1
(VOL 19, ISSUE 25 (JUNE 23, 1997) AT PAGE 70)

2 of 2 DOCUMENTS

InfoWorld

June 23, 1997

SECTION: NETWORKING: Product Reviews; Pg. 64d

LENGTH: 1067 words

HEADLINE: Aventail delivers highly secure, flexible VPN solution

BYLINE: By Lai-Han Szeto

BODY:
For secure remote-access needs, Aventail's MobileVPN 2.0 and AutoSocks 2.1 comprise a virtual private network (VPN) software solution that lets you monitor and maintain access to your central site via application-level proxies.

Most VPN products, such as Microsoft's Steelhead technology, Digital's AltaVista Tunnel, and Data Fellow's F-Secure, do not address security issues beyond initial log-ins, tending to be server-centric. Aventail has engineered a solution that is user-centric, taking a more in-depth approach to VPN implementation.

Boasting nearly unmatched interoperability with other security protocols, MobileVPN and AutoSocks succeed as a VPN solution, but not without drawbacks: Unidirectional data flow prohibits broadcasting and remote administration, and the system requires third-party products for specific IP-layer features, such as IPX encapsulation.

High level of security

Aventail has developed its own connectivity protocol, Socks 5, which represents the next step in the evolution of the well-known Socks 4 protocol. The addition of security protocols makes Socks 5 a viable VPN tool and a contender to Microsoft's Point to Point Tunneling Protocol (PPTP). Aventail implements the Socks 5 protocol in the Aventail Server, the engine of its VPN package. Socks 5 is based on directed architecture, as opposed to the tunneled architecture one usually associates with VPN technology.

The server establishes a unidirectional connection with a remote client (AutoSocks) or second host site. A secured user can read, write, and execute to the host Server site according to the user's permission profile, but the host cannot likewise carry out transactions on the user's machine. This

setup prevents an intruder from accessing both sites.

Unlike IP-based protocols such as IP Security Architecture (IPSec), a tunneling protocol currently in the draft stage, Socks 5 compels a user to pass permission requirements once that user passes the system perimeter. Once users traverse firewalls, Socks 5 limits access to specific parts of your host system. The system locks out users from directories and applications according to their permission profile.

Socks 5 performs encryption and authentication at the session layer (Layer 5) of the IP packet, enabling an interoperability unmatched by most of Aventail's competitors.

Aventail products support Challenge Handshake Authentication Protocol, Secure Sockets Layer, and Remote Access Dial-In User Service authentication. In addition, Aventail deploys an open architecture to further enhance the flexibility of its products. Key management is compliant with Public Key Cryptography Standards. Encryption is DES and triple-DES enabled. Recently, Aventail announced Socks 5 capability with the IPSec, PPTP, and Layer 2 Tunneling Protocol security protocols.

Outside authority

MobileVPN represents an achievement in usability. I ran my VPN server on Windows NT 4.0 and used a Windows 95 client unit running AutoSocks.

MobileVPN carries handy administrative tools such as Proxy Chaining and Credential Caching, as well as myriad conventional utilities for alias tables, filtering, and session parameters.

AutoSocks acts as the remote-access agent that intercepts application requests between the client application itself and the WinSock interface. It offers logging and configuration GUIs that resemble a miniature version of MobileVPN, minus the high-level host controls.

I installed both pieces with minimal hassle, minus a certificate authority component. Aventail has no plans to become a certificate authority vendor, leaving the task to third parties, such as VeriSign. Unfortunately, this extra service can cost from $290 to as much as $2,000 per year per server.

Add this to Aventail's tiered licensing scheme, and the bottom line becomes a little steeper than that of most conventional VPN solutions. Whether it is worth the cost depends on the complexity of your security policies.

Fluctuating protocols

Implementing VPNs is not for the faint of heart or pocketbook. Tunneling protocols are maturing even as I write. The key to maintaining a foothold in the market is flexibility. In general, developers are building modular products in anticipation of the Internet Engineering Task Force's final draft of IPSec. It is hard to say what will become of Socks 5 (or Socks 6), but for now it has found a little-explored niche in secured connectivity.

Although MobileVPN and AutoSocks lack bidirectional communication and IP-layer features, their open architecture makes them compatible with multiple standards and provides a high level of security.

Lai-Han Szeto (laihan_szeto@infoworld.com) is a contract analyst at the InfoWorld Test Center.

THE BOTTOM LINE: EXCELLENT

MobileVPN 2.0 and AutoSocks 2.1

This virtual private network (VPN) software combination offers a secure and easy-to-manage remote-access solution.

Pros: Excellent proxy-level management; flexible architecture that complements other VPN and security products.

Cons: Third-party products required for specific IP-layer features such as IPX encapsulation; no broadcasting or remote administration.

Aventail Corp., Seattle; (888) 762-5785 (toll-free), (206) 777-5600; fax: (206) 777-5656; http://www.aventail.com.

Price: $4,999 per server for fewer than 25 connections; $66 per client seat for fewer than 25 seats. (Tiered pricing available.)

Platforms: MobileVPN: Unix, Windows NT; AutoSocks: Unix, Windows 3.x, Windows 95, Windows NT.

LOAD-DATE: June 23, 1997

EXHIBIT E

TO MICHAEL FRATTO'S DECLARATION

FRATTO, "AVENTAIL VPN 2.5: NOT YOUR FATHER'S SOCKS," NETWORK COMPUTING, VOL. 8, NO. 18 (OCTOBER 1, 1997)

Network Computing

Aventail VPN 2.5: not your father's Socks. (virtual private network) (Socks protocol that consists of an application-layer proxy) (Software Review)(Evaluation)

Network Computing | October 1, 1997 | Fratto, Mike

Aventail VPN 2.5, a solution that includes Aventail VPN Server 2.5 and Aventail AutoSOCKS 2.2 client, lets you and your remote users make secure, authenticated connections over IP links. The Socks protocol is an application-layer proxy that relays TCP and User Datagram Protocol (UDP) packets from one network to another based on a server administrator-defined set of rules.

To use Socks, you need the server running on the edge of the network (VPN Server 2.5, in this case) and a client that redirects the connection on the client computer, such as AutoSOCKS 2.2. The Socks 5 Protocol, which Aventail Corp.'s VPN Server and AutoSOCKS use, provides authentication and authorization.

I tested beta versions of Aventail's latest VPN Server and the AutoSOCKS client in Network Computing's lab at Syracuse University and noted their tighter integration with Windows domains, and robust, secure authentication and authorization.

Trying On Socks for Size The VPN Server includes some new installation and management features for network administrators. With previous versions, you had to manually add users to the VPN Server. When a user tried to use the proxy service, the VPN Server authenticated the client and set up a secure session, then the user name/password in the NT Domain controller authenticated the user. Setting up initial access involved manually adding the users to the access control lists.

With version 2.5, you can manage Socks through the Windows NT Domain database. More important, you can add users both individually and in groups. The VPN Server accesses not only the domain in which the server is participating, but other domains that are visible to it.

Adding users is a snap. In the Internet Policy Manager Configuration Tool (the VPN Server management GUI), I added users from both the local server and the domain into a group alias. Each NT Domain is added manually as a resource and displayed as available. By drilling down through the users and groups, I selected individual users and groups and then added them to the selected window. Once the users were selected, I gave the group the name "NWC Domain Users" and closed the box. Wherever I needed to apply a rule to a set of users, I selected NWC Domain Users.

Once groups are created, you can begin applying rules to specific group aliases. However, you must be careful setting the filtering and access rules; when these rules are applied to NT Groups in a group alias, they affect all the users in the NT Group. You can create VPN Server-specific groups by adding individual users to groups in the Internet Policy Manager Configuration Tool.

Oddly enough, Aventail VPN Server does not offer any way to add individual users or NT Groups to the filter rules without first redefining them in the Internet Policy Manager. Here, all user and group management are meshed into one tab in the Internet Policy Manager, conveniently creating one place to make changes. This functionality reduces the probability for conflicting names in the rules, as well as the chance for creating loopholes in users' permissions. With VPN Server 2.5, you simply create a group alias and add users and groups across any number of domains quickly and easily.

Mike Fratto can be reached at mfratto@nwc.com.

Fratto, Mike

http://business.highbeam.com/4113/article-1G1-19805225/aventail-vpn-25-not-your-father-socks

---

EXHIBIT F

TO MICHAEL FRATTO'S DECLARATION

FRATTO, "FOOTLOOSE AND FANCY FREE WITH THREE SOCKS 5-BASED PROXY SERVERS," NETWORK COMPUTING, VOL. 9, ISSUE 11 (JUNE 15, 1998)

Footloose And Fancy Free With Three Socks 5-Based Proxy Servers

Network Computing | June 15, 1998 | Fratto, Mike

The sun is out, the birds are singing and we're lacing up our 'blades for a few hours of skating. Rollerblading is rough on the feet-your toes take a lot of abuse encased in stiff plastic and mylar netting. For relief, well-padded cotton socks do the trick-a simple, functional, utilitarian solution. Just slip 'em on and forget 'em.

Network security solutions would do well to follow this example: Keep it highly functional and simple in design. Add thorough logging and secure management, and it's much easier to tailor your security architecture.

When it comes to network security, firewalls solve many problems-and raise some, too. They keep the bad guys out, but also block legitimate users, or at least make it more difficult to gain access.

You could open holes in your firewall to let authorized users access resources from outside the network, but you also risk intruders sneaking through. Many firewalls have custom clients that will secure traffic over the firewall, but they add management complexity.

Here's where you can take a page from the 'bladers' book, and look for well-chosen socks-Socks 5, or the Authenticated Firewall Traversal protocol. It provides a way to securely allow users access across a firewall, regardless of direction, via a standard protocol. (For more information about Socks, see "Socks Version 5: The UnFirewall" at www.networkcomputing.com/905/905ws1.html.)

Socks 5 proxies sit between users and network servers. Unlike standard network requests, in which users access servers directly, users connected to a Socks 5 server pass (or proxy) requests to the server-end users never are connected directly to servers that are proxied. In this model, the Socks 5 proxy server can enforce user-access control policies, such as filtering destinations based on address and domain name. It also allows for content filtering.

Like all network and security devices, Socks servers require specific features for successful deployment: strong management, thorough logging and robust security. If you leverage network services such as user directories and SNMP management, it's a good indication that you'll be able to install and scale the server with little impact on your network.

For this review, we requested Socks 5 proxy servers that support RFC 1928, Socks Protocol Version 5 and RFC 1929 Username/Password Authentication for Socks 5. While you have the option to implement Socks 5 without authentication, doing so essentially defeats the purpose.

We tested Aventail Corp.'s Aventail VPN Server 2.6, Deerfield.com's WinGate 2.1 and Netscape Communications Corp.'s Proxy Server 3.5. Aventail's VPN Server took top honors with exceptional support for strong authentication and encryption, excellent access control, leverage of network services and a host of other features. Deerfield.com's WinGate and Netscape's Proxy Server both support RFC 1928 and RFC 1929, but neither offers data encryption or strong authentication via Socks 5.

Aventail Corp. Aventail VPN Server 2.6

Aventail bets the farm on Socks 5 security. The company leverages Socks' modular architecture to authenticate and encrypt sessions based on users rather than on IP addresses. Among the products we tested, VPN Server provides the broadest support for user authentication, data encryption and access control. Its relatively simple management platform greatly eases the complexities of configuring and managing the server. On the down side, configuration must be accomplished locally-currently there are no remote management capabilities, and therefore server management suffers. We also found the logging and reporting functions somewhat weak when stacked up against the WinGate and Proxy Server offerings.

Aventail provides a Socks 5 client called AutoSocks 2.6s. AutoSocks wraps around WinSock and "socksifies" connections based on a set of redirection rules, which can be as simple as stating that "any traffic bound for this network should be proxied." You can also configure applications that should not be proxied. The AutoSocks client is set up by the end user or the administrator through configuration files, which can be distributed by e-mail or disk. All of AutoSocks' configuration information is kept on the server, so you can maintain tight control over user access. We

successfully tested the AutoSocks client with all three servers in this review, as well as with NEC Corp.'s freeware SocksCAP client.

VPN Server is the only product in this roundup that offers encryption between the user and the server (this feature requires the AutoSocks client). At the first connection request, the server and client negotiate an SSL (Secure Sockets Layer) 3.0 connection, which is used to secure user authentication and configuration requests in transit. Once a user is authenticated, the server and client can establish other encryption routines, such as DES (Data Encryption Standard), triple-DES or RC4.

Aventail provides several options for user authentication: internal user lists, NT Domains, NDS, RADIUS and Unix /etc/password files. Users are grouped on the server for easier management when you're building access rules. All user management on the VPN Server is accomplished via groups-even if it's a group of one. We set up our server using NT Domains and created a few specific user groups. Adding users and NT groups was as simple as pointing the server to our PDC (Primary Domain Controller). Once the list was acquired, we could move users into assigned groups on the VPN Server.

After creating a set of rules and a security policy (see "Putting On The Socks: How We Tested" on page 116), applying them was a snap. With Aventail you tailor the rules to include "everyone" or custom groups, in which the rules are ordered hierarchically. Of course, you can also enforce asymmetric security by applying the rules to particular interfaces and specifying source networks. We set up our server to allow HTTP traffic from the local network to anywhere, but denied all incoming connections. We also set up Socks to chain connections to a second proxy server. Proxy chaining allows you to control access to your network as well as to your trading partner's network.

Aventail bundles URL filtering from CyberPatrol and SmartFilter. These managed URL filters allow you to block access to Web sites that match certain non-business-related criteria, such as those containing sports, entertainment and adult material. While testing, we found that we were blocked from accessing some adult and sports sites, but we were able to access www.playboy.com. In addition, we were denied access to the AltaVista search engine home page. After conferring with Aventail, we configured our server to perform reverse DNS lookups and set a ".com" domain alias to force IP lookup on commercial sites. Though this change blocked access to Playboy and other sites we intended to filter, it did not lift our AltaVista denial.

Lacking in Tracking Logging is central to effective network and security management. We found Aventail's logging was less appealing than either WinGate or Proxy Server. Most notably, all of Aventail VPN Server's connections are logged to a single file, which can grow significantly large as connections are made, and unlike WinGate, VPN Server has no automatic rollover.

VPN Server logs information either to a text file or to the Windows NT Event Log. We find text-based logging more useful, with greater detail than the Event Log provides. For example, when using the NT Event Log, you must tediously drill down into each event to determine what an actual event means. We prefer to use event numbers, which let you visually filter events quickly, without examining each Event Log entry.

The NT Event Log tracks some information, but filters are not included. To view filter information, you must use text-file logging. Similarly, if you're looking for custom reporting and accounting information, you will have to write custom scripts that parse the log and format the results.

Aventail says its next version of VPN Server, which should be available by the time you read this, will address these shortcomings. It will simultaneously log to the NT Event Log, a static file and a logging tool for real-time detailed logging. Aventail is also expanding the logging functionality to export log data in a format easier for spreadsheet and other reporting tools to accept.

Deerfield.com WinGate 2.1

WinGate is more than a Socks 5 proxy server; in addition to traditional proxy functionality, it offers full application proxy facilities. But Deerfield.com's solution focuses on access control rather than data encryption, and unlike the VPN Server and Proxy Server, it does not allow proxy chaining. WinGate offers excellent logging and incorporates solid packet filtering, but it does not let you filter based on content.

WinGate associates user names with IP addresses only for accounting purposes, either via GateKeeper, WinGate's client utility, or by setting up "Assumed Users." With GateKeeper, users must log into WinGate, where their current IP addresses will be associated with their user names. Using Assumed Users, any traffic arriving from a predefined IP address is automatically associated with a specific user. This approach is OK if you know users' IP addresses beforehand, but in the world of dynamic IP (via DHCP) or an ISP, this isn't a realistic option.

Think Globally, Act Locally We found the security and filtering configuration less straightforward than in the other two products we tested. Access control is set through system policy (filtering) tabs at either the global level or within individual services. Because WinGate is more than a simple proxy service, care should be taken when setting global access-control policies. Global policies will affect all services, such as the Socks 5 proxy, HTTP proxy, SMTP proxy,

etc. For greater control, we preferred to set access control at a service level. You must decide where to set policies-either globally or service by service-or you'll have a tough time getting correct permissions. WinGate expects to act as a proxy, so it looks for a rule to pass incoming data, even if there are conflicting rules. Unfortunately, WinGate offers no easy way to view configured filtering rules.

To control access to Internet sites, WinGate can set up a Ban List, a simplified filtering system that lets you take one of two security positions: "everything not allowed is denied" or "everything not denied is allowed." You configure the Ban List by allowing access to entire sites, regardless of the service the user is attempting to use. For example, we configured our WinGate to deny access to zdnet.com. For finer control, you can set up access permissions via the Advanced tab in the Socks service. For example, we blocked access to ZDnet's FTP site but not its Web site (though in Netscape it shows up as a network error rather than an access denial).

WinGate offers an HTTP proxy service, which lets you simplify management by setting up the Socks 5 service to use the HTTP access policies for HTTP requests received through Socks. Instead of making the same rules in both services or forcing users to configure their browsers for proxy access, leveraging an existing HTTP ensures HTTP access is uniformly applied. Unlike Proxy Server, WinGate doesn't make users configure an HTTP proxy in their browsers-as long as their HTTP traffic is "socksified."

To authenticate any user against WinGate, you must use GateKeeper, which allows WinGate to relate incoming user names with IP addresses. Without GateKeeper, users are not authenticated and appear as guests on WinGate's management station. Unauthenticated users are tracked in a separate guest log.

We found WinGate's logging to be very thorough, offering varying levels of detail that can be set for both users and the service. WinGate keeps two separate logs, allowing users to correlate events between users and service statuses. The service log tracks service-specific information, such as access, configuration changes, start-and-stop status, and errors or service events. User logs track specific access, showing each separate proxy access on its own line. We liked the way WinGate's logging calculates access charges based on byte counts. Charges are calculated per file transfer and can easily be accumulated by a third-party program. The system can be configured to roll over logs at preset intervals for easier file management.

Unlike in Proxy Server, users show up in the management station as they access services through WinGate. Guests do, too, but only as long as they are actually using a service. This allows real-time monitoring of who's on the system. Unfortunately, Socks 5 user sessions are treated as dynamic, when in fact they are static.

Netscape Communications Corp. Netscape Proxy Server 3.5

Like WinGate, Netscape's Proxy Server is more than a Socks 5 server; it lets you proxy HTTP, FTP and Gopher connections. Proxy Server is easy to configure and manage, and offers the most intuitive, straightforward management interface among the products we tested. If you're comfortable building packet-filter rules, for example, configuring Socks 5 filtering will be a snap. Although we found initial connections to be somewhat slower than with either VPN Server or WinGate, we suspect this was due to Web caching, not server load.

Setting up filtering was a breeze. With Proxy Server, you just set the address fields, port numbers and configuration. After setting our "everything not allowed is denied" rule, we were pleased to see Proxy Server's packet-filter rules processed in the order they're entered (from top to bottom); when Proxy Server finds a match, it processes the connection. You have good control over the way connections are made, as packet filtering is hierarchical rather than driven by connection or service type.

Unfortunately, we were unable to configure Proxy Server's URL filters to block access to specific Web sites. Unlike WinGate, which lets you apply an HTTP filter to the proxy, Proxy Server demands that all browsers be set up to proxy HTTP through Proxy Server. This defeats some of the advantages of using Socks 5-you must set up filtering in two places, or have users alter their local configurations to send HTTP to Proxy Server. Using Socks 5, we filtered HTTP tags, such as Java applets and JavaScript, without having to redirect users to the URL filter.

While testing Netscape's filter Rule Manager, we hit a snag attempting to set a filter rule specifying a destination address and port number. The destination address kept showing up in the source port field. We were offered the following solution: Edit the sock5.conf file by hand and reapply the changes to Proxy Server, then restart the Socks server. The rule continued to be swapped back and forth on each successive save and apply. Netscape promises a fix is forthcoming.

Logging for Socks 5 information was extensive, with great detail about who is making connections, where they are connecting and how much data is being transferred. A complete log entry has two lines: the request, which shows date and time of connection and destination, and a line showing how much data was sent and received while processing the connection. Denials of service are also logged, showing who is running up against filter rules and how often.

Mike Fratto can be reached at mfratto@nwc.com.

Putting On The Socks: How We Tested

In this review we were interested in the management and security issues surrounding Socks 5 implementations. Major stumbling blocks to the rollout of protocols such as Socks include the impact and management overhead involved when integrating them into an existing security framework. We particularly kept an eye on how each of these servers leverage existing network services for user management and event logging for auditing and accounting.

For our setup, we configured each solution to allow some internal users out onto the Internet, but only after they had authenticated to the Socks server. And we attempted to restrict HTTP access to non-business-related sites with sports, entertainment and adult content. In addition, we wanted to track usage (per user) and destination information. Access by external users to specific servers on our internal LAN was granted only after users authenticated to the Socks server. External users could access our internal Web server and internal FTP server for downloads only.

We set up each server on a 200-MHz Pentium Pro with 128 MB of RAM and two 3Com Corp. 3C509 10/100 network adapters. The servers straddled our internal and external networks, forcing all traffic to traverse the Socks server. We used similarly configured servers for the Socks proxy chain. A Cisco Systems 4700 router tied the networks together. Meanwhile, we installed Aventail Corp.'s AutoSocks 2.3 and NEC Corp.'s freeware SocksCAP on Windows 95 clients. Each client was configured similarly, and we redirected specific network traffic to the Socks server while directly connecting all other traffic.

June 15, 1998

Fratto, Mike

http://business.highbeam.com/4113/article-1G1-50078512/footloose-and-fancy-free-three-socks-5based-proxy-servers

---

EXHIBIT G

TO MICHAEL FRATTO'S DECLARATION

AVENTAIL CONNECT V3.01/2.51 ADMINISTRATOR'S GUIDE

Aventail

CONNECT

v3.01/v2.51

Administrator's Guide

Windows

Aventail

AVENTAIL CONNECT 3.01/2.51 ADMINISTRATOR'S GUIDE

808 Howell Street, Second Floor
Seattle, WA 98101
USA

http://www.aventail.com/

Printed in the United States of America.

TRADEMARKS AND COPYRIGHTS

Table of Contents

UTILITIES REFERENCE GUIDE

TROUBLESHOOTING

Introduction

Welcome to the Aventail Connect 3.01/2.51 secure Windows client for 16- and 32-bit Windows applications. The client component of the Aventail ExtraNet Center, Aventail Connect is a secure proxy client based on SOCKS 5, the IETF standard for authenticated firewall traversal. Aventail Connect delivers enhanced security and simplifies SOCKS deployment for users and network managers.

Aventail Connect redirects WinSock calls and reroutes them based upon a set of routing directives (rules) assigned when Aventail Connect is configured. (For more information about WinSock, TCP/IP, and general network communications, see “Getting Started.”)

On larger networks, Aventail Connect can address multiple SOCKS 5 servers based on end destination and type of service. This feature enables network administrators to effectively monitor and direct network traffic.

Aventail Connect is a proxy client, but when used with SSL it provides the ability to encrypt inbound or outbound information.

Features of Aventail Connect:

- Aventail Connect supports X.509 client certificates for strong authentication with SSL (when encryption is enabled)
- Automated Customizer utility simplifies client configuration, distribution, and installation
- SSL compression detects low bandwidth connections and compresses encrypted data (when encryption is enabled)
- Secure Extranet Explorer (via **Extranet Neighborhood** icon on desktop) allows users to securely access Windows or SMB hosts over an extranet connection (Windows 95, Windows 98, and Windows NT 4.0 only)
- Supports WinSock 2.0 (LSP) applications in Windows 98, and Windows NT 4.0, and WinSock 1.1 and WinSock 2.0 applications in Windows 95
- Supports WinSock 1.1 applications in Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51
- MultiProxy feature allows you to use a SOCKS server or an HTTP proxy to control outbound access
- Allows the use of port ranges for redirection rules
- Provides integration with SoftID™ and SecurID™ tokens
- Provides automated installation and uninstallation
- Credential cache timeout feature allows administrators to specify when credentials expire
- Provides optional password protection for configuration files
- Supports both SOCKS v4 and SOCKS v5 (RFC 1928 and RFC 1929) standards

- Enables network redirection through successive extranet (SOCKS) servers
- Includes a logging utility to troubleshoot problems with network connections
- Includes a Configuration wizard for simplified step-by-step creation of configuration files
- Allows internal network connections to pass through without interference
- Supports multiple authentication methods including SOCKS v4 identification, username/password, CHAP, CRAM, HTTP Basic (username/password), and SSL 3.0

SEE ALSO: *For more information on the differences between Aventail Connect 3.01 and Aventail Connect 2.51, see "What Does Aventail Connect Do?" in the Administrator's Guide.*

NOTE: *Not all versions of Aventail Connect have encryption enabled.*

.

ABOUT THIS DOCUMENT

This *Administrator's Guide* provides basic information about Aventail Connect. It includes entry-level data for non-technical users, plus installation, setup, and configuration information for network administrators. This information is also available via Aventail Connect Help and the Aventail Web site at **http://www.aventail.com/content/products/docs/**.

DOCUMENT ORGANIZATION

This document is divided into three main sections: *Administrator's Guide*, *Utilities Reference Guide*, and *Troubleshooting*.

The *Administrator's Guide* describes procedures for setting up, installing, and configuring Aventail Connect for individual and multiple networked workstations. It also describes how to create a customized Aventail Connect package for distribution to multiple users.

The *Utilities Reference Guide* describes the Aventail Connect system menu commands and utility programs. It contains detailed information about using the S5 Ping utility and the Logging Tool, and documents the authentication/encryption modules and settings.

The document concludes with *Troubleshooting* and the *Glossary*.

You can also use the Quick Start Card, a short document designed to help you install Aventail Connect to an individual workstation, and the Aventail Connect flowchart, at **http://www.aventail.com/contents/solutions/presentations/quickstart/vpnclient.pdf**.

DOCUMENT CONVENTIONS

The following typographic conventions are used in this document. Exceptions may be made for online material; for instance, italics may be difficult to read online.

Convention	Usage
`Courier font`	Filenames, extensions, directory names, keynames, and pathnames. Command-line commands, options, and portions of syntax that must be typed exactly as shown.
Bold	Dialog box controls (**Edit…** buttons), e-mail addresses (**support@aventail.com**), URLs, (**www.aventail.com**), and IP addresses (**165.121.6.26**).
Italic	Placeholders that represent information the user must insert.

SEE ALSO: *A reference to additional useful information.*

NOTE: *Information the user should be aware of to increase understanding and/or efficiency of the software.*

CAUTION: *An operational item that the user should be aware of to avoid a network policy/software conflict, or lapse, which may create a MINOR security flaw.*

WARNING: *An operational item that the user should be aware of to avoid a network policy/software conflict, or lapse, which may create a SERIOUS security flaw.*

AVENTAIL TECHNICAL SUPPORT

Contact Aventail Technical Support if you have questions about installation, configuration, or general usage of Aventail Connect. Refer to the Aventail Support Web site, at **http://www.aventail.com/index.phtml/support/online_support.phtml**, or the Aventail Knowledge Base, at **http://www.aventail.com/index.phtml?page_id=03110000**, for the latest technical notes and information. Refer to the `readme.txt` documentation for additional information not included in the *Administrator's Guide*.

Aventail Technical Support:
Web site: **http://www.aventail.com/index.phtml/support/index.phtml**
E-mail: **support@aventail.com**
Phone: 206.215.0078
Fax: 206.215.1120

ABOUT AVENTAIL CORPORATION

Aventail Corporation is the leading vendor of extranet software. Its extranet solutions allow organizations to secure their networked communications and manage their employees' access to the Internet. Building an extranet gives organizations the ability to dynamically create a private communication or data channel over the Internet. Aventail's adherence to open security standards simplifies extranet deployment, enables interoperability, and leverages corporations' existing network investments. Its extranet solutions allow companies to extend the reach of their corporate extranets to customers, partners, remote offices, and worldwide employees.

Aventail Corporation
808 Howell Street, Second Floor
Seattle, WA 98101
Phone:206.215.1111
Fax:206.215.1120
http://www.aventail.com/
info@aventail.com

An aventail is a piece of chainmail armor worn around the neck area. In the 14th century, knights wore an aventail to protect themselves while in combat. Today, Aventail continues the tradition of protection by allowing organizations to securely communicate over the Internet.

Administrator's Guide

This section includes procedural and background information on installing Aventail Connect on both single and networked workstations. It includes:

- "Getting Started," with brief explanations of network security and communications
- Definitions of SOCKS and Aventail Connect
- Aventail Connect platform and installation requirements, with an introduction to WinSock 2.0 and LSP architecture
- "Installing Aventail Connect," which includes network diagrams of Aventail ExtraNet Center and SOCKS v4-based server configurations
- Directions on how to create and edit configuration files, and an introduction to the Aventail Customizer

NOTE: *Aventail understands the importance of a flexible, easy-to-use installation process. If you have feedback regarding the Aventail Connect installation procedures, or if there are additional features you want to see implemented, please e-mail comments to* **support@aventail.com**. *Your input is appreciated.*

GETTING STARTED

If you are new to Aventail Connect technology, the following section will help you understand what Aventail Connect is and does, and its relationship to network security in general.

NETWORK SECURITY IN A NUTSHELL

Escalating security threats are forcing companies to seek ways to safeguard their corporate networks and the information they exchange. The first response to these concerns has been the development of security firewalls—software barriers that control the flow of information. But firewalls are not designed to handle complex security issues, such as monitoring network usage, providing private communication over public networks, and enabling remote users to gain secure access to internal network resources.

Enter SOCKS v5, an Internet Engineering Task Force (IETF)-approved security protocol targeted at securely traversing corporate firewalls. SOCKS was originally developed in 1990, and is now maintained by NEC. SOCKS acts as a circuit-level proxy mechanism that manages the flow and security of data traffic to and from your local area network (LAN) or extranet. An application whose traffic

is proxied by SOCKS is considered "socksified." SOCKS is more than a standard security firewall. Other features:

- Client Authentication: (SOCKS v5 only) Authentication allows network managers to provide selected user access to internal and external areas of a network.
- Traffic Encryption: (SOCKS v5 only) Encryption ensures that network traffic is private and secure.
- UDP Support: (SOCKS v5 only) User Datagram Protocol (UDP) traffic has traditionally been difficult to proxy, with the exception of SOCKS v5.
- Aventail Connect supports X.509 client certificates within SSL: Includes a Certificate wizard for generating and processing client certificate requests.
- Cross-Platform Support: Unlike many other security solutions, SOCKS can be used on various platforms, such as Windows NT, Windows 95, Windows 98, and various forms of UNIX.

NOTE: *Not all versions of Aventail Connect include the SSL module for encryption.*

WHAT IS AVENTAIL CONNECT?

Aventail Connect is the client component of the Aventail ExtraNet Center. Aventail Connect works with the Aventail ExtraNet Server, the SOCKS 5 server component of the Aventail ExtraNet Center. You can use Aventail Connect as a simple proxy client for managed outbound access, and for secure inbound access.

Aventail Connect automates the "socksification" of Transmission Control Protocol/Internet Protocol (TCP/IP) client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol. When you run Aventail Connect on your system, it automatically routes appropriate network traffic from a WinSock application to an extranet (SOCKS) server, or through successive servers. (WinSock is a Windows component that connects a Windows PC to the Internet using TCP/IP.) The SOCKS server then sends the traffic to the Internet or the external network. Network administrators can define a set of rules that route this traffic.

Aventail Connect is designed to run transparently on each workstation, without adding overhead to the user's desktop. In most cases, users will interact with Aventail Connect only when it prompts them to enter authentication credentials for a connection to a secure extranet (SOCKS) server. Users may also occasionally need to start and exit Aventail Connect, although network administrators often configure it to run automatically at startup. Aventail Connect does not require administrators to manually establish an encrypted tunnel; Aventail Connect can establish an encrypted tunnel automatically.

To understand Aventail Connect, you first need to understand a few basics of TCP/IP communications.

TCP/IP COMMUNICATIONS

Windows TCP/IP networking applications (such as telnet, e-mail, Web browsers, and ftp) use WinSock (Windows Sockets) to gain access to networks or the Internet. WinSock is the core component of TCP/IP under Windows, and is the interface that most Windows applications use to communicate to TCP/IP.

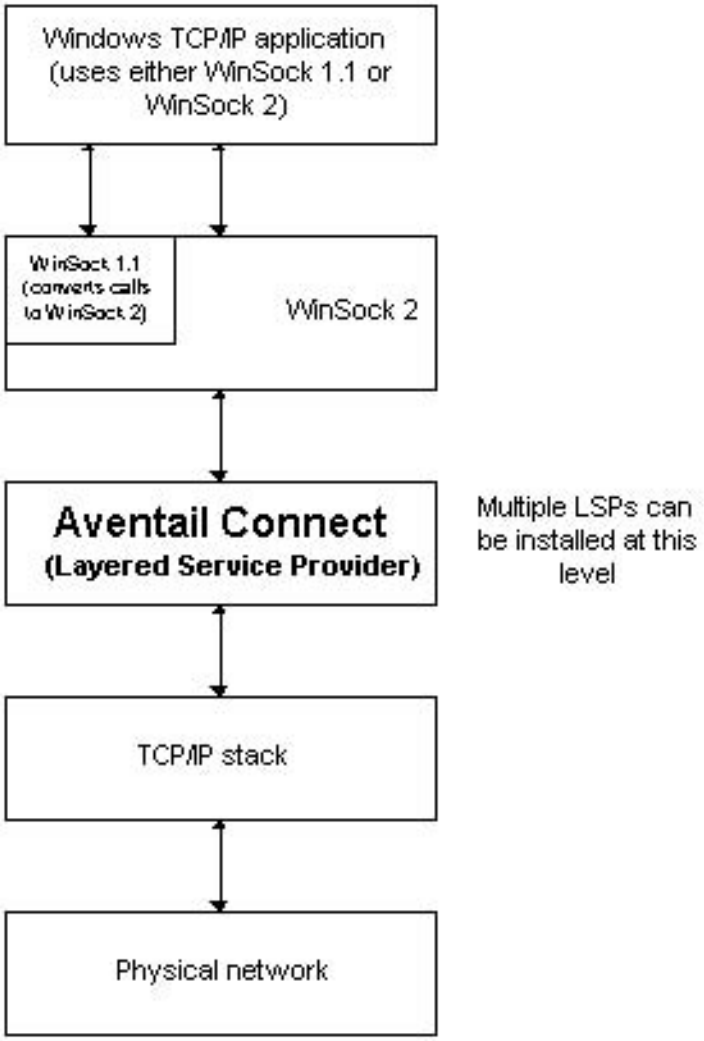
WINSOCK CONNECTION TO A REMOTE HOST

Via WinSock, an application goes through the following steps to connect to a remote host on the Internet or corporate extranet:

1. The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address. If the application already knows the IP address, this step is skipped.
2. The application requests a connection to the specified remote host. This causes the underlying stack to begin the TCP handshake, when two computers initiate communication with each other. When the handshake is complete, the application is notified that the connection is established, and data can then be transmitted and received.
3. The application sends and receives data.

WHAT DOES AVENTAIL CONNECT DO?

Aventail Connect slips in between WinSock and the underlying TCP/IP stack. (See diagram below.) As an application that sits between WinSock and the TCP/IP stack, Aventail Connect 3.01 is a Layered Service Provider (LSP). Aventail Connect can change data (compressing it or encrypting it, for example) before routing it to the TCP/IP stack for transport over the network. The routing is determined by the rules described in the configuration file.



Windows TCP/IP applications and Aventail Connect have no direct contact with one another; instead, each of them communicates through WinSock. Multiple LSP applications can be installed at the LSP level.

NOTE: *Aventail Connect does not alter or replace WinSock or any other core TCP/IP components (files) provided by the operating system.*

When the Aventail Connect LSP receives a connection request, it determines whether or not the connection needs to be redirected (to an Aventail ExtraNet Server) and/or encrypted (in SSL). When redirection and encryption are not necessary, Aventail Connect simply passes the connection request, and any subsequent transmitted data, to the TCP/IP stack.

The two most popular versions of WinSock are version 1.1 and version 2. Aventail Connect 3.01, like all LSPs, requires WinSock 2.0; WinSock 1.1 does not support LSPs. WinSock 2.0 includes backward-compatibility with all WinSock 1.1 applications. Not every platform supports WinSock 2.0 and its LSP structure.

- Windows 98 and Windows NT 4.0 support WinSock 2.0 natively. (Windows NT 4.0 requires Service Pack 3 or above, available from Microsoft.)
- Windows 95 supports WinSock 1.1. Windows 95 can also support WinSock 2.0, but you must install a Microsoft patch to add support for WinSock 2.0.
- Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51 do not support WinSock 2.0; they support only WinSock 1.1.

For those platforms that do not support WinSock 2.0 and LSP applications, Aventail includes Aventail Connect 2.51 on the Aventail Connect 3.01/2.51 CD. Aventail Connect 2.51 was designed for operating systems that support only WinSock 1.1. For Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51 operating systems, setup will install Aventail Connect 2.51. If you are working on a Windows 95 operating system, setup will detect whether you have installed the Microsoft Windows 95 WinSock 2.0 Update. If setup detects the Microsoft update, which upgrades Windows 95 to support WinSock 2.0, setup will install Aventail Connect 3.01. If setup does not detect the Microsoft update, it will install Aventail Connect 2.51.

The Aventail Connect 2.51 user interface is identical to that of Aventail Connect 3.01; however, Aventail Connect 3.01 includes MultiProxy (see "Multiple Firewall Traversal"). Aventail Connect 2.51 does not include MultiProxy.

In the future, more Windows applications may require WinSock 2.0.

During installation, setup determines which version of Aventail Connect to install. On WinSock 2.0 platforms, Aventail Connect 3.01 is installed. On WinSock 1.1 platforms, Aventail Connect 2.51 is installed. The following table shows how setup determines which version of Aventail Connect to install.

Operating System	WinSock Support	Aventail Connect Version Installed
Windows 98, Windows NT 4.0	WinSock 2.0	Aventail Connect 3.01
Windows 95	With Microsoft patch: WinSock 2.0	Aventail Connect 3.01
	Without Microsoft patch: WinSock 1.1	Aventail Connect 2.51
Windows 3.1, Windows for Workgroups 3.11, Windows NT 3.51	WinSock 1.1	Aventail Connect 2.51

You can create custom packages that include one or both versions of Aventail Connect (3.01 and 2.51) Setup will determine which version to install on each workstation. (For more information, see "Customizer.")

WINDOWS 95 AND WINSOCK

The Microsoft Windows 95 WinSock 2.0 Update upgrades WinSock 1.1 to WinSock 2.0 in Windows 95. This patch (filename `w95ws2setup.exe`) is available from the Microsoft Web site, at **http://www.microsoft.com/windows/downloads/contents/Updates/W95Sockets2/default.asp**. Unless you need specific Aventail Connect 3.01 features, Aventail recommends that you do not upgrade from WinSock 1.1 to WinSock 2.0. If you do not upgrade to WinSock 2.0, Aventail Connect 2.51 will be installed.

If you do need to install the Microsoft Windows 95 WinSock 2.0 Update, follow the instructions provided by Microsoft. Reboot your computer after upgrading, prior to installing Aventail Connect.

HOW DOES AVENTAIL CONNECT WORK?

The following three steps are identical to standard WinSock communications steps described above; however, nested inside them are additional actions and options introduced by Aventail Connect.

1. The application does a DNS lookup to convert the hostname to an IP address. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following:
   - If the hostname matches a local domain string or does not match a redirection rule, Aventail Connect passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack performs the lookup as if Aventail Connect were not running.
   - If the destination hostname matches a redirection rule domain name (i.e., the host is part of a domain we are proxying traffic to) then Aventail Connect creates a false DNS entry (HOSTENT) that it can recognize

during the connection request. Aventail Connect will forward the hostname to the extranet (SOCKS) server in step 2 and the SOCKS server performs the hostname resolution.

- If the DNS proxy option is enabled and the domain cannot be looked up directly, Aventail Connect creates a fake DNS entry that it can recognize later, and returns this to the calling application. The false entry tells Aventail Connect that the DNS lookup must be proxied, and that it must send the fully qualified hostname to the SOCKS server with the SOCKS connection request.

2. The application requests a connection to the remote host. This causes the underlying stack to begin the TCP handshake. When the handshake is complete, the application is notified that the connection is established and that data may now be transmitted and received. Aventail Connect does the following:

   a. Aventail Connect checks the connection request.

      - If the request contains a false DNS entry (from step 1), it will be proxied.
      - If the request contains a routable IP address, and the rules in the configuration file say it must be proxied, Aventail Connect will call WinSock to begin the TCP handshake with the server designated in the configuration file.
      - If the request contains a real IP address and the configuration file rule says that it does not need to be proxied, the request will be passed to WinSock and processing jumps to step 3 as if Aventail Connect were not running.

   b. When the connection is completed, Aventail Connect begins the SOCKS negotiation.

      - It sends the list of authentication methods enabled in the configuration file.
      - Once the server selects an authentication method, Aventail Connect executes the specified authentication processing.
      - It then sends the proxy request to the extranet (SOCKS) server. This includes either the IP address provided by the application or the DNS entry (hostname) provided in step 1.

   c. When the SOCKS negotiation is completed, Aventail Connect notifies the application. From the application's point of view, the entire SOCKS negotiation, including the authentication negotiation, is merely the TCP handshaking.

3 The application transmits and receives data.

   If an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server on behalf of the application. If data is being returned, Aventail Connect decrypts it so that the application sees cleartext data.

AVENTAIL CONNECT PLATFORM REQUIREMENTS

The following table lists the minimum system requirements for each of the platforms that Aventail Connect supports.

Platform	Processor	RAM	Extranet (SOCKS) Server
Windows 98; Windows NT 4.0 (requires Microsoft Service Pack 3 or above)	x86-based or Pentium personal computer	16 MB	Network-accessible SOCKS v4 or v5 compliant server
Windows 95; Windows NT 3.51	x86-based or Pentium personal computer	8 MB	Network-accessible SOCKS v4 or v5 compliant server
Windows 3.1; Windows for Workgroups 3.11	x86-based or Pentium personal computer	4 MB	Network-accessible SOCKS v4 or v5 compliant server

Aventail Connect 3.01 runs on the following operating systems:

- Windows 98
- Windows NT 4.0 (with Service Pack 3 or above, available from Microsoft)
- Windows 95, with the Microsoft WinSock 2.0 update (To install Aventail Connect 3.01, you must upgrade Windows 95 with the Microsoft WinSock 2.0 update prior to Aventail Connect installation and setup. If you do not install the Microsoft patch, Aventail Connect 2.51 will be installed. For more information, see "What Does Aventail Connect Do?".)

Aventail Connect 2.51 runs on the following operating systems:

- Windows 3.1
- Windows for Workgroups 3.11
- Windows NT 3.51
- Windows 95, without the Microsoft WinSock 2.0 update (If you do not upgrade Windows 95 with the Microsoft WinSock 2.0 update, Aventail Connect 2.51 will be installed. For more information, see "What Does Aventail Connect Do?".)

NOTE: *A WinSock-compatible 16- or 32-bit TCP/IP application must be installed and configured prior to running Aventail Connect. This can be the Microsoft-provided TCP/IP stack or a third-party TCP/IP stack.*

INTERFACE FEATURES

The following table lists the interface features for each platform. Each of these features is discussed in greater detail later in the *Administrator's Guide*.

Platform	Start Aventail Connect	Display System Menu	Open Secure Extranet Explorer	View Program Icon	Hide Program Icon
Windows 95, Windows 98, Windows NT 4.0	Start\Programs \Aventail Connect menu	Right-click **Aventail Connect** icon in system tray	Double-click **Extranet Neighborhood** icon on desktop	In system tray	Not available
Windows 3.1, Windows for Workgroups 3.11, Windows NT 3.51	**Aventail Connect** icon in Aventail Connect program group window	Click **Aventail Connect** icon in Aventail Connect program group window	Not available	Minimized on desktop	Configure during setup

INSTALLATION SOURCE MEDIA

Regardless of platform, Aventail Connect can be delivered on CD or as a network-delivered, self-extracting archive file.

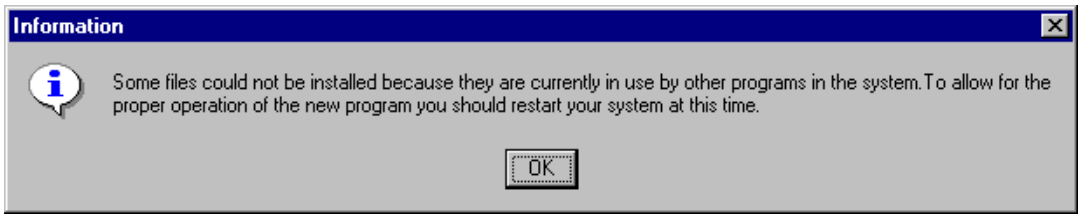
- **CD**: The CD contains the Aventail Connect setup program, `setup.exe`. The setup program allows for an administrative setup. It also contains the *Administrator's Guide* and the *User's Guide* in the `\docs` directory, formatted for Adobe® Acrobat Reader.
- **Network-delivered Source Media**: The network-delivered source media is a self-extracting archive containing the required disk/directory structure within the archive file. The executable automatically extracts the Aventail Connect installation files and initiates setup. The archive filename will be similar to `as30s.exe`. This archive, or package, will also be available on the CD (located in the Utilities directory) to be used with the Customizer application. For more information, see the "Customizer" section.

INSTALLING AVENTAIL CONNECT

After your Aventail ExtraNet Server is set up, Aventail Connect can be installed to a single workstation or to multiple networked workstations. In either case, you must perform an initial installation of the software and create one or more configuration (`.cfg`) files. This procedure is described under "Individual Installation." Once the initial installation is complete, you can then install to a series of networked computers using the instructions and information described under "Network Installation."

NOTE: *To install or uninstall Aventail Connect on Windows NT machines, you must have administrative privileges on the machine (but not necessarily on the domain).*

If you are upgrading from an earlier version of Aventail Connect (VPN Client or AutoSOCKS), the following message may appear on your screen if you install a custom setup package using Aventail Customizer. This is not an error message. If this message appears, click **OK** and reboot your computer.



CONFIGURATION FILES

Integral to the initial installation of Aventail Connect is deciding how SOCKS traffic will be redirected through the network. Network redirection rules (used to determine if and how SOCKS redirection will occur) are defined in the Aventail Connect configuration (`.cfg`) file. Configuration files are initially created at the end of the installation process; however, you can add, edit, and remove configuration files at any time using the Config Tool (in Windows 95, Windows 98, or Windows NT 4.0 via the **Aventail** icon in the system tray on the taskbar; in Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51 via the Aventail Program Group). The process of creating one or more configuration files is described under "Configuring Aventail Connect."

If you are installing Aventail Connect on multiple networked workstations, refer to "Network Installation" to determine the best method for maintaining and distributing configuration files. You can then proceed through the initial installation. The Installation Wizard will guide you through the steps, culminating with the option to create a configuration file.

CUSTOMIZED CONFIGURATION AND DISTRIBUTION

The Aventail Customizer is a utility that allows network administrators to customize Aventail Connect installation packages for distribution to multiple client work-

stations. Giving network administrators control over how setup packages are configured eliminates the need for end users to make installation and setup decisions at their workstations. The installation package is a self-extracting executable file. You can customize this file by adding license file, configuration file, or setup information for different authentication and encryption policies to meet various client-access needs of individuals or workgroups. You can customize configurations for multiple users and then distribute the package, providing easy access, download, and installation for users. You can reconfigure the Aventail Connect installation package anytime your network topology or security profiles change.

For more information about the Aventail Customizer, see the "Customizer" section.

INDIVIDUAL INSTALLATION

Before running setup, close all open Windows applications.

To install Aventail Connect

1. Installation procedures vary slightly, depending on which media source you use:
   - If you are installing directly from CD-ROM, run `setup.exe` from the Aventail Connect directory.
   - If you are installing from a network-delivered self-extracting archive, simply execute the archive file. This will extract the installation files and automatically launch the setup program.

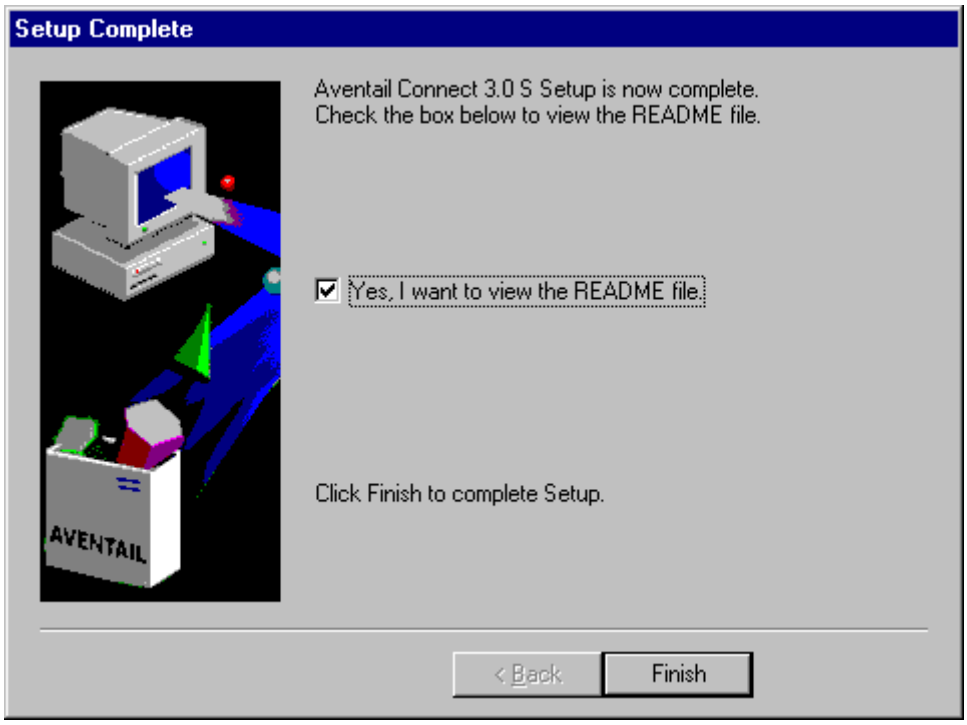
   The Aventail Connect installation wizard then guides you through the process of installing the Aventail Connect application.

NOTE: *You will be asked during the installation procedure if you would like Aventail Connect to be run automatically during startup. In most cases, you will select the **yes** option. Exceptions to this can be determined by the network administrator.*

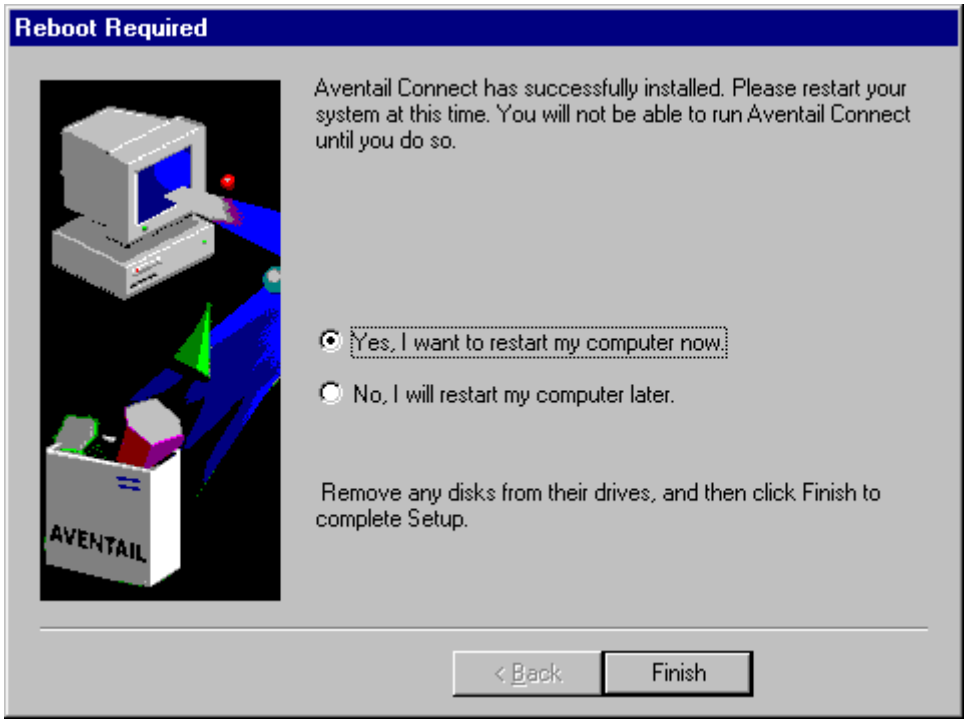
2. At the end of the setup program you can select the **Yes, I want to view the README file** box in the **Setup Complete** dialog box. This opens the `readme.txt` file, which contains the latest information on Aventail Connect.

-OR-

Simply click **Finish** to complete the setup program.



3. The setup program will then ask you if you want to restart your machine now or later.



4. After restarting your PC, Aventail Connect will launch automatically if, during installation, you chose "yes" when asked if Aventail Connect should be added to your startup directory. (If you selected the **no** option during installation, start Aventail Connect from the **Programs** menu.)
5. Aventail Connect will ask you if you want to run the configuration wizard.

   If you click **Yes,** then the configuration wizard will launch to help you create a new configuration file.

   If you click **No**, then Aventail Connect will ask you to select a configuration file.
6. After creating or selecting a configuration file, Aventail Connect will finish its installation procedure.

To uninstall Aventail Connect

The procedure to uninstall (remove) Aventail Connect varies depending on whether you are running a 16- or 32-bit Windows operating system.

- To uninstall Aventail Connect from Windows 95, Windows 98, and Windows NT 4.0, double-click **Add/Remove Programs** in the **Control Panel** window, click **Aventail Connect** on the list of programs on the **Install/Uninstall** tab, and click **Add/Remove**.
- To uninstall Aventail Connect on Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51, use the **Uninstall** icon in the Aventail Connect program group.

NETWORK INSTALLATION

In general, the process of installing Aventail Connect to multiple networked workstations involves selecting a file server to use, creating a staging area for the Aventail Connect software, and placing the Aventail Connect package in a shared network directory or other publicly accessible location. Additional options include adding a default configuration file, license file, certificate and roots files, and SEEHosts files. You must place Aventail Connect files on a network drive that can be accessed as a mapped drive or, for Microsoft networks, via a UNC path name (`\\computer_name\share_name\Connect`).

An executable archive file (with a filename similar to `as30s.exe`) automatically extracts the Aventail Connect installation files and initiates setup. This archive, or package, is located in the Utilities directory of the CD and can be used in conjunction with the Customizer application. The package can also be manually configured to suit your network specifications. The default package includes all of the core Aventail Connect files, but does not include the custom network information.

NETWORKED CONFIGURATION FILE SETUP

There are a number of ways to set up networked client configuration files. These are the most common:

- Client configuration file shared on a mapped network drive (Novell or Microsoft)
- Client configuration file distributed via a Microsoft UNC path and file-name
- Local client configuration file common for all users, but distributed via an Aventail Connect package

ADMINISTRATOR-MAINTAINED SHARED CONFIGURATION FILES

This is the most desirable configuration method—multiple workstations sharing one or more administrator-maintained configuration files located in a common directory. The network administrator maintains the configuration file, and the administrator can quickly adapt any changes to network topology through a single configuration file. For example:

- A single networked (usually read-only) configuration file is shared by more than one client workstation. This method is appropriate when multiple workstations share identical traffic routing rules.
- Multiple configuration files are shared by multiple workstations. This option is useful when you have workstations organized into functional groups (engineering, marketing, accounting, etc.) with group-specific redirection rules.

SHARED CONFIGURATION FILE DISTRIBUTION

Shared configuration files can be easily distributed and, if necessary, updated via the network. Aventail recommends that you test all configuration files before distribution.

To distribute a shared configuration file

There are three methods for distributing shared configuration files.

- Copy the file to a Microsoft or Novell network drive accessible by all users. Make sure that users configure Aventail Connect to load the configuration file located on the mapped drive. You can preconfigure this information for users from a package install.

-OR-

- Copy the file to a Microsoft Windows workstation supporting UNC-sharing for file resources. (Both the 16- and 32-bit versions of Aventail Connect support specification of the configuration file using the Microsoft UNCs.) This distribution method has all the benefits of placing the file on a network mapped drive with the added bonus of convenience—users do not need to actually map the network drive.

-OR-

- Create a shared configuration file to be installed on workstations during the standard Aventail Connect installation/upgrade process. (You can build the configuration file into a package with Customizer.) Whenever Aventail Connect is installed or updated, it will automatically copy the shared configuration file to the user's workstation and set Aventail Connect to use it.

You can create and distribute shared configuration files with the Aventail Customizer. This automated wizard allows you to create custom setup packages for multiple users and then store the packages in a networked directory, providing easy access, download, and installation for users. You can include multiple local and/or remote configuration files.

ADMINISTRATIVE SETUP

There are two ways to install Aventail Connect: from the setup program (`setup.exe`), or from a setup package that you create using the Aventail Customizer. The setup program (`setup.exe`) allows you to manually install Aventail Connect. With the Aventail Connect setup package, you can select options that will customize setup based on your unique network environment. You can customize the setup package through the Customizer Editor or the Customizer Wizard. The Customizer *Editor* is a dialog box that allows you to manually enter or modify information about your custom installation package. The Customizer *Wizard* walks you through each step of creating a custom installation package. Aside from the user-interface differences, the Customizer Wizard and the Customizer Editor are identical. You can use both the Customizer Wizard and the Customizer Editor to create or modify a setup package. For example, you can create a package using the Customizer Wizard, then modify it with the Customizer Editor.

CUSTOMIZER

The Aventail Customizer simplifies and customizes the installation and setup process. Network administrators can reconfigure the self-extracting executable installation package (included in the Customizer directory of the distribution CD) to meet the various client-access needs of individuals or workgroups. Customizer offers a centralized approach to network configuration; network administrators may select the unattended setup mode, which eliminates the need for individual users to answer any setup configuration questions. Specifying unattended mode will cause the setup program to automatically install using default values for any options not explicitly specified.

The setup program (`setup.exe`) allows users to select any available setup options during installation of Aventail Connect. Customizer modifies the setup control file of a custom package; this file controls all of the settings within the setup package, before users receive the setup package. With a customized package, users will receive an installation package based on the administrator's defined settings.

As Customizer allows you to select various options to suit your setup and installation needs, the size of the setup package will vary, depending on which options you select. If size of the setup package is a concern, select setup options carefully to keep the package size manageable.

The Aventail Connect CD includes both versions of Aventail Connect (3.01 and 2.51). You can create custom packages that include one or both versions of Aventail Connect; setup will determine which version to install on each workstation. (For more information, see "What Does Aventail Connect Do?")

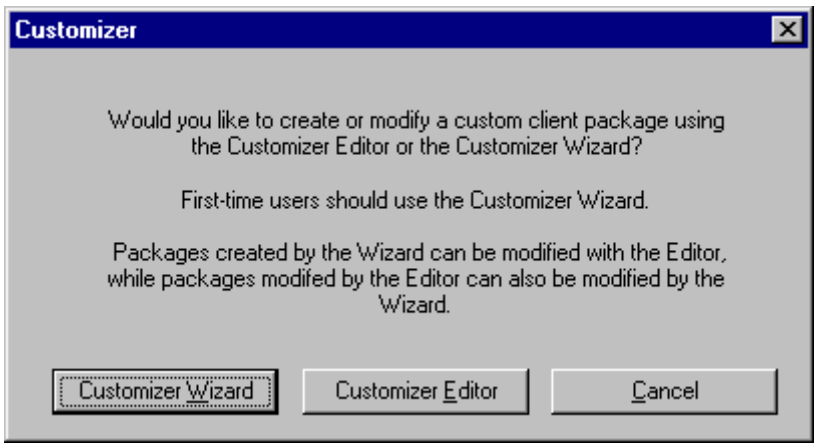
Aventail Connect requires a valid Aventail license file (`aventail.alf`) and one or more configuration (`.cfg`) files in order to function properly. Before installing Aventail Connect, make sure that users have these files. If users do not have a valid license file and/or configuration file(s), Aventail recommends that you include them in the installation package.

RUNNING CUSTOMIZER

The Customizer and the Aventail Connect installation package are included in the Customizer directory on the Aventail Connect CD. Before running Customizer, you must copy Customizer from the Aventail Connect CD to the local drive. You must also modify the Customizer attributes so it is not read-only.

To run Customizer, double-click the **Customizer** icon in the Customizer directory. To run Customizer from your hard drive, copy the Customizer and Aventail Connect directories into a common folder on the hard drive.

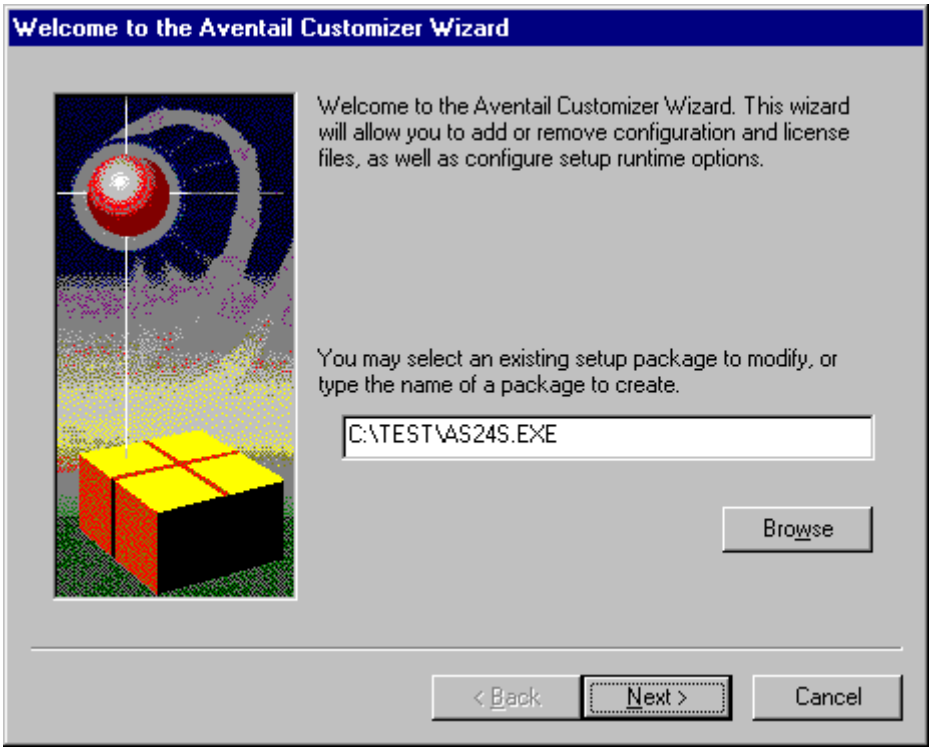
When you run Customizer, you will be prompted to select either the Customizer Wizard or the Customizer Editor.



- **Customizer Wizard:** This automated wizard walks you through the process of creating a new installation package or modifying an existing package. If you are unsure about which method to use, Aventail recommends that you use the Customizer Wizard.
- **Customizer Editor:** The Customizer Editor is a dialog box that allows you to manually enter information about the package you are creating or modifying.

CUSTOMIZER WIZARD

If you are using the Customizer Wizard to create a new setup package or modify an existing package, the Customizer Wizard will display a **Welcome...** screen, and will prompt you to enter the pathname of the package that you will be creating or modifying.



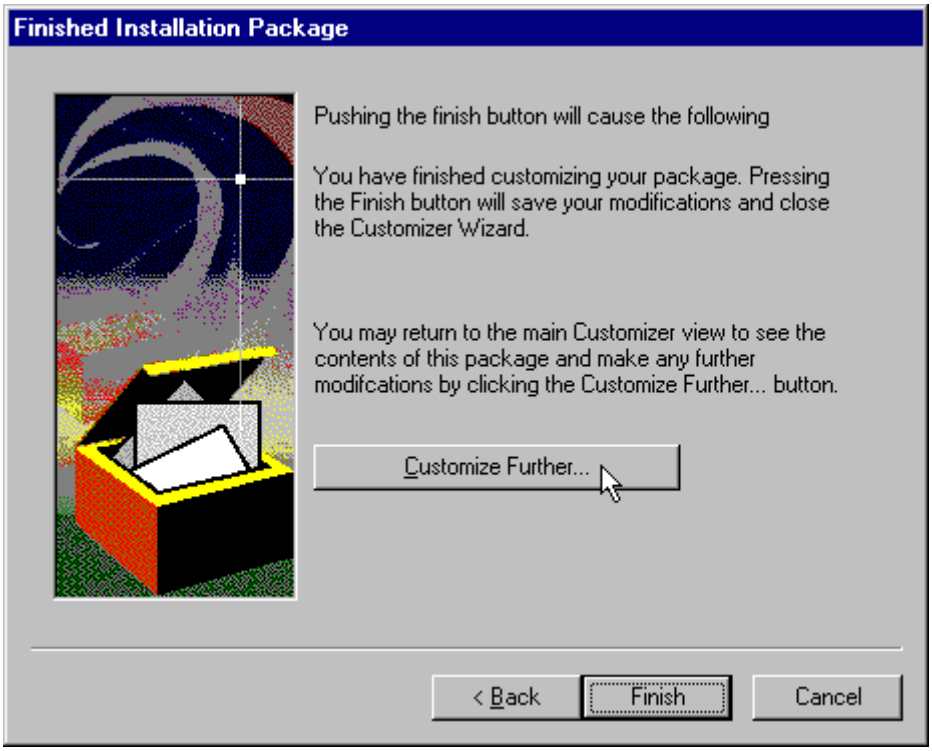
After you have specified the pathname of the package, the Customizer Wizard will prompt you to:

- Specify which platform(s) to support
- Add a license file, or leave an existing license file in the package
- Add or remove configuration files
- Select X.509 certificate files
- Select an extranet hosts (SEEHosts) file
- Specify a custom destination directory
- Specify whether or not to put program icons in a custom folder
- Enter command-line switches
- Specify whether or not to run setup in unattended mode
- Specify whether or not to add Aventail Connect to the startup directory
- Select any, all, or none of the following Aventail Connect components:
  - Extranet Neighborhood (Secure Extranet Explorer)
  - Configuration Tools (Config Tool and Configuration File command)
  - Diagnostic Tools (Logging Tool and S5 Ping)
  - Certificate Tools

  - Install 32-bit support only (on Windows NT 3.51)
- Select any, all, or none of the following authentication modules:
  - SSL (Secure Sockets Layer)
  - CRAM (Challenge Response Authentication Method)
  - CHAP (Challenge Handshake Authentication Protocol)
  - UNPW (Username/Password)
  - SOCKS 4
  - HTTP Basic (username/password)
- Specify whether or not to run a command after Setup

All of the features listed above are optional.

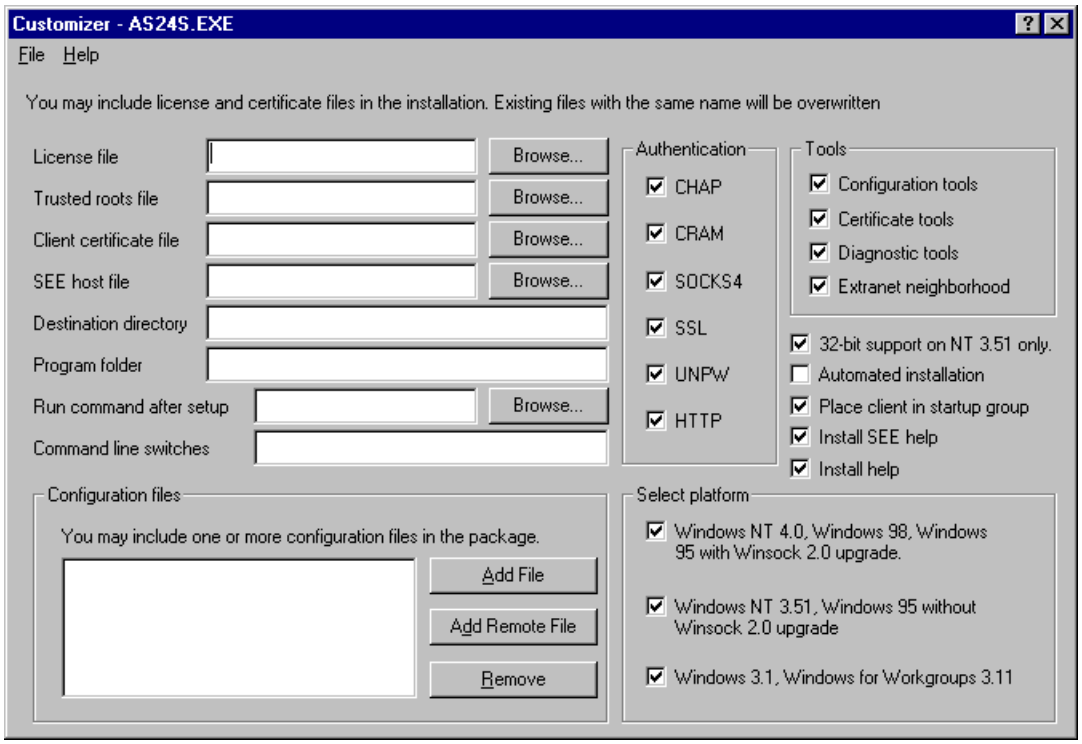
After entering or modifying the package information, the **Finished Installation Package** dialog box appears.



Clicking **Finish** saves your specifications and closes the Customizer Wizard. Clicking **Customize Further** allows you to view the **Customizer Editor** dialog box, where you can manually edit any of the information about your custom installation package.

CUSTOMIZER EDITOR

If you select the Customizer Editor as your tool to create a new setup package or modify an existing package, the **Customizer Editor** dialog box will appear. In this dialog box, you can manually enter or modify information about your custom installation package.



NOTE: *To view a list of tips on creating custom setup packages, click* ***Tips*** *on the* ***Help*** *menu in the* ***Customizer Editor*** *dialog box.*

After entering or editing your setup package information in the Customizer Editor, click **Save** (or **Save As**) on the **File** menu to save your changes. To close the **Customizer Editor** window, click **Exit** on the **File** menu.

The options in the Customizer Editor are identical to the options in the Customizer Wizard. These options are explained in the following paragraphs and tables.

Option	Settings	Default Setting
Pathname	Enter pathname	None
License file	Enter name of Aventail license file (must use `aventail.alf`)	None
Trusted roots file	Enter name of trusted roots file	None
Client certificate file	Enter name of file that contains certificate	None
Extranet (SEE) Hosts File	Enter name of extranet (SEE) hosts file	None
Destination directory	Enter name of destination directory	None
Program folder	Enter name of program folder	None
Run command after setup	Enter command to be run after setup	None
Command line switches	Enter command line switches	None
Configuration Files	Enter name(s) of local and/or remote configuration file(s) that Aventail Connect will use	None
Authentication Modules	SSL, CRAM, CHAP, UNPW, S4, or HTTP Basic	All
Tools	Configuration tools, Certificate tools, Diagnostic tools, or Extranet Neighborhood	All
32-bit support only, on Windows NT 3.51	Yes/No	Yes
Unattended Setup Mode/Automated installation	Yes/No	No
Add to Startup Directory	Yes/No	Yes
Install SEE help	Yes/No	Yes
Install help	Yes/No	Yes
Select platform	Windows NT 4.0, Windows 98, Windows 95 with WinSock 2.0 upgrade, Windows 95 without WinSock 2.0 upgrade, Windows NT 3.51, Windows 3.1, or Windows for Workgroups 3.11	All

The setup package options are discussed below.

- **Specify path for installation:** You can specify a path for installation, or you can select the default path. The default path for 32-bit operating systems is `c:\Program Files\Aventail\Connect`.

  For 16-bit-only operating systems, the default is `c:\Connect`.

NOTE: *If you are upgrading from an earlier version of Aventail Connect, Aventail Connect will install to the same directory that the earlier version of it was installed to.*

- **Platforms:** You must specify which operating systems need to be supported in the setup package. Aventail Connect 3.01 supports Windows 95 (with the Microsoft WinSock 2.0 update), Windows 98, and Windows NT 4.0 (with Service Pack 3 or above, available from Microsoft). Aventail Connect 2.51 supports Windows 95 (without the Microsoft WinSock 2.0 update), Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51. For more information, refer to "What Does Aventail Connect Do?"
- **Trusted Roots File and Certificate File:** If you want to use server certificates, you must include the trusted roots file that contains those certificates. If you want to use client certificates, you must specify the location of the file that contains the X.509 certificate.
- **Running Setup in Unattended Mode:** Unattended setup mode simplifies distribution of numerous client configuration files. The network administrator specifies all settings before users receive the Aventail Connect setup package file. No end-user input is required because the network administrator has already selected the setup options; users simply open the package file, which will automatically install on their workstations.

NOTE: *Specifying unattended setup mode will cause the setup package to automatically install using default values for any options not explicitly specified.*

- **Adding Aventail Connect to the Startup Directory:** If you choose to add Aventail Connect to the startup directory, Aventail Connect will automatically start when Windows starts.
- **Select Tools:** Aventail Connect gives you the option to install various components, including Extranet Neighborhood/Secure Extranet Explorer (SEE), configuration tools (Config Tool and Configuration File command), or diagnostic tools (Logging Tool and S5 Ping). The default value is to install all package components.
  - **Secure Extranet Explorer:** Secure Extranet Explorer (SEE) allows you to view your Extranet Neighborhood, which is accessed through

the **Extranet Neighborhood** icon on your desktop. Extranet Neighborhood functions much like Network Neighborhood, except Extranet Neighborhood allows you to browse, copy, move, and delete files from secured remote computers via an extranet, while Network Neighborhood displays all computers on your local network.

- **Config Tool:** The Aventail Connect Config Tool allows you to create configuration files that determine how network requests will be routed and which authentication protocols will be enabled. You can add, remove, or edit configuration files at any time. If necessary, you can create several configuration files for different users or user groups. If you want to prohibit end users from editing configuration files, do not include the Config Tool in the installation package.
- **S5 Ping:** S5 Ping allows you to use the ping and traceroute utilities, two diagnostic tools. The ping utility checks for network connectivity between two hosts and returns information about the quality of the connection. The traceroute utility checks for network connectivity by displaying information about routers between two hosts; it displays information for each hop.
- **Logging Tool:** The Logging Tool is a diagnostic utility that traces Aventail Connect activity. When running a trace, the Logging Tool displays errors, warnings, and information as Aventail Connect generates them. If necessary, the message list can be saved to a log file that can be used by Aventail Technical Support in troubleshooting technical problems. These traces are also useful when running Aventail Connect for the first time to ensure that network traffic is being routed appropriately.

- **Select Authentication Modules:** Aventail Connect lets you select any, all, or none of the following authentication modules: SSL, CRAM, CHAP, UN/PW, SOCKS v4, or HTTP Basic (username/password).
  - **Secure Sockets Layer:** Secure Sockets Layer (SSL) is a session-layer protocol for securing connections in a general, protocol-independent fashion.

NOTE: *In versions of Aventail Connect that do not include encryption, the Secure Sockets Layer (SSL) authentication module is not included.*

  - **CRAM:** The Challenge Response Authentication Method (CRAM) sends your username and password as clear text between extranet (SOCKS) servers, but encrypted between servers that support CRAM. Typically, CRAM subauthenticates within SSL, which provides both encryption and credential caching options.

NOTE: *In versions of Aventail Connect that do not include encryption, the CRAM authentication module is not included.*

- **CHAP:** The Challenge Handshake Authentication Protocol (CHAP) sends your username and password encrypted across the network to the destination server.
- **Username/Password:** The RFC 1928 (Internet standards document) Username/Password (UNPW) authentication protocol sends your username and password in clear text across the network to the destination server.
- **SOCKS 4 Identification:** Aventail Connect includes backward compatibility for the SOCKS 4 protocol. SOCKS 4 does not support password authentication, so only your username is sent, unencrypted, to the SOCKS server along with your connection request.
- **HTTP Basic (Username/Password):** The HTTP Basic authentication module enables username/password authentication against HTTP proxies that implement the RFC 2068 HTTP Basic authentication protocol.

NOTE: *Not all versions of Aventail Connect have encryption enabled.*

- **Configuration Files:** Aventail Connect needs at least one configuration (`.cfg`) file in order to function properly. The configuration file contains all of the authentication and traffic routing instructions that you specify. You can include one or more configuration files in the setup package; however, each configuration file must have a different name. If you include only one configuration file in a setup package, Aventail Connect will automatically use that configuration file. If, however, you include multiple configuration files, Aventail Connect will prompt users to select a configuration file at startup.

  You can include local configuration files, remote configuration files, or a combination of both. Local configuration files are included in the setup package and are installed on users' machines. If you include remote configuration files, pointers to those files are included in the package; the remote configuration files remain in their original location on the network, where they can be shared by multiple users.

  If your setup package does not already contain a configuration file, you can add a configuration file to the package. If your setup package contains one or more configuration files, you can remove or replace any or all of the existing configuration files, or you can leave them, unchanged, in the package. If you are upgrading from an earlier version of Aventail Connect, you may not need a new configuration file.
- **License Files:** Aventail Connect requires a valid license file in order to function properly. If your setup package contains a license file, you can remove or replace the existing license file, or you can leave it, unchanged, in the package. If your setup package does not contain a

license file, you can add one to the package. You must use the packaged Aventail license file, `aventail.alf`.

CAUTION: *Aventail Connect 3.01 and 2.51 use a different license (`.alf`) file format than earlier versions of Aventail Connect (VPN Client or AutoSOCKS) did. If you are upgrading from an earlier version of Aventail Connect (2.42 or earlier), you must include a new Aventail license file.*

- **Extranet (SEE) Hosts Files:** Secure Extranet Explorer (SEE) allows you to browse remote computers using Extranet Neighborhood. SEE requires a hosts file that specifies which Windows domains, WINS servers, and other computers are available in Extranet Neighborhood. The extranet hosts (SEEHosts) file is contained in the setup package. If you install SEE, this file is placed in the target directory. If you do not include a hosts file in the setup package, Aventail Connect will automatically create a hosts file on users' machines the first time they open Extranet Neighborhood. (Available only in Windows 95, Windows 98, and Windows NT 4.0.)

CREATING, LOADING, AND SAVING PACKAGES

You can create, load, or save custom setup packages through either the Customizer Editor or the Customizer Wizard.

To create a new package

There are two ways to create a new custom setup package:

- In the **Customizer Editor** window, select **File | New**.

-OR-

- Type the filename of a new package in the first window of the Customizer Wizard and click **Next**.

To load a package

There are two ways to load an existing setup package:

- In the **Customizer Editor** window, select **File | Open**, and then enter the filename of the package you want to load

-OR-

- Type the filename of the package in the first window of the Customizer Wizard and then click **Next**.

When you load a package, Customizer reads the setup control file to determine what information the package contains. Customizer uses this information to populate the **Customizer Editor** window. Customizer also reads the configuration file(s) into memory; configuration files are stored in memory to facilitate adding them to and removing them from a package.

To save changes to a package

There are two ways to save changes to a setup package:

- After making the desired changes to the package, click **Save** (or **Save As**) on the **File** menu in the **Customizer Editor** window

-OR-

- Click **Save Package** in the final window of the Customizer Wizard.

CUSTOMIZER TIPS

The following tips will help you use the Aventail Customizer more efficiently.

- **Keep the package size small:** You can control the size of your custom setup packages by selecting components carefully. To keep the package as small as possible, include only the options that you need, and support only the platforms (e.g., Windows 98, Windows NT 4.0, etc.) that your users work with. You may find that creating two separate, smaller packages is preferable to creating one larger package. For example, you might create one package that supports Windows 98 and Windows NT 4.0 operating systems, and another separate package that supports Windows 3.1 and Windows 95 operating systems.
- **Use descriptive package names:** When naming setup packages, assign descriptive, recognizable names that will help users identify the setup packages.
- **Select components carefully:** If you include the Config Tool in the package, users will be able to view and modify the settings in the Config Tool. Aventail recommends that, in most cases, you do not include the Config Tool in your custom setup package(s). Excluding options such as the Config Tool will eliminate users' ability to modify your settings, and will keep the package size smaller. However, the S5 Ping and Logging Tool utilities are useful diagnostic tools, and Aventail recommends including these options in the setup package whenever possible.
- **Install Aventail Connect 2.51 on Windows 95:** By default, Windows 95 does not support WinSock 2.0, but you can upgrade it to support WinSock 2.0 with a Microsoft patch. (The patch, `w95ws2setup.exe`, is available from Microsoft, at **http://www.microsoft.com/windows/downloads/contents/Updates/W95Sockets2/default.asp**. However, this procedure adds an extra step to the installation and setup process. Unless users need the Multi-Proxy feature, which is available only in Aventail Connect 3.01, Aventail recommends that you install Aventail Connect 2.51 rather than 3.01 on machines running the Windows 95 operating system.
- **Include a hosts file:** If you install Secure Extranet Explorer (SEE) without also installing a corresponding hosts file, SEE will automatically create a hosts file the first time that users open SEE. If you want to control which hosts users can view, Aventail recommends that you include a hosts file in the custom setup package.

- **Include a license file:** Aventail Connect requires a valid license file (`aventail.alf`) to function properly. Aventail Connect 3.01/2.51 uses a different license file than earlier versions of Aventail Connect (VPN Client or AutoSOCKS) did. If you are upgrading from an earlier version of Aventail Connect (2.42 or earlier), you must use the new Aventail license file, `aventail.alf`. Including this license file in the custom setup package is a simple way to install the license file.
- **Test each custom package:** Aventail recommends that you thoroughly test each custom setup package before distribution to users.

CONFIGURING AVENTAIL CONNECT

Create configuration files using the Config Tool or the Configuration wizard. You can launch either during the Aventail Connect installation or any time you want to add, modify, or remove a configuration file.

The steps for creating a new configuration file are:

1. Define the extranet (SOCKS) servers
2. Define the destinations (networks and hosts)
3. Specify redirection rules
4. Enter Local Name Resolution information (optional)
5. Manage authentication modules
6. Enable password protection (optional)

These procedures are described in the text below.

To launch the Config Tool

The Config Tool opens with the **Open Aventail Connect Configuration File** dialog box. After you select a configuration file or enter a new file name, the main window of the Config Tool appears.

1. Select the **Yes, I want to configure Aventail Connect** box in the **Setup Complete** dialog box (during installation).

-OR-

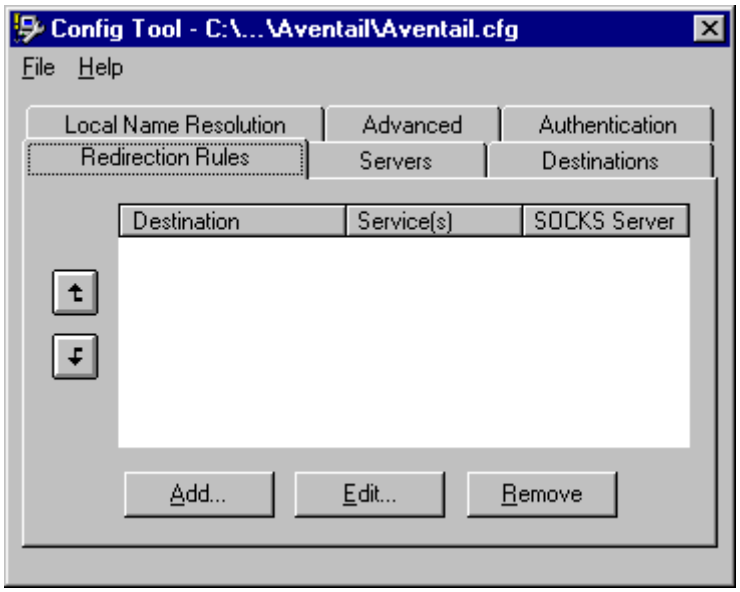
Right-click the **Aventail Connect** icon in the taskbar and click **Config Tool** (Windows 95, Windows 98, or Windows NT 4.0 programs menu option), or double-click the **Config Tool** icon in the Aventail Connect program group (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51).

2. If you are creating a new configuration file, enter a name for the configuration file

-OR-

Select the configuration file you want to open.

This displays the main window of the Config Tool.



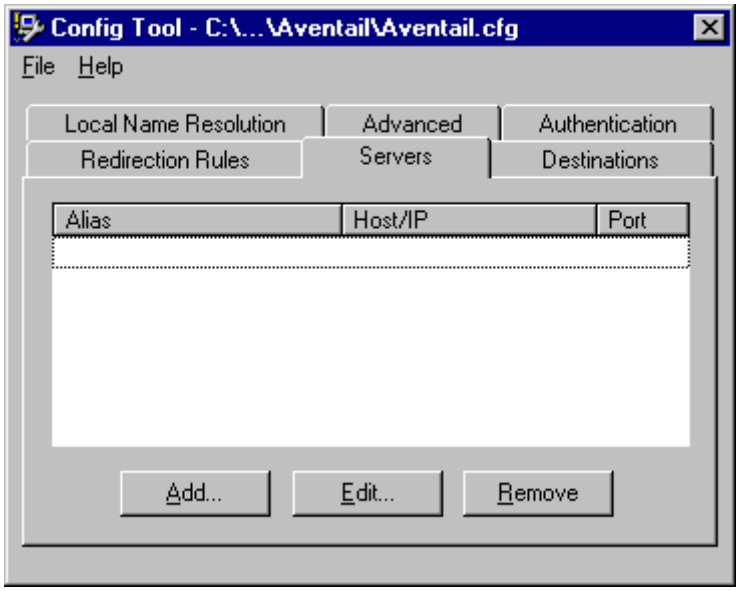
The **Config Tool** window contains six tabs. The properties defined on each tab can be edited at any time.

Tab	Function
Servers	Defines the extranet (SOCKS) server(s).
Destinations	Specifies the network and host addresses that will be routed through the SOCKS server(s).
Redirection Rules	Specifies how network requests are routed to the SOCKS server(s).
Local Name Resolution	(Optional) Specifies hostnames that will be resolved by the local workstation.
Authentication	Enables, disables, and sets properties for the authentication modules.
Advanced	Enables/disables extranet (SOCKS) traffic through successive SOCKS servers, enables/disables the Application Exclusion/ Inclusion List, secures selected applications, and sets credential cache timeouts.

You can change the width of any of the fields on the tabs by positioning the cursor over the dividing line between the fields on the field bar. When the cursor changes to a double-headed arrow, click and drag to resize the field.

DEFINE AN EXTRANET (SOCKS) SERVER

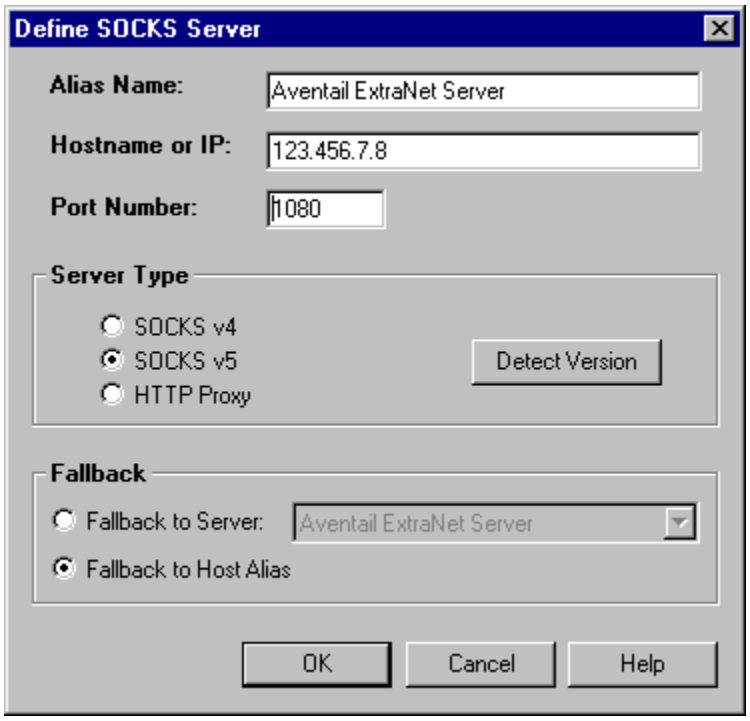
SOCKS servers are defined on the **Servers** tab in the Config Tool.



Field	Definition
Alias	The name you assign to the server.
Host/IP	The hostname or IP address of the server.
Port	The port on which the server is listening.

To add an extranet (SOCKS) server

1. On the **Servers** tab, click **Add…**. The **Define SOCKS Server** dialog box appears.



Field	Definition	
Alias Name	User-friendly alias for extranet (SOCKS) server.	
Hostname or IP	Actual hostname or full numeric IP address for SOCKS server.	
Port Number	SOCKS server port. Default value is 1080.	
Server Type	SOCKS v4	SOCKS Version 4.0.
	SOCKS v5	SOCKS Version 5.0.
	HTTP Proxy	HTTP proxy server.
	Detect Version	Detect SOCKS version number.
Fallback	Fallback to Server:	SOCKS server alias for redundant server.
	Fallback to Host Alias	Use DNS records for redundancy.

2. In the **Alias Name** box, type a user-friendly alias for the extranet (SOCKS) server. Do not leave this box blank.

3. In the **Hostname or IP address box**, type the actual hostname of the SOCKS server or its IP address.
4. In the **Port Number** box, type the extranet server's port number. If you do not enter a value, it defaults to the standard SOCKS port 1080.
5. Under "Server Type," select the version of SOCKS supported by the server. If you are unsure of the version, click **Detect Version**.

NOTE: *Typically you should select **SOCKS v5** unless the server can support only SOCKS v4.*

6. Under "Fallback," directly specify an extranet server for redundancy or use the Host Alias to specify an extranet server.

To edit extranet (SOCKS) server properties

- Select the extranet server you want to edit and click **Edit**.

  The **Define SOCKS server** dialog box appears with the selected server data filled in. Edit any of the information.

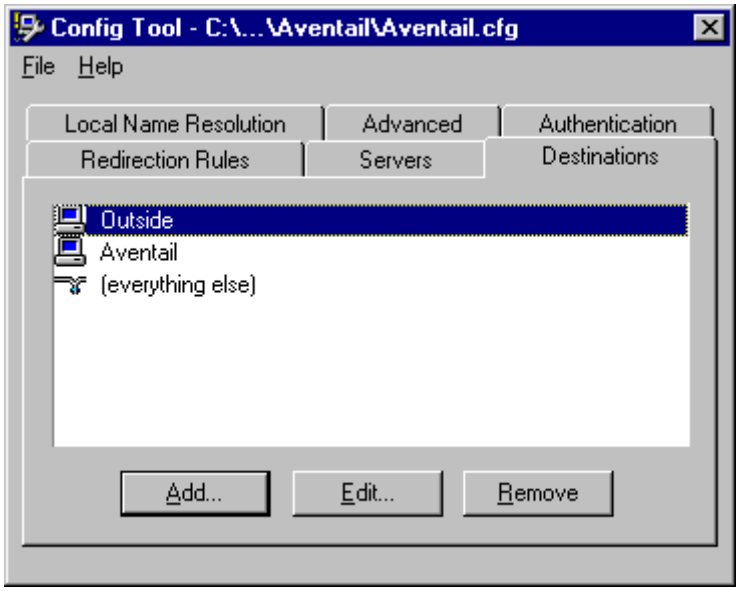
To remove an extranet (SOCKS) server definition

- Select the extranet server you want to remove and click **Remove**.

  The server is deleted from the list. Corresponding redirection rules will also be deleted.

DEFINE A DESTINATION

Destinations are defined on the **Destinations** tab in the Config Tool.



After one or more extranet servers are defined, add destinations to be routed through them.

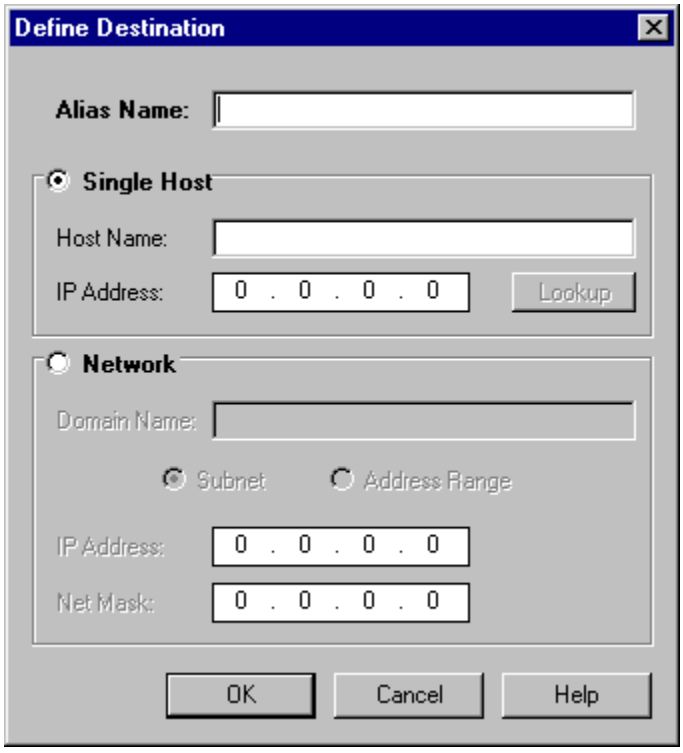
NOTE: *The "(everything else)" destination refers to all network and host addresses not otherwise defined. You cannot delete or modify "(everything else)".*

To add a destination

In the **Define Destination** dialog box, you can define subnets, individual host computers, or IP address ranges, and set up rules about redirecting some or none of the IP traffic to these defined destinations.

1. On the **Destinations** tab, click **Add…**.

The **Define Destination** dialog box appears.



Field	Definition	
Alias Name	User-friendly alias for destination network or host	
Single Host	A specific destination computer	
	Hostname	Actual name of destination network or host
	IP Address	Full numeric IP address
	Lookup	Look up IP address
Network	One or more computers in a network	
	Domain Name	Domain of the network
	Subnet	IP address and netmask address
	Address Range	Beginning and ending IP addresses From Starting IP address To Ending IP address

2. In the **Alias Name** box, type a user-friendly alias for the destination network or host.
3. Select either the **Single Host** or **Network** option:
   - Under "Single host," type the actual name of the host system and/or its full, numeric IP address. If you do not know the host's IP address, click **Lookup** to search for it.

-OR-

   - Under "Network," type the domain of the network and then select either **Address Range** or **Subnet**.

Use	To
Address Range	Enter a starting and ending IP address. All addresses between the two will be included as part of the destination. For example, a starting IP address of 192.1.1.0 and an ending IP address of 192.1.1.255 would include all hosts of the 192.1.1.x subnet.
Subnet	Enter an IP address and a netmask address. This is another way to specify a group of destinations. For example, an IP address of 192.1.1.0 and a net mask of 255.255.255.0 defines the same address range as shown above.

To edit a destination

- Select the destination you want to edit and click **Edit…**.

  The **Define Destination** dialog box appears with the selected destination data filled in. Edit the data as necessary.

To remove a destination

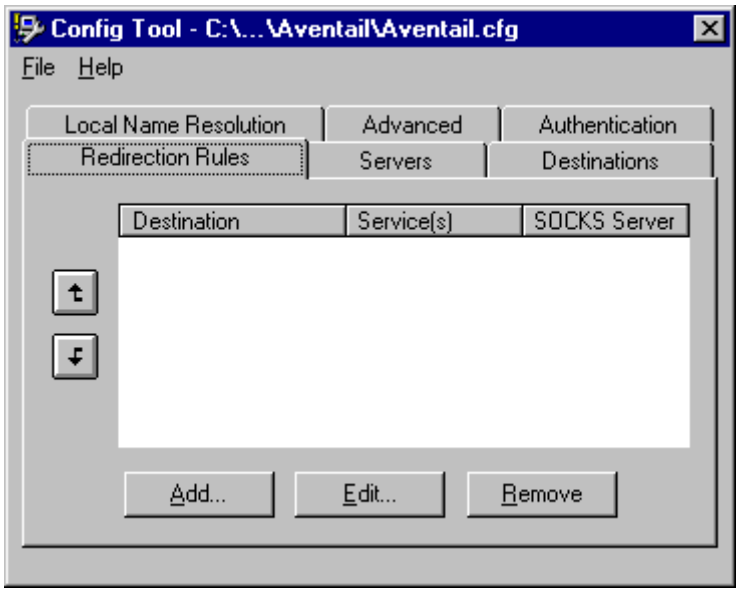
- Select the destination you want to remove and click **Remove**.

  The destination is deleted from the list. The corresponding redirection rules will also be deleted.

ENTER REDIRECTION RULES

Once servers and destinations are defined, you can specify how you want Aventail Connect to redirect (or deny) access to various hosts and services such as e-mail, FTP, and HTTP.

Redirection rules are specified on the **Redirection Rules** tab in the Config Tool.



.

Field	Definition
Destination	Destinations defined on the **Destinations** tab
Service	Type of Internet traffic
Proxy Redirection	Specify how to redirect traffic

You can change the width of any of the three fields by moving the cursor to the dividing line between the fields on the field bar. When the cursor changes to a double-headed arrow, click and drag to resize the field.

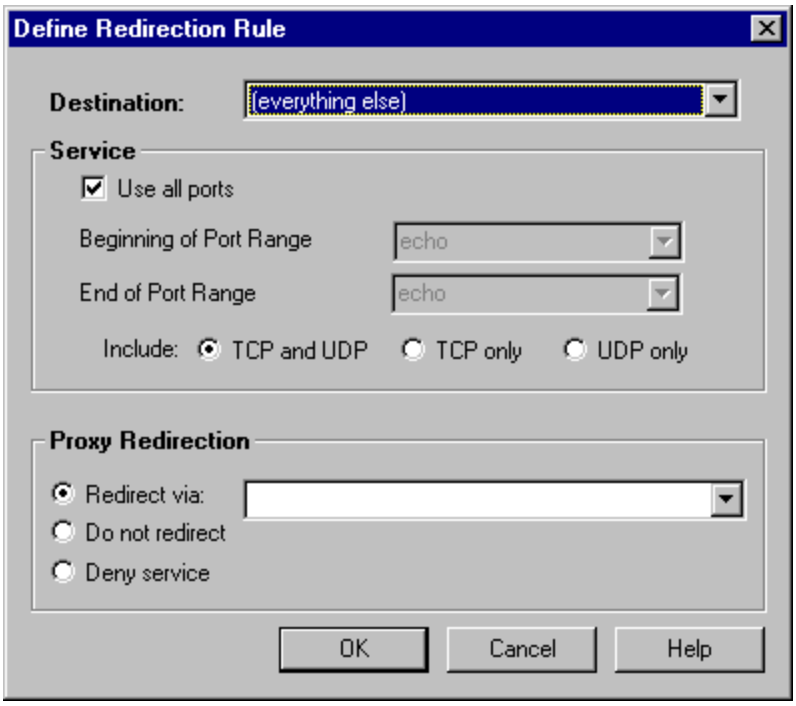
To add a redirection rule

As you add destinations, use the arrow buttons to prioritize them. List the most specific rules first and the general rules last.

NOTE: *Aventail Connect scans the list from the top down and uses the first matching rule it finds, so it is important to list the most specific rules first.*

1. On the **Redirection Rules** tab, click **Add**.

The **Define Redirection Rule** dialog box appears.



Field	Definition	
Destination	Host or server destination for message traffic.	
Service	Type of Internet traffic	
	Use all ports	Apply the defined rule to all ports.
	Beginning of port range	Apply the defined rule to this range of ports.
	End of port range	
	TCP and UDP	Apply the defined rule to both TCP and UDP traffic.
	TCP only	Apply the defined rule to TCP traffic only.
	UDP only	Apply the defined rule to UDP traffic only.
Proxy Redirection	Specify how to redirect traffic.	
	Redirect via	Redirect all traffic through the extranet server selected from the list.
	Do not redirect	Route traffic directly to the specified destination without being redirected through SOCKS.
	Deny service	Deny access to the specified destination. The network connection is blocked locally instead of at the server level.

2. Select a destination from the **Destination** list.
3. Under "Service," select the **Use all ports** box to apply the rule to all services. Otherwise, select a range of ports. To select a single port, enter that port number in both the **Beginning of port range** and **End of port range** boxes.
4. Under "Proxy Redirection," select one of three redirection options.

CAUTION: *If you select **Deny Service** and the user has edit control of the configuration file, the option can be circumvented by quitting Aventail Connect or by changing the option in the dialog box.*

To edit a redirection rule

- Select the redirection rule you want to edit and click **Edit…**.

  The **Define Redirection Rule** dialog box appears with the selected data filled in. Edit any of the information.

To remove a redirection rule

- Select the redirection rule you want to remove and click **Remove**.

  The redirection rule is deleted from the dialog box.

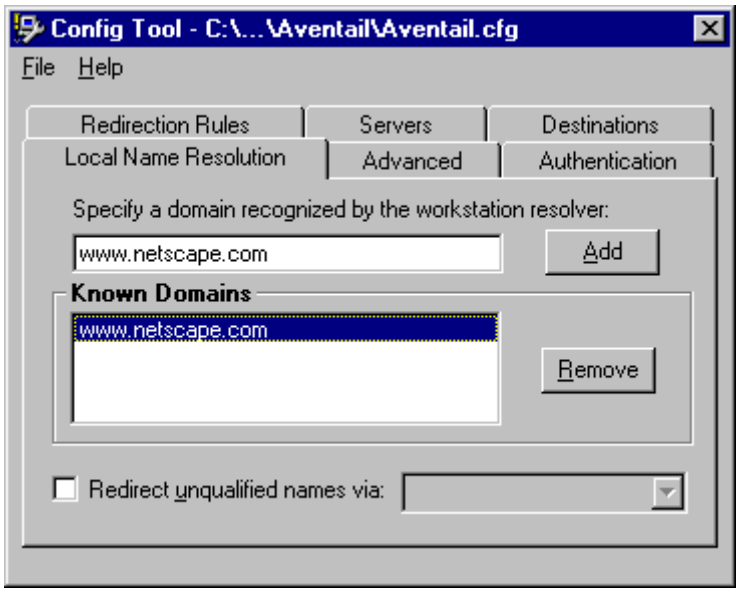
DEFINE LOCAL NAME RESOLUTION

Local Name Resolution instructs Aventail Connect to resolve hostnames locally without needing to venture on to the Internet. This optional feature offers you another level of control over how Aventail Connect performs name resolution.

The local workstation resolver is the name resolution component of the local TCP/IP stack. This feature acts as a shortcut; hostnames matching the strings defined in the **Local Name Resolution** dialog box are passed to the local resolver for name resolution instead of being proxied through the SOCKS v5 server.

For example, if **aventail.com** is added to the "Defined Strings" list, then a workstation attempting to connect to **www.aventail.com** would perform hostname resolution using the local TCP/IP stack.

Local Name Resolution is specified on the **Local Name Resolution** tab in the Config Tool.



Field	Definition
Specify a domain recognized by the workstation resolver	New domain name
Known Domains	List of domain names that can be resolved locally
Redirect unqualified names via	Pass through unqualified hostnames to the local resolver

To add a local domain name

- On the **Local Name Resolution** tab, type the new name in the **Specify a domain** box and click **Add…**.

  The new name is moved into the **Known Domains** box. It is now active.

To remove a local domain name

- Select the domain name you want to remove from the **Known Domains** box and click **Remove**.

  The domain name is removed from the list.

MANAGE AUTHENTICATION MODULES

SOCKS v5 servers often require user authentication before allowing access. Aventail Connect authentication modules display dialog boxes that prompt users to enter username and password information as well as other authentication credentials.

NOTE: *Not all versions of Aventail Connect have encryption enabled.*

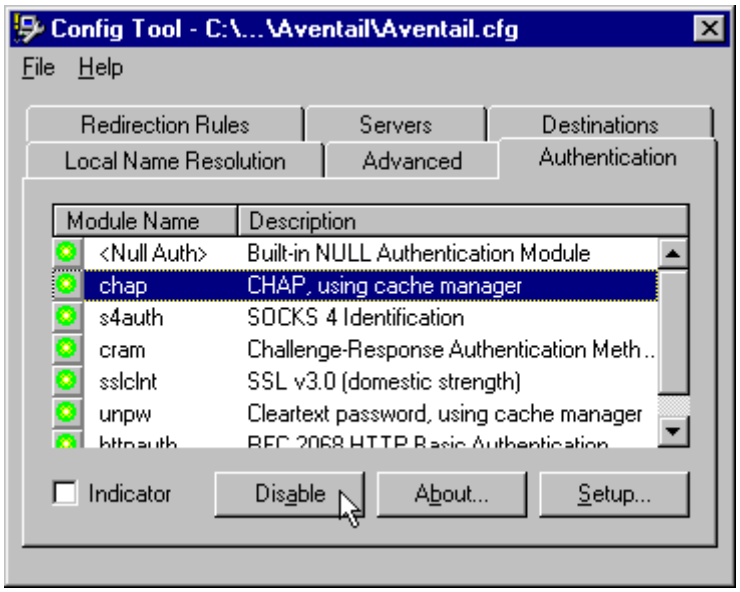
The current Aventail Connect authentication modules are SOCKS v4 Identification, Username/Password, Challenge Handshake Authentication Protocol (CHAP), Challenge Response Authentication Method (CRAM), Secure Sockets Layer (SSL), and HTTP Basic (username/password). Each of these authentication modules supports an Aventail Connect feature known as credential caching. Credential caching retains your authentication credentials once the extranet server has accepted them. Using credential caching, you can enter your credentials for an extranet server once per Aventail Connect session, rather than once for each individual connection (a tedious task for applications such as WWW browsers).

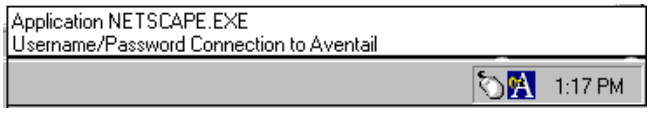
Aventail Connect can cache authentication credentials in memory, based on the option you select in the **Authentication** dialog box. Memory caching stores the credentials for the current session only. When you restart Aventail Connect or

Windows, the memory cache is flushed and you must reenter your credentials as prompted.

SEE ALSO: *For additional information on credential caching, see "Credential Cache Timeouts" in the "Advanced Tab Options" section of this Administrator's Guide.*

Authentication modules are managed and configured through the **Authentication** tab in the Config Tool.



Field	Definition
Module Name	The name of the authentication module on disk. `<Null Auth>` indicates that no authentication module will be used.
Description	The description of the authentication method.
Indicator	Check this option to display network traffic passing through a selected authentication/encryption module. See the example below (for Windows 95, Windows 98, and Windows NT 4.0). 

Each authentication module includes its own module-specific configuration. To view or edit a module's configuration, select the module from the list on the

Authentication tab and then click **Setup**. An options dialog box for the specific module will appear.

Enable and disable authentication modules with the **Disable/Enable** button. By default, the modules are all enabled. The green button next to the module name indicates an active module. This is the default state of all the modules. The green button changes to red when you disable the module.

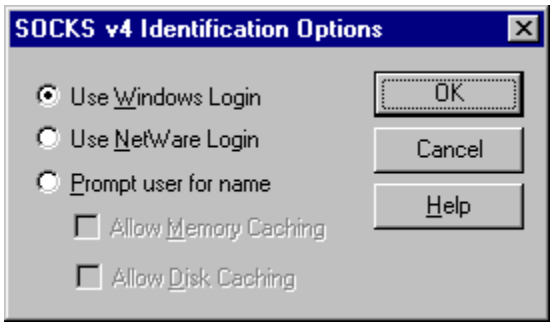
To configure the SOCKS 4 Identification module

Aventail Connect includes backward compatibility for the SOCKS 4 protocol. SOCKS 4 does not support password authentication, so only your username is sent unencrypted to the extranet (SOCKS) server along with your connection request.

Your username is determined by entries in the **SOCKS 4 Identification Module Configuration** dialog box.

1. On the **Authentication** tab in the Config Tool, click **s4auth** (SOCKS v4 Identification) and click **Setup**.

   The **SOCKS 4 Identification Options** dialog box appears.



Field	Description	
Use Windows Login	Identify users by their Windows Login names.	
Use NetWare Login	Identify users by their Novell NetWare Login names.	
Prompt user for name	Identify users by the names they enter for this specific purpose.	
	Allow Memory Caching	Stores credentials in memory for this session only. Cache is flushed upon restart; credentials must be reentered as prompted.
	Allow Disk Caching	This option is currently unavailable. (Stores credentials on disk for future sessions.)

2. When you select the **Prompt user for name** option, you must also select the desired caching option. (Currently only Memory Caching is available.)
3. After making appropriate selections, click **OK**.

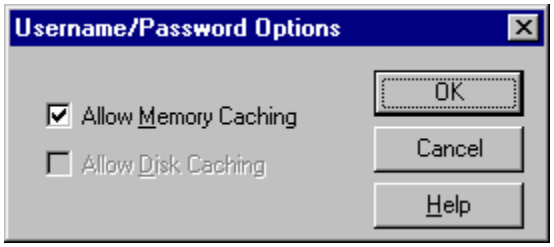
   The dialog box closes and the Config Tool reappears.

To configure the Username/Password authentication module

Aventail Connect supports the RFC 1928 (Internet standards document) username and password authentication protocol. This authentication method sends your username and password *in cleartext* across the network to the destination server. The **Username/Password authentication module** dialog box contains only credential caching options.

1. On the **Authentication** tab in the Config Tool, select **unpw** and click **Setup**.

   The **Username/Password Options** dialog box appears.



Field	Description
Allow memory caching	Stores credentials in memory for this session only. Cache is flushed upon restart; credentials must be reentered as prompted.
Allow Disk Caching	This option is currently unavailable. (Stores encrypted credentials on disk for future sessions.)

2. The selection defaults to **Allow Memory Caching**. Click **OK**.

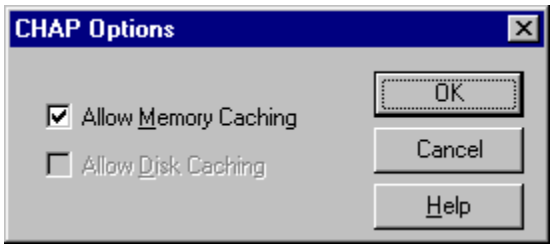
   The dialog box closes and the Config Tool reappears.

To configure the CHAP authentication module

Aventail Connect supports the Challenge Handshake Authentication Protocol (CHAP). This authentication method sends your username and password *encrypted* across the network to the destination server. The **CHAP authentication module** dialog box contains only credential caching options.

1. On the **Authentication** tab in the Config Tool, select **chap** and click **Setup**.

The **CHAP Options** dialog box appears.



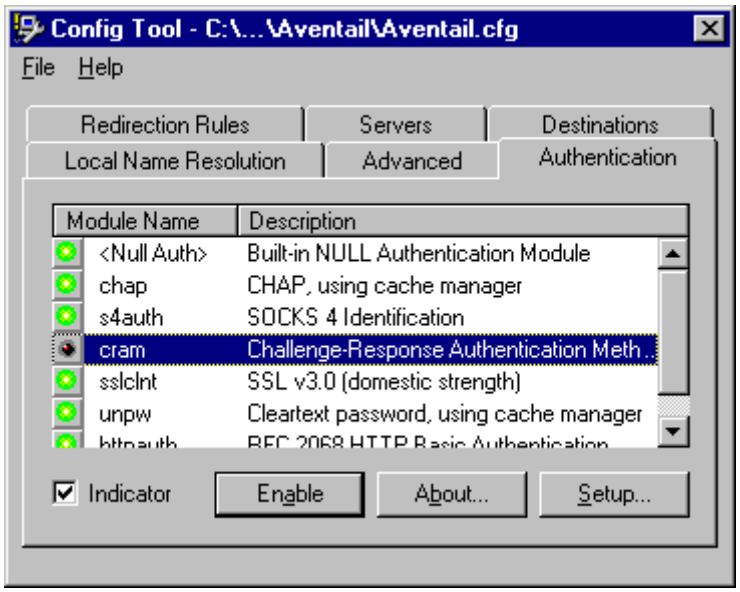
Field	Description
Allow memory caching	Stores credentials in memory for this session only. Cache is flushed upon restart; credentials must be reentered as prompted.
Allow disk caching	This option is currently unavailable. (Stores encrypted credentials on disk for future sessions.)

2. The selection defaults to **Allow Memory Caching**. Click **OK**.

   The dialog box closes and the Config Tool reappears.

To configure the CRAM authentication module

Aventail Connect supports the Challenge Response Authentication Method (CRAM). This authentication method sends your username and passcode as cleartext between extranet (SOCKS) servers, but *encrypted* between servers that support CRAM. Typically, CRAM subauthenticates within SSL, which provides both encryption and credential caching options.



You do not need to configure the CRAM authentication module. You can enable/disable it, by clicking on the **Disable/Enable** button. The button at the left of the module name will change from green to red, accordingly.

To configure the SSL security module

Aventail Connect supports Secure Sockets Layer (SSL) 3.0, a session-layer protocol for securing connections in a general, protocol-independent fashion.

NOTE: *Currently, SSL is a TCP-only enhancement. When using SSL with User Datagram Protocol (UDP) applications, bulk data is passed without encryption.*

Normally SSL servers are required to have an RSA key pair and a certificate. Aventail uses an RSA algorithm to create a cryptographic system: a private key (which, as the name suggests, is kept absolutely private and never shared) and a public key (which is widely published).

NOTE: *In versions of Aventail Connect that do not include encryption, SSL is not available.*

However, as the client, you normally must then establish some kind of relationship between your RSA public key and the identity of the server, so that somebody else cannot create their own RSA key information and use it to impersonate your server. *Certificates* establish this relationship. A certificate is essentially an electronic "statement" that verifies that a certain RSA public key is associated with a particular name.

Certificates are issued by a Certification Authority (CA), and are linked together to form a construct called a certificate *chain of authorities*, each one having a previous entity vouching for its identity. In practice, chains generally include two certificates: one confirming the identity of the server, and the other—a "root" certificate—containing the identity and public key of the CA.

Certificates contain special integrity checks and electronic signatures that verify that the certificate is genuine, was issued by a certification authority, and was not tampered with. Anybody can issue a certificate that says anything; the client must know who issued the certificate, and have some trust relationship in order to believe that it is in fact true. The client has a list of trusted CAs. A set of certificate chains can be structured as a tree, with new certificates stemming from old ones. A base CA is sometimes called the “root” or "trusted root" of this tree.

It is becoming common practice for both clients and servers to exchange certificate information. However, in Aventail Connect the client-side of this exchange is transparent. The client only needs to deal with the information from the server certificate and this is done through the SSL module.

The **SSL module** dialog box contains an initial set of options regarding the viewing of certificates.

1. On the **Authentication** tab in the Config Tool, select **sslclnt** (SSL 3.0) and click **Setup**.

   The **SSL Options** dialog box appears.

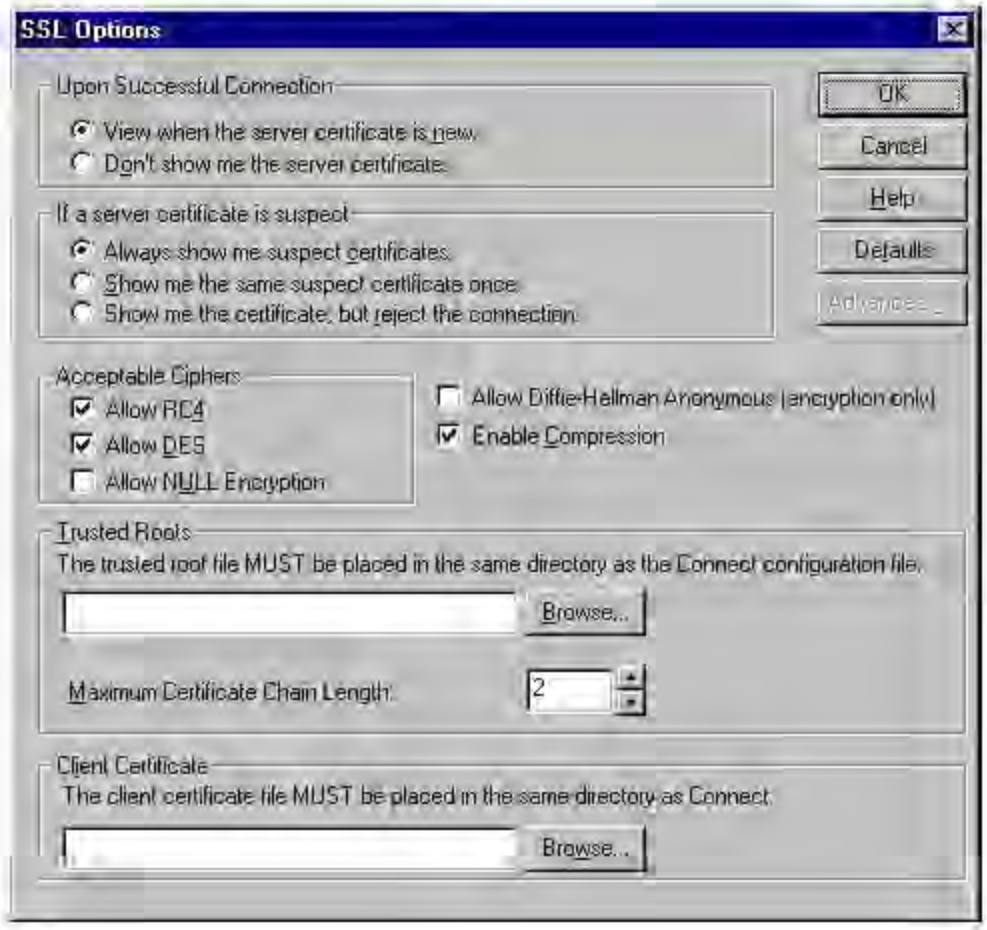
Field		Description
Upon Successful Connection		The certificate is valid.
	View when the server certificate is new.	Upon successful connection, display the server certificate if it has not been displayed during the current session.
	Do not show me the certificate.	Never display a valid server certificate.
If a server certificate is suspect		The certificate may not be valid.
	Always show me suspect certificates.	Each time Aventail Connect suspects a certificate may not be valid, show the certificate.
	Show me the same suspect certificate once.	Once a suspect certificate has been accepted by the user, do not display it again.
	Show me the certificate, but reject the connection.	Reject the connection, but display the suspect certificate.

2. Select an action that Aventail Connect must take once it accepts the validity of the server certificate. (Under normal circumstances, the server will provide Aventail Connect with a certificate to match one of Aventail Connect's trusted roots, if any exist):
   - **View when the server certificate is new:** Aventail Connect displays the certificate the first time it is seen. The certificate will not appear on subsequent connections to the same extranet server.
   - **Do not show me the server certificate:** Aventail Connect will never display a valid certificate.
3. Select an action that Aventail Connect must take if it receives a server certificate that is suspect:

- **Always show me suspect certificates**: Aventail Connect will display suspect certificates each time they are received. The **Certificate** dialog box will appear for each new connection to the server(s) sending a suspect certificate. (This option allows you to continue the connection despite the fact that the certificate is questionable.) The SSL module authenticates the server's certificate based on the following questions:
  - Is the certificate valid?
  - Did a trusted certificate authority (CA) issue the certificate?
  - Is the name established by the certificate the same as the name of the server for this connection?

  If a certificate does not pass all three tests, it is considered a suspect certificate.
- **Show me the same certificate once:** Aventail Connect will display a suspect certificate the first time that it is received. If you choose to maintain the connection, the questionable certificate will not be displayed again during the current session.
- **Show me the certificate, but reject the connection:** Aventail Connect will reject a connection if the certificate is suspect. It will display the certificate to allow you to view it.

4. Click **Advanced** in the dialog box to show the acceptable cipher (a cryptographic algorithm used to encrypt the data stream) options.



Field	Description	
Acceptable Ciphers		
Allow RC4	Offer the RC4 cipher to the server.	
Allow DES	Offer the DES cipher to the server.	
Allow NULL Encryption	Do not encrypt using SSL. SSL will be used to authenticate only.	
Allow Diffie-Hellman Anonymous	Do not authenticate the server; only do encryption.	
Enable Compression	Use SSL compression to improve performance when slower connections are detected.	
Trusted Roots	Select a certificate file that specifies trusted certificate chain roots, and specify the maximum allowable certificate-chain length. *NOTE: The trusted root file MUST be placed in the same directory as the Aventail Connect configuration file.*	
	Browse	Select the specific file
Client Certificate	Select a client certificate file. *NOTE: The client certificate MUST be placed in the same directory that Aventail Connect was installed to.*	
	Browse	Select the specific file

During the initial SSL connection, the client and the server negotiate which cipher to use. Checking a particular cipher in the dialog box does not mean that it will be used. Instead, each checked cipher is *offered* to the server, but the server determines which cipher to use. If the server requires a cipher that is not selected in this dialog box, the authentication will fail.

Any or all of the acceptable cipher options can be selected:

- **Allow RC4:** Aventail Connect encrypts the information using the RC4 cipher.
- **Allow DES:** Aventail Connect encrypts the information using the DES cipher.
- **Allow NULL Encryption:** Aventail Connect allows the server to select *no* encryption. Message integrity is still assured, but the data will be sent in cleartext.
- **Allow Diffie-Hellman Anonymous:** Aventail Connect will be able to communicate with the extranet (SOCKS) server without requiring a server certificate. The client and server will not exchange certificates, so there will be no authentication. The encryption will still be negotiated, and the data stream will still be encrypted (unless NULL encryption is chosen by the server).

- **Enable Compression:** To speed the encryption process and enhance overall performance, Aventail Connect will automatically compress encryption when a narrow bandwidth and/or slow modem are detected.

5. If necessary, add (or delete) a trusted root (`*.rot`) to (or from) the list of trusted roots by clicking **Browse**. Only the filename of the roots file loads via the **Browse** button, and not the pathname.

CAUTION: *The trusted root file must be in the same directory as the Aventail Connect configuration file.*

If Aventail Connect sends a client certificate to the server during the initial authentication exchange, it sends the certificate identified in the **Client Certificate** window. To load the client certificate, press **Browse** and then select the client certificate (`*.cer`) from the Aventail Connect directory. Only the filename of the certificate file loads via the **Browse** button, and not the pathname.

CAUTION: *The client certificate file must be placed in the Aventail Connect directory.*

When Aventail Connect receives a certificate from a server, it looks at the root of the certificate chain and matches it against the Aventail Connect list of trusted roots.

You can specify the maximum number of certificates in a certificate chain. The default maximum length is two certificates. In most instances, Aventail recommends allowing no more than two certificates to form a chain, although you can specify up to ten. The longer the certificate chain, the less secure the chain is.

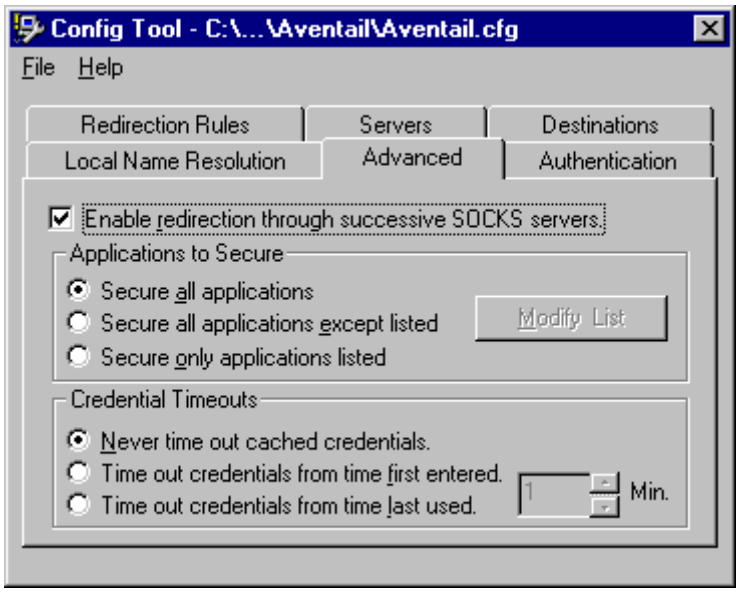
CAUTION: *In most instances, Aventail recommends allowing no more than two certificates in a certificate chain. Allowing more than two certificates can compromise security.*

6. After making appropriate selections, click **OK**.

   The dialog box closes and the Config Tool reappears.

ADVANCED TAB OPTIONS

The **Advanced** tab in the Config Tool contains three advanced options. In the **Advanced** tab, you can allow SOCKS tunneling through successive extranet (SOCKS) servers, secure selected applications, and set credential cache timeouts.



ALLOW SOCKS TUNNELING THROUGH SUCCESSIVE EXTRANET SERVERS

Once servers and destinations are defined, you can direct SOCKS traffic through successive extranet (SOCKS) servers.

On the **Advanced** tab in the Config Tool, select the **Enable redirection...** box to allow credential information to forward to successive extranet servers.

SECURE SELECTED APPLICATIONS

This option allows you to:

- secure all applications except those listed,
- secure only the applications that are listed,
- or secure all applications, enabling neither exclusion nor inclusion.

NOTE: *You can exclude and include only 32-bit applications. You cannot exclude and include 16-bit applications.*

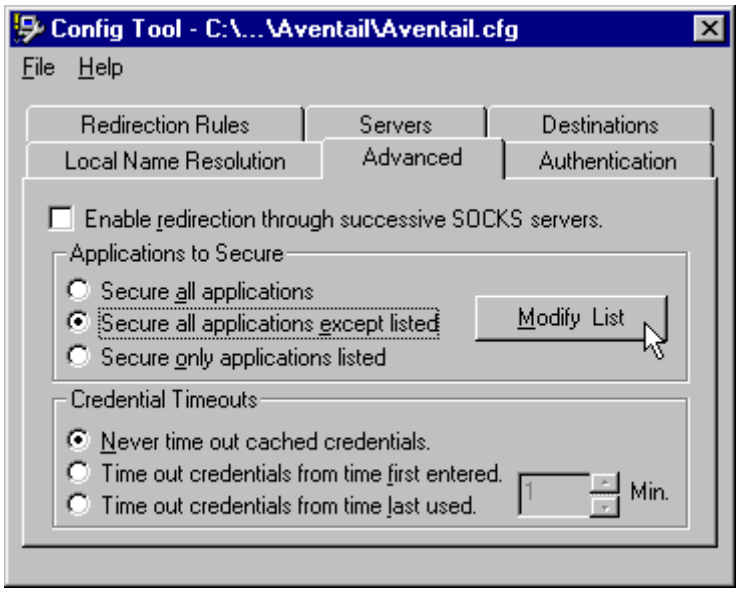
You can exclude or include specified applications in the Exclusion/Inclusion List. With the Exclusion/Inclusion List, you can secure all applications *except* those on the list, or you can secure *only* those applications on the list. The default setting is to secure (hook) *all* network applications.

Excluding Applications

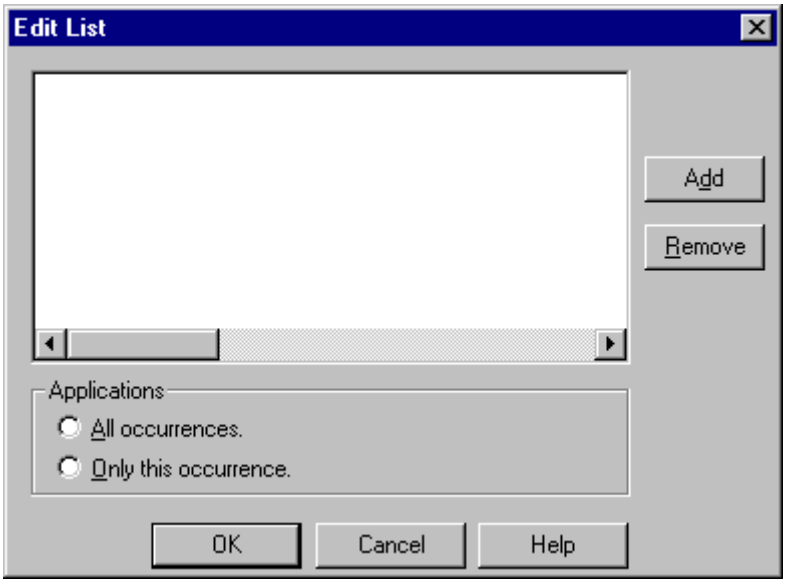
You can exclude specific applications through the Exclusion/Inclusion List. When you enable the "Secure all applications except listed" option, Aventail Connect will not proxy any applications that are on the Exclusion/Inclusion List.

To exclude an application

1. Under "Applications to Secure," select **Secure all applications except listed** and click **Modify List**.

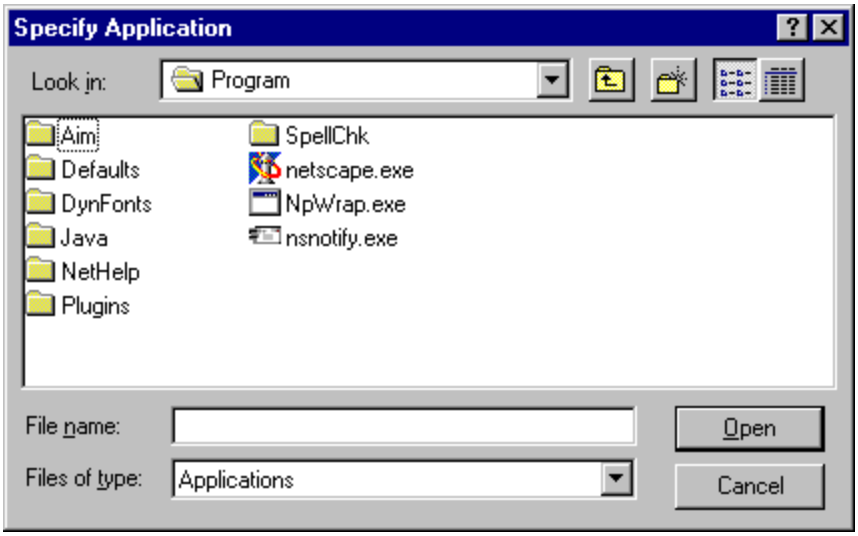


The **Edit List** dialog box appears.



2. Click **Add…**.

The **Specify Application** dialog box appears.



3. Highlight the application(s) to add to the Exclusion/Inclusion List, and then click **Open**.

   The **Specify Application** dialog box disappears and the applications are now in the **Edit List** dialog box.

4. In the **Edit List** dialog box, select **All occurrences** or **Only this occurrence**.

NOTE: *You may have more than one path (instance) of a specified filename (e.g.,* `ftp.exe`*). You can choose to exclude one specified application, with a fully qualified pathname (e.g.,* `C:\Windows\Sys32\ftp.exe`*), or all instances of a specified filename (e.g., all instances of* `ftp.exe`*).*

- **Only this occurrence:** Selecting this option excludes only the specified application.
- **All occurrences:** Selecting this option excludes all applications with the specified filename.

To undo application exclusion

1. Under "Applications to secure," select **Secure all applications except listed**, and then click **Modify List**.

   The **Edit List** dialog box appears.

2. Highlight the application you want to remove from the Exclusion/Inclusion List, and then click **Remove**.

   The application is removed from the Exclusion/Inclusion List.

Including Applications

You can include specific applications through the Exclusion/Inclusion List. When you enable the "Secure only applications listed" option, Aventail Connect will hook only those applications that are on the Exclusion/Inclusion List.

To include an application

1. Under "Applications to secure," select **Secure only applications listed**, and then click **Modify List**.

   The **Edit List** dialog box appears.

2. Click **Add**.

   The **Specify Application** dialog box appears.

3. Highlight the application(s) to add to the Exclusion/Inclusion List, and then click **Open**.

   The **Specify Application** dialog box disappears and the applications are now in the **Edit List** dialog box.

4. In the **Edit List** dialog box, select **All occurrences** or **Only this occurrence**.

NOTE: *You may have more than one instance of a specified application (e.g.,* `ftp.exe`*). You can choose to include one specified application, with a fully qualified pathname (e.g.,* `C:\Windows\Sys32\ftp.exe`*), or all instances of a specified application (e.g., all instances of* `ftp.exe`*).*

- **Only this occurrence:** Selecting this option excludes only the specified application.
- **All occurrences:** Selecting this option excludes all applications with the specified filename.

To undo application inclusion

1. Under "Applications to secure," select **Secure only applications listed**, and then click **Modify List**.

   The **Edit List** dialog box appears.

2. Highlight the application you want to remove from the Exclusion/Inclusion List, and then click **Remove**.

The application is removed from the Exclusion/Inclusion List.

Securing all Applications

You can secure *all* applications, enabling neither exclusion nor inclusion. When you secure all applications, Aventail Connect ignores any applications on the Exclusion/Inclusion List.

To secure all applications

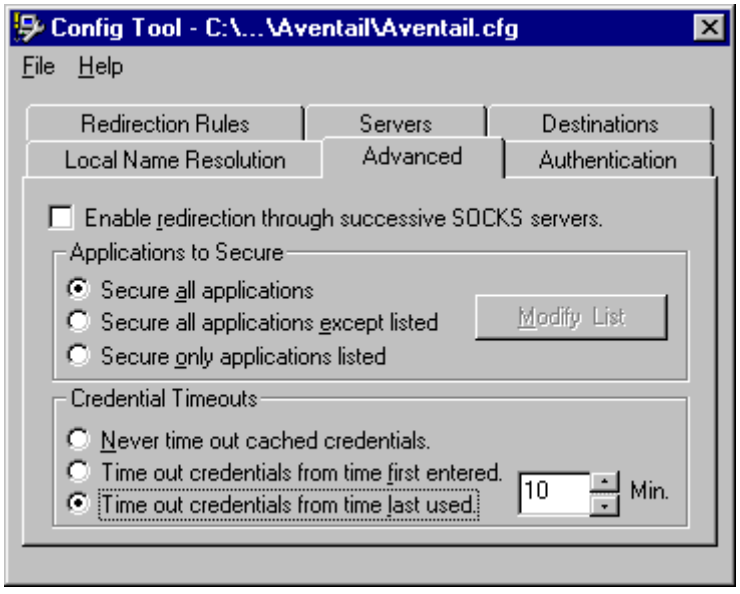
- On the **Advanced** tab, under "Applications to Secure," select **Secure all applications**.

NOTE: *Aventail Connect secures all applications by default. Unless you need to exclude or include specific applications, Aventail recommends that you use the default* ***Secure all applications*** *setting.*

CAUTION: *Microsoft Internet server products (including Microsoft Internet Information Server (IIS) and Microsoft Peer Web Server) include inetinfo.exe, which conflicts with Aventail Connect 3.01. To eliminate this conflict, exclude inetinfo.exe through the Application Exclusion/Inclusion List in the Config Tool.*

CREDENTIAL CACHE TIMEOUTS

With the credential cache timeout feature, you can control when credentials expire (time out). If a user has not made a connection to the extranet (SOCKS) server for a certain length of time (determined by the administrator), then the credentials will automatically be deleted from the credential cache. If a credential times out, the user must reauthenticate by entering the proper credentials before regaining access to the extranet. This feature can help to prevent unauthorized users from gaining access to secured areas.



There are three credential cache timeout options.

- **Never time out cached credentials:** Credentials never time out.

- **Time out credentials from time first entered:** Credentials time out ***x*** minutes after the user first entered the credentials (where "***x***" is the number of minutes you enter in the **Min.** box).
- **Time out credentials from time last used:** Credentials time out ***x*** minutes after the user last connected through the extranet server (where "***x***" is the number of minutes you enter in the **Min.** box).

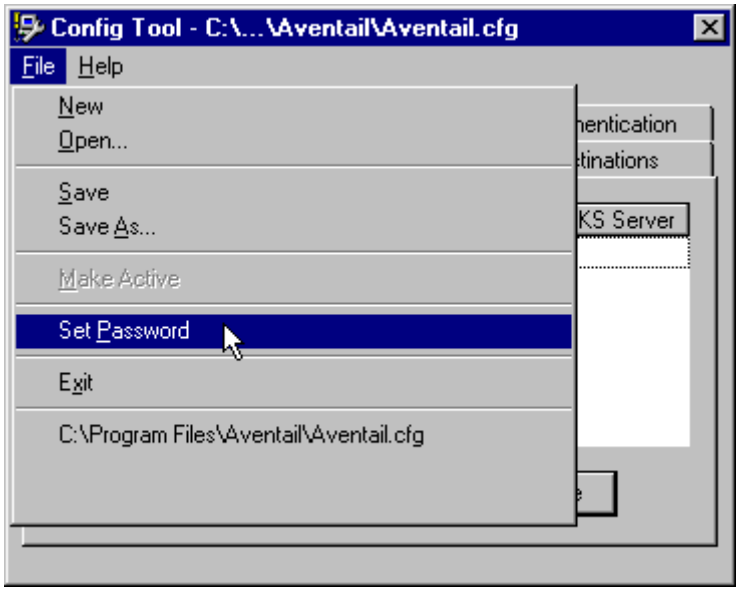
CAUTION: *If your mail program is configured to check for e-mail at regular intervals, the mail-checking frequency must be longer than the credential cache timeout. For example, if your mail program is configured to check for mail every ten minutes, you should set the credential cache to less than ten minutes.*

ENABLE PASSWORD PROTECTION

You can enable password protection for a configuration file. If you enable password protection, users will not be able to view or modify the configuration file without the assigned password. A password is not required to use the configuration file with Aventail Connect.

To enable password protection

1. From any tab of the Config Tool, select **File I Set Password**.



   The **Configuration File Password** dialog box will appear.

2. Enter the desired password.
3. Reenter the password to confirm, and then click **OK**.

To disable password protection

1. From any tab of the Config Tool, select **File l Set Password**.

   The **Configuration File Password** dialog box will appear.

2. Clear the password from both boxes, and then click **OK**.

NOTE: *If you save an existing configuration file using the **Save As** command, Aventail Connect will prompt you to enter the correct password for the configuration file.*

MULTIPLE FIREWALL TRAVERSAL

To gain access to your extranet, users may need to traverse multiple firewalls. In the simplest case, this involves an employee at a partner company gaining access to the Internet via an outbound proxy server at the partner company, and having an authenticated, encrypted, and controlled connection to your internal network via an Aventail ExtraNet Server. This capability is provided in Aventail Connect 3.01 by the Aventail MultiProxy feature. Aventail Connect can open connections through SOCKS servers, through HTTP proxies, or through proxy chaining.

- **MultiProxy with SOCKS Server:** Uses a SOCKS server to control outbound access.
- **MultiProxy with HTTP Proxy:** Uses an HTTP proxy to control outbound access.
- **Proxy Chaining:** Uses two Aventail ExtraNet Servers, where one Aventail ExtraNet Server acts as a client to another Aventail ExtraNet Server.

AVENTAIL MULTIPROXY

The Aventail MultiProxy feature allows Aventail Connect to traverse multiple firewalls by making connections through successive proxy servers. Aventail Connect makes a connection with each proxy server individually. Each proxy server forms a link in a chain that connects Aventail Connect to the final destination. Any or all of the proxy servers can apply authentication and access control rules. Proxies can be Aventail ExtraNet Servers, other SOCKS 5 servers, SOCKS 4 servers, or HTTP proxies.

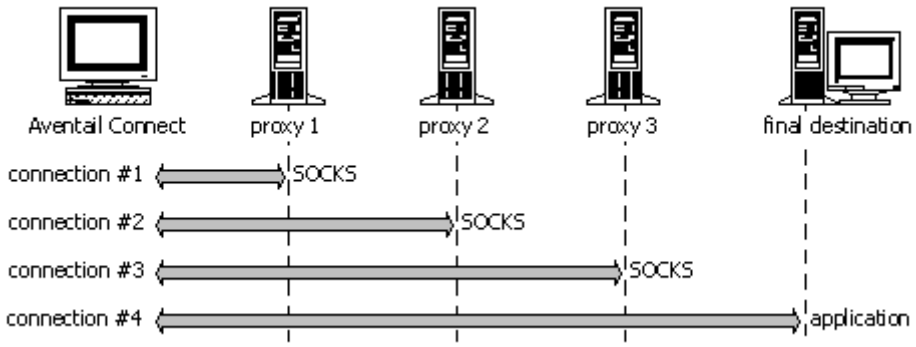
Using an HTTP proxy server to control outbound traffic eliminates the need to install a separate SOCKS server. This HTTP proxy can filter outbound connection requests and route those requests to the specified servers. MultiProxy supports RFC 2068 HTTP Basic (username/password) authentication. If your proxy uses HTTP Basic (username/password) authentication, Aventail Connect will store the username and password information in the credential cache, as it does with SOCKS servers.

NOTE: *The MultiProxy feature supports the use of HTTP proxies in Aventail Connect 3.01 only. HTTP proxies cannot be used in Aventail Connect 2.51.*

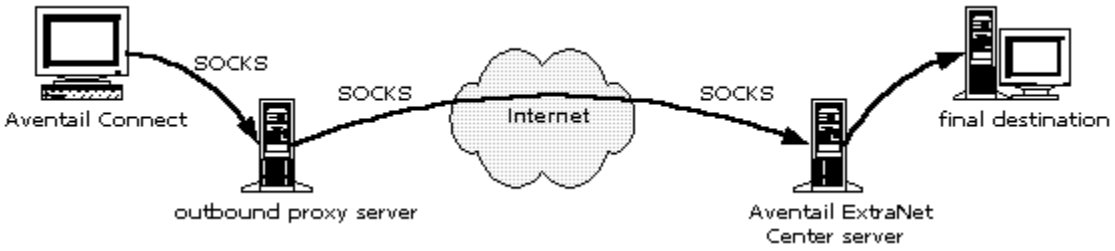
The steps for making a connection using MultiProxy are:

1. The client application requests access to the destination server.
2. Aventail Connect establishes a connection with the outbound server (SOCKS server or HTTP proxy). Aventail Connect then sends the access request to the outbound server, specifying the Aventail ExtraNet Server as the destination. The user authenticates with the outbound server, if necessary.
3. Aventail Connect instructs the outbound server to establish a connection with the Aventail ExtraNet Server on the specified port. The user authenticates with the Aventail ExtraNet Server, if necessary.
4. Aventail Connect instructs the Aventail ExtraNet Server to proxy its connection to the final destination.
5. Once the connection between the client and the Aventail ExtraNet Server is established, the outbound server simply relays the data.

The following example illustrates the connections made during a MultiProxy connection through three proxy servers.



In the following diagram, the Aventail ExtraNet Server acts as both a *destination* and a *server*. It is a destination because a proxy server routes traffic to it. It is a server because it routes traffic to the final destination.



CAUTION: *If using an HTTP proxy, you must configure your HTTP proxy and firewall to allow HTTPS/SSL connections to port 1080,* **OR** *you must run the Aventail ExtraNet Server on port 443 or port 563.*

Configuring Aventail MultiProxy

You have two options for configuring MultiProxy. You can configure Aventail Connect 3.01 to redirect all Internet traffic (including extranet traffic) through your outbound proxy, or you can configure Aventail Connect 3.01 to redirect only extranet traffic through your outbound proxy.

To configure Aventail MultiProxy

1. Create a destination ("Final destination").
2. Create a server ("Extranet server").
3. **To redirect only extranet traffic:** Create a destination ("Extranet server"), using the same information from step 2, above.

   -OR-

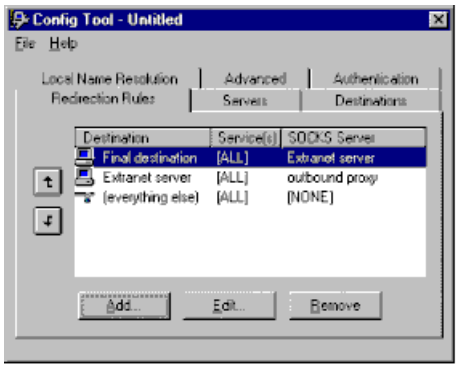
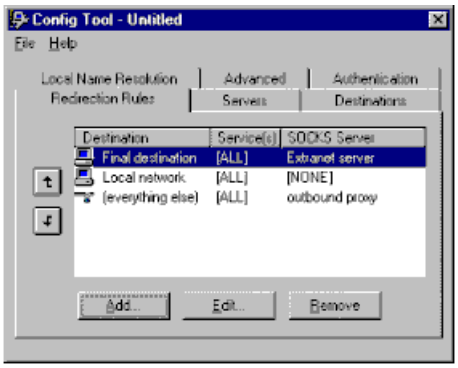
   To redirect all Internet traffic (including extranet traffic): Create a destination ("Local network," the network local to Aventail Connect).

NOTE: *If you have multiple domains or subnets, you may need to create multiple destinations.*

4. Create a server ("Outbound proxy"). This can be a SOCKS 5, SOCKS 4, or HTTP proxy server.
5. Create a redirection rule (Redirect "Final destination" through "Extranet server").
6. **To redirect only extranet traffic:** Create a redirection rule (Redirect "Extranet server" through "Outbound proxy"). Do not redirect "(everything else)."

   -OR-

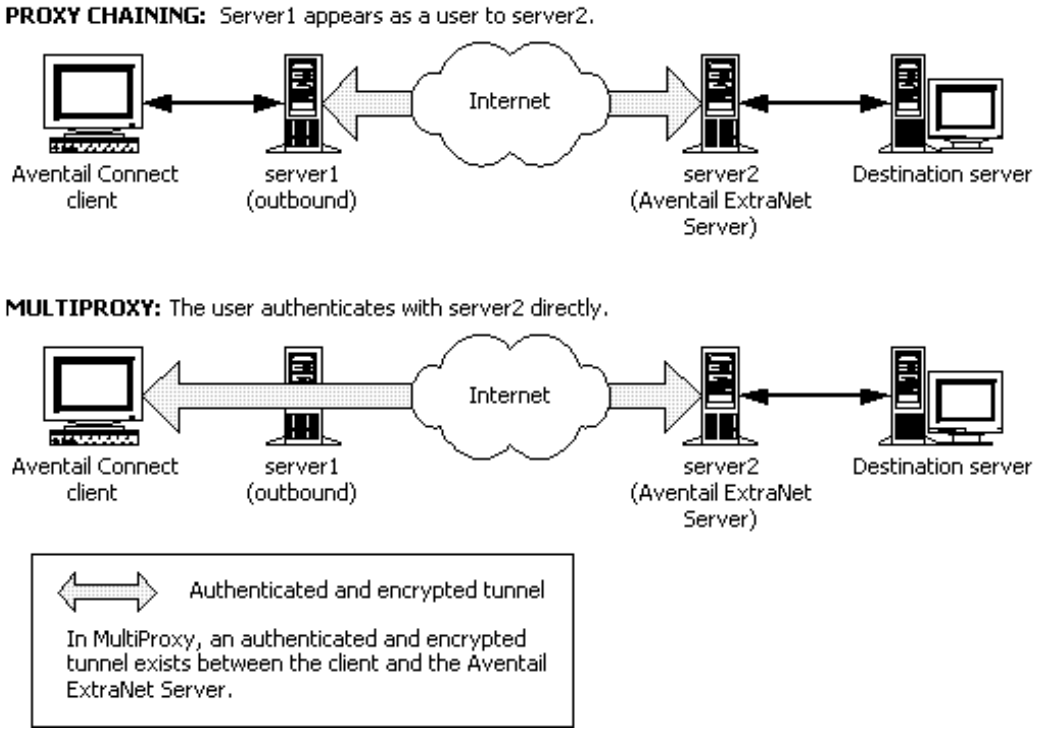
   To redirect all Internet traffic (including extranet traffic): Create a redirection rule (Do not redirect "Local network"). Redirect "(everything else)" through the outbound proxy. (**NOTE:** Your outbound proxy must belong to "Local network.")

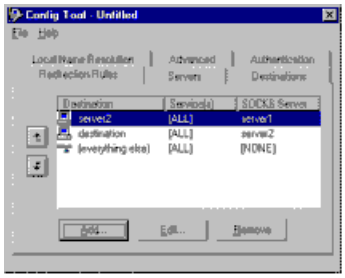
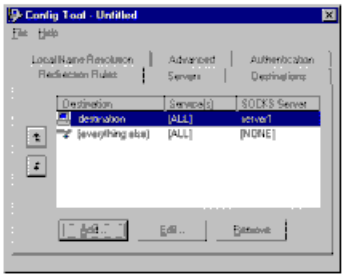
Redirect only extranet traffic	Redirect all Internet traffic (including extranet traffic)
	
Redirect only the extranet traffic through the outbound proxy. Leave all other traffic alone.	Redirect all Internet traffic through the outbound proxy. Leave only "Local network" traffic alone.

PROXY CHAINING

Proxy chaining is an Aventail ExtraNet Server feature. With proxy chaining, Aventail ExtraNet Servers forward connections for certain destinations to other proxy servers.

The following diagram and table illustrate the differences between MultiProxy and proxy chaining. In many cases, MultiProxy is the preferred method for traversing multiple firewalls. With MultiProxy, *each* proxy server can provide authentication, access control, and encryption.



Criteria	MultiProxy	Proxy Chaining
Server 1	Can be Aventail ExtraNet Server, other SOCKS 5 server, SOCKS 4 server, or HTTP proxy.	Must be Aventail ExtraNet Server.
Server 2	Must be Aventail ExtraNet Server.	Must be Aventail ExtraNet Server.
Authentication to Server 1	User authenticates (if necessary).	User authenticates.
Authentication to Server 2	User authenticates.	Server 1 authenticates automatically.
Trust model for Server 2	Not inherited. Each user must individually authenticate with Server 2.	Inherited from Server 1. Server 2 trusts everyone who authenticates to Server 1 equally.
Access control rules	Can be for specific users.	Treats everyone who authenticates to Server 1 equally.
Client configuration redirection rules		
Advantages	• Server 1 can be an Aventail ExtraNet Server, other SOCKS 5 server, SOCKS 4 server, or HTTP proxy. • Most secure, because no security policy is inherited from Server 1.	• Client is aware of Server 1 only. • User authenticates only once, to Server 1.
Disadvantages	• User may need to authenticate more than once. • Client must be aware of Server 1 and Server 2.	• All users connecting through Server 1 appear as a single user to Server 2.

HTTP Proxies and Web Browsers

Extranets often include Web pages that must be viewed with a Web browser. When a Web browser uses an HTTP proxy server, Aventail Connect sees connections being made to the HTTP proxy rather than to the final destination. Therefore, Aventail Connect cannot redirect the connections to the Aventail ExtraNet Server or provide authentication and encryption. For Aventail Connect to function properly, the Web browser cannot use the HTTP proxy to connect with sites protected in the extranet; this is because Aventail Connect must redirect and encrypt connections. The Web browser can still use the HTTP proxy to connect to sites that are not protected in the extranet.

If access to Web pages behind the Aventail ExtraNet Server requires users to connect through a Web browser (e.g., Microsoft Internet Explorer or Netscape Navigator), you must configure the Web browser to not use the HTTP proxy in the Web browser for those sites protected in the extranet.

When users need to access Web pages behind an Aventail ExtraNet Server, you must properly configure the Web browser.

Configuring Aventail Connect and the Web Browser

There are two approaches to configuring Aventail Connect for use with a Web browser.

- Configure the Web browser to not use the HTTP proxy for any traffic. (Aventail Connect redirects all connections through the outbound proxy.)

-OR-

- Configure the Web browser to not use the HTTP proxy for only those sites that are protected in the secure extranet. (Aventail Connect redirects only extranet connections through the outbound proxy.)

To use either approach, you must first configure Aventail Connect. The Aventail Connect configuration is the same for both approaches, whether you are configuring your browser to not use the HTTP proxy for all traffic or for protected sites only.

To configure Aventail Connect for use with a Web browser

1. In the **Servers** tab of the Config Tool, add the HTTP proxy as a server.
2. In the **Destinations** tab of the Config Tool, add the HTTP proxy as a destination.
3. In the **Redirection Rules** tab of the Config Tool, edit the "(everything else)" rule to redirect all traffic to the HTTP proxy server.
4. In the **Redirection Rules** tab, select the HTTP proxy and select the **Do not redirect** option.

CAUTION: *Make sure you do not redirect the outbound proxy. Redirecting the outbound server or proxy will instruct the outbound proxy to redirect traffic to itself, causing Aventail Connect to behave unpredictably.*

To configure the Web browser to not use the HTTP proxy for all traffic

After you have configured Aventail Connect by following the instructions above, configure the Web browser by using one of the following procedures.

- **Microsoft Internet Explorer**
  a. On the **View** menu, click **Internet Options.**
  b. Click the **Connection** tab.
  c. Click to clear the **Access the Internet using a proxy server** check box.
- **Netscape Navigator**
  a. On the **Edit** menu, click **Preferences**.
  b. Under "Category," click to expand **Advanced**, and then click **Proxies**.
  c. Select **Direct Connection to the Internet**, and then click **OK**.

To configure the Web browser to not use the HTTP proxy for protected sites only

After you have configured Aventail Connect, configure the Web browser by using one of the following procedures.

- **Microsoft Internet Explorer**
  a. On the **View** menu, click **Internet Options**.
  b. Click the **Connection** tab.
  c. Under "Proxy Server," click **Advanced**.
  d. In the **Exceptions** box, type the URL of each site that is in the protected extranet.
- **Netscape Navigator**
  a. On the **Edit** menu, click **Preferences**.
  b. Under "Category," click to expand **Advanced**, and then click **Proxies**.
  c. Select **Manual Proxy Configuration**, and then click **View**.
  d. In the **Exceptions** box, type the URL of each site that is in the protected extranet.

CONFIGURING THE HTTP PROXY

To allow SSL connections to destination ports other than 443 (https) and 563 (snews), you may need to configure your HTTP proxy. Typically, if you plan to connect to a SOCKS server on port 1080 using an HTTP proxy, you must change the HTTP proxy configuration.

To avoid changing the HTTP proxy configuration, you must run the destination Aventail ExtraNet Server on port 443 or port 563, and configure Aventail Connect accordingly.

Most HTTP proxies can allow connections to port 1080. The following instructions describe how to configure the Microsoft Proxy Server, Netscape Proxy Server, or Apache Web Server to allow port 1080 connections.

- **Microsoft Proxy Server 2.0:** Follow the Microsoft instructions at **http://support.microsoft.com/support/kb/articles/q184/0/28.asp**. You must modify a registry setting with `regedt32.exe`. (`regedit.exe` will not work; you must use `regedt32.exe`.)
- **Netscape Proxy Server 3.5:** Add the following to your `obj.conf` file:
  `<Object ppath="connect://*">` (*all ports*)
  `Service fn="connect" method="CONNECT"`
  `</Object>`
  To specify a particular port, add the following to your `obj.conf` file:
  `<Object ppath="connect://*:1080"`
- **Apache Web Server 1.3.2 (Linux) with Proxy Support:** The following two lines must be included in the `httpd.conf` file:

  `Proxy Requests On`

  `AllowCONNECT <port list>` (***NOTE:*** This feature is available only on version 1.3.2 and greater.)

THE CERTIFICATE WIZARD

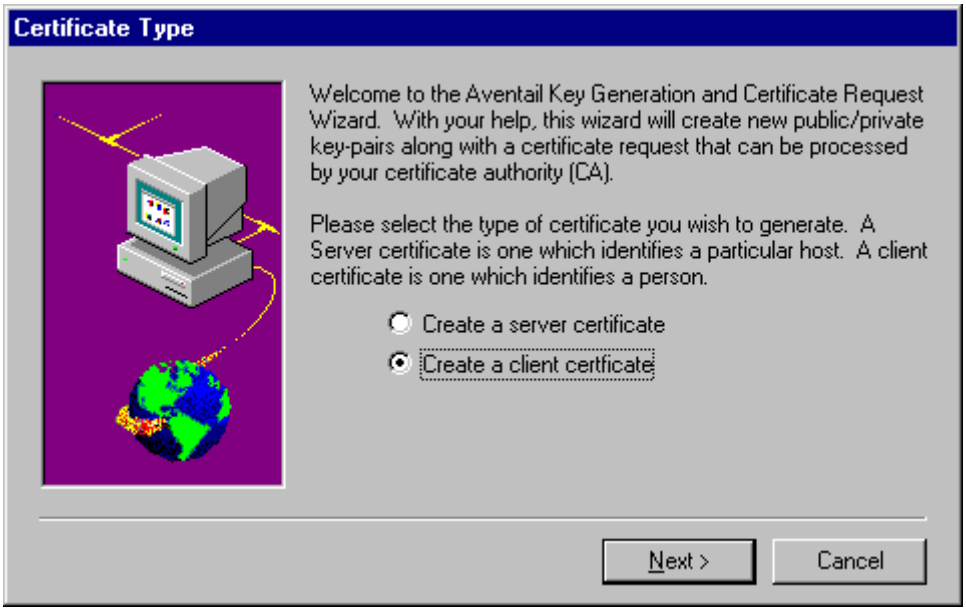
Aventail Connect supports client certificates and provides you with a certificate wizard to help *generate* and *process* a certificate. You start the certificate wizard through the Aventail Connect program group (via the **Start** button or Program Manager).

The Certificate wizard can create certificates for clients and servers. In this case, you are only interested in creating a client certificate. However, whether for client or server, you will need to run this wizard twice: Once to *generate* a Certificate Signing Request (CSR) to submit to your Certificate Authority (CA); the second time, to *process* the certificate file. If this is your first time in generating a certificate request, Aventail recommends that you complete the second step immediately after the first.

To generate the client key pair and Certificate Signing Request (CSR)

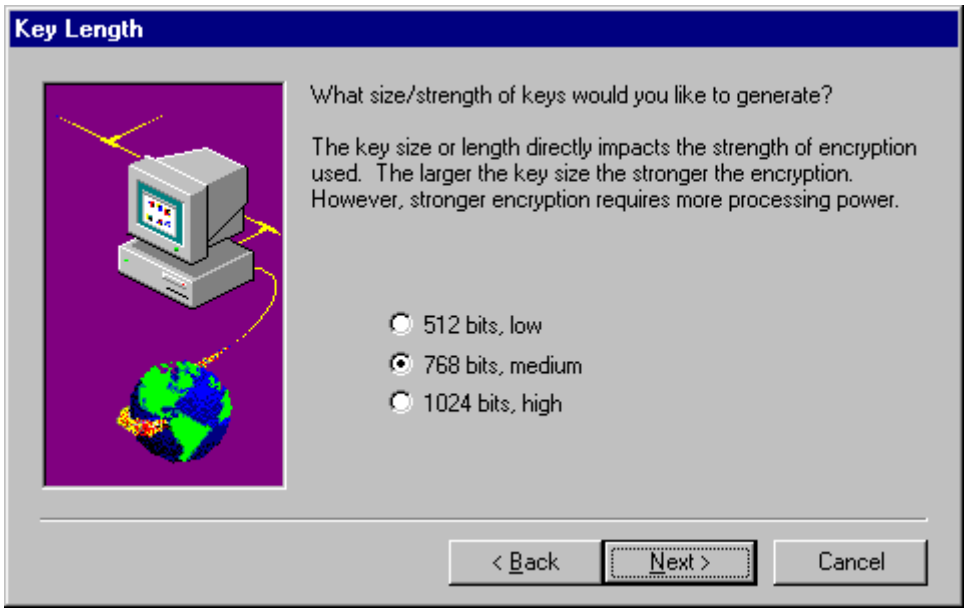
1. Select the certificate wizard from the Aventail Connect program group.

2. In the **Certificate Type** dialog box, select the **client certificate** option, and then click **Next**.

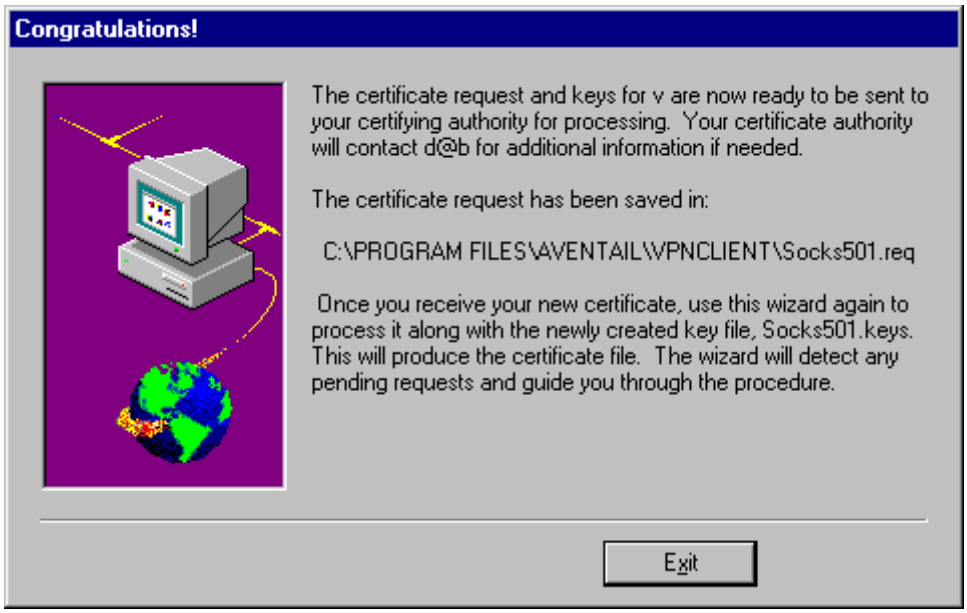


3. Provide the requested information by following the prompts in the subsequent dialog boxes.
4. In the **Key Length** dialog box, select the size of your key.

NOTE: *Not all CAs accept keys larger than 512 bits. It is prudent to know which key lengths your CA accepts prior to generating your key pair. For testing purposes use 512 bits.*



5. Once you have generated the random data, continue through the screen prompts until the **Congratulations!** screen, where you will see the name and path to the new certificate request.



To submit the Certificate Signing Request (CSR)

1. You are now ready to submit the CSR (the `*.req` file) to your CA (usually via e-mail).

   Aventail works with many certificate servers and certification authorities. Companies such as VeriSign (**www.verisign.com**) issue both client and server

certificates. The Aventail Web site (**www.aventail.com**) gives concise instructions (see "TechNotes") on how to use these programs with Aventail's certificate wizard. Aventail CSRs are generated in a standard PKCS #10 format.

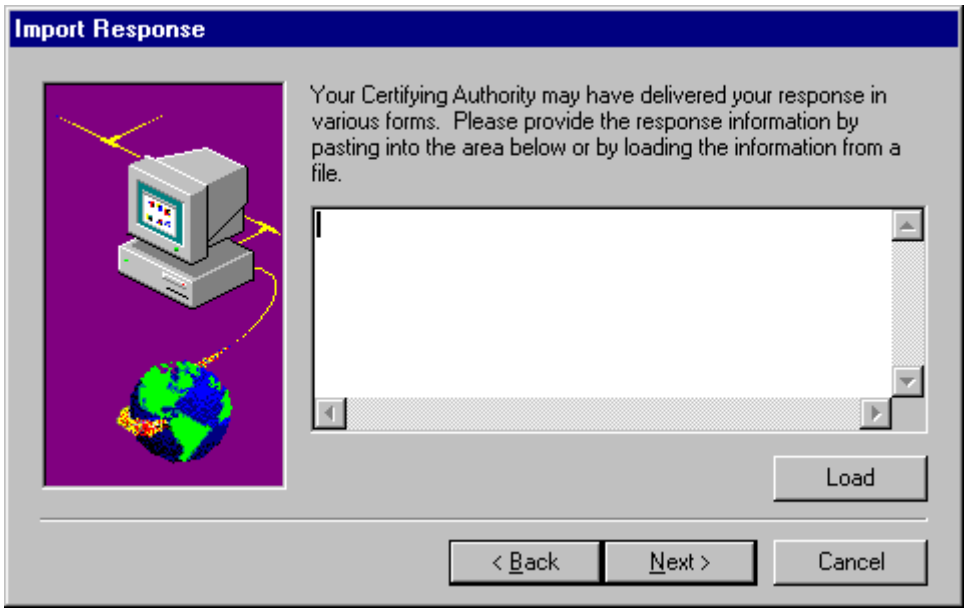
The CA will create a certificate file and issue a trusted roots file.

2. On receipt, copy (or place) the trusted roots file into the same directory as the configuration file, and the certificate file into the Aventail Connect directory.

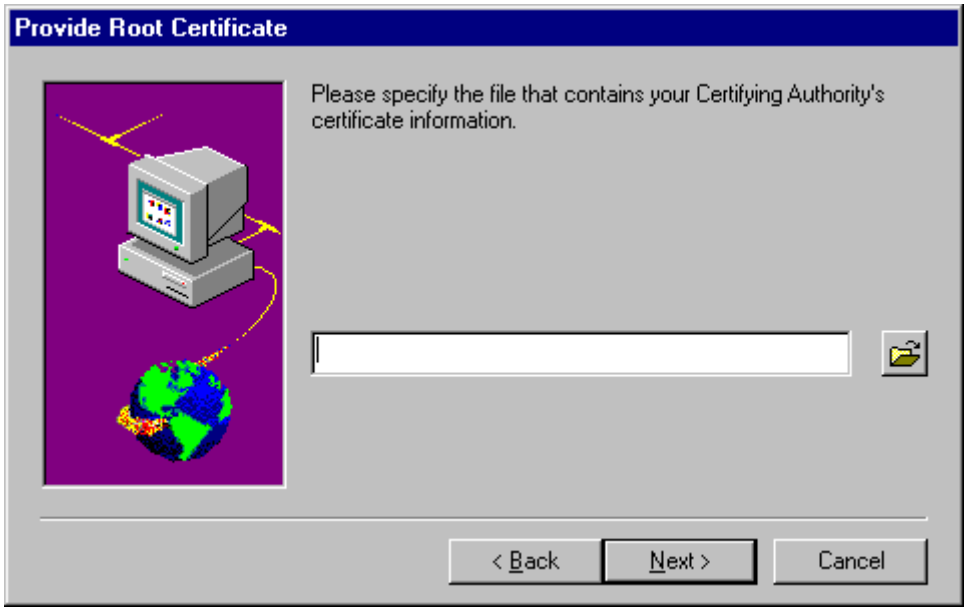
To process the CSR

1. Run the certificate wizard, again. The wizard will detect the pending requests. Click **Process**.

2. Follow the prompts on the following screens. You will be asked to paste or load the certificate response information in a window area. Click **Next**.



3. Provide the root certificate file name in the **Root Certificate** screen. The root certificate is your "trusted" root file. Click **Next**.



4. The **Summary** screen will identify the pathnames to your key file, and certificate.

   Verify that the roots file is in the same directory as the configuration file and that the certificate file is in the Aventail Connect directory.

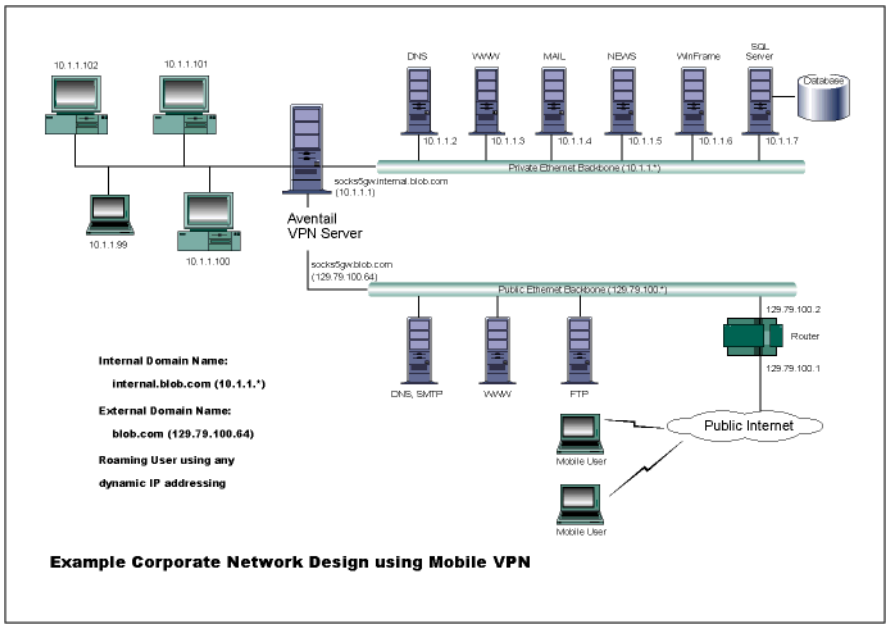
   You have successfully created a client certificate and key pair.

EXAMPLE NETWORK CONFIGURATION

The following section describes the setup of Aventail Connect in an example network configuration using the Aventail ExtraNet Server.

CONFIGURATION USING AVENTAIL EXTRANET SERVER

The following example network configurations show the Aventail ExtraNet Server configured for a Mobile Extranet environment and a Partner Extranet environment. This example emphasizes simplicity to facilitate easy adaptation to real world network designs.



The design used in the example above consists of two individual Ethernet segments, one public and one private. The public segment is used to host anonymous services available to the general public. The public access is provided through a router that is connected to the public Internet. The private segment is used to house all of the corporation's private network resources and data to be used only by internal company employees. The Aventail ExtraNet Server depicted in this example is used to provide secure and monitored access to the private LAN for mobile employees and partners. For security reasons the Aventail ExtraNet Server is configured such that operating system routing is disabled. Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the Aventail ExtraNet Server.

The mobile user workstations connected to the public Internet are the client workstations, onto which, Aventail Connect will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail ExtraNet Server unless they are running Aventail Connect. Depending on the security policy and the Aventail ExtraNet Server configuration, Aventail Connect will automatically proxy their allowed application traffic into the

private network. In this situation, Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then, based on the security policy, the Aventail ExtraNet Server will proxy mobile user traffic into the private network but only to those resources allowed. The client workstations we focus on in this section are Microsoft Windows based PCs.

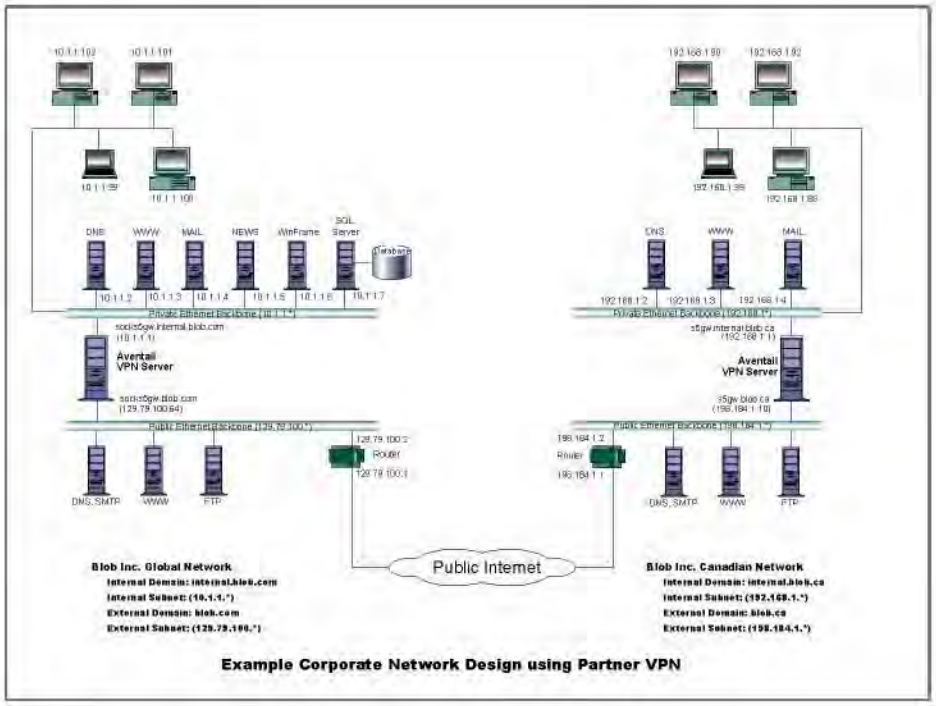
The Aventail ExtraNet Server in our example, has two network adapters configured to use the internal IP address of 10.1.1.1 and an external address of 129.79.100.64.

CAUTION: *Since the internal network address space is part of the IANA reserved address space (per BCP RFC 1918) routing MUST be disabled on this host and routing advertisements for this internal network MUST NOT be propagated to the outside world.*

User authentication and encryption on the Aventail ExtraNet Server require all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s). For this example, the Aventail ExtraNet Server encrypts all sessions with SSL.

SEE ALSO: *For additional information on how to configure the Aventail ExtraNet Server product, consult the* Aventail ExtraNet Server Administrator's Guide*.*

Installing and using Aventail Connect for remote access purposes differs a bit from its installation and use within a corporate network. First, configuration files need to be kept locally on the workstation or laptop. This is due to the inability to share a file server that allows direct access outside the perimeter of the private network. Second, not all traffic passes through to the Aventail ExtraNet Server. Only traffic destined for the internal network is authenticated and encrypted; all other traffic passes through Aventail Connect unchanged. For instance, browsing the Internet from the mobile user workstation occurs as if Aventail Connect is not even running in the background. Large sites with many mobile users will want to set up an internal file server for a network installation for all mobile users to easily install and configure Aventail Connect. For more information, refer to "Network Installation."



Example Corporate Network Design using Partner VPN

Utilities Reference Guide

This section explains:

- Commands on the System menu, including Close, Hide Icon (in Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51), Help, About, Credentials, and Configuration File
- How to use the Aventail Connect utilities, including the Config Tool, the Logging Tool, and S5 Ping, all displayed through the Utility Programs menu.
- How to use Secure Extranet Explorer (SEE)/Extranet Neighborhood.

System Menu Commands

Even though Aventail Connect requires little to no interaction with the user, there are commands on the Aventail Connect System menu. To display the System menu, right-click the **Aventail Connect** icon in the system tray on the taskbar (Windows 95, Windows 98, and Windows NT 4.0) or click the minimized **Aventail Connect** icon (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51).

Aventail Connect System Menu Commands

Menu Command	Function
Close	Closes Aventail Connect.
Hide Icon	Hides the **Aventail Connect** icon from view. Not available in Windows 95, Windows 98, and Windows NT 4.0.
Help	Accesses Help.
About	Displays Aventail Connect **About** box.
Credentials	Displays authentication credentials.
Configuration File	Selects a new configuration file via **Startup Options** dialog box.

Each of the commands is discussed below.

Close

This command closes Aventail Connect. Exiting Aventail Connect may limit access to certain remote hosts or prevent you from using certain WinSock applications.

HIDE ICON

This command hides the **Aventail Connect** icon from view (Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51 only). Aventail Connect will run in the background. *The **Hide Icon** command is not available in Windows 95, Windows 98, and Windows NT 4.0*.

HELP

This command accesses Aventail Connect Help.

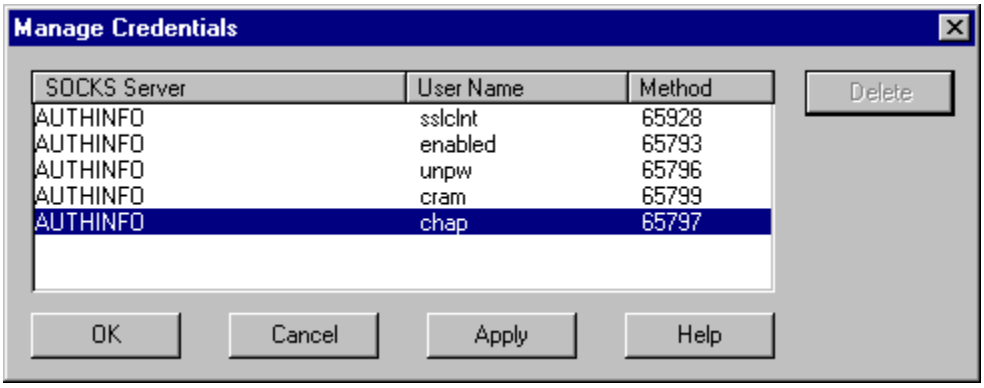
ABOUT

This command displays the Aventail Connect **About** box, which includes Aventail Connect software copyright notification, version information, and so on. Clicking **More** displays a list of files used by the current version of Aventail Connect.

CREDENTIALS

This command displays the **Manage Credentials** dialog box. Credentials include the information (such as username/password) that you enter when establishing a connection to an extranet (SOCKS) server requiring user authentication. (Aventail Connect prompts you with an authentication dialog box.) As long as your credentials are in memory, you can establish connections to associated extranet servers without needing to reenter your authentication information.

You cannot edit credential data fields; you can, however, delete individual credential entries. Aventail Connect will prompt you to enter updated authentication information when you reestablish a connection to the associated extranet server.



NOTE: *You cannot edit the "AUTHINFO" entries in the **Manage Credentials** dialog box. This information is for diagnostic purposes only.*

Field	Definition
SOCKS Server	Extranet (SOCKS) server name.
User Name	User name for the extranet server.
Method	Authentication method.

To delete a credential entry

Delete authentication credentials when they are no longer correct. After the credentials are deleted, you will be prompted to reenter them the next time you connect to the associated extranet server.

- Select the credential entry you want to delete and click **Delete**.

  This deletes the credential information.

To exit the Manage Credentials dialog box

- Click **OK** to accept changes to the credentials and close the dialog box.

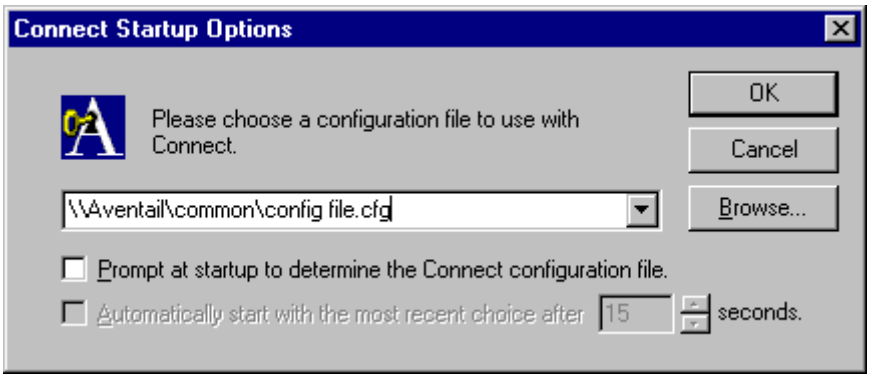
-OR-

- Click **Cancel** to close the dialog box without accepting any changes you might have entered.

NOTE: *Clicking **Apply** saves changes but keeps the dialog box open so you can keep working.*

CONFIGURATION FILE

This command lets you load a different configuration file via the Aventail Connect **Startup Options** dialog box.



For more information about the configuration file, refer to “Creating Configuration Files.”

To load a configuration file

Check with your network administrator before making any changes to the configuration.

- Select the configuration file you want to load (use the **Browse** button), and then click **OK**.
- If you want Aventail Connect to start automatically with your most recent choice of configuration file, select the **Automatically start...** check box, and then select the start delay (in seconds).

  The new configuration file transparently loads into Aventail Connect. You can close and restart Aventail Connect for your change to take effect, or wait the specified length of time if you selected the **Automatically start...** checkbox.

UTILITIES

To display the Utility Programs menu, right-click the **Aventail Connect** icon in the system tray on the taskbar (Windows 95, Windows 98, or Windows NT 4.0) or click the minimized **Aventail Connect** icon (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51).

Aventail Connect Utility Program Menu Commands.

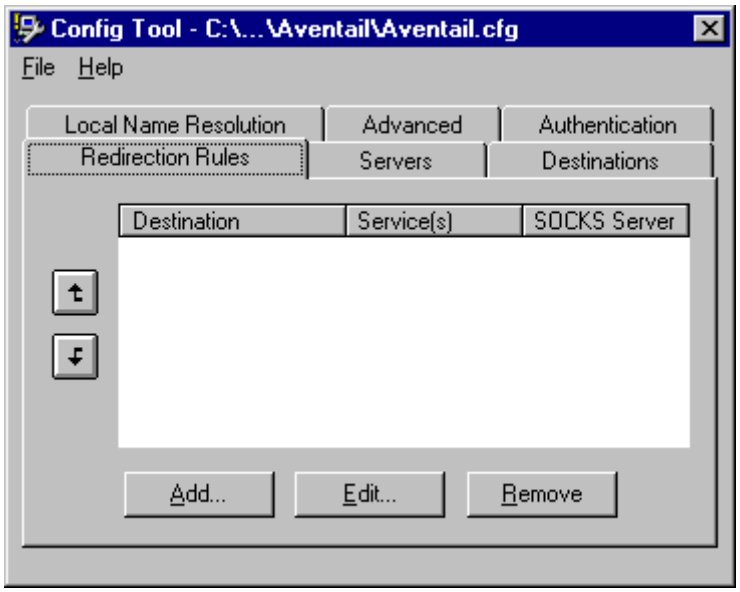
Menu Command	Function
Config Tool	Runs the Config Tool. (Optional)
Logging Tool	Runs the Logging Tool. (Optional)
S5 Ping	Runs the ping and traceroute utilities. (Optional)

Each of the commands is discussed below.

NOTE: *The **Config Tool**, **Logging Tool**, and **S5 Ping** commands are optional components and will only appear when the network administrator has included them in a custom setup package. They are discussed in the sections “Config Tool,” "Logging Tool," and "S5 Ping.”*

CONFIG TOOL

The Aventail Connect Config Tool creates configuration files that determine how network requests will be routed and which authentication protocols will be enabled. *(This option may not be available to all users if the network administrator has chosen not to install it.)*



Network administrators generally create configuration files during Aventail Connect installation. However, you can add, remove, or modify configuration files at any time. If necessary, you can create several configuration files for different users or user groups. Some configuration files may reside on a networked drive, accessible by multiple users. Other configuration files may be tailored to a specific user on an individual workstation. "Creating Configuration Files” discusses the Config Tool in detail.

LOGGING TOOL

The Logging Tool is an optional diagnostic utility for tracing Aventail Connect and WinSock activity. When running a trace, the Logging Tool displays errors, warnings, and information as Aventail Connect generates them. You can save the message list to a log file that Aventail Technical Support can use in troubleshooting technical problems, including Aventail Connect network, extranet (SOCKS) server, and WinSock application interoperability problems. Aventail Technical Support engineers may request that you perform a verbose trace, log it to a file,

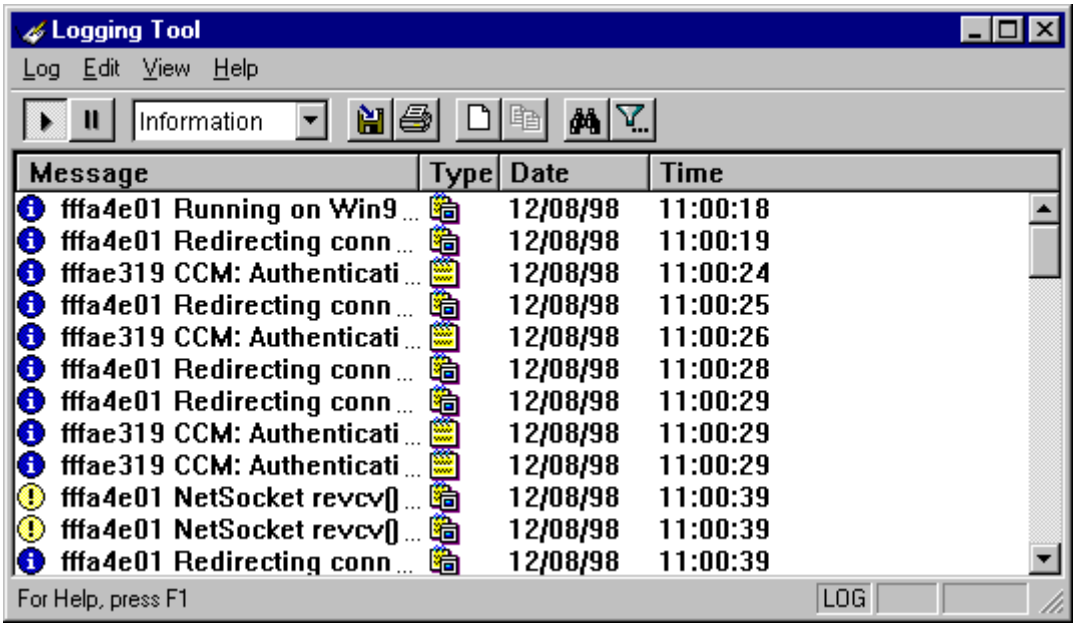
and e-mail it to them as an attachment. Log files are also useful when running Aventail Connect for the first time, to ensure that network traffic is being routed properly.

To trace Aventail Connect activity

1. Windows 95, Windows 98, or Windows NT 4.0: Either right-click the **Aventail Connect** icon (in the system tray on the taskbar) and click **Logging Tool**, or select **Start | Programs | Aventail Connect | Logging Tool**.

-OR-

Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51: From the Aventail Connect program group, double-click the **Logging Tool** program icon.



2. In the **Log** menu, click **Level** and select one of the five levels of information you want to trace.

-OR-

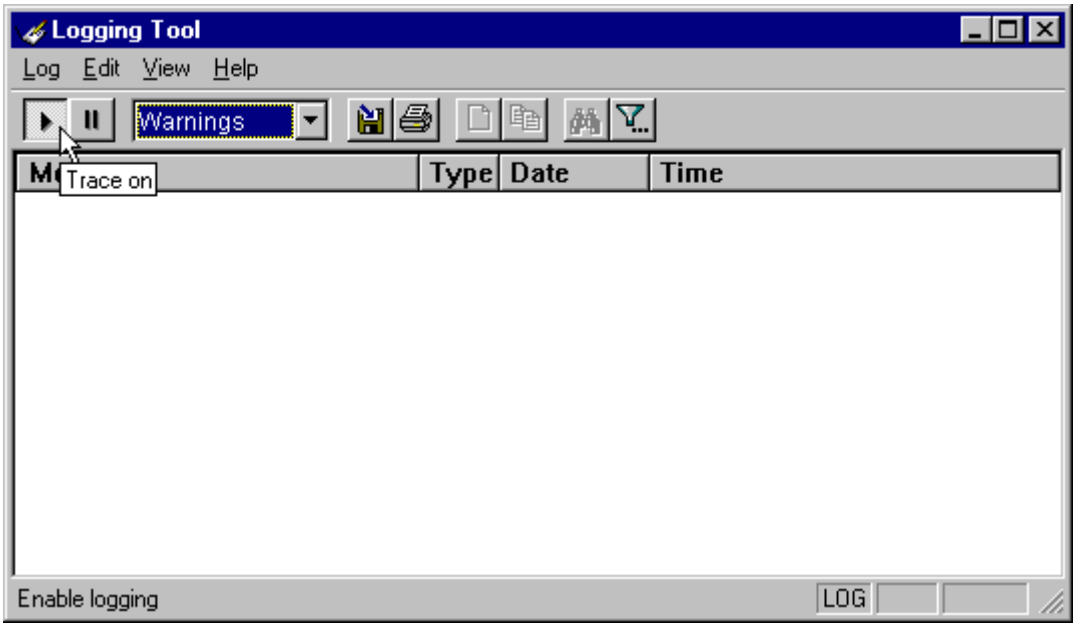
Select one of the five levels from the drop-down list on the toolbar.

Select	To Log
Fatal Errors	Fatal errors only
Errors	Errors and fatal errors only
Warnings	Errors and warnings only
Information	Errors, warning, and information
Verbose	All of the above, and more descriptive information on progress of connections

3. On the **Log** menu, click **Trace.**

-OR-

Click the **Trace On** button on the toolbar (shown below).

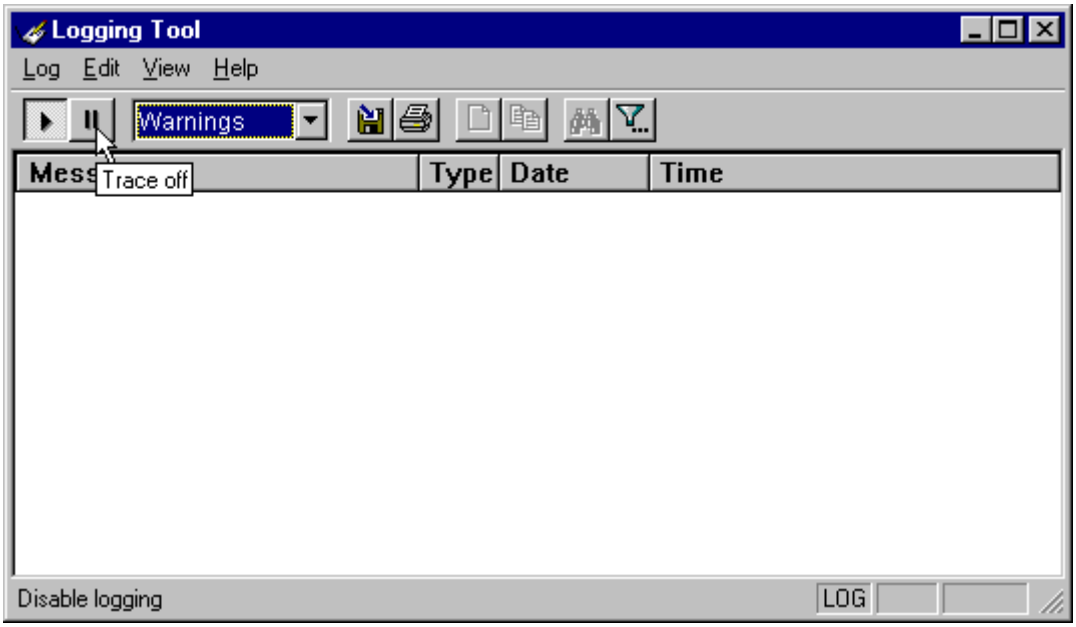


The log window will now record and display trace information as it is generated by Aventail Connect. You can tell when the trace function is active because messages are scrolling down the screen and the **Trace On** button is depressed.

4. When you are ready to stop the Trace function, click **Trace** on the **Log** menu.

-OR-

Click the **Trace Off** button on the toolbar (shown below).



The Trace function stops. You can now scroll through the results, print them, and/or save them to a file.

To save a log file

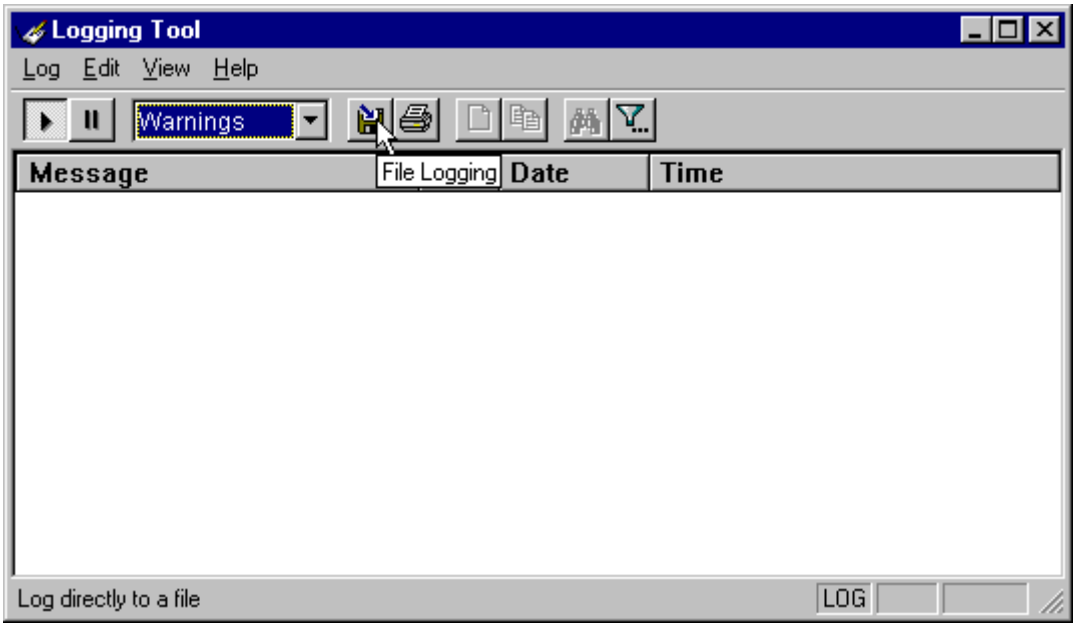
The Logging Tool allows you to append each new message to the end of a `.LOG` file during the trace, or save the contents of the log window at any time. If you save during a trace, Aventail Connect will append messages to the log file until you stop the log function. You must save data in the log window to retain it.

You cannot open a preexisting log file from within the log window. To open a pre-existing log file, you must open it in a text editor such as Notepad.

1. To save a log file as the data is being generated, click **Log to File** on the **Log** menu. Enter the filename in the **Select Log File** dialog box.

-OR-

Click the **File Logging** button on the toolbar (shown below).



2. Enter the filename in the **Select Log File** dialog box.

   - To save the contents of the log window at any time, click **Save As** on the **Log** menu and then enter the filename.

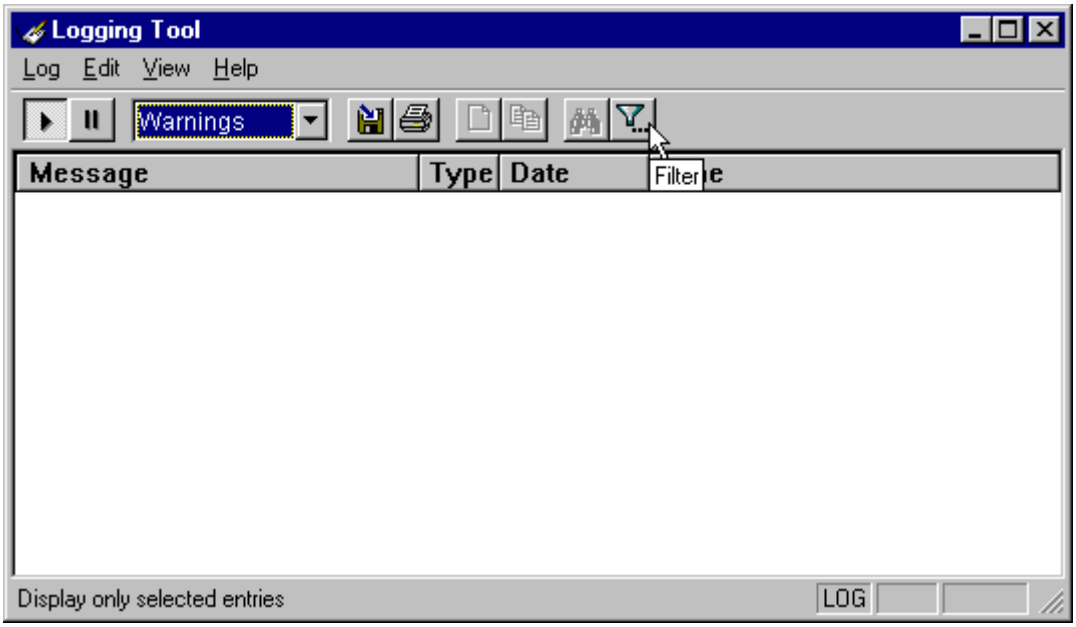
To filter messages in the log window

You can filter the contents of a log window by selecting the types of messages you want to view. By selecting a specific type of message, you can easily scan the information on-screen. If you save data to a log file, a view filter will not affect the file contents; it merely adjusts the screen display of those contents.

1. On the **View** menu, click **Filter Messages** to display the **Filter** dialog box

-OR-

Click the **Filter** button on the toolbar (shown below) to display the **Filter** dialog box.



NOTE: *The **Filter** function is an on/off toggle. If the filter is enabled, select **Filter Messages** to turn it off, then select it again to display the **Filter** dialog box.*

Field	Definition	
Categories	Select any of the five filters to display errors, fatal errors, warnings, information and/or verbose information in the log window.	
Log Type	Select the type of log to be filtered. (Currently, the only valid log type used in Aventail Connect is Miscellaneous.)	
Application Type*	32-bit	Show messages from 32-bit applications.
	16-bit	Show messages from 16-bit applications.
	*These options are disabled if you are running 16-bit Windows.	

2. Under "Categories," select one or more of the five filter check boxes. The log window will adjust the display based on your selection(s).
3. Under “Log Type,” select the log type to filter.
4. Under "Application Type," select one or both of the check boxes.

To change the view parameters

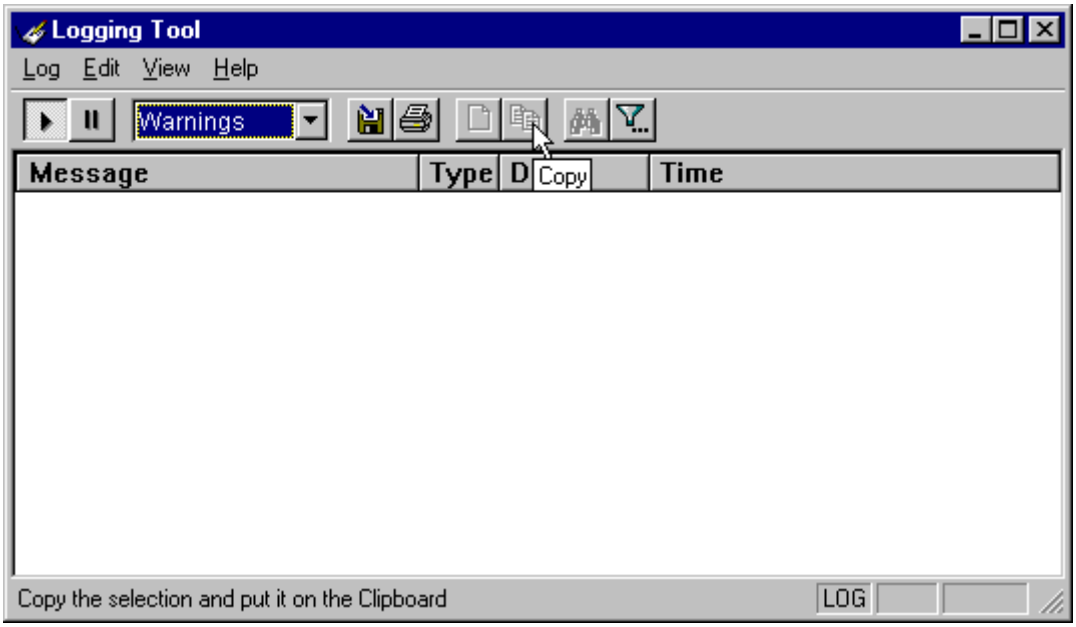
The display font and window options can be customized as follows:

- On the **View** menu, click **Font**. Enter your font preferences into the standard **Windows Font** dialog box.
- To display or hide the toolbar and status bar, click **Toolbar** and/or **Status Bar** on the **View** menu.

To copy the log window

You can copy the log window contents to the Windows Clipboard.

- To copy all of the log window contents to the Windows Clipboard, click **Select All** on the **Edit** menu. Then click **Copy** on the **Edit** menu, or click the **Copy** button on the toolbar.
- To copy selected messages to the Windows Clipboard, drag the mouse over the messages to highlight them. Then select **Copy** on the **Edit** menu or click the **Copy** button on the toolbar.



To print the log window

You can print the contents of the log window can be printed only in its entirety.

- On the **Log** menu, click **Print**.

-OR-

Click the **Print** button on the toolbar.

The entire contents of the window will print, regardless of whether you have specific messages selected. If you have filtered the display, only the filtered messages will print.

To find a specific message

The **Find** command will only work with data displayed in the window. If the display has been filtered, only the filtered messages will be searched. The **Find** dialog box remains active until you close it.

- On the **Edit** menu, click **Find**.

-OR-

Click the **Find** button on the toolbar.

Then enter your search parameters in the **Find** dialog box.

To clear the log window

Clear the log window contents when you are ready to execute a new trace.

- On the **Edit** menu, click **Clear All**.

-OR-

Click the **Clear All** button on the toolbar.

To close the log window

When you are ready to close the log window, make sure you have saved the contents of the trace for later reference. All settings are saved when you exit.

- On the **File** menu, click **Exit**.

S5 PING

Two of the most useful diagnostic tools in an administrator's arsenal are the ping and traceroute utilities.

- The ping utility checks for network connectivity between two hosts and returns information about the quality of the connection.
- The traceroute utility checks for network connectivity by displaying information about routers between two hosts. It displays information for each hop.

Ping and traceroute both use Internet Control Message Protocol (ICMP). SOCKS v5 is designed to handle TCP and UDP protocols; however, SOCKS v5 does not support ICMP. Because ping and traceroute are based on ICMP, there is no way to directly proxy a ping or traceroute request. To circumvent this problem, Aventail Connect provides a utility called S5 Ping.

S5 Ping determines whether a host outside of an extranet server is active. After a response from the host returns, the extranet server relays the data back to the client and displays it in the **S5 Ping** dialog box.

To launch S5 Ping

You can use S5 Ping whether or not Aventail Connect is running. However, if the server that you are connecting through requires authentication, you must load Aventail Connect before reconnecting.

- Windows 95, Windows 98, or Windows NT 4.0: Select **Start | Programs | Aventail Connect | S5 Ping**.

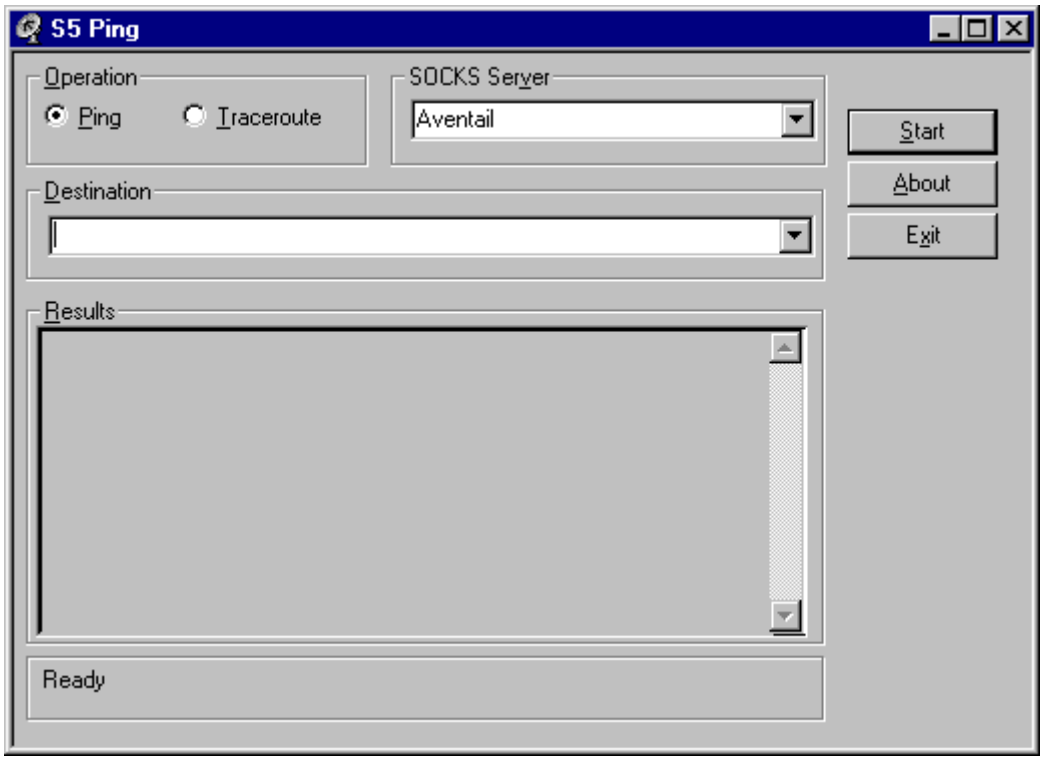
-OR-

Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51: From the Aventail Connect program group, double-click the **S5 Ping** program icon.

-OR-

If Aventail Connect is already running, right-click the **Aventail Connect** icon on the taskbar and click **S5 Ping** (Windows 95, Windows 98, or Windows NT 4.0), click the minimized **Aventail Connect** icon in the System menu (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51).

The **S5 Ping** dialog box appears.



NOTE: *S5 Ping will function without a properly configured Aventail Connect; however, the user will be required to type the information about the target extranet server and target host into the* ***SOCKS Server*** *and* ***Destination*** *boxes.*

Field	Definition
Operation	Select ping or traceroute.
SOCKS Server	The Extranet (SOCKS) server that will execute the operation. If Aventail Connect is already configured, this list will be preloaded with extranet servers from the configuration file.
Destination	The extranet server you want to ping (or traceroute). If Aventail Connect is already configured, this list will be preloaded with single host destinations defined in the configuration file. (See “Configuring Aventail Connect.”)
Results	The results of successful connection. The format of the results will vary based upon the extranet server platform.

S5 Ping can be used whether or not Aventail Connect is running. However, if the server that you are connecting through requires authentication, you must load

Aventail Connect before connecting. The network administrator may or may not make S5 Ping available to users during installation. In some cases, the **S5 Ping** command will not appear on the Aventail Connect System menu or in the program group.

Once the **S5 Ping** dialog box opens, you can execute a ping or traceroute network operation.

To run ping or traceroute using S5 Ping

S5 Ping has two modes of operation: ping and traceroute.

1. Under "Operation," select one of the two options, **Ping** or **Traceroute**.
2. Under "SOCKS Server," select an Aventail ExtraNet Server to carry out the operation. If no servers are listed (because S5 Ping did not locate an Aventail Connect configuration file), type the extranet server's hostname or IP address.
3. Under "Destination," select a single host destination to ping or traceroute. If no hosts are listed (because S5 Ping did not locate an Aventail Connect configuration file), type the hostname or IP address of the host you want to ping or traceroute.
4. Click **Start** to execute the operation. **Start** then changes to **Stop**. Results from any previous operation are cleared from the window.
5. If the extranet server requires authentication, you may be prompted with a server certificate or required to enter a username and password. (For more information about server certificates and username/password authentication, see "Manage Authentication Modules" in the *Administrator's Guide*.)
6. Once the connection to the host has been made, the information returned from the server will be displayed in the **Results** window.

To stop ping or traceroute

- Click **Stop**.

  This stops the operation and changes **Stop** to **Start**. The results of the operation remain displayed in the **S5 Ping** dialog box.

To exit S5 Ping

- Click **Exit**.

  This clears the results and closes the **S5 Ping** dialog box.

SECURE EXTRANET EXPLORER

Secure Extranet Explorer (SEE) allows you to view your Extranet Neighborhood, which is accessed through the **Extranet Neighborhood** icon on your desktop. The Extranet Neighborhood user interface resembles that of Network Neighborhood. However, while Network Neighborhood displays all computers on your local network, Extranet Neighborhood allows you to browse, copy, move, and delete files from remote computers via the Aventail Connect extranet connection. With Extranet Neighborhood, all interaction with the remote server can be secured. Network administrators determine which local and remote computers are available to users.

NOTE: *Some installations of Aventail Connect may not include SEE. Network administrators can decide whether or not to include SEE in a custom setup package.*

Extranet Neighborhood, a Windows Explorer shell extension, is a collection of Windows file servers and Windows NT domains. Network Neighborhood displays only those remote computers that the network administrator has specified. SEE requires a hosts file (SEEHosts) that determines which Windows file servers and NT domains are available. You can include a SEEHosts file with the Aventail Customizer tool. If users install a custom package that does not include a SEEHosts file, then the first time they open Extranet Neighborhood, SEE will create a SEEHosts file. For more information, see the “Customizer” section in the *Administrator’s Guide*.

Extranet Neighborhood offers Aventail Connect users a secure alternative to traditional file-browsing methods. Users can securely access computers from the desktop through Extranet Neighborhood (see icon below), or through Windows Explorer.

Generally, you will use Extranet Neighborhood to connect to a remote network through Aventail Connect. For example, you will use Extranet Neighborhood when:

- you are inside the office, on the corporate network, and you connect through an Aventail ExtraNet Server to your company’s remote site, or to another company’s network.
- you are outside the office, and you connect your laptop through an Aventail ExtraNet Server to your internal company network, or to another company’s network.

NOTE: *To use Extranet Neighborhood with remote hosts, Aventail Connect must be running and configured correctly.*

HOW EXTRANET NEIGHBORHOOD WORKS

Typically, with Windows networking, the Microsoft Windows Explorer and Network Neighborhood browse files using NetBIOS (NBT), over TCP. Network Neighborhood does not use the standard WinSock programming interface. This prevents Aventail Connect from redirecting TCP connections. Since Aventail Connect redirects only WinSock calls, it cannot redirect NBT calls.

To deliver a secured version of standard Windows browsing, Aventail Connect redirects NBT calls to WinSock. This allows Aventail Connect to redirect this traffic based on a set of redirection rules, as defined in the Aventail Connect configuration file.

Extranet Neighborhood can use either hosts files or Windows Internet Naming Service (WINS) servers to map a computer's Internet (host) name to its Windows machine name. Without a hosts file or a WINS server, Extranet Neighborhood cannot associate a computer's Internet name with its Windows machine name.

Extranet Neighborhood includes a browsing mode, which allows you to view a dynamic list of available Windows hosts. Hosts files provide a static list of hosts.

There are two basic methods for configuring Extranet Neighborhood.

- **Listing WINS Servers:** List only WINS servers for the domain(s) in the hosts file. You do not need to list individual hosts within the domain.
- **Listing Individual Hosts:** List every individual host in the hosts file that will be accessible to users.

LISTING WINS SERVERS

To use Extranet Neighborhood in the browsing mode, you must configure Extranet Neighborhood to use WINS, and you must identify the IP address (hostname) of the WINS server(s) and, possibly, the primary domain controller (PDC) for the domain. If you do not specify a WINS server, you will not be able to use Extranet Neighborhood in the browsing mode.

The PDC for the domain is required only if the destination network is not accessible by UDP. (For example, when using MultiProxy, the destination network is not UDP-accessible.) When Extranet Neighborhood is in browsing mode, it must be able to resolve the name of the host. If the destination network is UDP-accessible, then the WINS server is used to map a computer's Internet (host) name to its Windows machine name. If the destination network is not UDP-accessible, then Extranet Neighborhood uses the PDC and DNS to determine the host's address.

LISTING INDIVIDUAL HOSTS

To use Extranet Neighborhood in the static host list mode, you must define, in the hosts file, each individual host in the domain. This allows you to restrict access to designated hosts only. In the hosts file, you must specify the host's IP address or DNS name along with the Windows machine name. WINS and PDC are not used in this method.

INSTALLING EXTRANET NEIGHBORHOOD

When installed, Extranet Neighborhood appears on your desktop as an icon, and in Windows Explorer. You can open, move, copy, and delete files in Extranet Neighborhood just as you would in Network Neighborhood.

If you need to install Extranet Neighborhood, install it from the Aventail Connect CD. Or, if you downloaded your copy of Aventail Connect, run the downloaded executable package. When the **Installation Components and Sub-components** dialog box appears, select **Extranet Neighborhood** (located under **Components**). Continue with the installation process.

The default installation directory is `\Program Files\Aventail\Connect`.

NOTE: *Secure Extranet Explorer/Extranet Neighborhood is available only on Windows 95, Windows 98, and Windows NT 4.0 operating systems.*

CONFIGURING EXTRANET NEIGHBORHOOD

You can include a SEEHosts file with the Aventail Customizer tool. Only by installing a custom package will users have a local or remote hosts file automatically configured. If users install a custom package that does not include a SEEHosts file, the SEE Configuration wizard will run when users open Extranet Neighborhood for the first time. The SEE Configuration wizard walks you through the process of defining local or remote hosts files. Aventail recommends that you use the Customizer tool to distribute Extranet Neighborhood, bundled with a hosts file, in a custom setup package.

Extranet Neighborhood can automatically construct a hosts file from your local network or a remote network. Using the Search feature, Extranet Neighborhood can automatically "browse" available computers and build the local hosts file. The Search feature is available through the **Extranet Neighborhood Properties | Local** tab. Alternatively, you can enter the names of the available computers manually. The Search feature browses only those computers that are within your internal network. To search remote networks, you must manually enter the fully qualified hostname of each remote WINS server that is outside your Aventail ExtraNet Server. When using the Search feature, the same UDP restrictions described in "Listing WINS Servers" apply.

NOTE: *To use the Search feature, Aventail Connect must be running and configured correctly.*

Do not use the Search feature if you are using the WINS-browsing mode. The Search feature builds the local hosts file for all of the computers, which is not necessary with WINS. Use Search when creating a local hosts file using the "listing individual hosts" method.

NOTE: *When you click* ***Search****, you may see more than one domain in the resulting local hosts file. This is because Search includes trusted domains.*

To create a hosts file

Use this procedure if you have not yet created a hosts file.

1. Decide which method, listing WINS servers or listing all individual hosts, to use.
2. If no hosts file exists, launch Extranet Neighborhood (Extranet Neighborhood will prompt you automatically if you are running Extranet Neighborhood for the first time),

-OR-

Right-click the **Extranet Neighborhood** icon on your desktop and then click **Properties**.

3. Follow the on-screen instructions to create the hosts file.
4. To distribute the new hosts file, include the SEEHosts file in your custom setup package, if using the Customizer tool.

After creating the hosts file, users can browse only those domains and machines that the network administrator has included in that list of hosts. This list may be a local hosts file called "SEEHosts" and/or a remote host list, which is identified by `[share]\[path]\[filename]`.

NOTE: *To use the browsing mode, you must specify the domain's WINS server(s) in the local hosts file.*

CAUTION: *SEE cannot recognize share names that contain special characters (e.g.,* é*) or multiple spaces (e.g.,* Aventail Custom Computer*). SEE also will not recognize hidden one-letter share names (e.g.,* C$ *or* D$*).*

SEE CONFIGURATION METHODS

There are numerous methods for configuring SEE. The three most common methods are described below.

Local Hosts File Method

With this method, the hosts file contains a list of all domains and servers in the local hosts file. Every host is listed.

There are two ways to configure SEE using this method.

- In the **Extranet Neighborhood Properties l Local** tab, manually add each domain and host to the local hosts file

-OR-

- On the **Local** tab, click **Search**, click **Search Local Network**, and then search any remote networks, if necessary. SEE automatically builds a list of all hosts. You may delete hosts from the local hosts file if you do not want users to view them.

NOTE: *To view your domains, double-click the* ***Extranet Neighborhood*** *icon on your desktop. If you make changes to the hosts file, you can reload the* ***Extranet Neighborhood domains*** *window by pressing the* `F5` *key.*

Remote Hosts File Method

With this method, the local hosts file contains the path of the remote hosts file, and the remote hosts file contents are determined by which configuration method you use.

To use this method, first create the remote hosts file, and then create a local hosts file that points to the remote hosts file.

To configure SEE using the remote hosts file method

1. Create a local hosts file, using one of the methods listed above, and copy it to a central location. (This creates a remote hosts file; this file is not distributed with Aventail Connect.)
2. On the **Remote** tab, click **Add**, and then add a pointer to the remote hosts file that you created in Step 1. (This file is distributed with Aventail Connect.)

NOTE: *You can point to multiple remote hosts files on a single list.*

WINS Browsing Method

With this method, the hosts file contains a list of all domains, and the WINS servers for each domain. You do not need to list all of the computers.

To use this method, add each domain in the **Local** tab, specifying the primary WINS server and, if applicable, the secondary WINS server, and then select the **Make domain browsable** check box in the **Windows Domain** dialog box.

Choosing a Method

Each of the three methods has advantages and disadvantages. The table below lists pros and cons for each of the three methods.

Method	Advantages	Disadvantages
Local hosts file with individual computers	The administrator controls exactly which hosts the users can see. On slower connections, this method is fastest since you do not need to send a list of servers to the client.	The administrator must update the local hosts file if file servers are added to or removed from the domains.
Remote hosts file	• The administrator can edit the centrally stored hosts file whenever necessary. • If the hosts file is stored behind a firewall, SEE can go through an extranet server (using encryption and authentication) to reach it.	• Users are immediately prompted to enter authentication credentials upon opening SEE (because SEE must load the remote hosts file). • If a user loses network connectivity to the hosts file, SEE will not display the list of hosts/computers.
Local hosts file with WINS browsing	The administrator does not need to update the hosts file if new computers are added or removed.	• The administrator must update the local hosts file if domains are added or removed. • The administrator cannot control which computers appear in SEE; all computers in the NT domain are displayed. • On slower connections, this method is slower than other methods because a list of computers must be sent to the client.

You are not limited to using only one method for configuring SEE. You can use a combination of the various methods. For example:

- Use WINS browsing for some domains, and explicitly list hosts for other domains

-OR-

- Use multiple remote hosts files

-OR-

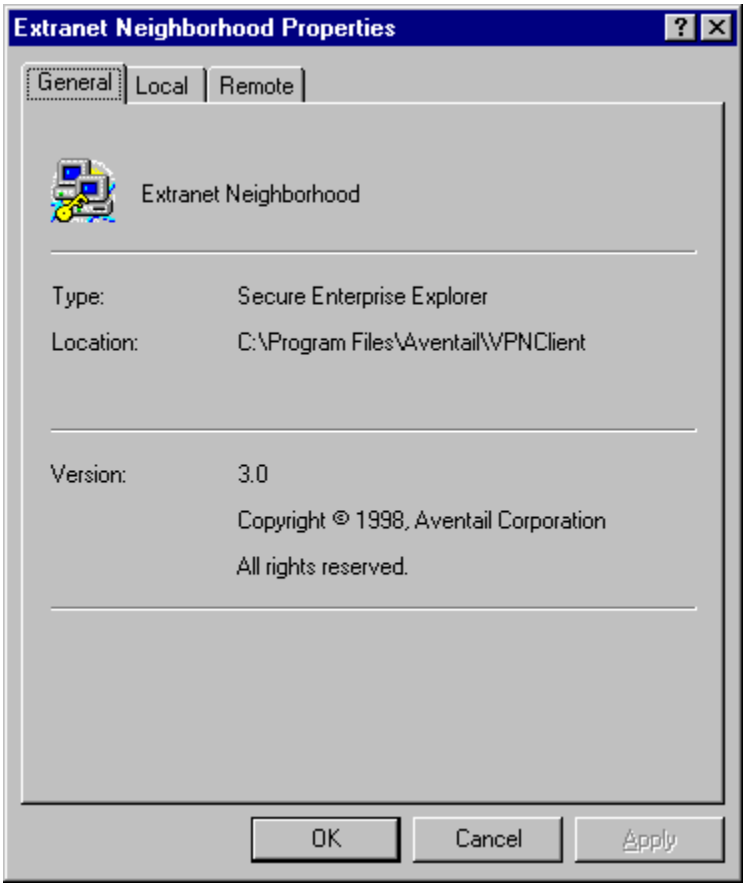
- Specify some computers in a local hosts file and others in a remote hosts file.

SEE PROPERTIES

To access information about the current configuration of SEE, or to make changes to that configuration, right-click the **Extranet Neighborhood** icon and click **Properties**, or click **View I Options** in any open **SEE** window. The **Extranet Neighborhood Properties** window will appear with the **General** tab selected.

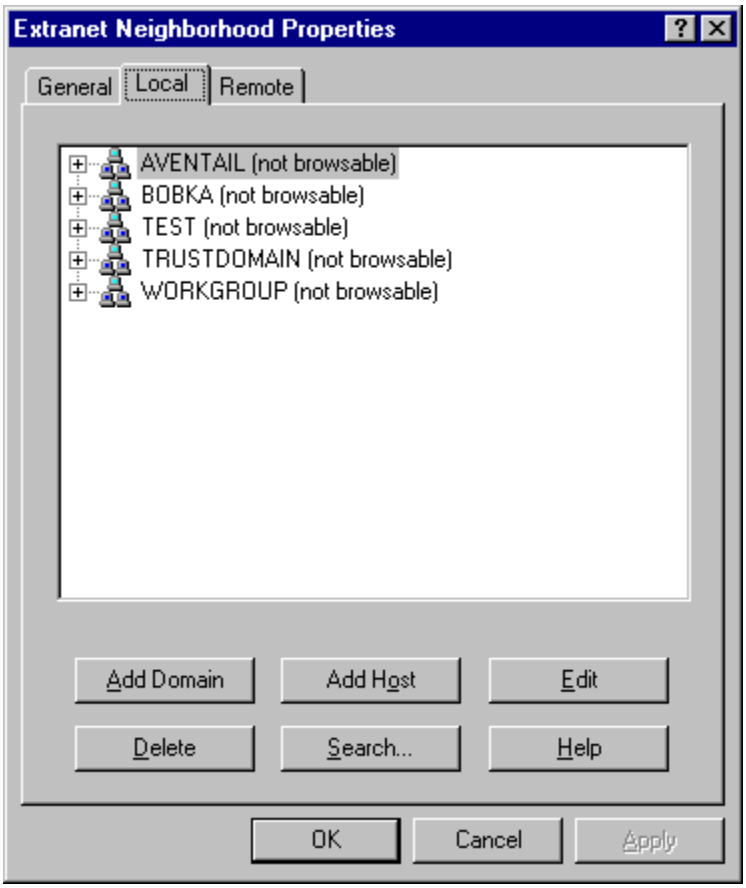
THE GENERAL TAB

The **General** tab displays information about the current configuration of SEE.



THE LOCAL TAB

The **Local** tab displays the computers that are listed in the local hosts file.



If you have specified a host in the local hosts file, you can add, edit, or remove computers or domains that appear in the **Local** tab. If you have specified hosts in the remote hosts file, they will not appear in this tab. To edit hosts in the remote hosts file, you must copy the file to your Aventail Connect directory, edit it, and then replace it in the remote hosts directory.

If you are using the WINS browsing mode, the individual computer names will not appear. Any hosts specified in remote hosts files, including WINS servers, will not appear in this tab.

The **Add Host** and **Add Domain** buttons allow you to add additional computers or domains in the **Add Host to Aventail** dialog box and the **Windows Domain** dialog box.

If no computers or domains appear in your **Local** tab, check the **Remote** tab. It is possible that your network administrator has configured Extranet Neighborhood with only a remote hosts file.

The Search feature can automatically browse available computers in local or remote domains and populate your local hosts file. Alternatively, you can enter the names of the hosts files manually.

NOTE: *To view your domains, double-click the **Extranet Neighborhood** icon on your desktop. To reload the hosts files in the **Extranet Neighborhood domains** window, press the* `F5` *key.*

NOTE: *In the **Local** tab, "browsable" domains do not show individual computers in them.*

Hosts File Locking

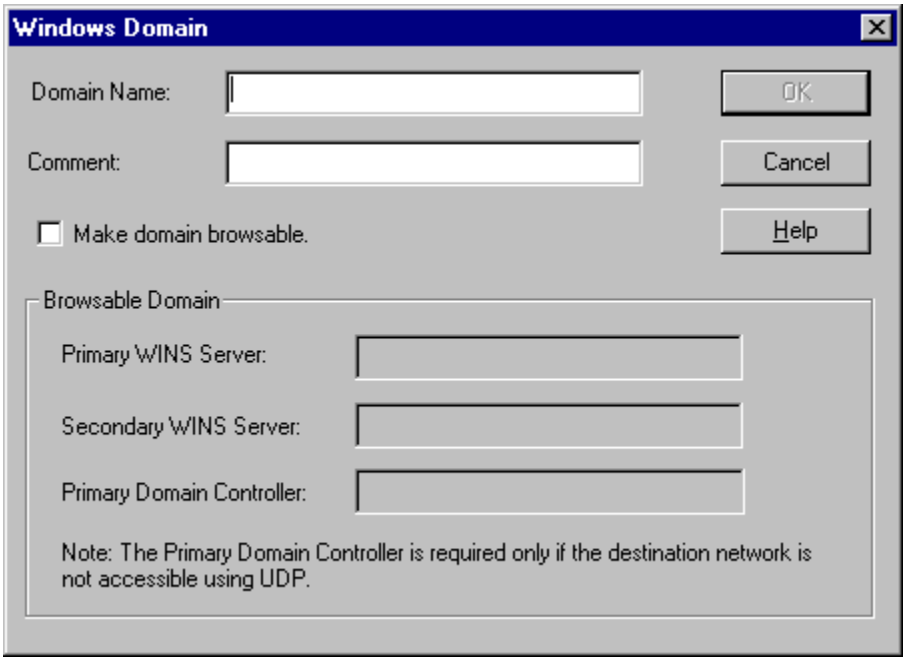
If the controls in this window are disabled (dimmed), then the hosts file has been "locked." The network administrator determines which, if any, hosts files are locked.

You can lock and unlock files from any **Extranet Neighborhood Properties** tab.

- To lock a file, use the **Ctrl+L** command.
- To unlock a file, use the **Ctrl+U** command.

Windows Domain Dialog Box

To open the **Windows Domain** dialog box, click **Add Domain** in the **Extranet Neighborhood Properties | Local** tab.



For each domain, you can either specify the WINS server names or specify each individual host that should appear in the domain. Listing WINS servers will result in a smaller, more manageable hosts file. You must add a domain before you can add hosts to that domain.

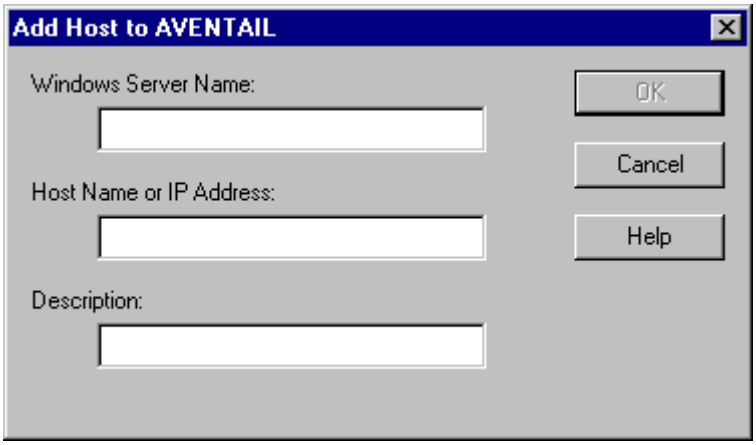
To make the specified domain “browsable,” enter WINS server information in the **Primary WINS Server** box and, if desired, the **Secondary WINS Server** box. In both of these boxes, you can enter either the server’s IP address or its fully qualified host name. You must also select the **Make domain browsable** check box. If you do not select the **Make domain browsable** check box, Extranet Neighborhood will display only those computers in the local or remote hosts file, even if you have specified a WINS server.

NOTE: *To use the browsing mode for a domain, you must specify the domain’s WINS server(s) in the hosts file. You must specify the WINS server(s) only if you want to use the browsing mode.*

To view your domains, double-click the **Extranet Neighborhood** icon on your desktop. To reload the hosts files in this screen, press the `F5` key.

Add Host to Aventail Dialog Box

To open the **Add Host to Aventail** dialog box, click **Add Host** on the **Extranet Neighborhood Properties | Local** tab.

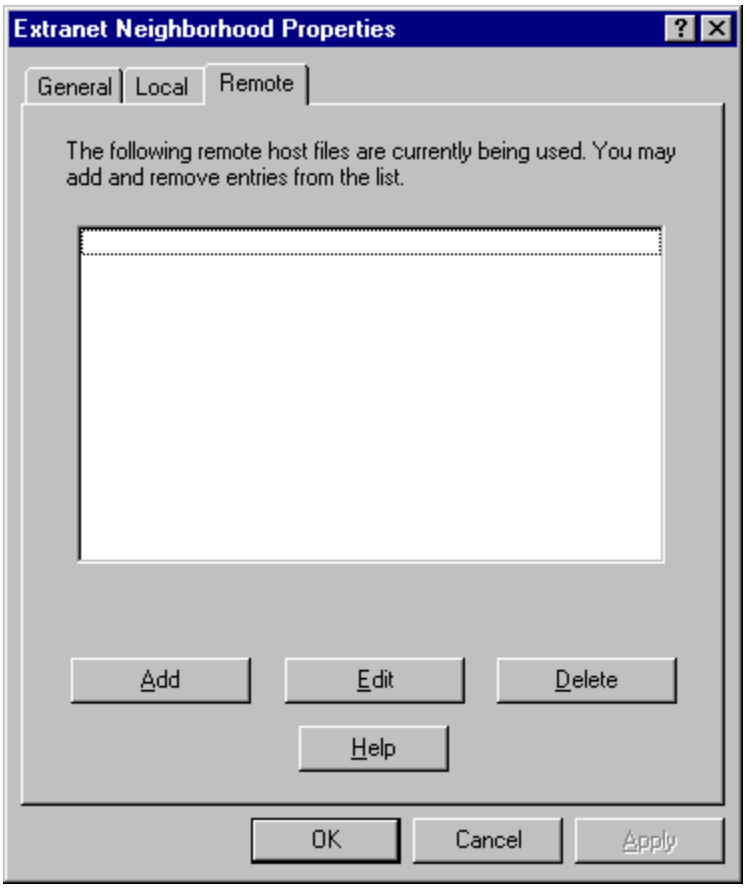


Aventail Connect automatically places hosts within the domain that is selected when you click **Add Host**. Select the correct domain before clicking **Add Host**. You must specify a domain before you can add hosts to that domain.

In the **Host name or IP address** box, be sure to enter the server’s Internet address, not its Windows machine name.

THE REMOTE TAB

If the network administrator has configured Extranet Neighborhood to use a remote hosts file, this tab displays the information about the currently configured remote hosts file(s). Server name, host name or address, pathname, and username are all configurable through the **Remote** tab.



Remote hosts files are always used in conjunction with a local hosts file. When you add a remote hosts file to the list, Extranet Neighborhood adds the path to the local hosts file. Extranet Neighborhood always has a single local hosts file; this file can include references to multiple remote hosts files.

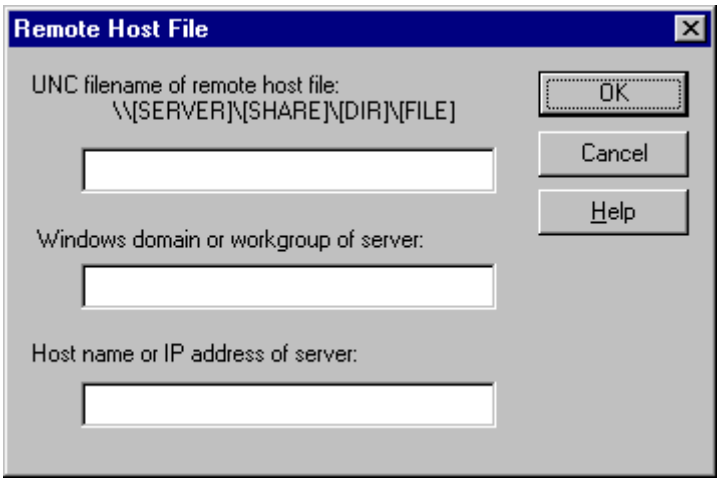
The most common configuration is one remote hosts file (with all domains and hosts in the remote hosts file) and one local hosts file that contains a pointer to the remote hosts file. If you want users to share a common hosts file, and if you want to simplify administration, use a remote hosts file.

To add entries to the list of remote hosts files, click **Add**. The **Remote Hosts File** dialog box appears, and you can type the names of the remote hosts file(s) you want to add.

NOTE: *To access remote hosts files, Aventail Connect must be running and configured correctly.*

Remote Hosts File Dialog Box

To open the **Remote Hosts File** dialog box, click **Add** on the **Remote** tab.



When entering the Universal Naming Convention (UNC) filename of the remote hosts file that you are adding, note that the [SERVER] name is the Windows machine name, not its IP address or hostname.

In the **Host name or IP address of Server** box, be sure to enter the server's Internet address, not its Windows machine name.

NOTE: *Extranet Neighborhood ignores any remote hosts files that it cannot access.*

Troubleshooting

Aventail Connect-related problems tend to fall into four categories: Installation, Network Connectivity, Configuration, and Application and TCP/IP Stack Interoperability.

AVENTAIL CONNECT INSTALLATION PROBLEMS

When the instructions in "Installing" in the *Administrator's Guide* are followed, Aventail Connect installation problems rarely occur. When they do occur, they are often the result of:

- **Toolbars, virus-checking utilities, or other Windows applications running during the installation**

  If any of these are running during a failed installation, close them, uninstall Aventail Connect, reboot, and then re-install Aventail Connect, ensuring that the toolbars, virus-checking utilities, or applications are not automatically restarted when the system reboots.

- **Insufficient RAM or free space on the volume to which Aventail Connect is being installed**

  If you suspect either of these as the cause of a failed installation, increase the available resources and retry the installation.

- **Corrupted Aventail Connect installation media, or corrupted or incomplete FTP of Aventail Connect self-extracting, executable installation file**

  If you suspect corrupted Aventail Connect installation diskettes as the cause of a failed installation, contact Aventail Technical Support (206.215.0078) for assistance in determining whether the files on the diskettes may have been corrupted and whether Aventail or your vendor must supply replacement diskettes.

  If you suspect a corrupted or incomplete FTP transfer of Aventail Connect installation files obtained over the Internet, retry the transfer, taking care to ensure that the FTP client is in binary mode and confirm that the transfer completes normally. Contact Aventail Technical Support to confirm that the byte size of the transferred installation file is correct.

- **Installation to a workstation on which Aventail Connect was running or from which a previous version of Aventail Connect was not completely uninstalled**

  If you suspect either of these circumstances as the cause of a failed installation, contact Aventail Technical Support.

- **Installation script errors**

  Aventail Connect is installed with InstallShield. If InstallShield reports errors during a failed installation, note the text of the error messages and the specific circumstances in which they occurred and contact Aventail Technical Support.

NETWORK CONNECTIVITY PROBLEMS

Before Aventail Connect can successfully redirect WinSock application connections:

1. The workstation on which Aventail Connect is installed must also have a properly installed, WinSock-compatible, TCP/IP stack running on it.

   This installation can be confirmed by successfully pinging the IP address of the workstation, from the workstation itself, using a WinSock ping application. If this test fails, the failure must be corrected before Aventail Connect can be tested and before Aventail Technical Support can provide assistance.

2. Basic TCP/IP network connectivity must exist between the client workstation on which Aventail Connect is installed and the extranet (SOCKS) server(s) to which it is configured to redirect connections.

   This connectivity can be confirmed by successfully pinging the extranet server(s) by IP address, from the client workstation. If this test fails, the failure must be corrected before Aventail Connect can be tested and before Aventail Technical Support can provide assistance.

3. Basic TCP/IP network connectivity must also exist between the extranet server(s) and the network host(s) to which the extranet server(s) are expected to proxy connections.

   This connectivity can be confirmed by successfully pinging the network host(s), by IP address, from the extranet server(s). If this test fails, the failure must be corrected before Aventail Connect can be tested and before Aventail Technical Support can provide assistance.

AVENTAIL CONNECT CONFIGURATION PROBLEMS

This section addresses troubleshooting of simple Aventail Connect configuration problems. Troubleshooting complex Aventail Connect configuration problems is beyond the scope of this section.

It is easiest to troubleshoot Aventail Connect configuration problems by creating and testing simple Aventail Connect configuration files, such as those that may be created with the Aventail Connect configuration wizard. However, all references to host and domain names must be removed from configuration files created with the wizard, before testing, to defer possible name resolution complications until the files can be demonstrated to work with IP addresses alone.

NOTE: *The IP address and SOCKS port number of the extranet (SOCKS) server(s) to which Aventail Connect must connect must be known before troubleshooting Aventail Connect configuration problems. Neither Aventail Connect, nor Aventail Technical Support, can discover the IP address or port number of the extranet server(s).*

When troubleshooting Aventail Connect configuration problems, confirm that the Aventail Connect configuration file that is currently selected in the **Configuration File** dialog box is the one intended for testing.

After selecting a configuration file to test, open the Aventail Connect Config Tool and:

1. Confirm that the extranet server has been correctly identified by IP address.

   Click the **Servers** tab, select the server alias and then click **Edit…**. Compare the IP address in the **Hostname or IP** box with that of the extranet server.

   If the extranet server is a SOCKS v5 server, click **SOCKS v4** in the “SOCKS Version” area of the **Servers** tab. Then click **Detect Version**. The selection will revert to **SOCKS v5**, indicating that Aventail Connect detected a SOCKS v5 server running at the IP address specified in the **Hostname or IP** box.

   If, on the other hand, the extranet server is a SOCKS v4 server, click **SOCKS v5** in the “SOCKS Version” area. Then click **Detect Version**. The selection will revert **SOCKS v4**, indicating that Aventail Connect detected a SOCKS v4 server running at the IP address specified in the **Hostname or IP** box.

   If **Detect Version** fails to detect an extranet server of either version, it is possible that no extranet server is running on the host identified in the **Hostname or IP** box. Contact your extranet server administrator to confirm that the extranet server is running at the address specified.

2. Confirm that all Aventail Connect authentication modules are enabled.

   Click the **Authentication** tab and confirm that the “traffic light” icons for all of the authentication Modules are green, indicating that the modules are enabled. Enabling all the modules configures Aventail Connect to attempt any form of authentication demanded by the extranet server or null (no) authentication. Note the form of authentication demanded by the extranet server and, if necessary, obtain the proper authentication credentials, such as an extranet server username and password, from the extranet server administrator.

3. Confirm that the network hosts to which the extranet server is expected to proxy connections are within a redirected destination.

   Click the **Destinations** tab, select the destination that includes the network host to which the extranet server is expected to proxy connections, and then click **Edit…**. Confirm that the definition of the Destination includes the network host.

   Next, click the **Redirection Rules** tab. Confirm that connections to the Destination are configured to be redirected by the extranet server.

After making any necessary changes to the Aventail Connect configuration, restart Aventail Connect and then restart any WinSock applications before testing the new configuration.

APPLICATION AND TCP/IP STACK INTEROPERABILITY PROBLEMS

Aventail Connect is intended to "automatically socksify" all "well-behaved" WinSock applications. Occasionally, you may find WinSock applications that Aventail Connect does not socksify, due to interoperability problems with the application.

Aventail Connect is also intended to run on all WinSock-compliant Microsoft Windows TCP/IP stacks. Aventail Connect does not alter or replace WinSock or any other core TCP/IP components (files) provided by the operating system. Occasionally, you may find WinSock stacks on which Aventail Connect does not run as expected, due to interoperability problems with the stack.

If you suspect an application or stack interoperability problem, report it to Aventail Technical Support. Aventail will make every reasonable effort to resolve interoperability problems.

AVENTAIL CONNECT TRACE LOGGING

Aventail Connect includes a Logging Tool for tracing Aventail Connect and WinSock activity. Aventail Connect traces are often useful in troubleshooting Aventail Connect network, extranet server, and WinSock application interoperability problems. Aventail Technical Support engineers may request that you perform a verbose trace, log it to a file, and e-mail it to them as an attachment.

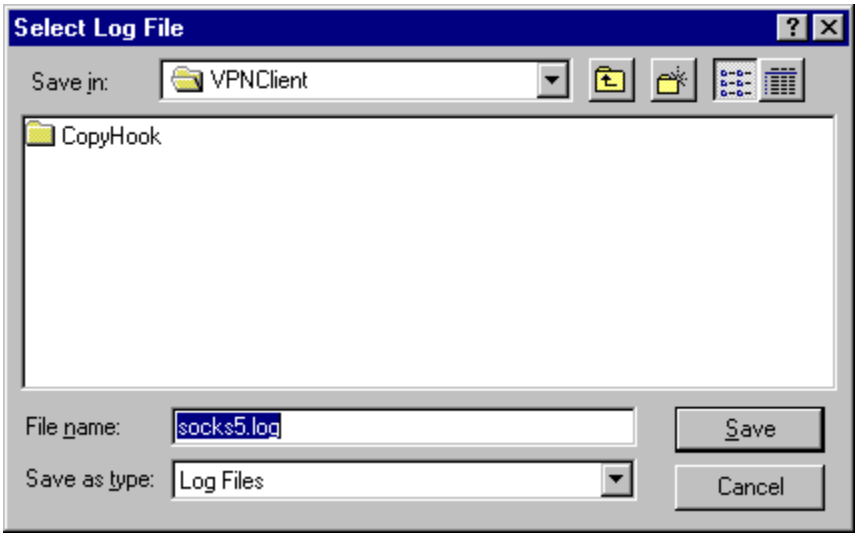
To run an Aventail Connect trace

1. Close any WinSock applications that are running on the workstation.
2. If Aventail Connect is running, close it and then restart it.
3. Start an Aventail Connect trace.

   In Windows 95, Windows 98, and Windows NT 4.0, right-click the minimized **Aventail Connect** icon in the system tray, and click **Logging Tool**. In Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51, double-click the **Logging Tool** icon in the Aventail program group. The Aventail Connect **Logging Tool** window will open, as illustrated in Figure 1, below.
4. On the **Log** menu, confirm that the **Trace** command is checked. If it is not, click **Trace** to enable it.

To save an Aventail Connect trace to a file

1. On the **Log** menu, confirm that the **Log To File** command is checked. If it is not, click **Log To File** to enable it.
2. The **Select Log File** dialog box (shown below) appears. Enter a file name and click **Save**.



ERROR MESSAGES

Occasionally, you may see an error message while running Aventail Connect. The following table explains some of the more common Aventail Connect error messages.

Error Message	Meaning
Setup has determined that your computer does not have this support and needs the WinSock 2.0 patch, available from Microsoft.	SETUP: To install Aventail Connect 3.01, you must first install the Microsoft WinSock 2.0 upgrade.
The patch is available for download on the Microsoft Web site, at **www.microsoft.com/windows/downloads/contents/Updates/W95Sockets2/default.asp**.	SETUP: Location of the Microsoft WinSock 2.0 upgrade.
You must have administrator privileges to install.	SETUP: On Windows NT machines, you must have administrative privileges to install or uninstall Aventail Connect.

Error Message	Meaning
Setup has detected that a previous installation of (....) is present. Would you like to continue and upgrade to (....)? Pressing NO will leave your existing installation intact and will cause Setup to terminate.	SETUP: Retain the previous installation of Aventail Connect by pressing NO. Replace with the newer installation by pressing YES.
The package does not contain the necessary 3.01 files. Please contact your administrator.	SETUP: Setup cannot find the necessary Aventail Connect 3.01 files.
The package does not contain the necessary 2.51 files. Please contact your administrator.	SETUP: Setup cannot find the necessary Aventail Connect 2.51 files.
The file you have selected is not a valid Aventail setup file. Would you like to create it?	CUSTOMIZER: Create a new setup file, or retain a previous setup file.
Customizer must be run from a valid Customize directory. Your changes will not be saved.	CUSTOMIZER: Must run Customizer from a valid Customize directory.
The Connect executable does not have a valid Aventail digital signature.	The specified signature is not valid.
Connect cannot find your license file, aventail.alf.	Aventail Connect cannot find a valid Aventail license file, aventail.alf.
Connect cannot load because your license file does not contain a license.	The license file exists, but it contains no license.
This version of Connect does not support HTTP servers.	Aventail Connect 2.51 does not support HTTP servers.

REPORTING AVENTAIL CONNECT PROBLEMS

Report Aventail Connect problems to Aventail Technical Support by completing and submitting an Online Support form on the Support page of the Aventail Web site, **http://www.aventail.com**.

Glossary

ALIAS

User-friendly name for destination network or host computer.

AUTHENTICATION

A method for identifying a user in order to establish access to a system resource or network. Authentication information such as username/password is entered via prompts.

CERTIFICATE

A certificate is essentially an electronic "statement" which verifies that a certain RSA public key is associated with a particular name. Certificates are issued by a Certification Authority (CA).

CLIENT

A program or Internet service that sends commands to and receive information from a corresponding program known as a server. Most Internet services run as client/server programs.

CONFIGURATION FILE

A file of information containing traffic redirection rules used to determine if and how SOCKS redirection should occur.

CREDENTIALS

Credentials include the information (such as username/password) that you enter when establishing a connection to a SOCKS server requiring user authentication.

DOMAIN

Internet name for a network or computer system.

ENCRYPTION

A security procedure that converts data into a format which can be read only by the intended recipient computer.

EXTRANET

A network that is partially accessible to outsiders.

FIREWALL

Software or hardware barriers that control the flow of information to Private networks.

GATEWAY

A communications device/program that passes data between networks.

HACKER

A person who enjoys using computers and has a thorough understanding of how they work, as well as the networks they run on. Often used to mean "cracker," the correct term for someone who accesses computer systems without authorization.

HOST

A server connected to the Internet.

IETF

Internet Engineering Task Force: An open community of network designers, vendors, etc. who resolve protocol and architectural issues for the quickly evolving Internet.

INTERNET PROTOCOL (IP)

The basic data transfer protocol used for the Internet. Information such as the address of the sender and the recipient is inserted into an electronic “packet” which is then transmitted.

INTRANET

A network that is internal to a company or organization.

LAN

Local area network

LAYERED SERVICE PROVIDER (LSP)

A program that is installed just below WinSock 2.0, allowing two-way communication between the WinSock 2.0-compatible application and the underlying TCP/IP stack. An LSP can redirect and/or change data before sending the data to the operating system’s TCP/IP stack for transport over the network.

LOG WINDOW

The window of the Logging Tool which shows alerts, messages, and warnings generated by Aventail Connect.

PING

A utility that determines if a remote host computer is up. ping sends data packets to the host. If the packets are not returned, the host is down.

PROTOCOL

Rules and procedures used to exchange information between networks and computer systems.

REDIRECTION RULES

Rules defined in the configuration file which specify how network requests are routed to SOCKS servers.

ROUTER

A device that transmits traffic between networks

SERVER

A networked computer that shares resources with other computers. Servers “serve up” information to clients.

SMB

Server Message Block. A message format used by DOS and Windows for sharing files, directories, and other resources.

SOCKS

SOCKS is a security protocol. It acts as a proxy mechanism that manages the flow and security of data traffic to and from your local area network or intranet.

SSL

Security Sockets Layer. An authentication and encryption protocol.

TRACEROUTE

A utility that traces the routing of data over the Internet to a specific computer. Traceroute sends a data packet and then lists the intermediate host computers that it traverses on it’s way to the destination machine.

TRANSMISSION CONTROL PROTOCOL (TCP)

A means of sending data over the Internet with guaranteed delivery.

TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL (TCP/IP)

A suite of protocols the Internet uses to provide for services such as e-mail, ftp, and telnet.

USER DATAGRAM PROTOCOL (UDP)

A means of sending data over the Internet without guaranteed delivery. Also known as “connectionless” protocol, it is used for data such as RealAudio®.

UNIVERSAL NAMING CONVENTION (UNC)

A way of accessing a file or directory on another computer. For example: // host/share/directory/file (“share” refers to the alias used to make the resource available.)

VIRUS

A self-replicating code segment that can infect a computer or network, causing minor to major damage

VPN

Virtual Private Network: A secure channel used to transmit data over a public network

WINSOCK

Windows Sockets. A Windows component that connects a Windows PC to the Internet using TCP/IP.

WORKSTATION

Any computer connected to a network.

X.509

An ISO format standard for client and server certificates.

W

X

EXHIBIT H

TO MICHAEL FRATTO'S DECLARATION

PR NEWSWIRE, "AVENTAIL SHIPS DIRECTORY-ENABLED EXTRANET SOLUTION; AVENTAIL EXTRANET CENTER V3.1 AVAILABLE AT WWW.AVENTAIL.COM." (AUGUST 9, 1998)

Aventail Ships Directory-enabled Extranet Solution; Aventail Extranet Center V3.1 Available At www.aventail.com.

Publication: **Business Wire** Publish date: **August 9, 1999**

SEATTLE--(BUSINESS WIRE)--Aug. 9, 1999--

Aventail Corporation, the leading provider of Extranet Management and Security (EMS) solutions, announced today that they have shipped the latest versions of its award-winning product, Aventail ExtraNet Center(tm).

This latest offering simplifies extranet user management by including broader support for Public Key Infrastructure (PKI) and Lightweight Directory Access Protocol (LDAP)-enabled directories as well as automatic client updating. With these enhancements, Aventail has greatly simplified how large multi-enterprise organizations deploy and manage world-class extranets.

"I was quite impressed with the latest version of Aventail ExtraNet Center and its ability to seamlessly integrate with various LDAP directories and PKI environments," stated Ken Aull, Technical Fellow for TRW. "Aventail ExtraNet Center is a perfect tool for any enterprise organization conducting business-to-business commerce and collaboration. Aventail's standards-based solution reassures corporations that their extranet will work now as well as in the future."

Simplifying User Management

Aventail ExtraNet Center's latest features enable deployments that are easily scalable for any number of extranet users. These new features include:

-- LDAP-enabled Authentication and Authorization: Aventail ExtraNet

Center v3.1 allows corporations implementing various LDAP

directories, including Netscape (NYSE:AOL) Directory Server, IBM

(NYSE:IBM) SecureWay Directory, and Lotus Domino, to authorize

users and groups from an authoritative directory. Aventail's LDAP

implementation complements its existing support for NDS and

Bindery, RADIUS, Windows NT Domain, UNIX Passwrd Files, and

Security Dynamics' ACE/Server. -- Increased Support for Emerging PKI Standards: Adding broader PKI

support gives users more choices when using digital certificates.

These include the ability to acquire a certificate via a browser

(PKCS #12) and support for smart card and other device-based

authentication (PKCS #11) such as SPYRUS' Rosetta Smart Card. The

expanded suppourID, Hewlett-Packards' (NYSE:HWP) Authorization Server, and

x.509 certificates from VeriSign (Nasdaq:VRSN), Netscape

(Nasdaq:NSCP), GTE (Nasdaq:GTE), and Microsoft (Nasdaq:MSFT). -- Automated Configuration Updates: Aventail ExtraNet Center also

includes the ability to easily configure, distribute, and

automatically update client configuration files. Utilizing

Aventail Customizer(tm), administrators can easily configure and distribute pre-packaged clients via e-mail, FTP, HTTP, or application deployment products such as Microsoft's SMS. After deployment, Aventail's AutoUpdate(tm) allows new configuration files to be automatically installed without user intervention at an interval set by the administrator. This easy-to-use administrative tool reiterates Aventail's efforts in providing a transparent client for business partners, suppliers, consultants, and customers.

Pricing and Availability

Aventail ExtraNet Center is currently shipping on Windows NT and Solaris. Additional platform support will be available at the end of August.

Aventail ExtraNet Center is available through Aventail's worldwide sales team and Aventail Extranet Advantage VAR partners as well as leading security vendors such as Hewlett-Packard and BullSoft. Pricing begins at $10,000 depending on client requirements.

A Comprehensive Solution for Extranet Management and Security

Aventail ExtraNet Center is a client/server software solution that includes integrated encryption, authentication and authorization services using the popular IETF standards SSL and SOCKS v5.

Aventail ExtraNet Center has the ability to seamlessly integrate into any existing infrastructure, making it less costly to install and easier to implement and support than other extranet solutions. Aventail ExtraNet Center works with any IP-based application, including legacy host, mainframe, Java, CORBA-based, custom corporate, and client/server applications from vendors such as SAP (NYSE:SAP), BAAN (Nasdaq:BAANF), Oracle (Nasdaq:ORCL), and PeopleSoft (Nasdaq: PSFT). Aventail ExtraNet Center can also traverse any firewall, such as Check Point Software Ltd.'s (Nasdaq:CHKP) Firewall-1/VPN-1, AXENT Technologies' (Nasdaq:AXNT) Raptor Firewall, and IBM's (NYSE:IBM) eNetwork Firewall.

About Aventail Corporation

Founded in 1996, Aventail has quickly emerged as the leading provider of EMS solutions for the Global 2000. Aventail's solutions allow organizations to securely extend their enterprise resources to strategic partners, suppliers, customers, consultants and other key individuals over public IP networks.

Leading corporations are using Aventail's solutions to help them increase their competitive advantage, raise profits, and leverage investments in existing and future enterprise systems. Aventail's solutions are currently deployed at companies such as Aetna, Bear Stearns, Kodak, Hewlett-Packard, IBM, IKON, Marriott, and Xerox. With a strong reputation for providing highly secure and easy-to-manage software solutions, Aventail has received numerous industry awards from publications and industry analyst firms such as Giga Information Group, InfoWorld, Network Computing, LAN Times, BYTE Magazine, Software Digest, and Computer Reseller News.

Aventail Corporation is privately held and headquartered in Seattle, Washington. For more information on the company or to download a trial version of Aventail ExtraNet Center, please visit www.aventail.com, or contact the company directly at 206-215-1111, 877-AVENTAIL, or info@aventail.com. Information on Aventail can also be obtained through Yahoo (Nasdaq:YHOO), Infoseek (Nasdaq:SEEK), Lycos (Nasdaq:LCOS), and Excite (Nasdaq:XCIT).

Aventail is a registered trademark of Aventail Corporation. Aventail ExtraNet Center, Aventail Customizer and Aventail AutoUpdate are trademarks of Aventail Corporation. All other trademarks are the property of their respective owners.

EXHIBIT I

TO MICHAEL FRATTO'S DECLARATION

"INTRANET APPLICATIONS: BRIEFS," NETWORK WORLD, AT PAGE 55 (OCTOBER 19, 1998)

Wireless e-mail: Must have or pie in the sky?
Paul McNamara
Network World; Oct 19, 1998; 15, 42; ABI/INFORM Global
pg. 55

Briefs

■ **Aventail Corp.** *last week introduced the Aventail ExtraNet Center 3.0. This* **client/server package** *provides access controls, user-based authentication and key-certificate management and active filtering for business partners and suppliers who communicate over the Internet.* **The Aventail ExtraNet Center,** *which starts at $7,995, is available for Windows NT 4.0, Linux 2.X, and Unix platforms from Digital, Sun and Hewlett-Packard.*
✆ Aventail: (206) 215-1111

■ *John Manley, Canadian Minister of Industry, recently announced the government of* **Canada** *wants to make it easier to export products with encryption features in order to* **encourage electronic commerce.** *Canada will streamline export procedures with a one-time review process for even the strongest encryption, without requiring key-recovery features. More information is available at the Canadian government's Web site at http://info.ic.gc.ca/cmb/welcomeic.nsf/Pages/release/e.htm.*

Manley *opens Canada for export.*

■ *IBM has released the beta of its* **HotMedia Web multimedia toolkit,** *a set of Java applets and assembly tools that let Web developers add sound and video clips to Web presentations. IBM is integrating HotMedia into the* **IBM** *electronic commerce server,* **net.commerce.** *Several other electronic catalog companies, including iCat, InterShop and Open Market, are beta-testing HotMedia with an eye toward the same goal. Now available for free download at ir.software.ibm.com/netmedia, HotMedia will have a licensing fee when it formally ships by year-end.*

Wireless e-mail: Must have or pie in the sky?

By Paul McNamara
Cambridge, Mass.

Messaging vendors have been tripping over one another recently in a mad dash to provide wireless e-mail support for popular handheld devices and cell phones.

However, whether this product development and marketing frenzy signals the next big thing or wishful thinking on the part of vendors depends upon whom you ask.

Christopher Herot, senior director of the Mobile Communications Group at Lotus, insists his company's internal research has detected both current demand and significant growth potential for wireless e-mail.

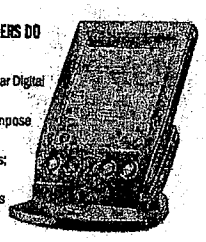
"About 20% of customers say they're under pressure to provide pager connection or mobile e-mail connection today," says Herot.

When those customers were asked if they will require such support within the next year, that demand "goes up to 25% and within three years it's almost everybody," he contends.

The Cambridge, Mass., IBM subsidiary recently unveiled its Wireless Domino Access program, which allows customers to access e-mail wirelessly from a Domino Server using the 3Com PalmPilot, Palm III or IBM WorkPad. The software works in conjunction with a Minstrel Wireless IP Modem from Novatel Wireless and a microbrowser from Unwired Planet.

WIRELESS DOMINO LETS PALM USERS DO THE FOLLOWING:

- Get IP access to Domino via Cellular Digital Packet Data.
- Read, reply, forward, fax and compose e-mail messages.
- View and fax calendar schedules; create new entries.
- Search for names and addresses in directories.



Lotus has been particularly active but by no means alone among messaging vendors that are enhancing their wireless and remote-access offerings, often with the help of partnerships and third-party providers.

Microsoft and Novell, for example, have been pitching various mobile mechanisms for accessing their respective Exchange and GroupWise servers.

Customers can expect to hear more about such features, according to David Ferris, an analyst with Ferris Research in San Francisco.

"Lotus wants to provide good support for mobile professionals and is taking a leading role among leading software vendors in providing that support," Ferris says.

"PDA integration will be a hot area for users and vendors alike over the next couple of years," he adds.

Right now, at least, it remains easier to find skeptics and window shoppers than customers who are widely deploying these wireless

See **Messaging,** *page 56*

In-Site

Swiss bank battens down Web hatches

By Ellen Messmer
Zurich, Switzerland

Mindful of hackers determined to break into Web servers, Union Bank of Switzerland (UBS) took a long, hard look at how to securely offer its wide array of financial services on the Internet when the Swiss banking giant entered online banking earlier this year.

Aware of the critical nature of banking transactions, UBS opted for a customized Web server built according to the U.S. military's B1 operating system security rating, which calls for mandatory access controls and compartmentalized services. UBS not only ordered a Web server built to military security specifications, but it also integrated a home-grown Web authentication application, Benutzbewachtigungssysteme, into the system.

The Web became an issue when UBS business units began clamoring to offer banking services globally via the 'Net and demanded that the UBS IT division find a way to do it, says Silvano Caliaro, executive director of UBS IT services. Caliaro oversees a staff of 4,000 supporting the UBS TCP/IP network and applications worldwide.

UBS' Caliaro *helps protect vital information.*

"The pressure from the business managers was very high," he notes. "Our experts asked questions of the business managers, and we felt we needed to develop this secure server."

After a review of proposals, UBS last year picked Champaign, Ill., company Argus Systems Group to build the Web server. Argus, which has sold a B1-accredited trusted operating system for four years, spent several months building the Web server for UBS.

"Our Gibraltar operating system and Web server module is installed on a standard off-the-shelf Solaris system," explains Argus President Randy Sandone. The advantage of the B1 architecture is it diminishes the hacker's ability to exploit buffer overflows to gain root access.

Gibraltar, which encrypts data between the user and the UBS back-end systems, provides isolated compartments for running multiple applications to access this legacy data. On the Web server, UBS is running four applications

See **UBS,** *page 56*

Get more online:

- **An overview of the Argus Gibraltar server used by UBS.**
- **A look at how U.S. banks are testing digital certificates.**

www.nwfusion.com

EXHIBIT J

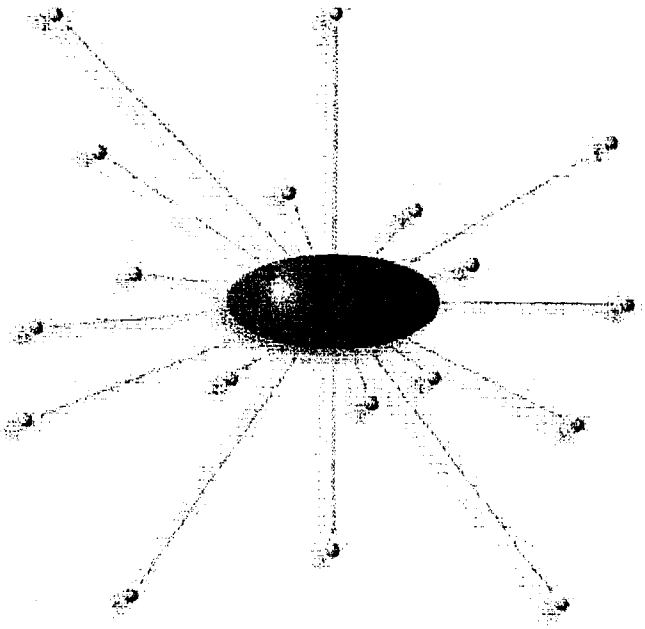
TO MICHAEL FRATTO'S DECLARATION

AVENTAIL CONNECT V3.1/V2.6 ADMINISTRATOR'S GUIDE

Aventail

CONNECT

v3.1/v2.6



Administrator's Guide

Windows

Aventail

AVENTAIL CONNECT 3.1/2.6 ADMINISTRATOR'S GUIDE

808 Howell Street, Second Floor
Seattle, WA 98101
USA

http://www.aventail.com/

Printed in the United States of America.

TRADEMARKS AND COPYRIGHTS

Table of Contents

UTILITIES REFERENCE GUIDE

TROUBLESHOOTING

Introduction

Welcome to the Aventail Connect 3.1/2.6 secure Windows client for 16- and 32-bit Windows applications. The client component of the Aventail ExtraNet Center, Aventail Connect is a secure proxy client based on SOCKS 5, the IETF standard for authenticated firewall traversal. Aventail Connect delivers enhanced security and simplifies SOCKS deployment for users and network managers.

Aventail Connect redirects WinSock calls and reroutes them based upon a set of routing directives (rules) assigned when Aventail Connect is configured. (For more information about WinSock, TCP/IP, and general network communications, see "Getting Started.")

On larger networks, Aventail Connect can address multiple SOCKS 5 servers based on end destination and type of service. This feature enables network administrators to effectively monitor and direct network traffic.

Aventail Connect is a proxy client, but when used with SSL it provides the ability to encrypt inbound or outbound information.

Features of Aventail Connect:

- Aventail Connect supports X.509 client certificates for strong authentication with SSL (when encryption is enabled)
- Automated Customizer utility simplifies client configuration, distribution, and installation
- SSL compression detects low bandwidth connections and compresses encrypted data (when encryption is enabled)
- Secure Extranet Explorer (via **Extranet Neighborhood** icon on desktop) allows users to securely access Windows or SMB hosts over an extranet connection (Windows 95, Windows 98, and Windows NT 4.0 only)
- Supports WinSock 2 (LSP) applications in Windows 98, and Windows NT 4.0, and WinSock 1.1 and WinSock 2 applications in Windows 95
- Supports WinSock 1.1 applications in Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51
- MultiProxy feature allows you to use a SOCKS server or an HTTP proxy to control outbound access
- Allows the use of port ranges for redirection rules
- Provides integration with SoftID™ and SecurID™ tokens
- Provides automated installation and uninstallation
- Credential cache timeout feature allows administrators to specify when credentials expire
- Provides optional password protection for configuration files
- Supports both SOCKS v4 and SOCKS v5 (RFC 1928 and RFC 1929) standards

- Enables network redirection through successive extranet (SOCKS) servers
- Includes a logging utility to troubleshoot problems with network connections
- Includes a Configuration wizard for simplified step-by-step creation of configuration files
- Allows internal network connections to pass through without interference
- Supports multiple authentication methods including SOCKS v4 identification, username/password, CHAP, CRAM, HTTP Basic (username/password), and SSL 3.0

NOTE: *Not all versions of Aventail Connect have encryption enabled.*

About This Document

This *Administrator's Guide* provides basic information about Aventail Connect. It includes entry-level data for non-technical users, plus installation, setup, and configuration information for network administrators. This information is also available via Aventail Connect Help and the Aventail Web site at **http://www.aventail.com/content/products/docs/**.

Document Organization

This document is divided into three main sections: *Administrator's Guide*, *Utilities Reference Guide*, and *Troubleshooting*.

The *Administrator's Guide* describes procedures for setting up, installing, and configuring Aventail Connect for individual and multiple networked workstations. It also describes how to create a customized Aventail Connect package for distribution to multiple users.

The *Utilities Reference Guide* describes the Aventail Connect system menu commands and utility programs. It contains detailed information about using the S5 Ping utility and the Logging Tool, and documents the authentication/encryption modules and settings.

The document concludes with *Troubleshooting* and the *Glossary*.

You can also use the Quick Start Card, a short document designed to help you install Aventail Connect to an individual workstation, and the Aventail Connect flowchart, at **http://www.aventail.com/contents/solutions/presentations/quickstart/vpnclient.pdf**.

DOCUMENT CONVENTIONS

The following typographic conventions are used in this document. Exceptions may be made for online material; for instance, italics may be difficult to read online.

Convention	Usage
`Courier font`	Filenames, extensions, directory names, keynames, and pathnames. Command-line commands, options, and portions of syntax that must be typed exactly as shown.
Bold	Dialog box controls (**Edit...** buttons), e-mail addresses (**support@aventail.com**), URLs, (**www.aventail.com**), and IP addresses (**165.121.6.26**).
Italic	Placeholders that represent information the user must insert.

SEE ALSO: *A reference to additional useful information.*

NOTE: *Information the user should be aware of to increase understanding and/or efficiency of the software.*

CAUTION: *An operational item that the user should be aware of to avoid a network policy/software conflict, or lapse, which may create a MINOR security flaw.*

WARNING: *An operational item that the user should be aware of to avoid a network policy/software conflict, or lapse, which may create a SERIOUS security flaw.*

AVENTAIL TECHNICAL SUPPORT

Contact Aventail Technical Support if you have questions about installation, configuration, or general usage of Aventail Connect. Refer to the Aventail Support Web site, at **http://www.aventail.com/index.phtml/support/online_support.phtml**, or the Aventail Knowledge Base, at **http://www.aventail.com/index.phtml?page_id=03110000**, for the latest technical notes and information. Refer to the `readme.txt` documentation for additional information not included in the *Administrator's Guide*.

Aventail Technical Support:
Web site: **http://www.aventail.com/index.phtml/support/index.phtml**
E-mail: **support@aventail.com**
Phone: 206.215.0078
Fax: 206.215.1120

ABOUT AVENTAIL CORPORATION

Aventail Corporation is the leading vendor of extranet software. Its extranet solutions allow organizations to secure their networked communications and manage their employees' access to the Internet. Building an extranet gives organizations the ability to dynamically create a private communication or data channel over the Internet. Aventail's adherence to open security standards simplifies extranet deployment, enables interoperability, and leverages corporations' existing network investments. Its extranet solutions allow companies to extend the reach of their corporate extranets to customers, partners, remote offices, and worldwide employees.

Aventail Corporation
808 Howell Street, Second Floor
Seattle, WA 98101
Phone:206.215.1111
Fax:206.215.1120
http://www.aventail.com/
info@aventail.com

An aventail is a piece of chainmail armor worn around the neck area. In the 14th century, knights wore an aventail to protect themselves while in combat. Today, Aventail continues the tradition of protection by allowing organizations to securely communicate over the Internet.

Administrator's Guide

This section includes procedural and background information on installing Aventail Connect on both single and networked workstations. It includes:

- "Getting Started," with brief explanations of network security and communications
- Definitions of SOCKS and Aventail Connect
- Aventail Connect platform and installation requirements, with an introduction to WinSock 2 and LSP architecture
- "Installing Aventail Connect," which includes network diagrams of Aventail ExtraNet Center and SOCKS v4-based server configurations
- Directions on how to create and edit configuration files, and an introduction to the Aventail Customizer

NOTE: *Aventail understands the importance of a flexible, easy-to-use installation process. If you have feedback regarding the Aventail Connect installation procedures, or if there are additional features you want to see implemented, please e-mail comments to* **support@aventail.com**. *Your input is appreciated.*

GETTING STARTED

If you are new to Aventail Connect technology, the following section will help you understand what Aventail Connect is and does, and its relationship to network security in general.

NETWORK SECURITY IN A NUTSHELL

Escalating security threats are forcing companies to seek ways to safeguard their corporate networks and the information they exchange. The first response to these concerns has been the development of security firewalls—software barriers that control the flow of information. But firewalls are not designed to handle complex security issues, such as monitoring network usage, providing private communication over public networks, and enabling remote users to gain secure access to internal network resources.

Enter SOCKS v5, an Internet Engineering Task Force (IETF)-approved security protocol targeted at securely traversing corporate firewalls. SOCKS was originally developed in 1990, and is now maintained by NEC. SOCKS acts as a circuit-level proxy mechanism that manages the flow and security of data traffic to and from your local area network (LAN) or extranet. An application whose traffic

is proxied by SOCKS is considered "socksified." SOCKS is more than a standard security firewall. Other features:

- Client Authentication: (SOCKS v5 only) Authentication allows network managers to provide selected user access to internal and external areas of a network.
- Traffic Encryption: (SOCKS v5 only) Encryption ensures that network traffic is private and secure.
- UDP Support: (SOCKS v5 only) User Datagram Protocol (UDP) traffic has traditionally been difficult to proxy, with the exception of SOCKS v5.
- Aventail Connect supports X.509 client certificates within SSL.
- Cross-Platform Support: Unlike many other security solutions, SOCKS can be used on various platforms, such as Windows NT, Windows 95, Windows 98, and various forms of UNIX.

NOTE: *Not all versions of Aventail Connect include the SSL module for encryption.*

WHAT IS AVENTAIL CONNECT?

Aventail Connect is the client component of the Aventail ExtraNet Center. Aventail Connect works with the Aventail ExtraNet Server, the SOCKS 5 server component of the Aventail ExtraNet Center. You can use Aventail Connect as a simple proxy client for managed outbound access, and for secure inbound access.

Aventail Connect automates the "socksification" of Transmission Control Protocol/Internet Protocol (TCP/IP) client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol. When you run Aventail Connect on your system, it automatically routes appropriate network traffic from a WinSock (Windows sockets) application to an extranet (SOCKS) server, or through successive servers. (WinSock is a Windows component that connects a Windows PC to the Internet using TCP/IP.) The SOCKS server then sends the traffic to the Internet or the external network. Network administrators can define a set of rules that route this traffic.

Aventail Connect is designed to run transparently on each workstation, without adding overhead to the user's desktop. In most cases, users will interact with Aventail Connect only when it prompts them to enter authentication credentials for a connection to a secure extranet (SOCKS) server. Users may also occasionally need to start and exit Aventail Connect, although network administrators often configure it to run automatically at startup. Aventail Connect does not require administrators to manually establish an encrypted tunnel; Aventail Connect can establish an encrypted tunnel automatically.

To understand Aventail Connect, you first need to understand a few basics of TCP/IP communications.

TCP/IP Communications

Windows TCP/IP networking applications (such as telnet, e-mail, Web browsers, and ftp) use WinSock to gain access to networks or the Internet. WinSock is the core component of TCP/IP under Windows, and is the interface that most Windows applications use to communicate to TCP/IP.

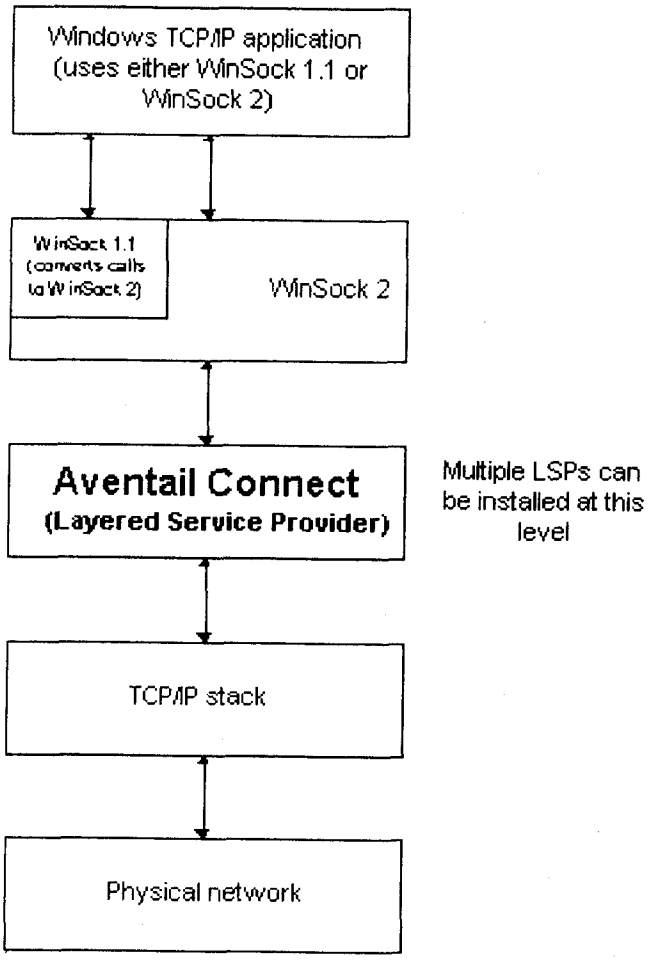
WinSock Connection to a Remote Host

Via WinSock, an application goes through the following steps to connect to a remote host on the Internet or corporate extranet:

1. The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address or, in rare cases, it will do a reverse DNS lookup to convert the IP address into a hostname. If the application already knows the IP address, this step is skipped.
2. The application requests a connection to the specified remote host. This causes the underlying stack to begin the TCP handshake, when two computers initiate communication with each other. When the handshake is complete, the application is notified that the connection is established, and data can then be transmitted and received.
3. The application sends and receives data.

WHAT DOES AVENTAIL CONNECT DO?

Aventail Connect slips in between WinSock and the underlying TCP/IP stack. (See diagram below.) As an application that sits between WinSock and the TCP/IP stack, Aventail Connect 3.1 is a Layered Service Provider (LSP). Aventail Connect can change data (compressing it or encrypting it, for example) before routing it to the TCP/IP stack for transport over the network. The routing is determined by the rules described in the configuration file.



Windows TCP/IP applications and Aventail Connect have no direct contact with one another; instead, each of them communicates through WinSock. Multiple LSP applications can be installed at the LSP level.

NOTE: Aventail Connect does not alter or replace WinSock or any other core TCP/IP components (files) provided by the operating system.

When the Aventail Connect LSP receives a connection request, it determines whether or not the connection needs to be redirected (to an Aventail ExtraNet Server) and/or encrypted (in SSL). When redirection and encryption are not necessary, Aventail Connect simply passes the connection request, and any subsequent transmitted data, to the TCP/IP stack.

The two most popular versions of WinSock are versions 1.1 and 2. Aventail Connect 3.1, like all LSPs, requires WinSock 2; WinSock 1.1 does not support LSPs. WinSock 2 includes backward-compatibility with all WinSock 1.1 applications. Not every platform supports WinSock 2 and its LSP structure.

- Windows 98 and Windows NT 4.0 support WinSock 2 natively. (Windows NT 4.0 requires Service Pack 3 or above, available from Microsoft.)
- Windows 95 supports WinSock 1.1. Windows 95 can also support WinSock 2, but you must install a patch (available from Microsoft) to add support for WinSock 2.
- Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51 do not support WinSock 2; they support only WinSock 1.1.

For those platforms that do not support WinSock 2 and LSP applications, Aventail includes Aventail Connect 2.6 on the Aventail Connect 3.1/2.6 CD. Aventail Connect 2.6 was designed for operating systems that support only WinSock 1.1. On Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51 operating systems, setup will install Aventail Connect 2.6. If you are working on a Windows 95 operating system, setup will detect whether you have installed the Microsoft Windows 95 WinSock 2 Update. If setup detects the Microsoft update, which upgrades Windows 95 to support WinSock 2, setup will install Aventail Connect 3.1. If setup does not detect the Microsoft update, it will install Aventail Connect 2.6.

The Aventail Connect 2.6 user interface is identical to that of Aventail Connect 3.1; however, Aventail Connect 3.1 includes MultiProxy functionality (see "Multiple Firewall Traversal"). Aventail Connect 2.6 does not include MultiProxy.

In the future, more Windows applications may require WinSock 2.

During installation, setup determines which version of Aventail Connect to install. On WinSock 2 platforms, Aventail Connect 3.1 is installed. On WinSock 1.1 platforms, Aventail Connect 2.6 is installed. The following table shows how setup determines which version of Aventail Connect to install.

Operating System	WinSock Support	Aventail Connect Version Installed
Windows 98, Windows NT 4.0	WinSock 2	Aventail Connect 3.1
Windows 95	With Microsoft patch: WinSock 2	Aventail Connect 3.1
	Without Microsoft patch: WinSock 1.1	Aventail Connect 2.6
Windows 3.1, Windows for Workgroups 3.11, Windows NT 3.51	WinSock 1.1	Aventail Connect 2.6

You can create custom packages that include one or both versions of Aventail Connect (3.1 and 2.6). Setup will determine which version to install on each workstation. (For more information, see "Customizer.")

Windows 95 and WinSock

The Microsoft Windows 95 WinSock 2 Update upgrades WinSock 1.1 to WinSock 2 in Windows 95. This patch (filename `w95ws2setup.exe`) is available from the Microsoft Web site, at **http://www.microsoft.com/Windows95/downloads/contents/wuadmintools/s_wunetworkingtools/W95Sockets2/default.asp**. Unless you need specific Aventail Connect 3.1 features, Aventail recommends that you do not upgrade from WinSock 1.1 to WinSock 2. If you do not upgrade to WinSock 2, Aventail Connect 2.6 will be installed on Windows 95 systems.

If you do need to install the Microsoft Windows 95 WinSock 2 Update, follow the instructions provided by Microsoft. Reboot your computer after upgrading, prior to installing Aventail Connect.

How Does Aventail Connect Work?

The following three steps are identical to standard WinSock communications steps described above; however, nested inside them are additional actions and options introduced by Aventail Connect.

1. The application does a DNS lookup to convert the hostname to an IP address or, in rare cases, it will do a reverse DNS lookup to convert the IP address to a hostname. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following:
   - If the hostname matches a local domain string or does not match a redirection rule, Aventail Connect passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack performs the lookup as if Aventail Connect were not running.

- If the destination hostname matches a redirection rule domain name (i.e., the host is part of a domain we are proxying traffic to) then Aventail Connect creates a false DNS entry (HOSTENT) that it can recognize during the connection request. Aventail Connect will forward the hostname to the extranet (SOCKS) server in step 2 and the SOCKS server performs the hostname resolution.
- If the DNS proxy option is enabled and the domain cannot be looked up directly, Aventail Connect creates a false DNS entry that it can recognize later, and returns this to the calling application. The false entry tells Aventail Connect that the DNS lookup must be proxied, and that it must send the fully qualified hostname to the SOCKS server with the SOCKS connection request.

CAUTION: *The reverse DNS process can create unexpected delays, causing Aventail Connect to behave unpredictably. Aventail recommends that you do not enable this option unless you specifically require the Reverse DNS functionality.*

2. The application requests a connection to the remote host. This causes the underlying stack to begin the TCP handshake. When the handshake is complete, the application is notified that the connection is established and that data may now be transmitted and received. Aventail Connect does the following:

   a. Aventail Connect checks the connection request.
      - If the request contains a false DNS entry (from step 1), it will be proxied.
      - If the request contains a routable IP address, and the rules in the configuration file say it must be proxied, Aventail Connect will call WinSock to begin the TCP handshake with the server designated in the configuration file.
      - If the request contains a real IP address and the configuration file rule says that it does not need to be proxied, the request will be passed to WinSock and processing jumps to step 3 as if Aventail Connect were not running.

   b. When the connection is completed, Aventail Connect begins the SOCKS negotiation.
      - It sends the list of authentication methods enabled in the configuration file.
      - Once the server selects an authentication method, Aventail Connect executes the specified authentication processing.
      - It then sends the proxy request to the extranet (SOCKS) server. This includes either the IP address provided by the application or the DNS entry (hostname) provided in step 1.

   c. When the SOCKS negotiation is completed, Aventail Connect notifies the application. From the application's point of view, the entire SOCKS

negotiation, including the authentication negotiation, is merely the TCP handshaking.

3. The application transmits and receives data.

   If an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server on behalf of the application. If data is being returned, Aventail Connect decrypts it so that the application sees cleartext data.

AVENTAIL CONNECT PLATFORM REQUIREMENTS

The following table lists the minimum system requirements for each of the platforms that Aventail Connect supports.

Platform	Processor	RAM	SOCKS Server
Windows 98; Windows NT 4.0 (requires Microsoft Service Pack 3 or above)	x86-based or Pentium personal computer	16 MB	Network-accessible SOCKS v4 or v5 compliant server
Windows 95; Windows NT 3.51	x86-based or Pentium personal computer	8 MB	Network-accessible SOCKS v4 or v5 compliant server
Windows 3.1; Windows for Workgroups 3.11	x86-based or Pentium personal computer	4 MB	Network-accessible SOCKS v4 or v5 compliant server

Aventail Connect 3.1 runs on the following operating systems:

- Windows 98
- Windows NT 4.0 (with Service Pack 3 or above, available from Microsoft)
- Windows 95, with the Microsoft WinSock 2 update (To install Aventail Connect 3.1, you must upgrade Windows 95 with the Microsoft WinSock 2 update prior to Aventail Connect installation and setup. If you do not install the Microsoft patch, Aventail Connect 2.6 will be installed. For more information, see "What Does Aventail Connect Do?".)

Aventail Connect 2.6 runs on the following operating systems:

- Windows 3.1
- Windows for Workgroups 3.11
- Windows NT 3.51
- Windows 95, without the Microsoft WinSock 2 update (If you do not upgrade Windows 95 with the Microsoft WinSock 2 update, Aventail Connect 2.6 will be installed. For more information, see "What Does Aventail Connect Do?".)

NOTE: A WinSock-compatible 16- or 32-bit TCP/IP application must be installed and configured prior to running Aventail Connect. This can be the Microsoft-provided TCP/IP stack or a third-party TCP/IP stack.

INTERFACE FEATURES

The following table lists the interface features for each platform. Each of these features is discussed in greater detail later in the *Administrator's Guide*.

Platform	Start Aventail Connect	Display System Menu	Open Secure Extranet Explorer	View Program Icon	Hide Program Icon
Windows 95, Windows 98, Windows NT 4.0	Start\Programs \Aventail Connect menu	Right-click **Aventail Connect** icon in system tray	Double-click **Extranet Neighborhood** icon on desktop	In system tray	Not available
Windows 3.1, Windows for Workgroups 3.11, Windows NT 3.51	**Aventail Connect** icon in Aventail Connect program group window	Click **Aventail Connect** icon in Aventail Connect program group window	Not available	Minimized on desktop	Configure during setup

INSTALLATION SOURCE MEDIA

Regardless of platform, Aventail Connect can be delivered on CD or as a network-delivered, self-extracting archive file.

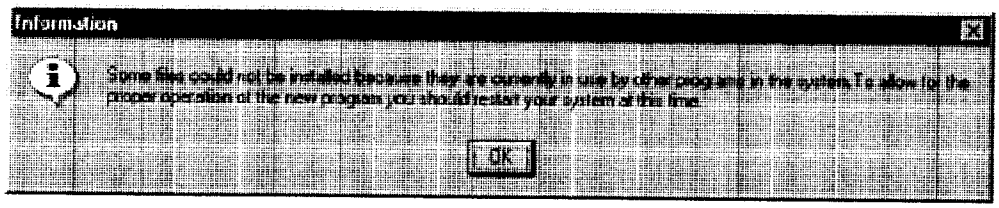
- **CD**: The CD contains the Aventail Connect setup program, `setup.exe`. The setup program allows for an administrative setup. It also contains the *Administrator's Guide* and the *User's Guide* in the `\docs` directory, formatted for Adobe® Acrobat Reader.
- **Network-delivered Source Media**: The network-delivered source media is a self-extracting archive containing the required disk/directory structure within the archive file. The executable automatically extracts the Aventail Connect installation files and initiates setup. The archive filename will be similar to `as31s.exe`. This archive, or package, will also be available on the CD (located in the **Utilities** directory) to be used with the Customizer application. For more information, see the "Customizer" section.

INSTALLING AVENTAIL CONNECT

After your Aventail ExtraNet Server is set up, Aventail Connect can be installed to a single workstation or to multiple networked workstations. In either case, you must perform an initial installation of the software and create one or more configuration (.cfg) files. This procedure is described under "Individual Installation." Once the initial installation is complete, you can then install to a series of networked computers using the instructions and information described under "Network Installation."

NOTE: *To install or uninstall Aventail Connect on Windows NT machines, you must have administrative privileges on the machine (but not necessarily on the domain).*

If you are upgrading from an earlier version of Aventail Connect (Aventail VPN Client or Aventail AutoSOCKS), the following message may appear on your screen if you install a custom setup package using Aventail Customizer. This is not an error message. If this message appears, click **OK** and reboot your computer.



CONFIGURATION FILES

Integral to the initial installation of Aventail Connect is deciding how SOCKS traffic will be redirected through the network. Network redirection rules (used to determine if and how SOCKS redirection will occur) are defined in the Aventail Connect configuration (.cfg) file. Configuration files are initially created at the end of the installation process; however, you can add, edit, and remove configuration files at any time using the Config Tool (in Windows 95, Windows 98, or Windows NT 4.0 via the **Aventail** icon in the system tray on the taskbar; in Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51 via the Aventail Program Group). The process of creating one or more configuration files is described under "Configuring Aventail Connect."

If you are installing Aventail Connect on multiple networked workstations, refer to "Network Installation" to determine the best method for maintaining and distributing configuration files. You can then proceed through the initial installation. The Installation Wizard will guide you through the steps, culminating with the option to create a configuration file.

CUSTOMIZED CONFIGURATION AND DISTRIBUTION

The Aventail Customizer is a utility that allows network administrators to customize Aventail Connect installation packages for distribution to multiple client workstations. Giving network administrators control over how setup packages are configured eliminates the need for end users to make installation and setup decisions at their workstations. The installation package is a self-extracting executable file. You can customize this file by adding license file, configuration file, or setup information for different authentication and encryption policies to meet various client-access needs of individuals or workgroups. You can customize configurations for multiple users and then distribute the package, providing easy access, download, and installation for users. You can reconfigure the Aventail Connect installation package anytime your network topology or security profiles change.

For more information about the Aventail Customizer, see the "Customizer" section.

INDIVIDUAL INSTALLATION

Before running setup, close all open Windows applications.

To install Aventail Connect

1. Installation procedures vary slightly, depending on which media source you use:
   - If you are installing directly from CD-ROM, run `setup.exe` from the Aventail Connect directory.
   - If you are installing from a network-delivered self-extracting archive, simply execute the archive file. This will extract the installation files and automatically launch the setup program.

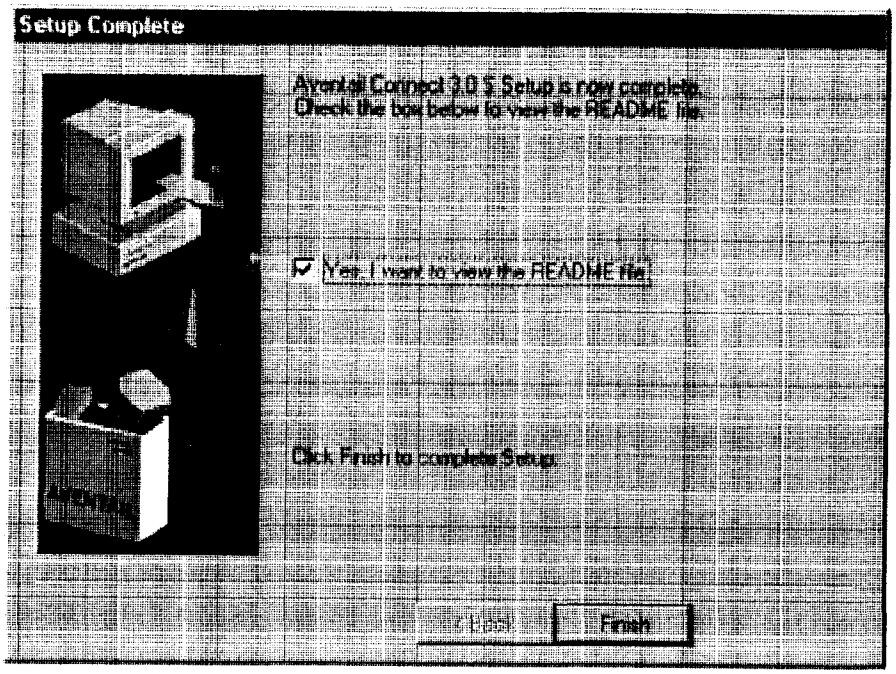
   The Aventail Connect Installation Wizard then guides you through the process of installing the Aventail Connect application.

NOTE: *You will be asked during the installation procedure if you would like Aventail Connect to be run automatically during startup. In most cases, you will select* ***yes****. Exceptions to this can be determined by the network administrator.*

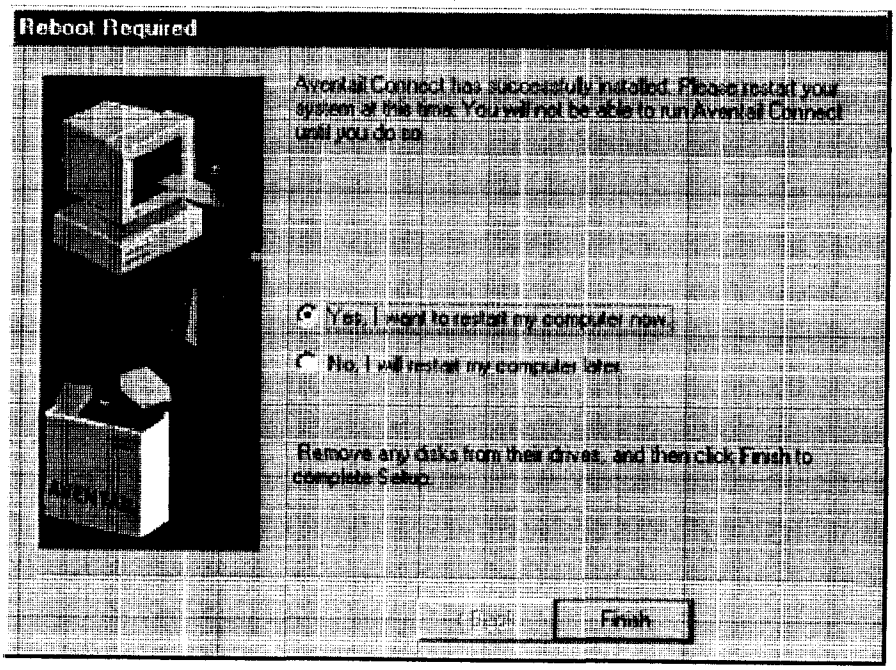
2. At the end of the setup program, you can select **Yes, I want to view the README file** in the **Setup Complete** dialog box. This opens the `readme.txt` file, which contains the latest information on Aventail Connect.

-OR-

Simply click **Finish** in the **Setup Complete** dialog box to complete the setup program.



3. The setup program will then ask you if you want to restart your machine now or later.



4. After restarting your PC, Aventail Connect will launch automatically if, during installation, you selected **Yes** when asked if Aventail Connect should be added to your startup directory. (If, during installation, you specified that Aventail Connect *not* be added to the startup directory, start Aventail Connect from the **Programs** menu.)
5. Aventail Connect will ask you if you want to run the configuration wizard.

   If you click **Yes,** then the configuration wizard will launch to help you create a new configuration file.

   If you click **No**, then Aventail Connect will ask you to select a configuration file.
6. After creating or selecting a configuration file, Aventail Connect will finish its installation procedure.

To uninstall Aventail Connect

The procedure to uninstall (remove) Aventail Connect varies depending on whether you are running a 16- or 32-bit Windows operating system.

- To uninstall Aventail Connect from Windows 95, Windows 98, and Windows NT 4.0, double-click **Add/Remove Programs** in the **Control Panel** window, click **Aventail Connect** on the list of programs on the **Install/Uninstall** tab, and then click **Add/Remove**.
- To uninstall Aventail Connect on Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51, use the **Uninstall** icon in the Aventail Connect program group.

NETWORK INSTALLATION

In general, the process of installing Aventail Connect to multiple networked workstations involves selecting a file server to use, creating a staging area for the Aventail Connect software, and placing the Aventail Connect package in a shared network directory or other publicly accessible location. Additional options include adding a default configuration file, license file, certificate and roots files, and SEEHosts files. You must place Aventail Connect files on a network drive that can be accessed as a mapped drive or, for Microsoft networks, via a UNC path name (`\\computer name\share name\Connect`).

An executable archive file (with a filename similar to `as31s.exe`) automatically extracts the Aventail Connect installation files and initiates setup. This archive, or package, is located in the Utilities directory of the CD and can be used in conjunction with the Customizer application. (For more information, see "Customizer.") The package can also be manually configured to suit your network specifications. The default package includes all of the core Aventail Connect files, but does not include the custom network information.

NETWORKED CONFIGURATION FILE SETUP

There are a number of ways to set up networked client configuration files. These are the most common:

- **Remote UNC:** Remote client configuration file on a Windows share using UNC path and filename (e.g., `\\internal\common\a.cfg`)
- **Local Configuration File:** Local client configuration file common for all users, but distributed via a locally stored Aventail Connect package
- **Remote Web Server:** Remote configuration files stored on a Web server using URL (e.g., **http://internal/a.cfg**)

Configuration file setup method	Location	Advantages	Disadvantages
Remote UNC	Windows share using UNC path and filename	• Configuration file can be centrally maintained. • No local caching required.	• File server must be on local network. If file server is unavailable, Aventail Connect will not function.
Local Configuration File	Locally stored setup package	• Does not require network connection; configuration file is always available.	• Configuration files cannot be centrally maintained.
Remote Web Server	Web server	• Configuration file can be centrally maintained. • Connection to Web server can be made across the Internet, and can traverse proxies. • Supports authentication and encryption. • If Web server is unavailable, locally cached copy can be used.	• Requires Web server. • Requires network connection for updates.

ADMINISTRATOR-MAINTAINED SHARED CONFIGURATION FILES

This is the most desirable configuration method—multiple workstations sharing one or more administrator-maintained configuration files located in a common directory. The network administrator maintains the configuration file, and the administrator can quickly adapt any changes to network topology through a single configuration file. For example:

- A single networked (usually read-only) configuration file is shared by more than one client workstation. This method is appropriate when multiple workstations share identical traffic routing rules.
- Multiple configuration files are shared by multiple workstations. This option is useful when you have workstations organized into functional groups (engineering, marketing, accounting, etc.) with group-specific redirection rules.

Shared Configuration File Distribution

Shared configuration files can be easily distributed and, if necessary, updated via the network or a Web server. Aventail recommends that you test all configuration files before distribution.

You can distribute shared configuration files with the Aventail Customizer. This automated wizard allows you to create custom setup packages for multiple users and then store the packages in a networked directory, providing easy access, download, and installation for users. You can include multiple local and/or remote configuration files. For more information, refer to the "Customizer" section.

To distribute a shared configuration file

There are three methods for distributing shared configuration files.

- **Remote UNC:** Copy the file to a Microsoft or Novell network drive accessible by all users, or to a Microsoft Windows workstation supporting UNC-sharing for file resources. (Both the 16- and 32-bit versions of Aventail Connect support specification of the configuration file using the Microsoft UNCs.) If you copy the file to a network drive, make sure that users configure Aventail Connect to load the configuration file located on the mapped drive. You can preconfigure this information for users from a package install.

-OR-

- **Local Configuration File:** Create a shared configuration file to be installed on workstations during the standard Aventail Connect installation/upgrade process. Whenever Aventail Connect is installed or updated, it will automatically copy the shared configuration file to the user's workstation and set Aventail Connect to use it.

-OR-

- **Web Server:** Copy the file to a Web server. The Web server can be directly accessible to the workstation, or it can be behind a proxy server. To keep configuration files secure, you can redirect the configuration file connection, authenticated and encrypted, across firewalls.

Storing Remote Configuration Files on a Web Server

When you specify the remote configuration file in Aventail Connect, include the entire URL (e.g., http://aventail.com/server1/config.cfg). You can specify this URL in the **Aventail Connect Configuration File** dialog box, or with Customizer.

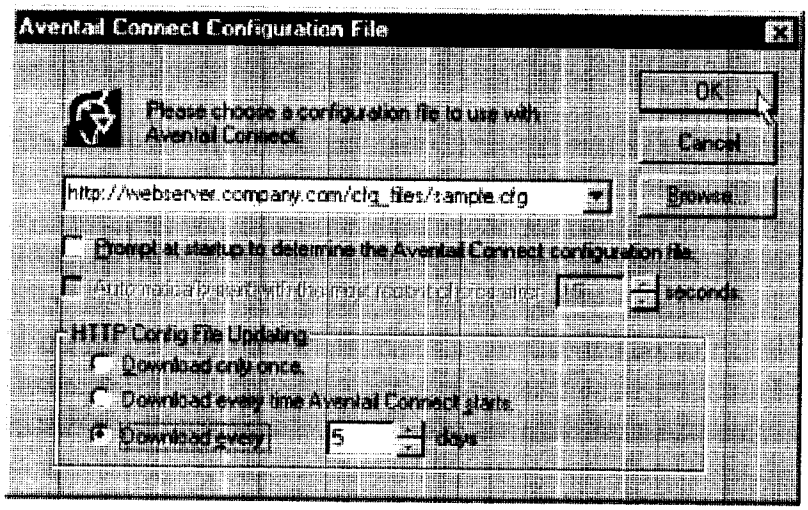
Aventail Connect keeps a temporary local copy of the remote configuration file in its program directory, with the filename _ashttpX.cfg, where X is a number between 0 and 9. Keeping a local copy of the remote configuration file allows the connection to the Web server to be proxied (with authentication and encryption) if necessary. Whenever the remote configuration file needs to be downloaded, Aventail Connect will check the cached copy of the configuration file to determine whether redirection is necessary.

Aventail Connect can download remote configuration files either every time Aventail Connect starts or on a scheduled basis. You can configure this setting in the **Aventail Connect Configuration File** dialog box, or when adding a remote configuration file to a custom installation package with Customizer. When you add a remote configuration file with Customizer, a cached copy of the file can automatically be added to the package.

To store remote configuration files on a Web server

1. Place an Aventail Connect configuration file on a Web server.
2. If redirection through a proxy server is required to reach the Web server, configure Aventail Connect to use a configuration file that can access the Web server. If redirection is not required, skip this step.
3. With Aventail Connect running, select **Configuration File** from the system tray menu.

   The **Aventail Connect Configuration File** dialog box will open.
4. Enter the URL and filename of the configuration file, e.g., **http://webserver.company.com/cfg_files/sample.cfg**. Click **OK**.



5. Under "HTTP Config File Updating," specify how often Aventail Connect will download the configuration file. Click **OK**.

   The configuration file will automatically be downloaded, and Aventail Connect will begin using it immediately. A local copy of the configuration file will be cached in the Aventail Connect program directory.

ADMINISTRATIVE SETUP

There are two ways to install Aventail Connect: from the setup program (`setup.exe`), or from a setup package that you create using the Aventail Customizer. The setup program (`setup.exe`) allows you to manually install Aventail

Connect. With the Aventail Connect setup package, you can select options that will customize setup based on your unique network environment. You can customize the setup package through the Customizer Editor or the Customizer Wizard. The Customizer *Editor* is a dialog box that allows you to manually enter or modify information about your custom installation package. The Customizer *Wizard* walks you through each step of creating a custom installation package. Aside from the user-interface differences, the Customizer Wizard and the Customizer Editor are identical. You can use both the Customizer Wizard and the Customizer Editor to create or modify a setup package. For example, you can create a package using the Customizer Wizard, then modify it with the Customizer Editor.

CUSTOMIZER

The Aventail Customizer simplifies and customizes the installation and setup process. Network administrators can reconfigure the self-extracting executable installation package (included in the Customizer directory of the distribution CD) to meet the various client-access needs of individuals or workgroups. Customizer offers a centralized approach to network configuration; network administrators can select the *unattended setup mode*, which eliminates the need for individual users to answer any setup configuration questions. Specifying unattended mode will cause the setup program to automatically install using default values for any options not explicitly specified.

The setup program (`setup.exe`) allows users to select any available setup options during installation of Aventail Connect. Customizer modifies the setup control file of a custom package; this file controls all of the settings within the setup package, before users receive the setup package. With a customized package, users will receive an installation package based on the administrator's defined settings. (For more information, see "Network Installation.")

As Customizer allows you to select various options to suit your setup and installation needs, the size of the setup package will vary, depending on which options you select. If size of the setup package is a concern, select setup options carefully to keep the package size manageable.

The Aventail Connect CD includes both versions of Aventail Connect (3.1 and 2.6). You can create custom packages that include one or both versions of Aventail Connect; setup will determine which version to install on each workstation. (For more information, see "What Does Aventail Connect Do?")

Aventail Connect requires a valid Aventail license file (`aventail.alf`) and one or more configuration (`.cfg`) files in order to function properly. Before installing Aventail Connect, make sure that users have these files. If users do not have a valid license file and/or configuration file(s), Aventail recommends that you include them in the installation package.

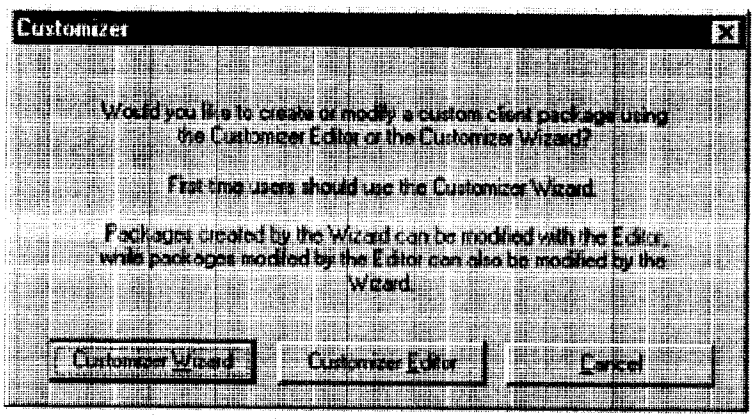
RUNNING CUSTOMIZER

The Customizer and the Aventail Connect installation package are included in the Customizer directory on the Aventail Connect CD. Before running Custom-

izer, you must copy Customizer from the Aventail Connect CD to the local drive. You must also modify the Customizer attributes so it is not read-only.

To run Customizer, double-click the **Customizer** icon in the Customizer directory. To run Customizer from your hard drive, copy the Customizer and Aventail Connect directories into a common folder on the hard drive.

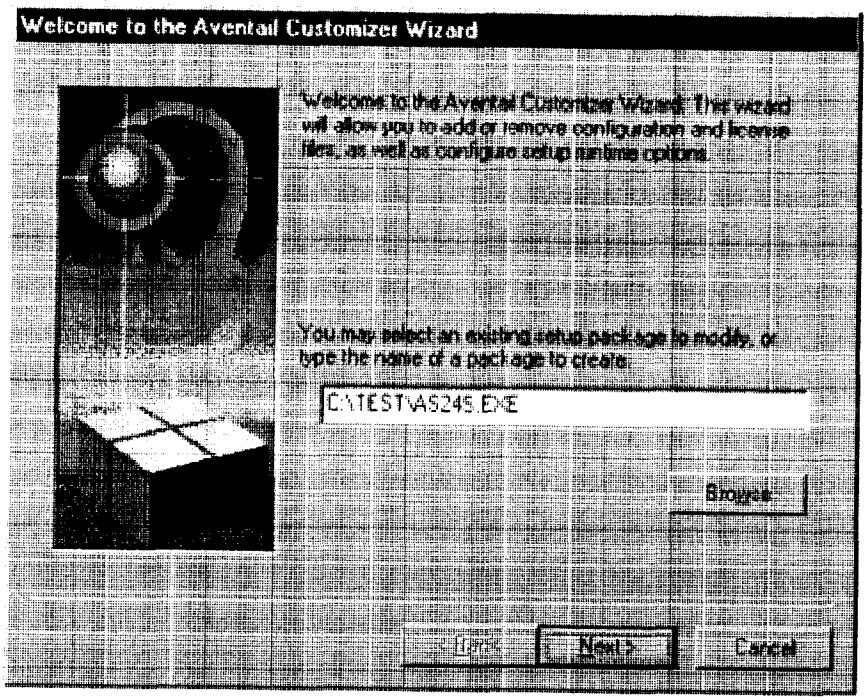
When you run Customizer, you will be prompted to select either the Customizer Wizard or the Customizer Editor.



- **Customizer Wizard:** This automated wizard walks you through the process of creating a new installation package or modifying an existing package. If you are unsure about which method to use, Aventail recommends that you use the Customizer Wizard.
- **Customizer Editor:** The Customizer Editor is a dialog box that allows you to manually enter information about the package you are creating or modifying.

CUSTOMIZER WIZARD

If you are using the Customizer Wizard to create a new setup package or modify an existing package, the Customizer Wizard will display a **Welcome...** screen, and will prompt you to enter the pathname of the package that you will be creating or modifying.



After you have specified the pathname of the package, the Customizer Wizard will prompt you to:

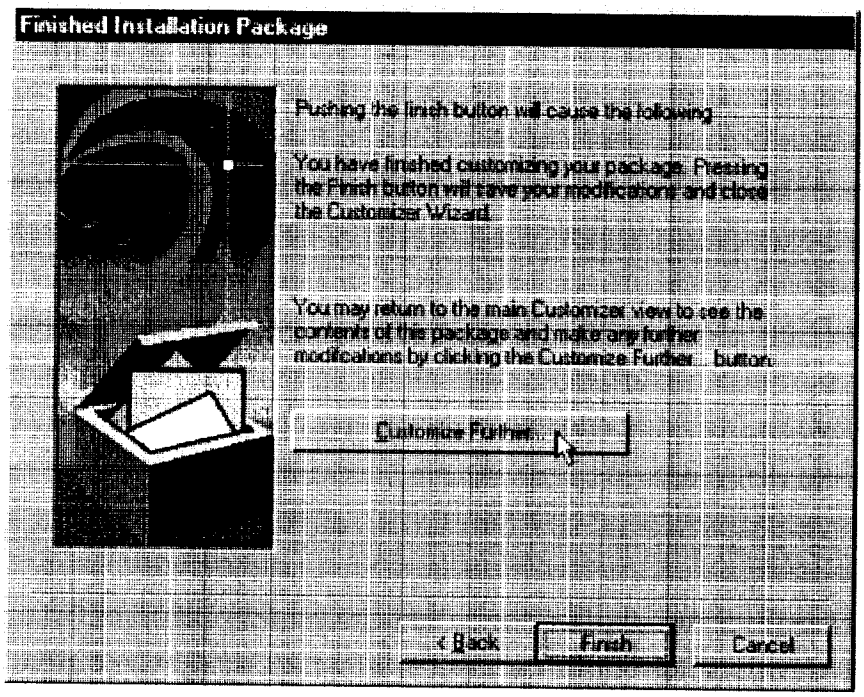
- Specify which platform(s) to support
- Add a license file, or leave an existing license file in the package
- Add or remove configuration files
- Select X.509 certificate files
- Select an extranet hosts (SEEHosts) file
- Specify a custom destination directory
- Specify whether or not to put program icons in a custom folder
- Enter command-line switches
- Specify whether or not to run setup in unattended mode
- Specify whether or not to add Aventail Connect to the startup directory
- Select any, all, or none of the following Aventail Connect components:
  - Extranet Neighborhood (Secure Extranet Explorer)
  - Configuration Tools (Config Tool and Configuration File command)
  - Diagnostic Tools (Logging Tool and S5 Ping)
  - Certificate Tools

- Install 32-bit support only (on Windows NT 3.51)

- Select any, all, or none of the following authentication modules:
  - SSL (Secure Sockets Layer)
  - CRAM (Challenge Response Authentication Method)
  - CHAP (Challenge Handshake Authentication Protocol)
  - UNPW (Username/Password)
  - SOCKS 4
  - HTTP Basic (username/password)
- Specify whether or not to run a command after setup

All of the features listed above are optional.

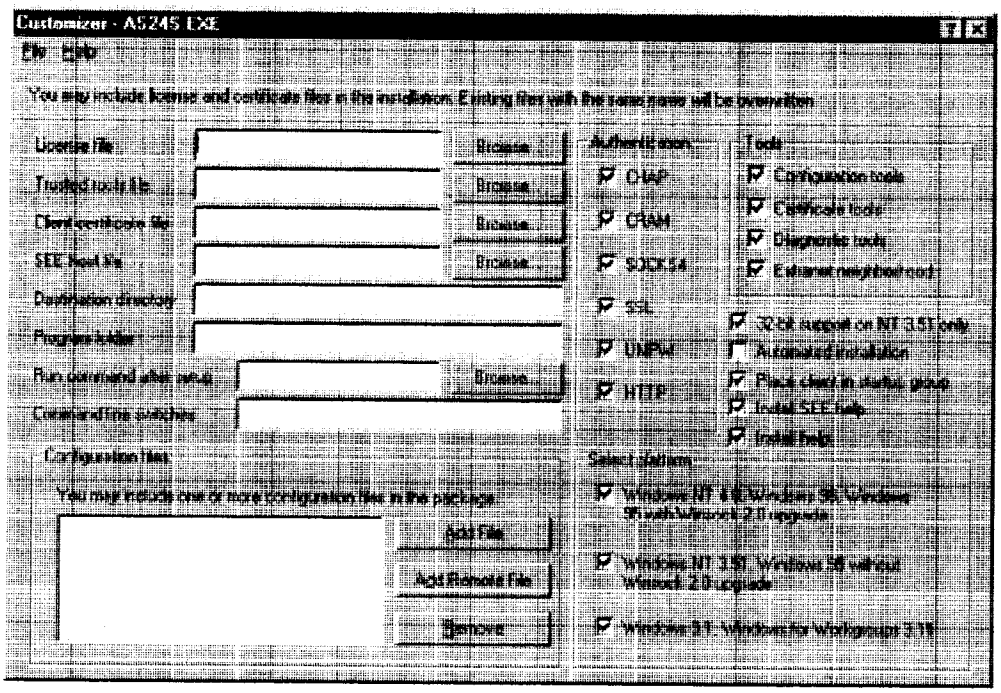
After entering or modifying the package information, the **Finished Installation Package** dialog box appears.



Clicking **Finish** saves your specifications and closes the Customizer Wizard. Clicking **Customize Further** allows you to view the **Customizer Editor** dialog box, where you can manually edit any of the information about your custom installation package.

CUSTOMIZER EDITOR

If you select the Customizer Editor as your tool to create a new setup package or modify an existing package, the **Customizer Editor** dialog box will appear. In this dialog box, you can manually enter or modify information about your custom installation package.



NOTE: *To view a list of tips on creating custom setup packages, click* ***Tips*** *on the* ***Help*** *menu in the* ***Customizer Editor*** *dialog box.*

After entering or editing your setup package information in the Customizer Editor, click **Save** (or **Save As**) on the **File** menu to save your changes. To close the **Customizer Editor** window, click **Exit** on the **File** menu.

The options in the Customizer Editor are identical to the options in the Customizer Wizard. These options are explained in the following paragraphs and tables.

Option	Settings	Default Setting
Pathname	Enter pathname	None
License file	Enter name of Aventail license file (must use `aventail.alf`)	None
Trusted roots file	Enter name of trusted roots file	None
Client certificate file	Enter name of file that contains certificate	None
Extranet (SEE) Hosts File	Enter name of extranet (SEE) hosts file	None
Destination directory	Enter name of destination directory	None
Program folder	Enter name of program folder	None
Run command after setup	Enter command to be run after setup	None
Command line switches	Enter command line switches	None
Configuration Files	Enter name(s) of local and/or remote configuration file(s) that Aventail Connect will use	None
Authentication Modules	SSL, CRAM, CHAP, UNPW, S4, or HTTP Basic	All
Tools	Configuration tools, Certificate tools, Diagnostic tools, or Extranet Neighborhood	All
32-bit support only, on Windows NT 3.51	Yes/No	Yes
Unattended setup mode/automated installation	Yes/No	No
Add to Startup Directory	Yes/No	Yes
Install SEE help	Yes/No	Yes
Install help	Yes/No	Yes
Select platform	Windows NT 4.0, Windows 98, Windows 95 with WinSock 2 upgrade, Windows 95 without WinSock 2 upgrade, Windows NT 3.51, Windows 3.1, or Windows for Workgroups 3.11	All

The setup package options are discussed below.

- **Specify path for installation:** You can specify a path for installation, or you can select the default path. The default path for 32-bit operating systems is `c:\Program Files\Aventail\Connect`.
  For 16-bit-only operating systems, the default is `c:\Connect`.

NOTE: If you are upgrading from an earlier version of Aventail Connect, Aventail Connect will install to the same directory that the earlier version of it was installed to.

- **Platforms:** You must specify which operating systems need to be supported in the setup package. Aventail Connect 3.1 supports Windows 95 (with the Microsoft WinSock 2 update), Windows 98, and Windows NT 4.0 (with Service Pack 3 or above, available from Microsoft). Aventail Connect 2.6 supports Windows 95 (without the Microsoft WinSock 2 update), Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51. For more information, refer to "What Does Aventail Connect Do?"
- **Trusted Roots File and Certificate File:** If you want to use server certificates, you must include the trusted roots file that contains those certificates. If you want to use client certificates, you must specify the location of the file that contains the X.509 certificate.
- **Running Setup in Unattended Mode:** Unattended setup mode simplifies distribution of numerous client configuration files. The network administrator specifies all settings before users receive the Aventail Connect setup package file. No end-user input is required because the network administrator has already selected the setup options; users simply open the package file, which will automatically install on their workstations.

NOTE: Specifying unattended setup mode will cause the setup package to automatically install using default values for any options not explicitly specified.

- **Adding Aventail Connect to the Startup Directory:** If you choose to add Aventail Connect to the startup directory, Aventail Connect will automatically start when Windows starts.
- **Select Tools:** Aventail Connect gives you the option to install various components, including Extranet Neighborhood/Secure Extranet Explorer (SEE), configuration tools (Config Tool and Configuration File command), or diagnostic tools (Logging Tool and S5 Ping). The default value is to install all package components.
  - **Secure Extranet Explorer:** Secure Extranet Explorer (SEE) allows you to view your Extranet Neighborhood, which is accessed through the **Extranet Neighborhood** icon on your desktop. Extranet Neighborhood functions much like Network Neighborhood, except Extranet

Neighborhood allows you to browse, copy, move, and delete files from secured remote computers via an extranet, while Network Neighborhood displays all computers on your local network.

- **Config Tool:** The Aventail Connect Config Tool allows you to create configuration files that determine how network requests will be routed and which authentication protocols will be enabled. You can add, remove, or edit configuration files at any time. If necessary, you can create several configuration files for different users or user groups. If you want to prohibit end users from editing configuration files, do not include the Config Tool in the installation package.
- **S5 Ping:** S5 Ping allows you to use the ping and traceroute utilities, two diagnostic tools. The ping utility checks for network connectivity between two hosts and returns information about the quality of the connection. The traceroute utility checks for network connectivity by displaying information about routers between two hosts; it displays information for each hop.
- **Logging Tool:** The Logging Tool is a diagnostic utility that traces Aventail Connect activity. When running a trace, the Logging Tool displays errors, warnings, and information as Aventail Connect generates them. If necessary, the message list can be saved to a log file that can be used by Aventail Technical Support in troubleshooting technical problems. These traces are also useful when running Aventail Connect for the first time to ensure that network traffic is being routed appropriately.

- **Select Authentication Modules:** Aventail Connect lets you select any, all, or none of the following authentication modules: SSL, CRAM, CHAP, UN/PW, SOCKS v4, or HTTP Basic (username/password).
  - **Secure Sockets Layer:** Secure Sockets Layer (SSL) is a session-layer protocol for securing connections in a general, protocol-independent fashion.

NOTE: *In versions of Aventail Connect that do not include encryption, the Secure Sockets Layer (SSL) authentication module is not included.*

  - **CRAM:** The Challenge Response Authentication Method (CRAM) sends your username and password as clear text between extranet (SOCKS) servers, but encrypted between servers that support CRAM. Typically, CRAM subauthenticates within SSL, which provides both encryption and credential caching options.

NOTE: *In versions of Aventail Connect that do not include encryption, the CRAM authentication module is not included.*

- **CHAP:** The Challenge Handshake Authentication Protocol (CHAP) sends your username and password encrypted across the network to the destination server.
- **Username/Password:** The RFC 1928 (Internet standards document) Username/Password (UNPW) authentication protocol sends your username and password in clear text across the network to the destination server.
- **SOCKS 4 Identification:** Aventail Connect includes backward compatibility for the SOCKS 4 protocol. SOCKS 4 does not support password authentication, so only your username is sent, unencrypted, to the SOCKS server along with your connection request.
- **HTTP Basic (Username/Password):** The HTTP Basic authentication module enables username/password authentication against HTTP proxies that implement the RFC 2068 HTTP Basic authentication protocol.

NOTE: *Not all versions of Aventail Connect have encryption enabled.*

- **Configuration Files:** Aventail Connect needs at least one configuration (`.cfg`) file in order to function properly. The configuration file contains all of the authentication and traffic routing instructions that you specify. You can include one or more configuration files in the setup package; however, each configuration file must have a different name. If you include only one configuration file in a setup package, Aventail Connect will automatically use that configuration file. If, however, you include multiple configuration files, Aventail Connect will prompt users to select a configuration file at startup.

  You can include local configuration files, remote configuration files, or a combination of both. Local configuration files are included in the setup package and are installed on users' machines. If you include remote configuration files, pointers to those files are included in the package; the remote configuration files remain in their original location on the network, where they can be shared by multiple users.

  If your setup package does not already contain a configuration file, you can add a configuration file to the package. If your setup package contains one or more configuration files, you can remove or replace any or all of the existing configuration files, or you can leave them, unchanged, in the package. If you are upgrading from an earlier version of Aventail Connect, you may not need a new configuration file.
- **License Files:** Aventail Connect requires a valid license file in order to function properly. If your setup package contains a license file, you can remove or replace the existing license file, or you can leave it, unchanged, in the package. If your setup package does not contain a

license file, you can add one to the packaged Aventail license file, `aventail.alf`.

CAUTION: Aventail Connect 3.1 and 2.6 use a different license (`.alf`) file format than earlier versions of Aventail Connect (VPN Client or AutoSOCKS) did. If you are upgrading from an earlier version of Aventail Connect (v2.42 or earlier), you must include a new Aventail license file.

- **Extranet (SEE) Hosts Files:** Secure Extranet Explorer (SEE) allows you to browse remote computers using Extranet Neighborhood. SEE requires a hosts file that specifies which Windows domains, WINS servers, and other computers are available in Extranet Neighborhood. The extranet hosts (SEEHosts) file is contained in the setup package. If you install SEE, this file is placed in the target directory. If you do not include a hosts file in the setup package, Aventail Connect will automatically create a hosts file on users' machines the first time they open Extranet Neighborhood. (Available only in Windows 95, Windows 98, and Windows NT 4.0.)

CREATING, LOADING, AND SAVING PACKAGES

You can create, load, or save custom setup packages through either the Customizer Editor or the Customizer Wizard.

To create a new package

There are two ways to create a new custom setup package:

- In the **Customizer Editor** window, select **File | New**.

-OR-

- Type the filename of a new package in the first window of the Customizer Wizard and click **Next**.

To load a package

There are two ways to load an existing setup package:

- In the **Customizer Editor** window, select **File | Open**, and then enter the filename of the package you want to load

-OR-

- Type the filename of the package in the first window of the Customizer Wizard and then click **Next**.

When you load a package, Customizer reads the setup control file to determine what information the package contains. Customizer uses this information to populate the **Customizer Editor** window. Customizer also reads the configuration file(s) into memory; configuration files are stored in memory to facilitate adding them to and removing them from a package.

To save changes to a package

There are two ways to save changes to a setup package:

- After making the desired changes to the package, click **Save** (or **Save As**) on the **File** menu in the **Customizer Editor** window

-OR-

- Click **Save Package** in the final window of the Customizer Wizard.

CUSTOMIZER TIPS

The following tips will help you use the Aventail Customizer more efficiently.

- **Keep the package size small:** You can control the size of your custom setup packages by selecting components carefully. To keep the package as small as possible, include only the options that you need, and support only the platforms (e.g., Windows 98, Windows NT 4.0, etc.) that your users work with. You may find that creating two separate, smaller packages is preferable to creating one larger package. For example, you might create one package that supports Windows 98 and Windows NT 4.0 operating systems, and another separate package that supports Windows 3.1 and Windows 95 operating systems.
- **Use descriptive package names:** When naming setup packages, assign descriptive, recognizable names that will help users identify the setup packages.
- **Select components carefully:** If you include the Config Tool in the package, users will be able to view and modify the settings in the Config Tool. Aventail recommends that, in most cases, you do not include the Config Tool in your custom setup package(s). Excluding options such as the Config Tool will eliminate users' ability to modify your settings, and will keep the package size smaller. However, the S5 Ping and Logging Tool utilities are useful diagnostic tools, and Aventail recommends including these options in the setup package whenever possible.
- **Install Aventail Connect 2.6 on Windows 95:** By default, Windows 95 does not support WinSock 2, but you can upgrade it to support WinSock 2 with a Microsoft patch. (The patch, `w95ws2setup.exe`, is available from Microsoft, at **http://www.microsoft.com/Windows95/downloads/contents/wuadmintools/s_wunetworkingtools/W95Sockets2/default.asp**. However, this procedure adds an extra step to the installation and setup process. Unless users need the MultiProxy feature, which is available only in Aventail Connect 3.1, Aventail recommends that you install Aventail Connect 2.6 rather than 3.1 on machines running the Windows 95 operating system.
- **Include a hosts file:** If you install Secure Extranet Explorer (SEE) without also installing a corresponding hosts file, SEE will automatically create a hosts file the first time that users open SEE. If you want to control which hosts users can view, Aventail recommends that you include a hosts file in the custom setup package.

- **Include a license file:** Aventail Connect requires a valid license file (`aventail.alf`) to function properly. Aventail Connect 3.1/2.6 uses a different license file than earlier versions of Aventail Connect (VPN Client or AutoSOCKS) did. If you are upgrading from an earlier version of Aventail Connect (v2.42 or earlier), you must use the new Aventail license file, `aventail.alf`. Including this license file in the custom setup package is a simple way to install the license file.
- **Test each custom package:** Aventail recommends that you thoroughly test each custom setup package before distribution to users.

CONFIGURING AVENTAIL CONNECT

Create configuration files using the Config Tool or the Configuration wizard. You can launch either during the Aventail Connect installation or any time you want to add, modify, or remove a configuration file.

The steps for creating a new configuration file are:

1. Define the SOCKS servers
2. Define the destinations (networks and hosts)
3. Specify redirection rules
4. Enter Name Resolution information (optional)
5. Manage authentication modules
6. Enable password protection (optional)

These procedures are described in the text below.

To launch the Config Tool

The Config Tool opens with the **Open Aventail Connect Configuration File** dialog box. After you select a configuration file or enter a new file name, the main window of the Config Tool appears.

1. Select the **Yes, I want to configure Aventail Connect** box in the **Setup Complete** dialog box (during installation).

-OR-

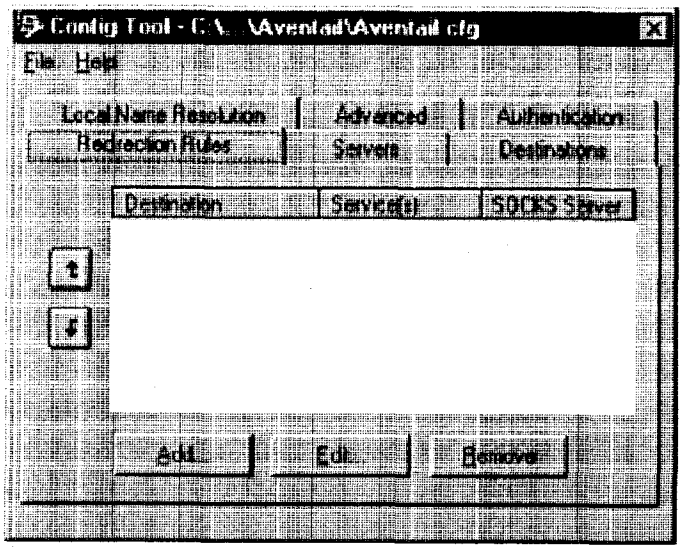
Right-click the **Aventail Connect** icon in the taskbar and click **Config Tool** (Windows 95, Windows 98, or Windows NT 4.0 programs menu option), or double-click the **Config Tool** icon in the Aventail Connect program group (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51).

2. If you are creating a new configuration file, enter a name for the configuration file

-OR-

Select the configuration file you want to open.

This displays the main window of the Config Tool.



The **Config Tool** window contains six tabs. The properties defined on each tab can be edited at any time.

Tab	Function
Servers	Defines the extranet (SOCKS) server(s).
Destinations	Specifies the network and host addresses that will be routed through the SOCKS server(s).
Redirection Rules	Specifies how network requests are routed to the SOCKS server(s).
Name Resolution	(Optional) Specifies hostnames that will be resolved by the local workstation.
Authentication	Enables, disables, and sets properties for the authentication modules.
Advanced	Enables/disables extranet (SOCKS) traffic through successive SOCKS servers, enables/disables the Application Exclusion/ Inclusion List, secures selected applications, and sets credential cache timeouts.

You can change the width of any of the fields on the tabs by positioning the cursor over the dividing line between the fields on the field bar. When the cursor changes to a double-headed arrow, click and drag to resize the field.

Aventail Connect 3.1 allows you to create or modify a configuration file and then immediately use it, without needing to restart Aventail Connect and any Aventail-processed applications. When you modify a configuration file, Aventail Connect can re-read the updated configuration file; all applications being processed by

Aventail Connect will then immediately begin using the new configuration information.

When you make a modified configuration file active, Aventail Connect will save the current (modified) configuration file, update the registry, and load the selected configuration file. Aventail Connect will begin using the modified configuration file with any subsequent TCP connection requests, and/or any subsequent UDP activity.

NOTE: *The configuration file "refresh" feature is supported in Aventail Connect 3.1 only. It is not supported in Aventail Connect 2.6. To activate modified configuration files in Aventail Connect 2.6, you must first shut down and restart Aventail Connect and all applications being processed through Aventail Connect.*

To load a modified configuration file for immediate use

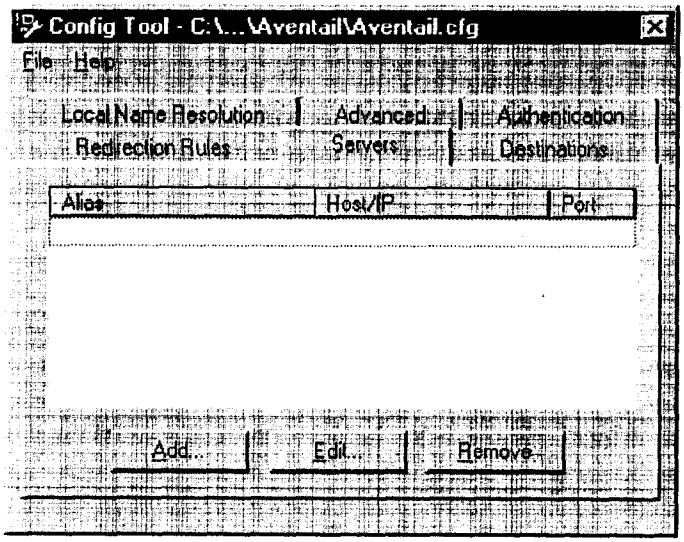
- With the newly modified configuration file open, select **Make Active** from the **File** menu of the Config Tool

  -OR-

- From the system tray menu, select **Configuration File**, and select (or enter the name of) the configuration file that you want to use. Click **OK**.

DEFINE AN EXTRANET (SOCKS) SERVER

SOCKS servers are defined on the **Servers** tab in the Config Tool.



Field	Definition
Alias	The name you assign to the server.
Host/IP	The hostname or IP address of the server.
Port	The port on which the server is listening.

Aventail Connect 3.1 allows you to set a server fallback timeout for every Aventail ExtraNet Server. If a primary SOCKS server is down, or otherwise unable to accept connections, Aventail Connect can fall back to a secondary server. You can set the server fallback timeout, in seconds, on a server-by-server basis. If you do set a server fallback timeout, each connection to a primary server must be completed within the specified length of time or else the connection will fall back to the secondary server.

NOTE: *Server fallback timeouts are supported in Aventail Connect 3.1 only. You cannot set a server fallback timeout in Aventail Connect 2.6; you must let the TCP/IP stack time out.*

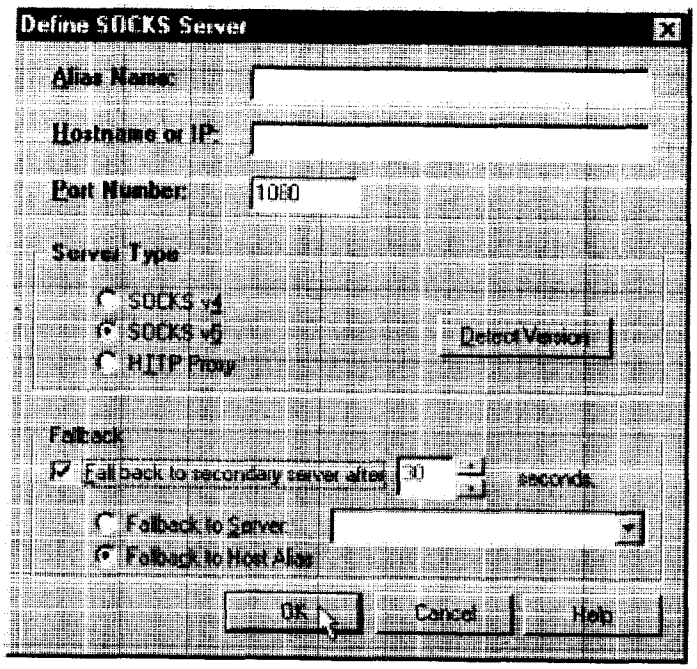
NOTE: *Aventail Connect can fall back to only one server. For example, Aventail Connect could fall back from Server A (primary server) to Server B (secondary server). Aventail Connect could not, however, fall back from Server A to Server B to Server C.*

During normal operation, if you configure Aventail Connect to fall back to a secondary server, connections will be directed to the primary server. If the primary server does not respond or accept the connection by the end of the fallback timeout period, the connection will be redirected to the secondary server. If the secondary server accepts the connection, all subsequent connections will automatically be directed to the secondary server. The secondary server is generally meant to be used only when the primary server is unable to accept connections. To prevent the secondary server from automatically becoming the default server for all subsequent connection, Aventail Connect will check the primary server's status every ten minutes. If the primary server is back up and able to accept connection, all subsequent connections will be routed through the primary server.

CAUTION: *Do not enable the server fallback option if you are using plug gateways.*

To add an extranet (SOCKS) server

1. On the **Servers** tab, click **Add...**. The **Define SOCKS Server** dialog box appears.



Field	Definition	
Alias Name	User-friendly alias for extranet (SOCKS) server.	
Hostname or IP	Actual hostname or full numeric IP address for SOCKS server.	
Port Number	SOCKS server port. Default value is 1080.	
Server Type	SOCKS v4	SOCKS Version 4.0.
	SOCKS v5	SOCKS Version 5.0.
	HTTP Proxy	HTTP proxy server.
	Detect Version	Detect SOCKS version number.
Fallback	Fall back to secondary server after x seconds	Server fallback timeout period (in seconds).
	Fall back to Server:	SOCKS server alias for redundant server.
	Fall back to Host Alias	Use DNS records for redundancy.

2. In the **Alias Name** box, type a user-friendly alias for the extranet (SOCKS) server. Do not leave this box blank.
3. In the **Hostname or IP address box**, type the actual hostname of the SOCKS server or its IP address.
4. In the **Port Number** box, type the extranet server's port number. If you do not enter a value, it defaults to the standard SOCKS port 1080.
5. Under "Server Type," select the version of SOCKS supported by the server. If you are unsure of the version, click **Detect Version**.

NOTE: *Typically you should select **SOCKS v5** unless the server can support only SOCKS v4.*

6. If you want to use a fallback server, select **Fall back to secondary server after...** under "Fallback." Either select **Fall back to server** and directly specify an extranet server for redundancy, or select **Fall back to host alias**. Select or enter, in seconds, the fallback timeout period. Click **OK**.

To edit extranet (SOCKS) server properties

- Select the extranet server you want to edit and click **Edit**.

  The **Define SOCKS server** dialog box appears with the selected server data filled in. Edit any of the information, and then click **OK**.

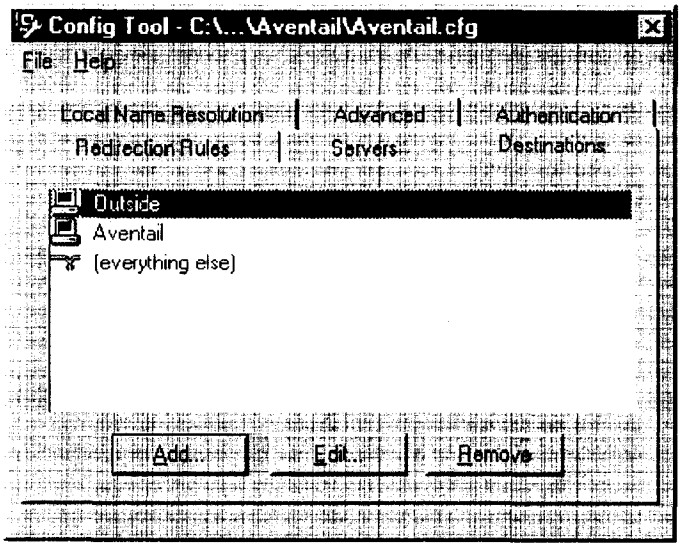
To remove an extranet (SOCKS) server definition

- Select the extranet server you want to remove and click **Remove**.

  The server is deleted from the list. Corresponding redirection rules will also be deleted.

DEFINE A DESTINATION

Destinations are defined on the **Destinations** tab in the Config Tool.



After one or more SOCKS servers are defined, add destinations to be routed through them.

NOTE: *The "(everything else)" destination refers to all network and host addresses not otherwise defined. You cannot delete or modify "(everything else)."*

WILDCARDS IN HOSTNAME DEFINITIONS

Aventail Connect supports the use of wildcard characters in destination hostnames. You can use wildcards when defining named destinations (hostnames); you cannot use wildcards when defining numerical destinations, such as IP addresses or subnet masks.

Acceptable wildcard characters are "?" and "*" (where "?" represents one character, and "*" represents any number of characters). For example:

`e*tra.in.aventail.com` **matches** `extra.in.aventail.com`
`e?tra.in.aventail.com` **matches** `extra.in.aventail.com`
`e?ra.in.aventail.com` **does NOT match** `extra.in.aventail.com`

You can use any combination of "?" and "*" characters between each set of periods. However, each section must contain at least one non-wildcard character. For example, the following destination names would be allowed:

```
e?t?a.in.aventail.com
*xtr?.in.aventail.com
e???a.in.ave*.com
e*.in.*tail.com
```

The following destination names, however, would not be allowed:

```
extra.*.aventail.com
*.*.aventail.com
extra.in.*.com
```

CAUTION: *You cannot use a wildcard character, or a series of wildcard characters, to represent multiple sections. Any wildcard character in a section can represent characters within that section only. For example:*

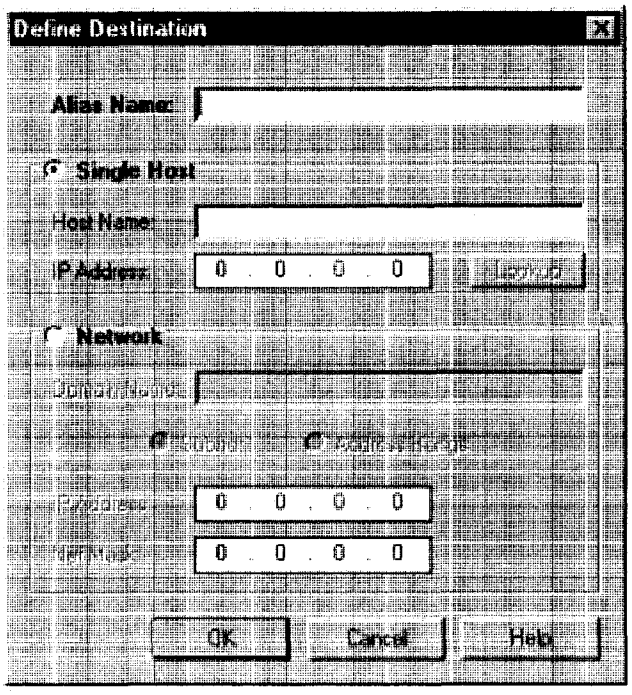
`e*.in.aventail.com` ***matches*** `extra.in.aventail.com`
`e*.aventail.com` ***does NOT match*** `extra.in.aventail.com`

To add a destination

In the **Define Destination** dialog box, you can define subnets, individual host computers, or IP address ranges, and set up rules about redirecting some or none of the IP traffic to these defined destinations.

1. On the **Destinations** tab, click **Add...**.

The **Define Destination** dialog box appears.



Field	Definition	
Alias Name	User-friendly alias for destination network or host	
Single Host	A specific destination computer	
	Hostname	Actual name of destination network or host
	IP Address (*optional*)	Full numeric IP address
	Lookup	Look up IP address
Network	One or more computers in a network	
	Domain Name	Domain of the network
	Subnet (*optional*)	IP address and netmask address
	Address Range (*optional*)	Beginning and ending IP addresses From Starting IP address To Ending IP address

CAUTION: *The **IP Address**, **Subnet**, and **Address Range** fields are all optional. However, in order to apply redirection rules when connecting by IP address, you must enter IP address and subnet information.*

2. In the **Alias Name** box, type a user-friendly alias for the destination network or host.
3. Select either the **Single Host** or **Network** option:
   - Under "Single host," type the actual name of the host system and/or its full, numeric IP address. If you do not know the host's IP address, click **Lookup** to search for it.

-OR-

   - Under "Network," type the domain of the network and then, if applicable, select either **Address Range** or **Subnet**.

Use	To
Address Range	Enter a starting and ending IP address. All addresses between the two will be included as part of the destination. For example, a starting IP address of 192.1.1.0 and an ending IP address of 192.1.1.255 would include all hosts of the 192.1.1.x subnet.
Subnet	Enter an IP address and a netmask address. This is another way to specify a group of destinations. For example, an IP address of 192.1.1.0 and a net mask of 255.255.255.0 defines the same address range as shown above.

To edit a destination

- Select the destination you want to edit and click **Edit...**.

  The **Define Destination** dialog box appears with the selected destination data filled in. Edit the data as necessary.

To remove a destination

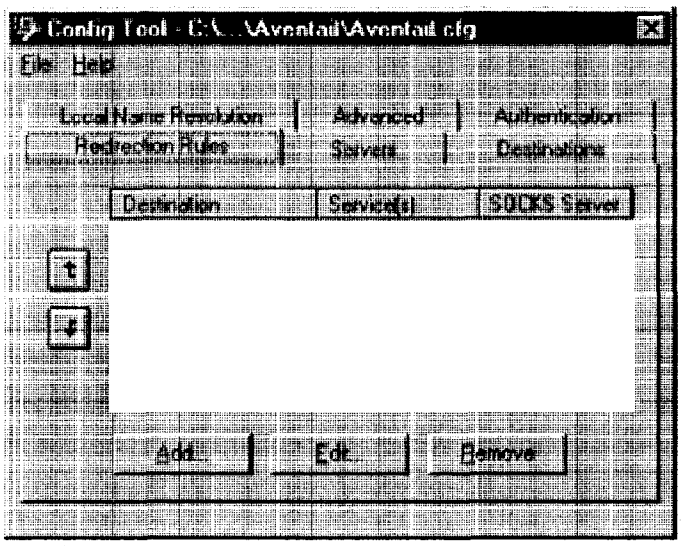
- Select the destination you want to remove and click **Remove**.

  The destination is deleted from the list. The corresponding redirection rules will also be deleted.

ENTER REDIRECTION RULES

Once servers and destinations are defined, you can specify how you want Aventail Connect to redirect (or deny) access to various hosts and services such as e-mail, FTP, and HTTP.

Redirection rules are specified on the **Redirection Rules** tab in the Config Tool.



Field	Definition
Destination	Destinations defined on the **Destinations** tab
Service	Type of Internet traffic
Proxy Redirection	Specify how to redirect traffic

You can change the width of any of the three fields by moving the cursor to the dividing line between the fields on the field bar. When the cursor changes to a double-headed arrow, click and drag to resize the field.

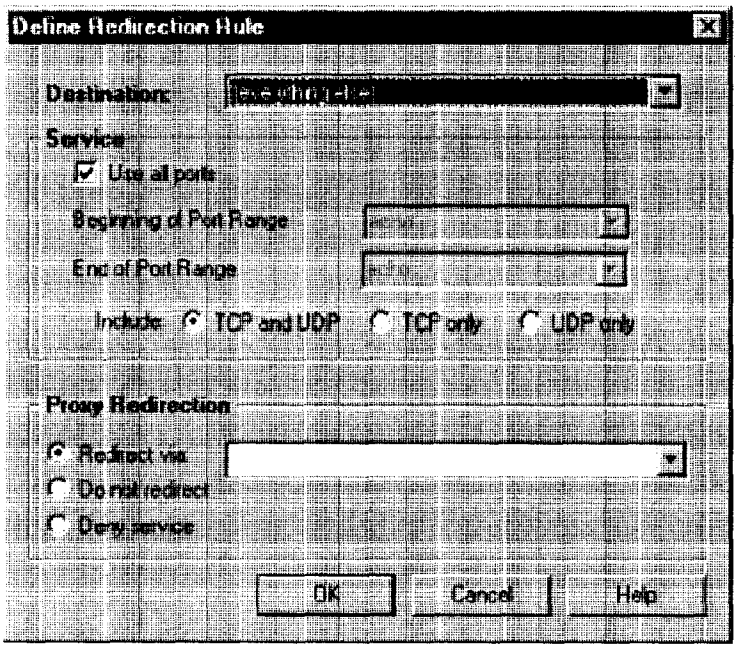
To add a redirection rule

As you add destinations, use the arrow buttons to prioritize them. List the most specific rules first and the general rules last.

NOTE: *Aventail Connect scans the list from the top down and uses the first matching rule it finds, so it is important to list the most specific rules first.*

1. On the **Redirection Rules** tab, click **Add**.

The **Define Redirection Rule** dialog box appears.



Field	Definition	
Destination	Host or server destination for message traffic.	
Service	Type of Internet traffic	
	Use all ports	Apply the defined rule to all ports.
	Beginning of port range	Apply the defined rule to this range of ports.
	End of port range	
	TCP and UDP	Apply the defined rule to both TCP and UDP traffic.
	TCP only	Apply the defined rule to TCP traffic only.
	UDP only	Apply the defined rule to UDP traffic only.
Proxy Redirection	Specify how to redirect traffic.	
	Redirect via	Redirect all traffic through the extranet server selected from the list.
	Do not redirect	Route traffic directly to the specified destination without being redirected through SOCKS.
	Deny service	Deny access to the specified destination. The network connection is blocked locally instead of at the server level.

2. Select a destination from the **Destination** list.
3. Under "Service," select the **Use all ports** box to apply the rule to all services. Otherwise, select a range of ports. To select a single port, enter that port number in both the **Beginning of port range** and **End of port range** boxes.
4. Under "Proxy Redirection," select one of three redirection options.

*CAUTION: If you select **Deny Service** and the user has edit control of the configuration file, the option can be circumvented by quitting Aventail Connect or by changing the option in the dialog box.*

To edit a redirection rule

- Select the redirection rule you want to edit and click **Edit...**.

  The **Define Redirection Rule** dialog box appears with the selected data filled in. Edit any of the information.

To remove a redirection rule

- Select the redirection rule you want to remove and click **Remove**.

  The redirection rule is deleted from the dialog box.

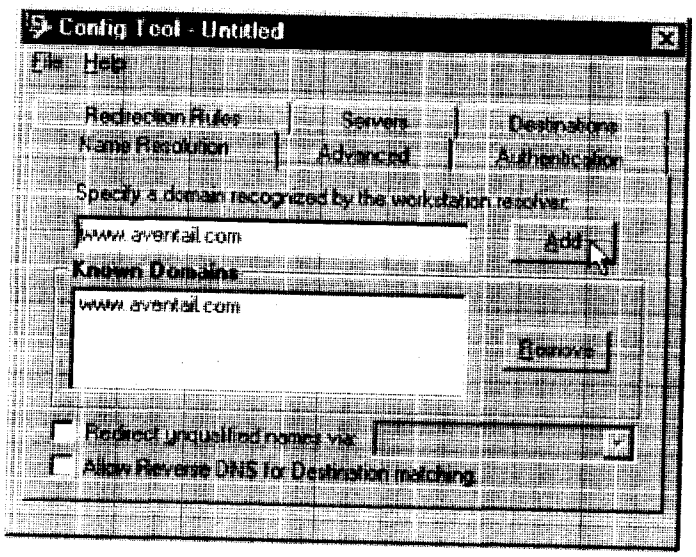
DEFINE NAME RESOLUTION

Name Resolution instructs Aventail Connect to resolve hostnames locally without needing to venture on to the Internet. This optional feature offers you another level of control over how Aventail Connect performs name resolution.

The local workstation resolver is the name resolution component of the local TCP/IP stack. This feature acts as a shortcut; hostnames matching the strings defined in the **Name Resolution** dialog box are passed to the local resolver for name resolution instead of being proxied through the SOCKS v5 server.

For example, if **aventail.com** is added to the Defined Strings list, then a workstation attempting to connect to **www.aventail.com** would perform hostname resolution using the local TCP/IP stack.

Name Resolution is specified on the **Name Resolution** tab in the Config Tool.



Field	Definition
Specify a domain recognized by the workstation resolver	New domain name
Known Domains	List of domain names that can be resolved locally
Redirect unqualified names via	Pass through unqualified hostnames to the local resolver
Allow Reverse DNS for destination matching	Enable Reverse DNS (converts IP addresses into hostnames)

To add a local domain name

- On the **Name Resolution** tab, type the new name in the **Specify a domain** box and click **Add...**.
- If necessary, select **Allow Reverse DNS for destination matching**.

  The new name is moved into the **Known Domains** box. It is now active.

CAUTION: *The reverse DNS process can create unexpected delays, causing Aventail Connect to behave unpredictably. Aventail recommends that you do not enable this option unless you specifically require the Reverse DNS functionality.*

To remove a local domain name

- Select the domain name you want to remove from the **Known Domains** box and click **Remove**.

  The domain name is removed from the list.

MANAGE AUTHENTICATION MODULES

SOCKS v5 servers often require user authentication before allowing access. Aventail Connect authentication modules display dialog boxes that prompt users to enter username and password information as well as other authentication credentials.

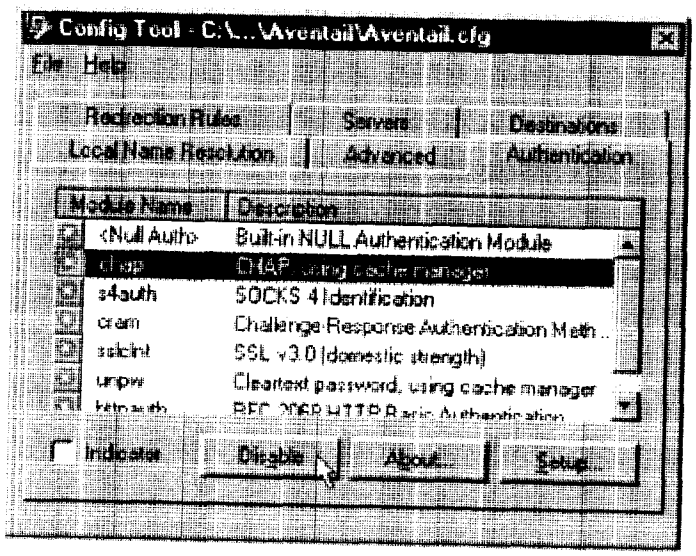
NOTE: *Not all versions of Aventail Connect have encryption enabled.*

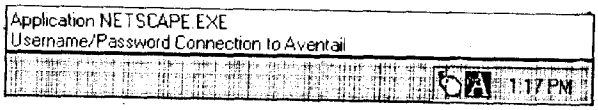
The current Aventail Connect authentication modules are SOCKS v4 Identification, Username/Password, Challenge Handshake Authentication Protocol (CHAP), Challenge Response Authentication Method (CRAM), Secure Sockets Layer (SSL), and HTTP Basic (username/password). Each of these authentication modules supports an Aventail Connect feature known as credential caching. Credential caching retains your authentication credentials once the extranet server has accepted them. Using credential caching, you can enter your credentials for an extranet server once per Aventail Connect session, rather than once for each individual connection (a tedious task for applications such as WWW browsers).

Aventail Connect can cache authentication credentials in memory, based on the option you select in the **Authentication** dialog box. Memory caching stores the credentials for the current session only. When you restart Aventail Connect or Windows, the memory cache is flushed and you must reenter your credentials as prompted.

SEE ALSO: *For additional information on credential caching, see "Credential Cache Timeouts" in the "Advanced Tab Options" section of this Administrator's Guide.*

Authentication modules are managed and configured through the **Authentication** tab in the Config Tool.



Field	Definition
Module Name	The name of the authentication module on disk. `<Null Auth>` indicates that no authentication module will be used.
Description	The description of the authentication method.
Indicator	Check this option to display network traffic passing through a selected authentication/encryption module. See the example below (for Windows 95, Windows 98, and Windows NT 4.0). 

Each authentication module includes its own module-specific configuration. To view or edit a module's configuration, select the module from the list on the **Authentication** tab and then click **Setup**. An options dialog box for the specific module will appear.

Enable and disable authentication modules with the **Disable/Enable** button. By default, the modules are all enabled. The green button next to the module name indicates an active module. This is the default state of all the modules. The green button changes to red when you disable the module.

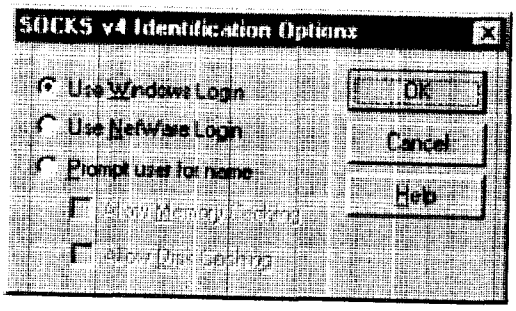
To configure the SOCKS 4 Identification module

Aventail Connect includes backward compatibility for the SOCKS 4 protocol. SOCKS 4 does not support password authentication, so only your username is sent unencrypted to the extranet (SOCKS) server along with your connection request.

Your username is determined by entries in the **SOCKS 4 Identification Module Configuration** dialog box.

1. On the **Authentication** tab in the Config Tool, click **s4auth** (SOCKS v4 Identification) and click **Setup**.

   The **SOCKS 4 Identification Options** dialog box appears.



Field	Description	
Use Windows Login	Identify users by their Windows Login names.	
Use NetWare Login	Identify users by their Novell NetWare Login names.	
Prompt user for name	Identify users by the names they enter for this specific purpose.	
	Allow Memory Caching	Stores credentials in memory for this session only. Cache is flushed upon restart; credentials must be reentered as prompted.
	Allow Disk Caching	This option is currently unavailable. (Stores credentials on disk for future sessions.)

2. When you select the **Prompt user for name** option, you must also select the desired caching option. (Currently only Memory Caching is available.)
3. After making appropriate selections, click **OK**.

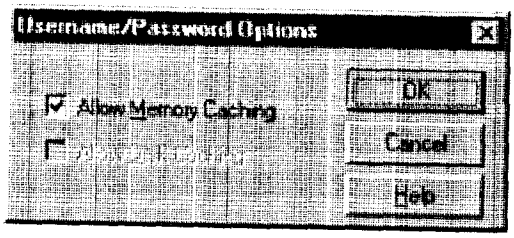
   The dialog box closes and the Config Tool reappears.

To configure the Username/Password authentication module

Aventail Connect supports the RFC 1928 (Internet standards document) username and password authentication protocol. This authentication method sends your username and password *in cleartext* across the network to the destination server. The **Username/Password authentication module** dialog box contains only credential caching options.

1. On the **Authentication** tab in the Config Tool, select **unpw** and click **Setup**.

   The **Username/Password Options** dialog box appears.



Field	Description
Allow memory caching	Stores credentials in memory for this session only. Cache is flushed upon restart; credentials must be reentered as prompted.
Allow Disk Caching	This option is currently unavailable. (Stores encrypted credentials on disk for future sessions.)

2. The selection defaults to **Allow Memory Caching**. Click **OK**.

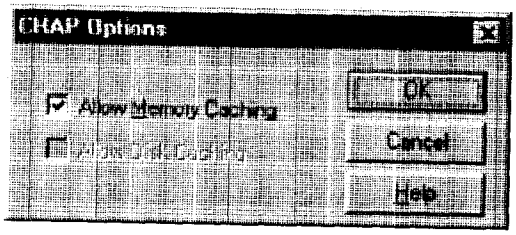
   The dialog box closes and the Config Tool reappears.

To configure the CHAP authentication module

Aventail Connect supports the Challenge Handshake Authentication Protocol (CHAP). This authentication method sends your username and password *encrypted* across the network to the destination server. The **CHAP authentication module** dialog box contains only credential caching options.

1. On the **Authentication** tab in the Config Tool, select **chap** and click **Setup**.

The **CHAP Options** dialog box appears.



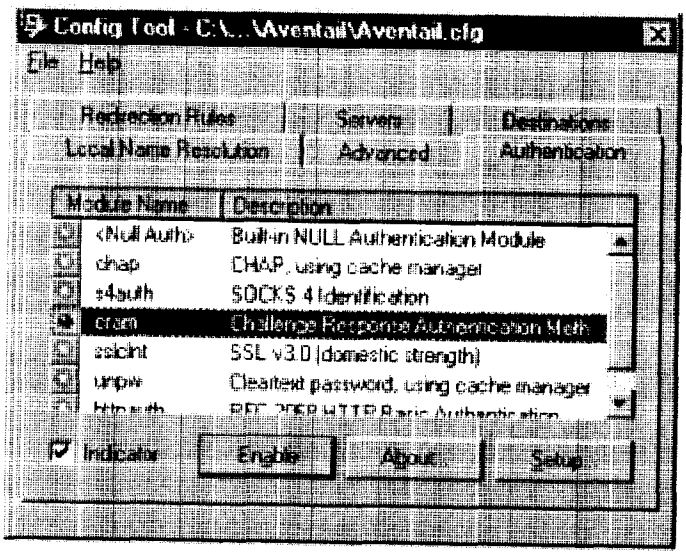
Field	Description
Allow memory caching	Stores credentials in memory for this session only. Cache is flushed upon restart; credentials must be reentered as prompted.
Allow disk caching	This option is currently unavailable. (Stores encrypted credentials on disk for future sessions.)

2. The selection defaults to **Allow Memory Caching**. Click **OK**.

   The dialog box closes and the Config Tool reappears.

To configure the CRAM authentication module

Aventail Connect supports the Challenge Response Authentication Method (CRAM). This authentication method sends your username and passcode as cleartext between extranet (SOCKS) servers, but *encrypted* between servers that support CRAM. Typically, CRAM subauthenticates within SSL, which provides both encryption and credential caching options.



You do not need to configure the CRAM authentication module. You can enable/disable it, by clicking on the **Disable/Enable** button. The button at the left of the module name will change from green to red, accordingly.

To configure the SSL security module

Aventail Connect supports Secure Sockets Layer (SSL) v3.0, a session-layer protocol for securing connections in a general, protocol-independent fashion.

NOTE: *Currently, SSL is a TCP-only enhancement. When using SSL with User Datagram Protocol (UDP) applications, bulk data is passed without encryption.*

Normally SSL servers are required to have an RSA key pair and a certificate. Aventail uses an RSA algorithm to create a cryptographic system: a private key (which, as the name suggests, is kept absolutely private and never shared) and a public key (which is widely published).

NOTE: *In versions of Aventail Connect that do not include encryption, SSL is not available.*

However, as the client, you normally must then establish some kind of relationship between your RSA public key and the identity of the server, so that somebody else cannot create their own RSA key information and use it to impersonate your server. *Certificates* establish this relationship. A certificate is essentially an electronic "statement" that verifies that a certain RSA public key is associated with a particular name.

Certificates are issued by a Certification Authority (CA), and are linked together to form a construct called a certificate *chain of authorities*, each one having a previous entity vouching for its identity. In practice, chains generally include two certificates: one confirming the identity of the server, and the other—a "root" certificate—containing the identity and public key of the CA.

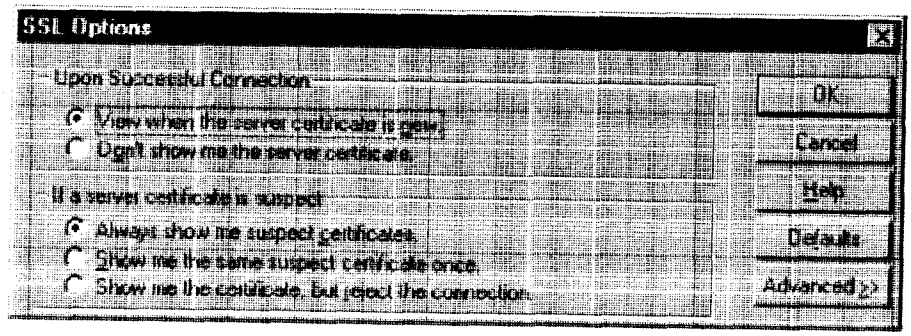
Certificates contain special integrity checks and electronic signatures that verify that the certificate is genuine, was issued by a certification authority, and was not tampered with. Anybody can issue a certificate that says anything; the client must know who issued the certificate, and have some trust relationship in order to believe that it is in fact true. The client has a list of trusted CAs. A set of certificate chains can be structured as a tree, with new certificates stemming from old ones. A base CA is sometimes called the "root" or "trusted root" of this tree.

It is becoming common practice for both clients and servers to exchange certificate information. However, in Aventail Connect the client-side of this exchange is transparent. The client only needs to deal with the information from the server certificate and this is done through the SSL module.

The **SSL module** dialog box contains an initial set of options regarding the viewing of certificates.

1. On the **Authentication** tab in the Config Tool, select **sslclnt** (SSL v3.0) and click **Setup**.

   The **SSL Options** dialog box appears.



Field		Description
Upon Successful Connection		The certificate is valid.
	View when the server certificate is new.	Upon successful connection, display the server certificate if it has not been displayed during the current session.
	Do not show me the certificate.	Never display a valid server certificate.
If a server certificate is suspect		The certificate may not be valid.
	Always show me suspect certificates.	Each time Aventail Connect suspects a certificate may not be valid, show the certificate.
	Show me the same suspect certificate once.	Once a suspect certificate has been accepted by the user, do not display it again.
	Show me the certificate, but reject the connection.	Reject the connection, but display the suspect certificate.

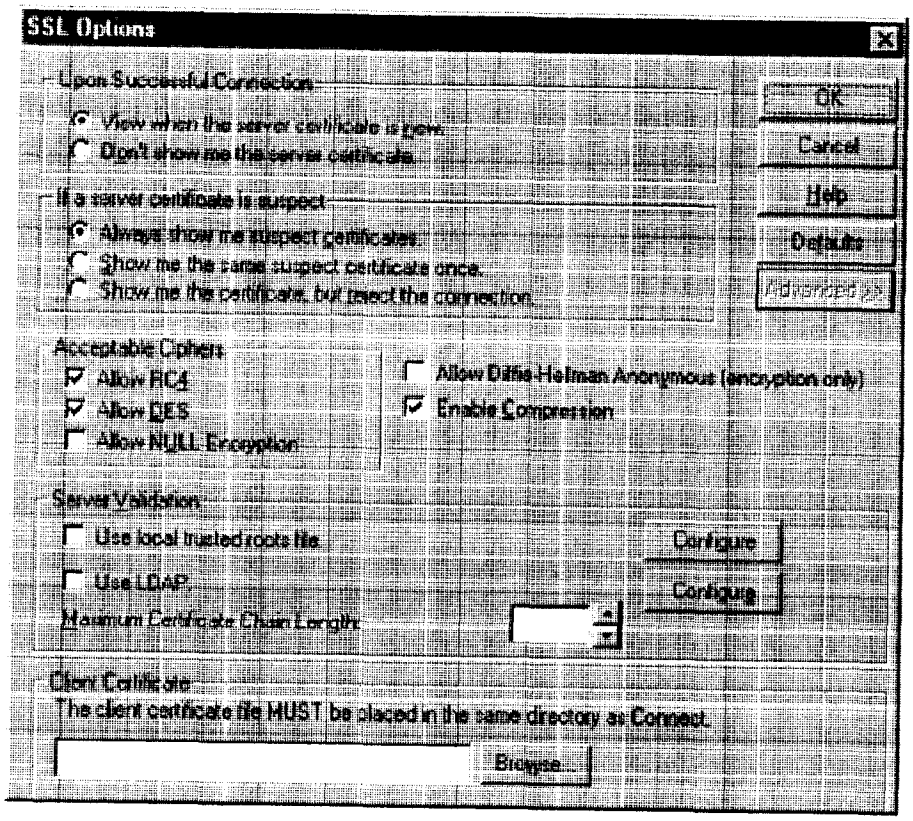
2. Select an action that Aventail Connect must take once it accepts the validity of the server certificate. (Under normal circumstances, the server will provide Aventail Connect with a certificate to match one of Aventail Connect's trusted roots, if any exist):
   - **View when the server certificate is new:** Aventail Connect displays the certificate the first time it is seen. The certificate will not appear on subsequent connections to the same extranet server.
   - **Do not show me the server certificate:** Aventail Connect will never display a valid certificate.
3. Select an action that Aventail Connect must take if it receives a server certificate that is suspect:
   - **Always show me suspect certificates**: Aventail Connect will display suspect certificates each time they are received. The **Certificate** dialog box will appear for each new connection to the server(s) sending a suspect certificate. (This option allows you to continue the connection despite the fact that the certificate is questionable.) The SSL module authenticates the server's certificate based on the following questions:
     - Is the certificate valid?
     - Did a trusted certificate authority (CA) issue the certificate?
     - Is the name established by the certificate the same as the name of the server for this connection?

   If a certificate does not pass all three tests, it is considered a suspect certificate.
   - **Show me the same certificate once:** Aventail Connect will display a suspect certificate the first time that it is received. If you choose to

maintain the connection, the questionable certificate will not be displayed again during the current session.

- **Show me the certificate, but reject the connection:** Aventail Connect will reject a connection if the certificate is suspect. It will display the certificate to allow you to view it.

4. Click **Advanced** in the dialog box to show the acceptable cipher (a cryptographic algorithm used to encrypt the data stream) options.



Field	Description	
Acceptable Ciphers		
Allow RC4	Offer the RC4 cipher to the server.	
Allow DES	Offer the DES cipher to the server.	
Allow NULL Encryption	Do not encrypt using SSL. SSL will be used to authenticate only.	
Allow Diffie-Hellman Anonymous	Do not authenticate the server; only do encryption.	
Enable Compression	Use SSL compression to improve performance when slower connections are detected.	
Server Validation		
Trusted Roots	Use a trusted roots file to validate trusted certificate chain roots. *NOTE: The trusted roots file MUST be placed in the same directory as the Aventail Connect configuration file.*	
	Configure	Configure trusted roots
LDAP	Use an LDAP server to validate trusted certificates.	
	Configure	Configure LDAP
Maximum Chain Length	Specify the maximum allowable certificate-chain length.	
Client Certificate	Select a client certificate file. *NOTE: The client certificate MUST be placed in the same directory that Aventail Connect was installed to.*	
	Browse	Select the specific file

During the initial SSL connection, the client and the server negotiate which cipher to use. Checking a particular cipher in the dialog box does not mean that it will be used. Instead, each checked cipher is *offered* to the server, but the server determines which cipher to use. If the server requires a cipher that is not selected in this dialog box, the authentication will fail.

Any or all of the acceptable cipher options can be selected:

- **Allow RC4:** Aventail Connect encrypts the information using the RC4 cipher.
- **Allow DES:** Aventail Connect encrypts the information using the DES cipher.
- **Allow NULL Encryption:** Aventail Connect allows the server to select *no* encryption. Message integrity is still assured, but the data will be sent in cleartext.
- **Allow Diffie-Hellman Anonymous:** Aventail Connect will be able to communicate with the extranet (SOCKS) server without requiring a

server certificate. The client and server will not exchange certificates, so there will be no authentication. The encryption will still be negotiated, and the data stream will still be encrypted (unless NULL encryption is chosen by the server).

- **Enable Compression:** To speed the encryption process and enhance overall performance, Aventail Connect will automatically compress encryption when a narrow bandwidth and/or slow modem are detected.

5. If necessary, add (or delete) a trusted roots (`*.rot`) file and/or an LDAP server definition.

To add or remove a trusted root

a. In the **SSL Options—Advanced** dialog box, under "Server Validation," select **Use local trusted roots file**, and then click **Configure**.

The **Trusted Roots** dialog box will appear.

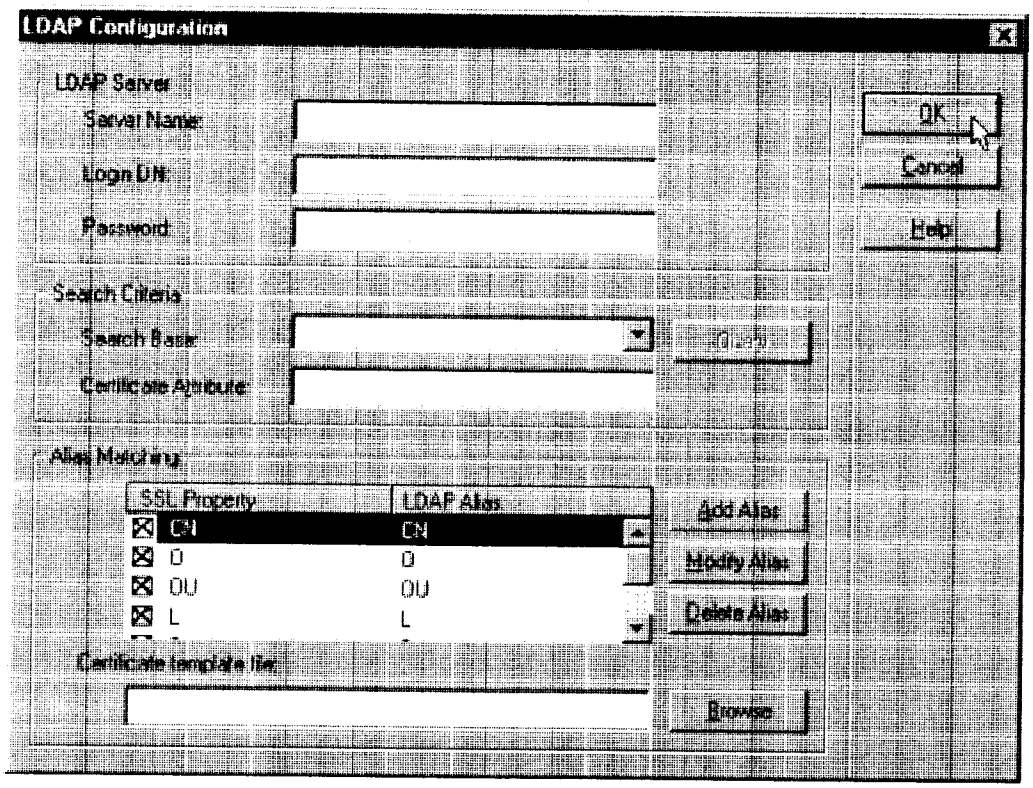
b. Enter the name of the trusted roots file, or click **Browse** to search for the file, and then click **OK**.

CAUTION: The trusted root file must be in the same directory as the Aventail Connect configuration file.

To configure LDAP

a. In the **SSL Options—Advanced** dialog box, under "Server Validation," select **Use LDAP**, and then click **Configure**.

The **LDAP Configuration** dialog box appears.



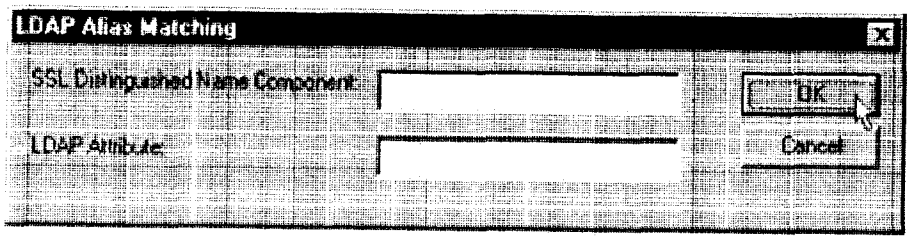
Field	Description	
LDAP Server		
Server Name	Enter the LDAP server hostname.	
Login DN	Enter the login DN (distinguished name) for the LDAP server.	
Password	Enter the password for the LDAP server.	
Search Criteria		
Search Base	Enter the DN to use as the search base.	
	Query	Search available DNs to use as search base.
Certificate Attribute	Enter the certificate attribute.	
Alias Matching		
SSL Property/LDAP Alias	Names of SSL property and corresponding LDAP alias.	
Add Alias	Add an LDAP alias/SSL property.	
Modify Alias	Modify an LDAP alias.	
Delete Alias	Delete an LDAP alias/SSL property.	
Certificate template file:	*(Optional)* Enter name of certificate file to use as template.	
	Browse	Search available certificate files.

b. Under "LDAP Server," enter the LDAP server name, and the DN and password that you want to log in under.

c. Under "Search Criteria," enter or select the DN to use as the search base, and enter the certificate attribute. (In most cases, the certificate attribute will be "usercertificate.")

d. Under "Alias Matching," select the SSL properties that you want to use as search criteria.

   If necessary, you can modify any of the LDAP aliases to map to the SSL properties. To modify an LDAP alias, click **Modify Alias**. In the **LDAP Alias Matching** dialog box, enter the LDAP Attribute that will map to the SSL Distinguished Name Component. You can also **Add** or **Remove** an SSL property/LDAP alias in the **LDAP Alias Matching** dialog box.



In the **Certificate template file:** box, you can specify a certificate file to use as a template. If you specify a certificate template file, Aventail Connect will automatically populate the "SSL Property/LDAP Alias" box with the attributes used in the specified certificate template file.

e. Click **OK**.

6. If Aventail Connect sends a client certificate to the server during the initial authentication exchange, it sends the certificate identified in the **Client Certificate** window. To load the client certificate, press **Browse** and then select the client certificate (`*.cer`) from the Aventail Connect directory. Only the filename of the certificate file loads via the **Browse** button, and not the pathname.

CAUTION: *The client certificate file must be placed in the Aventail Connect directory.*

When Aventail Connect receives a certificate from a server, it looks at the root of the certificate chain and matches it against the Aventail Connect list of trusted roots.

You can specify the maximum number of certificates in a certificate chain. The default maximum length is two certificates. In most instances, Aventail recommends allowing no more than two certificates to form a chain, although you can specify up to ten. The longer the certificate chain, the less secure the chain is.

CAUTION: *In most instances, Aventail recommends allowing no more than two certificates in a certificate chain. Allowing more than two certificates can compromise security.*

7. After making appropriate selections, click **OK**.

PKCS #12 Certificates for User Authentication

Aventail Connect supports PKCS #12-formatted X.509 client certificates for SSL authentication. PKCS #12-formatted certificates are stored in a portable format for easy exchange between applications. You can generate client certificates by enrolling with a public-key infrastructure (PKI), such as VeriSign OnSite. You can then use your Web browser to export the client certificate to a PKCS #12 file in

the Aventail program directory. When users connect to an Aventail ExtraNet Server for the first time, they will be prompted to select a certificate.

To export a PKCS #12-formatted X.509 certificate

1. Using a Web browser and a CA, such as VeriSign Onsite, obtain a client certificate.
2. Export the certificate to a file in the Aventail program directory. You can use any filename. This step varies from browser to browser.

   Microsoft Internet Explorer 4.01

   a. Select **View|Internet Options...|Content|Certificates|Personal...**.
   b. Select the certificate that you want to export, and click **Export...**.
   c. Specify a password to protect the certificate.
   d. Save the file to the Aventail Connect program directory.

CAUTION: On Windows NT, Microsoft Internet Explorer 4.01 does not export PKCS #12 certificates properly. This problem was corrected in Microsoft Internet Explorer 5.0.

   Microsoft Internet Explorer 5.0

   a. Select **Tools|Internet Options...|Content|CertIficates|Personal...**.
   b. Select the certificate that you want to export, and click **Export...**.
   c. In the Certificate Export Wizard, click **Export the Private Key**.
   d. Specify a password to protect the certificate.
   e. Select the PKCS #12 format.
   f. Select **Include all certificates in the certificate path if possible**.
   g. Save the file to the Aventail Connect program directory.

   Netscape Navigator 4.5

   a. Click the Lock icon in the lower-left corner of the main Netscape Navigator window.
   b. Select **Certificate|Yours**.
   c. Select the certitficate that you want to export, and click **Export**.
   d. Specify a password to protect the certificate.
   e. Save the file to the Aventail Connect program directory.

3. Use an Aventail Connect configuration file and server setup that forces the user to authenticate using client certificates. Configure the Aventail ExtraNet Server.
4. Initiate a connection that forces the user to authenticate. You will be prompted for a certificate file. Select the certificate that you just exported, and then click **OK**.

PKCS #11 Smart Cards for User Authentication

Aventail Connect can use client certificates that are stored on PKCS #11-compatible smart cards for SSL authentication. Currently, Aventail Connect supports the DataKey and Spyrus Rosetta smart cards.

Aventail Connect will be prompted for a file (or smart card) containing certificate information only when the SOCKS server requests client authentication using a certificate. If a SOCKS server requests client authentication with a certificate, and no certificate is already specified for that host, the user will be prompted to select a certificate. You can configure passwords or PINs to be cached to memory, or you can specify that users enter passwords or PINs each time they use a smart card to authenticate.

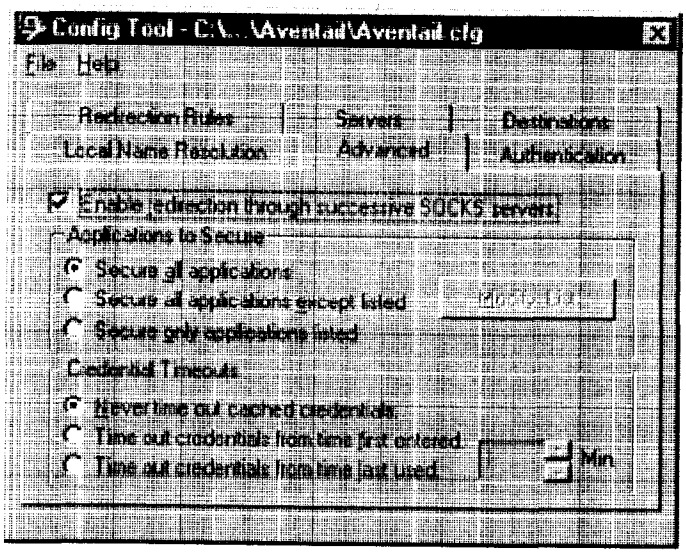
To configure PKCS #11 smart-card user authentication

1. Use a smart card with an X.509 certificate stored on it.
2. Install the appropriate smart card software on the user's computer.
3. Include the public certificates of the CA (and any intermediary CAs) for the client certificate in the trusted roots file that Aventail Connect is configured to use.
4. Configure Aventail Connect to redirect to an Aventail ExtraNet Server that requires client certificates.
5. Initiate a connection.
6. When prompted, specify whether you want to authenticate with a client certificate that is stored in a file, a client certificate that is stored on a smart card, or no client certificate at all.
7. Aventail Connect will prompt you for the path of the dynamic link library (DLL) for the smart card's PKCS #11 module. This is the same DLL that is used with Netscape Navigator. Enter the DLL pathname and click **OK**.
8. Aventail Connect will display a list of all detected smart cards on the system. If you have not yet inserted your smart card into the appropriate reader, insert it and click **Refresh List**.
9. Select your smart card and click **OK**.
10. If the smart card is protected with a PIN, you will be prompted to enter it.
11. Select the private key you want to use, and click **OK**.

NOTE: *Once you specify a smart card token or client certificate to be used with a server, this setting will be remembered indefinitely. To reset the setting, select* ***Credentials*** *from the Aventail Connect system tray menu, select (highlight) the credentials, and click* ***Delete****. Your PIN will not be remembered.*

ADVANCED TAB OPTIONS

The **Advanced** tab in the Config Tool contains three advanced options. In the **Advanced** tab, you can allow SOCKS tunneling through successive extranet (SOCKS) servers, secure selected applications, and set credential cache time-outs.



ALLOW SOCKS TUNNELING THROUGH SUCCESSIVE EXTRANET SERVERS

Once servers and destinations are defined, you can direct SOCKS traffic through successive extranet (SOCKS) servers.

On the **Advanced** tab in the Config Tool, select the **Enable redirection...** box to allow credential information to forward to successive extranet servers.

SECURE SELECTED APPLICATIONS

This option allows you to:

- secure all applications except those listed,
- secure only the applications that are listed,
- or secure all applications, enabling neither exclusion nor inclusion.

NOTE: *You can exclude and include only 32-bit applications. You cannot exclude and include 16-bit applications.*

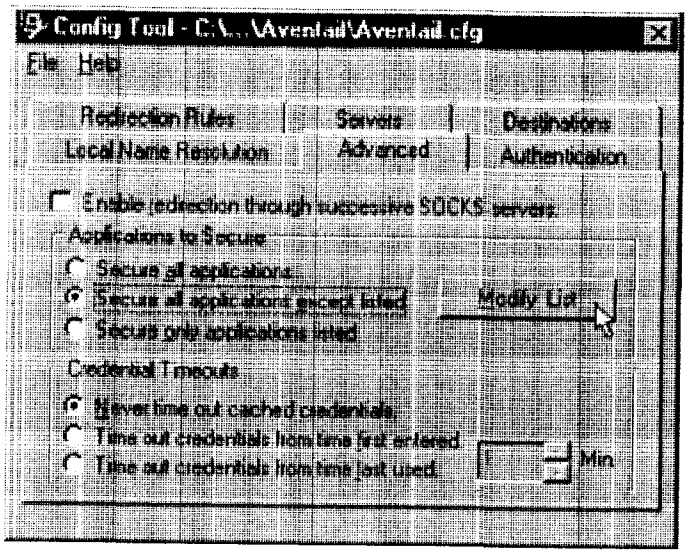
You can exclude or include specified applications in the Exclusion/Inclusion List. With the Exclusion/Inclusion List, you can secure all applications *except* those on the list, or you can secure *only* those applications on the list. The default setting is to secure (hook) *all* network applications.

Excluding Applications

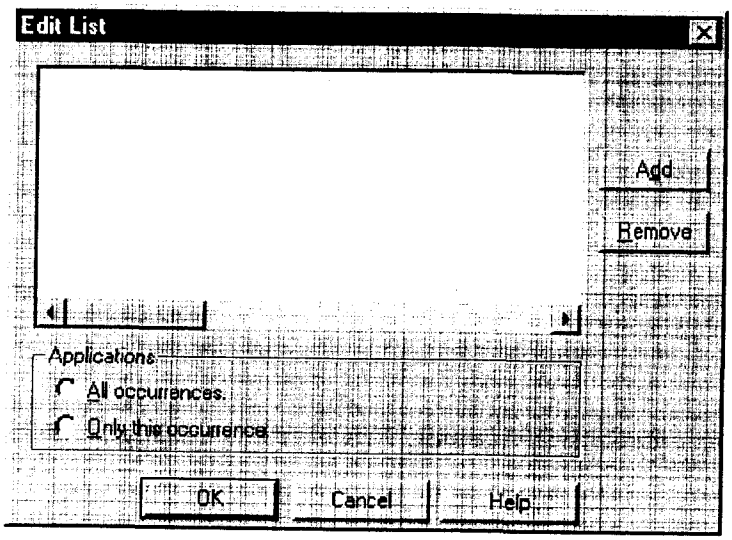
You can exclude specific applications through the Exclusion/Inclusion List. When you enable the "Secure all applications except listed" option, Aventail Connect will not proxy any applications that are on the Exclusion/Inclusion List.

To exclude an application

1. Under "Applications to Secure," select **Secure all applications except listed** and click **Modify List**.

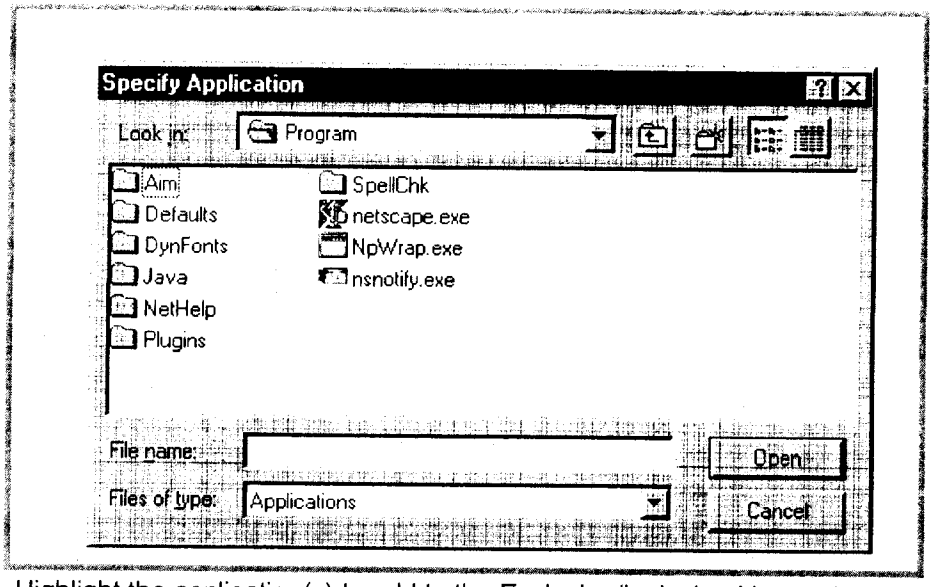


The **Edit List** dialog box appears.



2. Click **Add...**.

The **Specify Application** dialog box appears.



3. Highlight the application(s) to add to the Exclusion/Inclusion List, and then click **Open**.

   The **Specify Application** dialog box disappears and the applications are now in the **Edit List** dialog box.

4. In the **Edit List** dialog box, select **All occurrences** or **Only this occurrence**.

NOTE: *You may have more than one path (instance) of a specified filename (e.g.,* `ftp.exe`*). You can choose to exclude one specified application, with a fully qualified pathname (e.g.,* `C:\Windows\Sys32\ftp.exe`*), or all instances of a specified filename (e.g., all instances of* `ftp.exe`*).*

- **Only this occurrence:** Selecting this option excludes only the specified application.
- **All occurrences:** Selecting this option excludes all applications with the specified filename.

To undo application exclusion

1. Under "Applications to secure," select **Secure all applications except listed**, and then click **Modify List**.

   The **Edit List** dialog box appears.

2. Highlight the application you want to remove from the Exclusion/Inclusion List, and then click **Remove**.

   The application is removed from the Exclusion/Inclusion List.

Including Applications

You can include specific applications through the Exclusion/Inclusion List. When you enable the "Secure only applications listed" option, Aventail Connect will hook only those applications that are on the Exclusion/Inclusion List.

To include an application

1. Under "Applications to secure," select **Secure only applications listed**, and then click **Modify List**.

   The **Edit List** dialog box appears.

2. Click **Add**.

   The **Specify Application** dialog box appears.

3. Highlight the application(s) to add to the Exclusion/Inclusion List, and then click **Open**.

   The **Specify Application** dialog box disappears and the applications are now in the **Edit List** dialog box.

4. In the **Edit List** dialog box, select **All occurrences** or **Only this occurrence**.

NOTE: *You may have more than one instance of a specified application (e.g., `ftp.exe`). You can choose to include one specified application, with a fully qualified pathname (e.g., `C:\Windows\Sys32\ftp.exe`), or all instances of a specified application (e.g., all instances of `ftp.exe`).*

- **Only this occurrence:** Selecting this option excludes only the specified application.
- **All occurrences:** Selecting this option excludes all applications with the specified filename.

To undo application inclusion

1. Under "Applications to secure," select **Secure only applications listed**, and then click **Modify List**.

   The **Edit List** dialog box appears.

2. Highlight the application you want to remove from the Exclusion/Inclusion List, and then click **Remove**.

The application is removed from the Exclusion/Inclusion List.

Securing all Applications

You can secure *all* applications, enabling neither exclusion nor inclusion. When you secure all applications, Aventail Connect ignores any applications on the Exclusion/Inclusion List.

To secure all applications

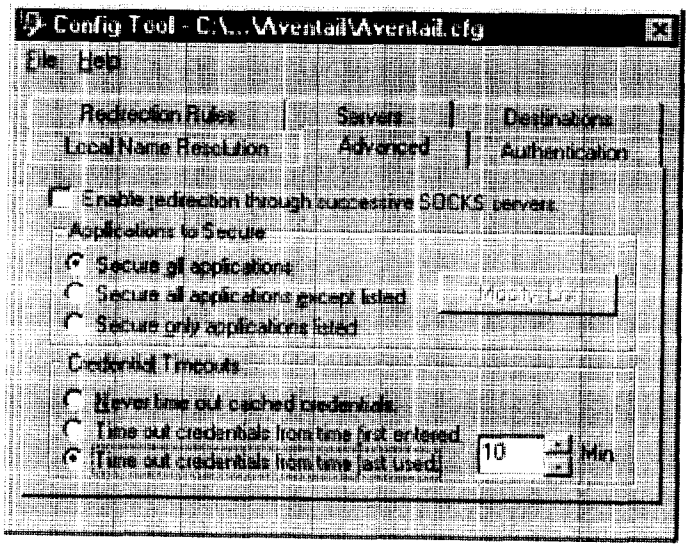
- On the **Advanced** tab, under "Applications to Secure," select **Secure all applications**.

*NOTE: Aventail Connect secures all applications by default. Unless you need to exclude or include specific applications, Aventail recommends that you use the default **Secure all applications** setting.*

CAUTION: Microsoft Internet server products (including Microsoft Internet Information Server (IIS) and Microsoft Peer Web Server) include inetinfo.exe, which conflicts with Aventail Connect 3.1. To eliminate this conflict, exclude inetinfo.exe through the Application Exclusion/Inclusion List in the Config Tool.

CREDENTIAL CACHE TIMEOUTS

With the credential cache timeout feature, you can control when credentials expire (time out). If a user has not made a connection to the extranet (SOCKS) server for a certain length of time (determined by the administrator), then the credentials will automatically be deleted from the credential cache. If a credential times out, the user must reauthenticate by entering the proper credentials before regaining access to the extranet. This feature can help to prevent unauthorized users from gaining access to secured areas.



There are three credential cache timeout options.

- **Never time out cached credentials:** Credentials never time out.

- **Time out credentials from time first entered:** Credentials time out ***x*** minutes after the user first entered the credentials (where "***x***" is the number of minutes you enter in the **Min.** box).
- **Time out credentials from time last used:** Credentials time out ***x*** minutes after the user last connected through the extranet server (where "***x***" is the number of minutes you enter in the **Min.** box).

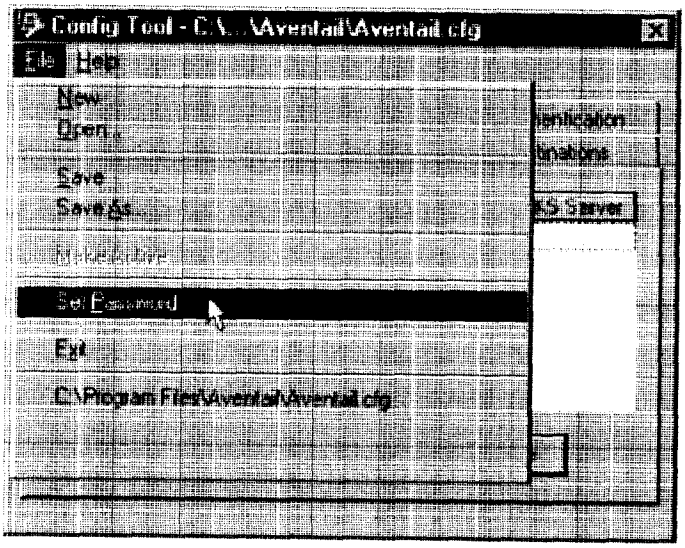
CAUTION: *If your mail program is configured to check for e-mail at regular intervals, the mail-checking frequency must be longer than the credential cache timeout. For example, if your mail program is configured to check for mail every ten minutes, you should set the credential cache to less than ten minutes.*

ENABLE PASSWORD PROTECTION

You can enable password protection for a configuration file. If you enable password protection, users will not be able to view or modify the configuration file without the assigned password. A password is not required to use the configuration file with Aventail Connect.

To enable password protection

1. From any tab of the Config Tool, select **File | Set Password**.



The **Configuration File Password** dialog box will appear.

2. Enter the desired password.
3. Reenter the password to confirm, and then click **OK**.

To disable password protection

1. From any tab of the Config Tool, select **File | Set Password**.

   The **Configuration File Password** dialog box will appear.

2. Clear the password from both boxes, and then click **OK**.

NOTE: *If you save an existing configuration file using the* ***Save As*** *command, Aventail Connect will prompt you to enter the correct password for the configuration file.*

MULTIPLE FIREWALL TRAVERSAL

To gain access to your extranet, users may need to traverse multiple firewalls. In the simplest case, this involves an employee at a partner company gaining access to the Internet via an outbound proxy server at the partner company, and having an authenticated, encrypted, and controlled connection to your internal network via an Aventail ExtraNet Server. This capability is provided in Aventail Connect 3.1 by the Aventail MultiProxy feature. Aventail Connect can open connections through SOCKS servers, through HTTP proxies, or through proxy chaining.

- **MultiProxy with SOCKS Server:** Uses a SOCKS server to control outbound access.
- **MultiProxy with HTTP Proxy:** Uses an HTTP proxy to control outbound access.
- **Proxy Chaining:** Uses two Aventail ExtraNet Servers, where one Aventail ExtraNet Server acts as a client to another Aventail ExtraNet Server.

AVENTAIL MULTIPROXY

The Aventail MultiProxy feature allows Aventail Connect to traverse multiple firewalls by making connections through successive proxy servers. Aventail Connect makes a connection with each proxy server individually. Each proxy server forms a link in a chain that connects Aventail Connect to the final destination. Any or all of the proxy servers can apply authentication and access control rules. Proxies can be Aventail ExtraNet Servers, other SOCKS 5 servers, SOCKS 4 servers, or HTTP proxies.

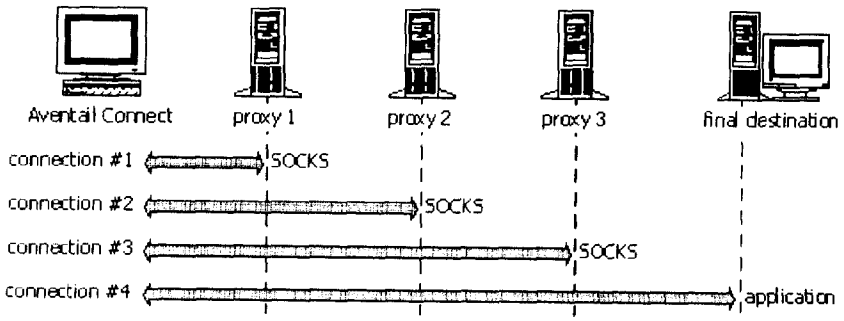
Using an HTTP proxy server to control outbound traffic eliminates the need to install a separate SOCKS server. This HTTP proxy can filter outbound connection requests and route those requests to the specified servers. MultiProxy supports RFC 2068 HTTP Basic (username/password) authentication. If your proxy uses HTTP Basic (username/password) authentication, Aventail Connect will store the username and password information in the credential cache, as it does with SOCKS servers.

NOTE: *The MultiProxy feature supports the use of HTTP proxies in Aventail Connect 3.1 only. HTTP proxies cannot be used in Aventail Connect 2.6.*

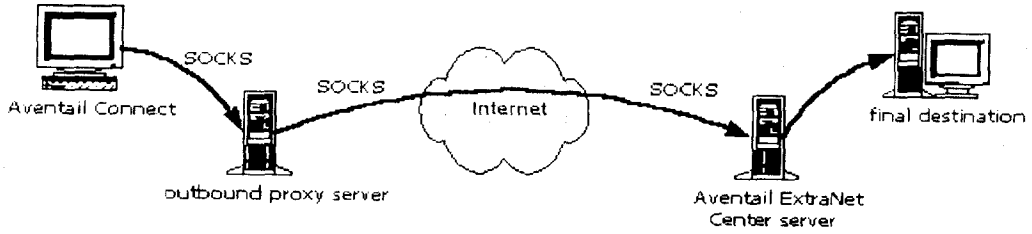
The steps for making a connection using MultiProxy are:

1. The client application requests access to the destination server.
2. Aventail Connect establishes a connection with the outbound server (SOCKS server or HTTP proxy). Aventail Connect then sends the access request to the outbound server, specifying the Aventail ExtraNet Server as the destination. The user authenticates with the outbound server, if necessary.
3. Aventail Connect instructs the outbound server to establish a connection with the Aventail ExtraNet Server on the specified port. The user authenticates with the Aventail ExtraNet Server, if necessary.
4. Aventail Connect instructs the Aventail ExtraNet Server to proxy its connection to the final destination.
5. Once the connection between the client and the Aventail ExtraNet Server is established, the outbound server simply relays the data.

The following example illustrates the connections made during a MultiProxy connection through three proxy servers.



In the following diagram, the Aventail ExtraNet Server acts as both a *destination* and a *server*. It is a destination because a proxy server routes traffic to it. It is a server because it routes traffic to the final destination.



CAUTION: If using an HTTP proxy, you must configure your HTTP proxy and firewall to allow HTTPS/SSL connections to port 1080, **OR** *you must run the Aventail ExtraNet Server on port 443 or port 563.*

Configuring Aventail MultiProxy

You have two options for configuring MultiProxy. You can configure Aventail Connect 3.1 to redirect all Internet traffic (including extranet traffic) through your outbound proxy, or you can configure Aventail Connect 3.1 to redirect only extranet traffic through your outbound proxy.

To configure Aventail MultiProxy

1. Create a destination ("Final destination").
2. Create a server ("Extranet server").
3. **To redirect only extranet traffic:** Create a destination ("Extranet server"), using the same information from step 2, above.

   -OR-

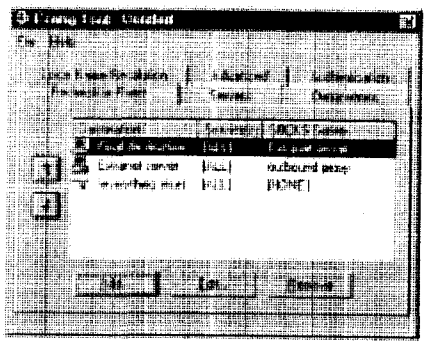
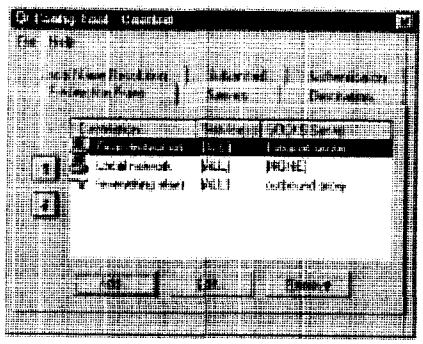
   To redirect all Internet traffic (including extranet traffic): Create a destination ("Local network," the network local to Aventail Connect).

NOTE: If you have multiple domains or subnets, you may need to create multiple destinations.

4. Create a server ("Outbound proxy"). This can be a SOCKS 5, SOCKS 4, or HTTP proxy server.
5. Create a redirection rule (Redirect "Final destination" through "Extranet server").
6. **To redirect only extranet traffic:** Create a redirection rule (Redirect "Extranet server" through "Outbound proxy"). Do not redirect "(everything else)."

   -OR-

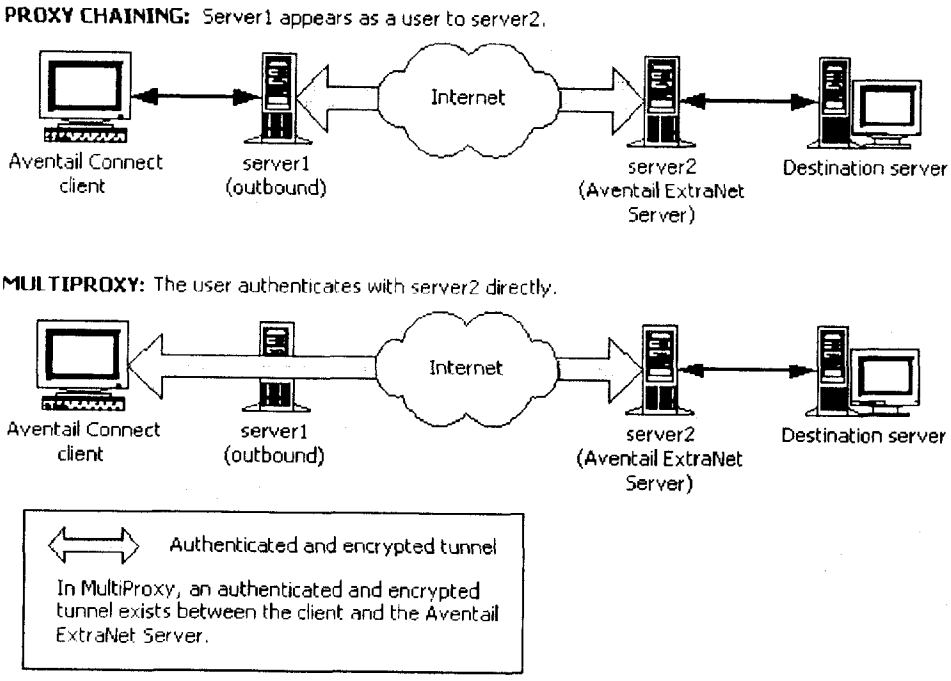
   To redirect all Internet traffic (including extranet traffic): Create a redirection rule (Do not redirect "Local network"). Redirect "(everything else)" through the outbound proxy. (**NOTE:** Your outbound proxy must belong to "Local network.")

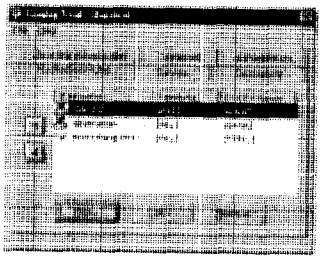
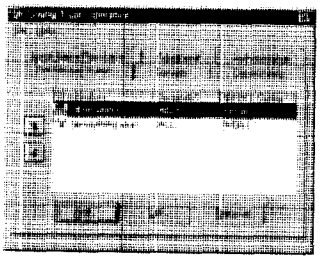
Redirect only extranet traffic	Redirect all Internet traffic (including extranet traffic)
	
Redirect only the extranet traffic through the outbound proxy. Leave all other traffic alone.	Redirect all Internet traffic through the outbound proxy. Leave only "Local network" traffic alone.

PROXY CHAINING

Proxy chaining is an Aventail ExtraNet Server feature. With proxy chaining, Aventail ExtraNet Servers forward connections for certain destinations to other proxy servers.

The following diagram and table illustrate the differences between MultiProxy and proxy chaining. In many cases, MultiProxy is the preferred method for traversing multiple firewalls. With MultiProxy, *each* proxy server can provide authentication, access control, and encryption.



Criteria	MultiProxy	Proxy Chaining
Server 1	Can be Aventail ExtraNet Server, other SOCKS 5 server, SOCKS 4 server, or HTTP proxy.	Must be Aventail ExtraNet Server.
Server 2	Must be Aventail ExtraNet Server.	Must be Aventail ExtraNet Server.
Authentication to Server 1	User authenticates (if necessary).	User authenticates.
Authentication to Server 2	User authenticates.	Server 1 authenticates automatically.
Trust model for Server 2	Not inherited. Each user must individually authenticate with Server 2.	Inherited from Server 1. Server 2 trusts everyone who authenticates to Server 1 equally.
Access control rules	Can be for specific users.	Treats everyone who authenticates to Server 1 equally.
Client configuration redirection rules		
Advantages	• Server 1 can be an Aventail ExtraNet Server, other SOCKS 5 server, SOCKS 4 server, or HTTP proxy. • Most secure, because no security policy is inherited from Server 1.	• Client is aware of Server 1 only. • User authenticates only once, to Server 1.
Disadvantages	• User may need to authenticate more than once. • Client must be aware of Server 1 and Server 2.	• All users connecting through Server 1 appear as a single user to Server 2.

HTTP Proxies and Web Browsers

Extranets often include Web pages that must be viewed with a Web browser. When a Web browser uses an HTTP proxy server, Aventail Connect sees connections being made to the HTTP proxy rather than to the final destination. Therefore, Aventail Connect cannot redirect the connections to the Aventail ExtraNet Server or provide authentication and encryption. For Aventail Connect to function properly, the Web browser cannot use the HTTP proxy to connect with sites protected in the extranet; this is because Aventail Connect must redirect and encrypt connections. The Web browser can still use the HTTP proxy to connect to sites that are not protected in the extranet.

If access to Web pages behind the Aventail ExtraNet Server requires users to connect through a Web browser (e.g., Microsoft Internet Explorer or Netscape Navigator), you must configure the Web browser to not use the HTTP proxy in the Web browser for those sites protected in the extranet.

When users need to access Web pages behind an Aventail ExtraNet Server, you must properly configure the Web browser.

Configuring Aventail Connect and the Web Browser

There are two approaches to configuring Aventail Connect for use with a Web browser.

- Configure the Web browser to not use the HTTP proxy for any traffic. (Aventail Connect redirects all connections through the outbound proxy.)

-OR-

- Configure the Web browser to not use the HTTP proxy for only those sites that are protected in the secure extranet. (Aventail Connect redirects only extranet connections through the outbound proxy.)

To use either approach, you must first configure Aventail Connect. The Aventail Connect configuration is the same for both approaches, whether you are configuring your browser to not use the HTTP proxy for all traffic or for protected sites only.

To configure Aventail Connect for use with a Web browser

1. In the **Servers** tab of the Config Tool, add the HTTP proxy as a server.
2. In the **Destinations** tab of the Config Tool, add the HTTP proxy as a destination.
3. In the **Redirection Rules** tab of the Config Tool, edit the "(everything else)" rule to redirect all traffic to the HTTP proxy server.
4. In the **Redirection Rules** tab, select the HTTP proxy and select the **Do not redirect** option.

CAUTION: *Make sure you do not redirect the outbound proxy. Redirecting the outbound server or proxy will instruct the outbound proxy to redirect traffic to itself, causing Aventail Connect to behave unpredictably.*

To configure the Web browser to not use the HTTP proxy for all traffic

After you have configured Aventail Connect by following the instructions above, configure the Web browser by using one of the following procedures.

- **Microsoft Internet Explorer**
  a. On the **View** menu, click **Internet Options.**
  b. Click the **Connection** tab.
  c. Click to clear the **Access the Internet using a proxy server** check box.
- **Netscape Navigator**
  a. On the **Edit** menu, click **Preferences**.
  b. Under "Category," click to expand **Advanced**, and then click **Proxies**.
  c. Select **Direct Connection to the Internet**, and then click **OK**.

To configure the Web browser to not use the HTTP proxy for protected sites only

After you have configured Aventail Connect, configure the Web browser by using one of the following procedures.

- **Microsoft Internet Explorer**
  a. On the **View** menu, click **Internet Options**.
  b. Click the **Connection** tab.
  c. Under "Proxy Server," click **Advanced**.
  d. In the **Exceptions** box, type the URL of each site that is in the protected extranet.
- **Netscape Navigator**
  a. On the **Edit** menu, click **Preferences**.
  b. Under "Category," click to expand **Advanced**, and then click **Proxies**.
  c Select **Manual Proxy Configuration**, and then click **View**.
  d. In the **Exceptions** box, type the URL of each site that is in the protected extranet.

CONFIGURING THE HTTP PROXY

To allow SSL connections to destination ports other than 443 (https) and 563 (snews), you may need to configure your HTTP proxy. Typically, if you plan to connect to a SOCKS server on port 1080 using an HTTP proxy, you must change the HTTP proxy configuration.

To avoid changing the HTTP proxy configuration, you must run the destination Aventail ExtraNet Server on port 443 or port 563, and configure Aventail Connect accordingly.

Most HTTP proxies can allow connections to port 1080. The following instructions describe how to configure the Microsoft Proxy Server, Netscape Proxy Server, or Apache Web Server to allow port 1080 connections.

- **Microsoft Proxy Server 2.0:** Follow the Microsoft instructions at **http://support.microsoft.com/support/kb/articles/q184/0/28.asp**. You must modify a registry setting with `regedt32.exe`. (`regedit.exe` will not work; you must use `regedt32.exe`.)
- **Netscape Proxy Server 3.5:** Add the following to your `obj.conf` file:
  ```
  <Object ppath="connect://*"> (all ports)
  Service fn="connect" method="CONNECT"
  </Object>
  ```
  To specify a particular port, add the following to your `obj.conf` file:
  ```
  <Object ppath="connect://*:1080"
  ```
- **Apache Web Server 1.3.2 (Linux) with Proxy Support:** The following two lines must be included in the `httpd.conf` file:

  ```
  Proxy Requests On
  ```

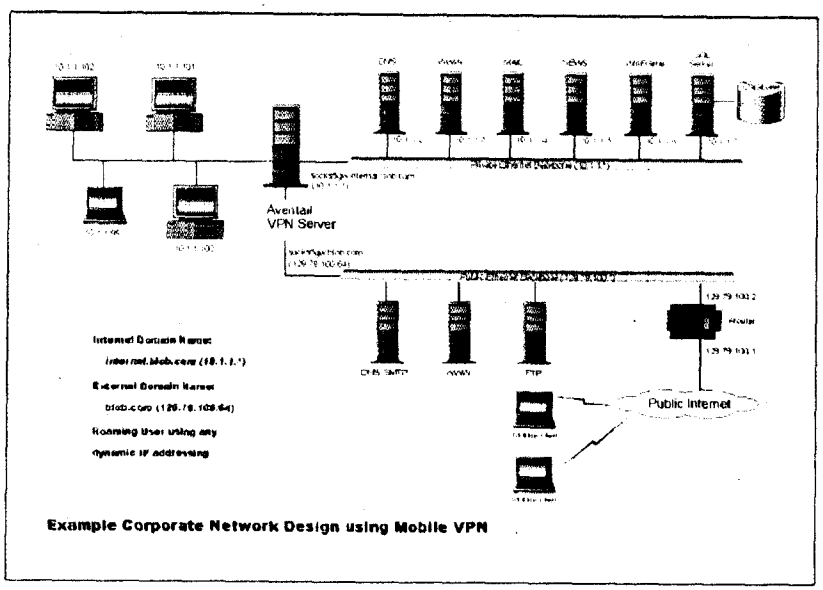
  `AllowCONNECT <port list>` (***NOTE:*** This feature is available only on version 1.3.2 and greater.)

EXAMPLE NETWORK CONFIGURATION

The following section describes the setup of Aventail Connect in an example network configuration using the Aventail ExtraNet Server.

CONFIGURATION USING AVENTAIL EXTRANET SERVER

The following example network configurations show the Aventail ExtraNet Server configured for a Mobile Extranet environment and a Partner Extranet environment. This example emphasizes simplicity to facilitate easy adaptation to real world network designs.



Example Corporate Network Design using Mobile VPN

The design used in the example above consists of two individual Ethernet segments, one public and one private. The public segment is used to host anonymous services available to the general public. The public access is provided through a router that is connected to the public Internet. The private segment is used to house all of the corporation's private network resources and data to be used only by internal company employees. The Aventail ExtraNet Server depicted in this example is used to provide secure and monitored access to the private LAN for mobile employees and partners. For security reasons the Aventail ExtraNet Server is configured such that operating system routing is disabled. Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the Aventail ExtraNet Server.

The mobile user workstations connected to the public Internet are the client workstations, onto which, Aventail Connect will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail ExtraNet Server unless they are running Aventail Connect. Depending on the security policy and the Aventail ExtraNet Server configuration, Aventail Connect will automatically proxy their allowed application traffic into the private network. In this situation, Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then, based on the security policy, the Aventail ExtraNet Server will proxy mobile user traffic into the private network but only to those resources allowed. The client workstations we focus on in this section are Microsoft Windows based PCs.

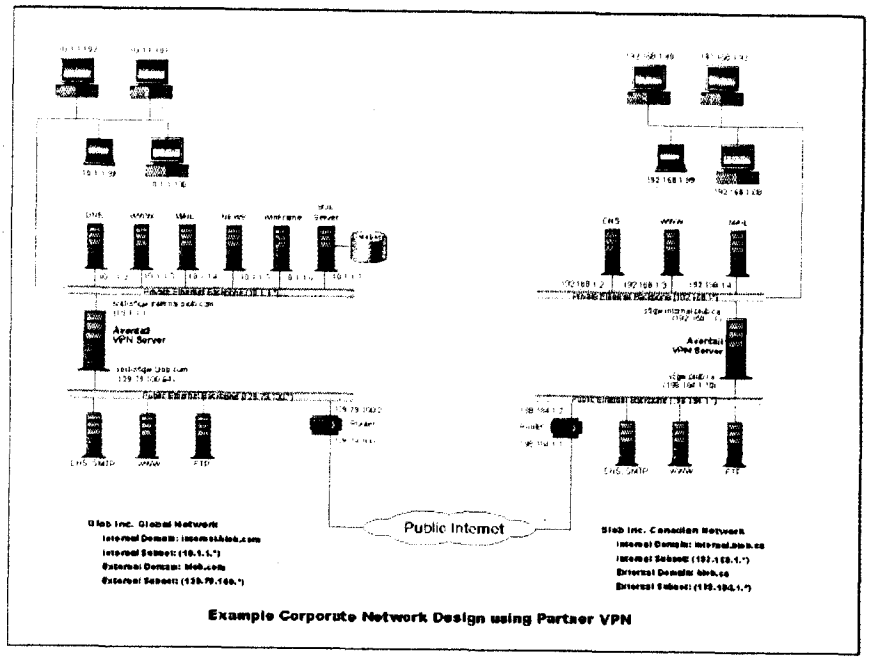
The Aventail ExtraNet Server in our example, has two network adapters configured to use the internal IP address of 10.1.1.1 and an external address of 129.79.100.64.

CAUTION: *Since the internal network address space is part of the IANA reserved address space (per BCP RFC 1918) routing MUST be disabled on this host and routing advertisements for this internal network MUST NOT be propagated to the outside world.*

User authentication and encryption on the Aventail ExtraNet Server require all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s). For this example, the Aventail ExtraNet Server encrypts all sessions with SSL.

SEE ALSO: *For additional information on how to configure the Aventail ExtraNet Server product, consult the* Aventail ExtraNet Server Administrator's Guide.

Installing and using Aventail Connect for remote access purposes differs a bit from its installation and use within a corporate network. First, configuration files need to be kept locally on the workstation or laptop. This is due to the inability to share a file server that allows direct access outside the perimeter of the private network. Second, not all traffic passes through to the Aventail ExtraNet Server. Only traffic destined for the internal network is authenticated and encrypted; all other traffic passes through Aventail Connect unchanged. For instance, browsing the Internet from the mobile user workstation occurs as if Aventail Connect is not even running in the background. Large sites with many mobile users will want to set up an internal file server for a network installation for all mobile users to easily install and configure Aventail Connect. For more information, refer to "Network Installation."



Example Corporate Network Design using Partner VPN

Utilities Reference Guide

This section explains:

- Commands on the System menu, including Close, Hide Icon (in Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51), Help, About, Credentials, and Configuration File
- How to use the Aventail Connect utilities, including the Config Tool, the Logging Tool, and S5 Ping, all displayed through the Utility Programs menu.
- How to use Secure Extranet Explorer (SEE)/Extranet Neighborhood.

System Menu Commands

Even though Aventail Connect requires little to no interaction with the user, there are commands on the Aventail Connect System menu. To display the System menu, right-click the **Aventail Connect** icon in the system tray on the taskbar (Windows 95, Windows 98, and Windows NT 4.0) or click the minimized **Aventail Connect** icon (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51).

Aventail Connect System Menu Commands

Menu Command	Function
Close	Closes Aventail Connect.
Hide Icon	Hides the **Aventail Connect** icon from view. Not available in Windows 95, Windows 98, and Windows NT 4.0.
Help	Accesses Help.
About	Displays Aventail Connect **About** box.
Credentials	Displays authentication credentials.
Configuration File	Selects new configuration file via **Aventail Connect Configuration File** dialog box.

Each of the commands is discussed below.

Close

This command closes Aventail Connect. Exiting Aventail Connect may limit access to certain remote hosts or prevent you from using certain WinSock applications.

HIDE ICON

This command hides the **Aventail Connect** icon from view (Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51 only). Aventail Connect will run in the background. *The **Hide Icon** command is not available in Windows 95, Windows 98, and Windows NT 4.0.*

HELP

This command accesses Aventail Connect Help.

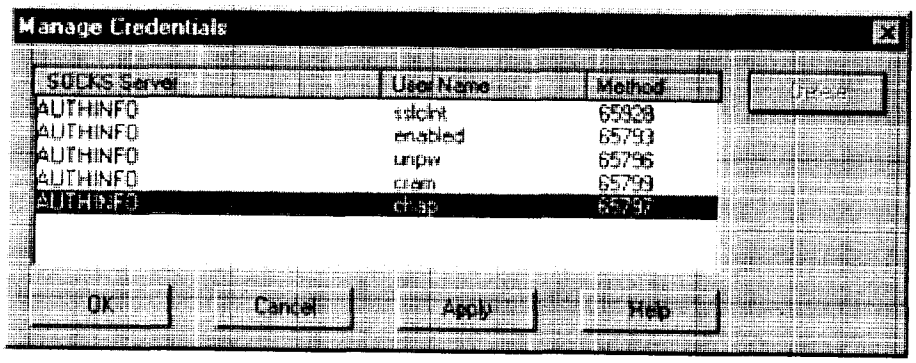
ABOUT

This command displays the Aventail Connect **About** box, which includes Aventail Connect software copyright notification, version information, and so on. Clicking **More** displays a list of files used by the current version of Aventail Connect.

CREDENTIALS

This command displays the **Manage Credentials** dialog box. Credentials include the information (such as username/password) that you enter when establishing a connection to an extranet (SOCKS) server requiring user authentication. (Aventail Connect prompts you with an authentication dialog box.) As long as your credentials are in memory, you can establish connections to associated extranet servers without needing to reenter your authentication information.

You cannot edit credential data fields; you can, however, delete individual credential entries. Aventail Connect will prompt you to enter updated authentication information when you reestablish a connection to the associated extranet server.



NOTE: *You cannot edit the "AUTHINFO" entries in the* ***Manage Credentials*** *dialog box. This information is for diagnostic purposes only.*

Field	Definition
SOCKS Server	Extranet (SOCKS) server name.
User Name	User name for the extranet server.
Method	Authentication method.

To delete a credential entry

Delete authentication credentials when they are no longer correct. After the credentials are deleted, you will be prompted to reenter them the next time you connect to the associated extranet server.

- Select the credential entry you want to delete and click **Delete**.

  This deletes the credential information.

To exit the Manage Credentials dialog box

- Click **OK** to accept changes to the credentials and close the dialog box.

-OR-

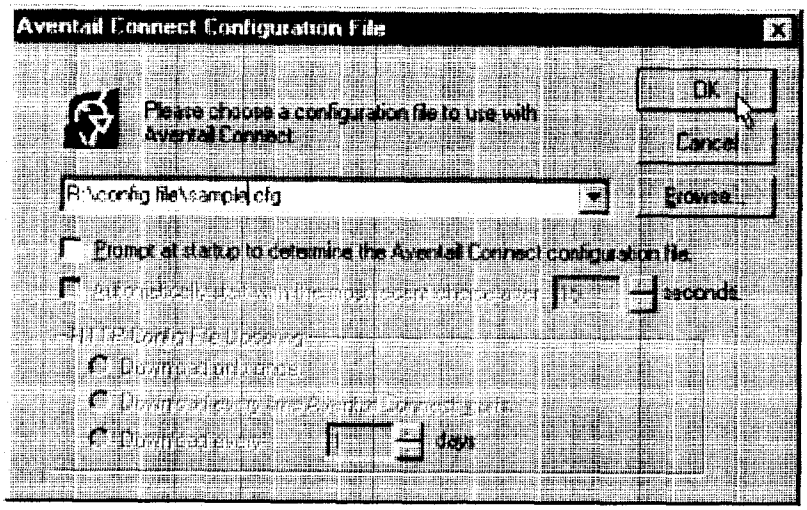
- Click **Cancel** to close the dialog box without accepting any changes you might have entered.

NOTE: *Clicking* ***Apply*** *saves changes but keeps the dialog box open so you can keep working.*

CONFIGURATION FILE

This command lets you load a different configuration file via the **Aventail Connect Configuration File** dialog box. Aventail Connect 3.1 allows you to use a new or modified configuration file immediately, without needing to restart Aventail Connect and any Aventail-processed applications.

For more information about the configuration file, refer to "Configuring Aventail Connect."



To load a configuration file

- Select the configuration file you want to load (use the **Browse** button), and then click **OK**.
- If you want Aventail Connect to start automatically with your most recent choice of configuration file, select the **Automatically start...** check box, and then select the start delay (in seconds).

  The new configuration file transparently loads into Aventail Connect. You can close and restart Aventail Connect for your change to take effect, or wait the specified length of time if you selected the **Automatically start...** checkbox.

UTILITIES

To display the Utility Programs menu, right-click the **Aventail Connect** icon in the system tray on the taskbar (Windows 95, Windows 98, or Windows NT 4.0) or click the minimized **Aventail Connect** icon (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51).

Aventail Connect Utility Program Menu Commands.

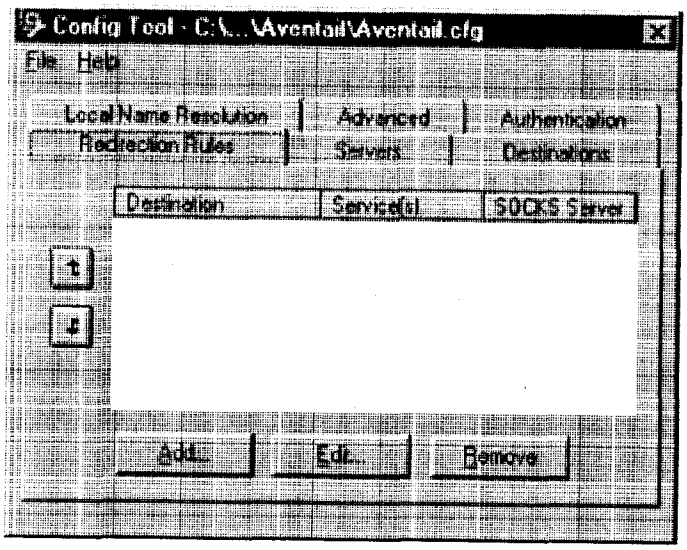
Menu Command	Function
Config Tool	Runs the Config Tool. (Optional)
Logging Tool	Runs the Logging Tool. (Optional)
S5 Ping	Runs the ping and traceroute utilities. (Optional)

Each of the commands is discussed below.

*NOTE: The **Config Tool**, **Logging Tool**, and **S5 Ping** commands are optional components and will only appear when the network administrator has included them in a custom setup package. They are discussed in the sections "Config Tool," "Logging Tool," and "S5 Ping."*

CONFIG TOOL

The Aventail Connect Config Tool creates configuration files that determine how network requests will be routed and which authentication protocols will be enabled. *(This option may not be available to all users if the network administrator has chosen not to install it.)*



Network administrators generally create configuration files during Aventail Connect installation. However, you can add, remove, or modify configuration files at any time. If necessary, you can create several configuration files for different users or user groups. Some configuration files may reside on a networked drive, accessible by multiple users. Other configuration files may be tailored to a specific user on an individual workstation. "Configuring Aventail Connect" discusses the Config Tool in detail.

LOGGING TOOL

The Logging Tool is an optional diagnostic utility for tracing Aventail Connect and WinSock activity. When running a trace, the Logging Tool displays errors, warnings, and information as Aventail Connect generates them. You can save the message list to a log file that Aventail Technical Support can use in troubleshooting technical problems, including Aventail Connect network, extranet (SOCKS) server, and WinSock application interoperability problems. Aventail Technical Support engineers may request that you perform a verbose trace, log it to a file,

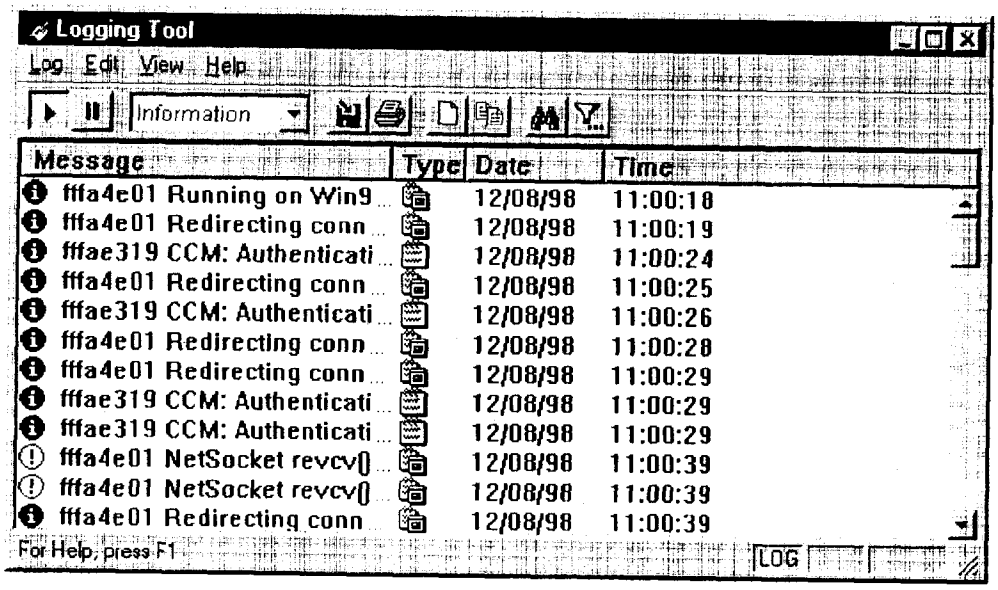
and e-mail it to them as an attachment. Log files are also useful when running Aventail Connect for the first time, to ensure that network traffic is being routed properly.

To trace Aventail Connect activity

1. Windows 95, Windows 98, or Windows NT 4.0: Either right-click the **Aventail Connect** icon (in the system tray on the taskbar) and click **Logging Tool**, or select **Start | Programs | Aventail Connect | Logging Tool**.

-OR-

Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51: From the Aventail Connect program group, double-click the **Logging Tool** program icon.



2. In the **Log** menu, click **Level** and select one of the five levels of information you want to trace.

-OR-

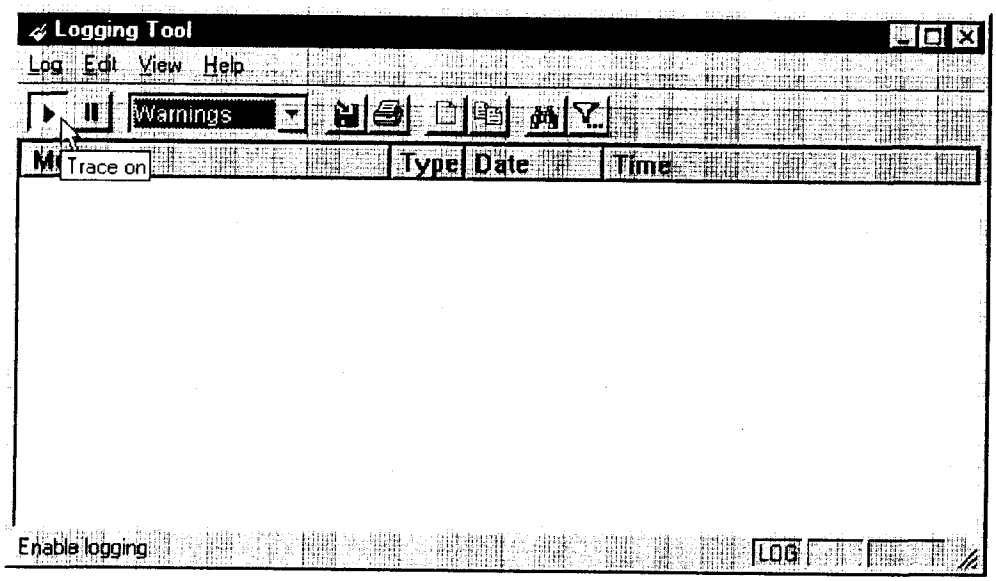
Select one of the five levels from the drop-down list on the toolbar.

Select	To Log
Fatal Errors	Fatal errors only
Errors	Errors and fatal errors only
Warnings	Errors and warnings only
Information	Errors, warning, and information
Verbose	All of the above, and more descriptive information on progress of connections

3. On the **Log** menu, click **Trace.**

-OR-

Click the **Trace On** button on the toolbar (shown below).

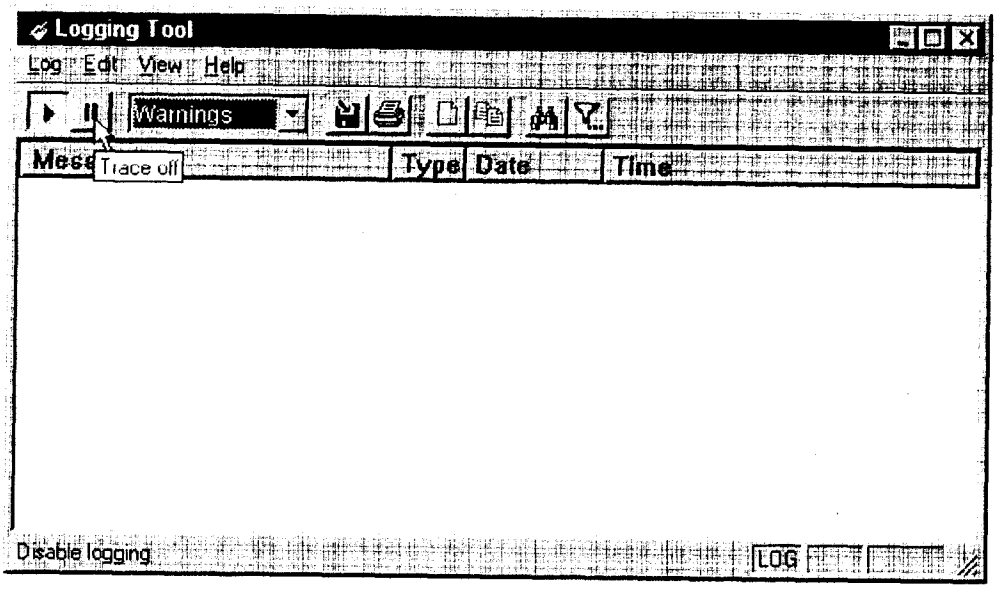


The log window will now record and display trace information as it is generated by Aventail Connect. You can tell when the trace function is active because messages are scrolling down the screen and the **Trace On** button is depressed.

4. When you are ready to stop the Trace function, click **Trace** on the **Log** menu.

-OR-

Click the **Trace Off** button on the toolbar (shown below).



The Trace function stops. You can now scroll through the results, print them, and/or save them to a file.

To save a log file

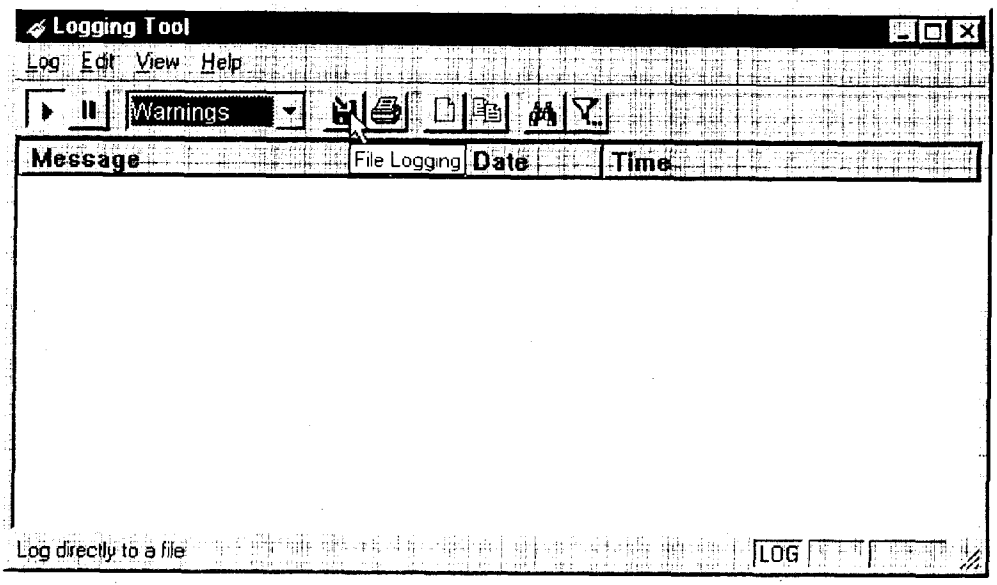
The Logging Tool allows you to append each new message to the end of a .LOG file during the trace, or save the contents of the log window at any time. If you save during a trace, Aventail Connect will append messages to the log file until you stop the log function. You must save data in the log window to retain it.

You cannot open a preexisting log file from within the log window. To open a pre-existing log file, you must open it in a text editor such as Notepad.

1. To save a log file as the data is being generated, click **Log to File** on the **Log** menu. Enter the filename in the **Select Log File** dialog box.

-OR-

Click the **File Logging** button on the toolbar (shown below).



2. Enter the filename in the **Select Log File** dialog box.
   - To save the contents of the log window at any time, click **Save As** on the **Log** menu and then enter the filename.

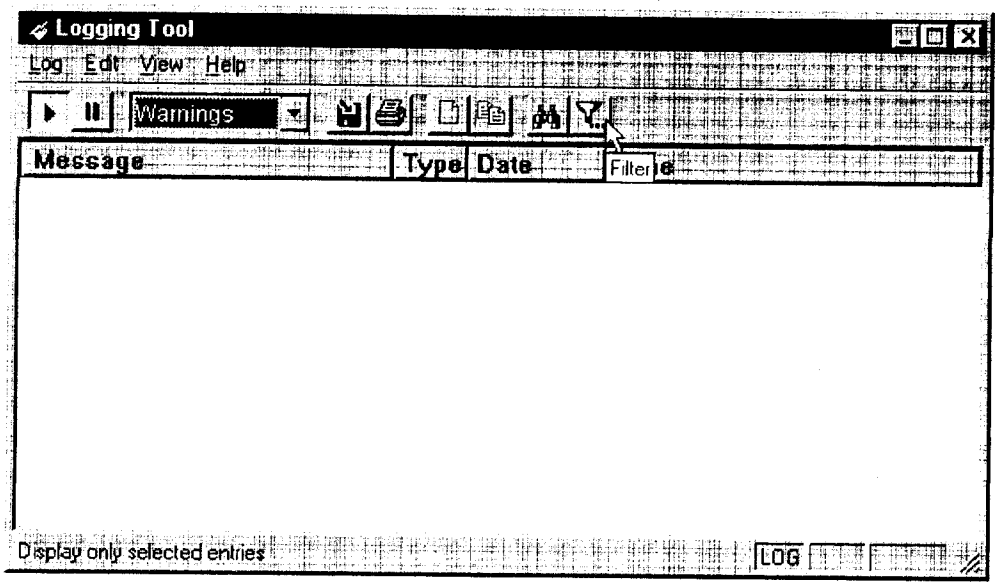
To filter messages in the log window

You can filter the contents of a log window by selecting the types of messages you want to view. By selecting a specific type of message, you can easily scan the information on-screen. If you save data to a log file, a view filter will not affect the file contents; it merely adjusts the screen display of those contents.

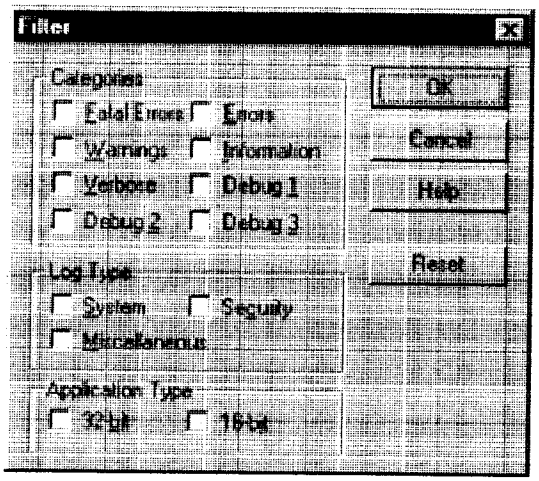
1. On the **View** menu, click **Filter Messages** to display the **Filter** dialog box

-OR-

Click the **Filter** button on the toolbar (shown below) to display the **Filter** dialog box.



NOTE: *The **Filter** function is an on/off toggle. If the filter is enabled, select **Filter Messages** to turn it off, then select it again to display the **Filter** dialog box.*



Field	Definition	
Categories	Select any of the five filters to display errors, fatal errors, warnings, information and/or verbose information in the log window.	
Log Type	Select the type of log to be filtered. (Currently, the only valid log type used in Aventail Connect is Miscellaneous.)	
Application Type*	32-bit	Show messages from 32-bit applications.
	16-bit	Show messages from 16-bit applications.
	*These options are disabled if you are running 16-bit Windows.	

2. Under "Categories," select one or more of the five filter check boxes. The log window will adjust the display based on your selection(s).
3. Under "Log Type," select the log type to filter.
4. Under "Application Type," select one or both of the check boxes.

To change the view parameters

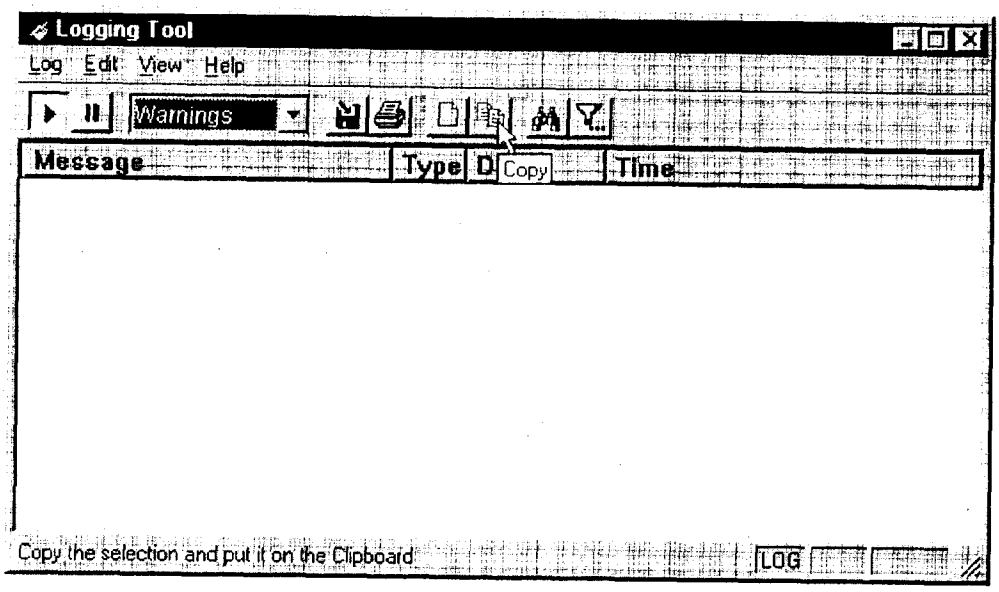
The display font and window options can be customized as follows:

- On the **View** menu, click **Font**. Enter your font preferences into the standard **Windows Font** dialog box.
- To display or hide the toolbar and status bar, click **Toolbar** and/or **Status Bar** on the **View** menu.

To copy the log window

You can copy the log window contents to the Windows Clipboard.

- To copy all of the log window contents to the Windows Clipboard, click **Select All** on the **Edit** menu. Then click **Copy** on the **Edit** menu, or click the **Copy** button on the toolbar.
- To copy selected messages to the Windows Clipboard, drag the mouse over the messages to highlight them. Then select **Copy** on the **Edit** menu or click the **Copy** button on the toolbar.



To print the log window

You can print the contents of the log window can be printed only in its entirety.

- On the **Log** menu, click **Print**.

-OR-

Click the **Print** button on the toolbar.

The entire contents of the window will print, regardless of whether you have specific messages selected. If you have filtered the display, only the filtered messages will print.

To find a specific message

The **Find** command will only work with data displayed in the window. If the display has been filtered, only the filtered messages will be searched. The **Find** dialog box remains active until you close it.

- On the **Edit** menu, click **Find**.

-OR-

Click the **Find** button on the toolbar.

Then enter your search parameters in the **Find** dialog box.

To clear the log window

Clear the log window contents when you are ready to execute a new trace.

- On the **Edit** menu, click **Clear All**.

-OR-

Click the **Clear All** button on the toolbar.

To close the log window

When you are ready to close the log window, make sure you have saved the contents of the trace for later reference. All settings are saved when you exit.

- On the **File** menu, click **Exit**.

S5 PING

Two of the most useful diagnostic tools in an administrator's arsenal are the ping and traceroute utilities.

- The ping utility checks for network connectivity between two hosts and returns information about the quality of the connection.
- The traceroute utility checks for network connectivity by displaying information about routers between two hosts. It displays information for each hop.

Ping and traceroute both use Internet Control Message Protocol (ICMP). SOCKS v5 is designed to handle TCP and UDP protocols; however, SOCKS v5 does not support ICMP. Because ping and traceroute are based on ICMP, there is no way to directly proxy a ping or traceroute request. To circumvent this problem, Aventail Connect provides a utility called S5 Ping.

S5 Ping determines whether a host outside of an extranet server is active. After a response from the host returns, the extranet server relays the data back to the client and displays it in the **S5 Ping** dialog box.

To launch S5 Ping

You can use S5 Ping whether or not Aventail Connect is running. However, if the server that you are connecting through requires authentication, you must load Aventail Connect before reconnecting.

- Windows 95, Windows 98, or Windows NT 4.0: Select **Start | Programs | Aventail Connect | S5 Ping**.

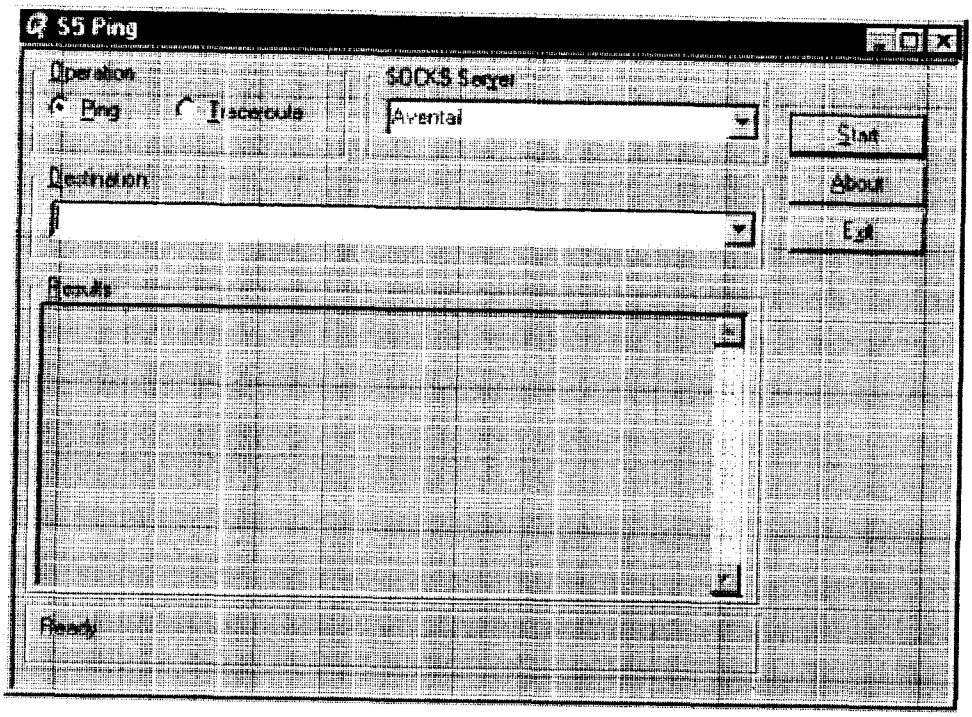
-OR-

Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51: From the Aventail Connect program group, double-click the **S5 Ping** program icon.

-OR-

If Aventail Connect is already running, right-click the **Aventail Connect** icon on the taskbar and click **S5 Ping** (Windows 95, Windows 98, or Windows NT 4.0), click the minimized **Aventail Connect** icon in the System menu (Windows 3.1, Windows for Workgroups 3.11, or Windows NT 3.51).

The **S5 Ping** dialog box appears.



NOTE: *S5 Ping will function without a properly configured Aventail Connect; however, the user will be required to type the information about the target extranet server and target host into the* ***SOCKS Server*** *and* ***Destination*** *boxes.*

Field	Definition
Operation	Select ping or traceroute.
SOCKS Server	The Extranet (SOCKS) server that will execute the operation. If Aventail Connect is already configured, this list will be preloaded with extranet servers from the configuration file.
Destination	The extranet server you want to ping (or traceroute). If Aventail Connect is already configured, this list will be preloaded with single host destinations defined in the configuration file. (See "Configuring Aventail Connect.")
Results	The results of successful connection. The format of the results will vary based upon the extranet server platform.

S5 Ping can be used whether or not Aventail Connect is running. However, if the server that you are connecting through requires authentication, you must load

Aventail Connect before connecting. The network administrator may or may not make S5 Ping available to users during installation. In some cases, the **S5 Ping** command will not appear on the Aventail Connect System menu or in the program group.

Once the **S5 Ping** dialog box opens, you can execute a ping or traceroute network operation.

To run ping or traceroute using S5 Ping

S5 Ping has two modes of operation: ping and traceroute.

1. Under "Operation," select one of the two options, **Ping** or **Traceroute**.
2. Under "SOCKS Server," select an Aventail ExtraNet Server to carry out the operation. If no servers are listed (because S5 Ping did not locate an Aventail Connect configuration file), type the extranet server's hostname or IP address.
3. Under "Destination," select a single host destination to ping or traceroute. If no hosts are listed (because S5 Ping did not locate an Aventail Connect configuration file), type the hostname or IP address of the host you want to ping or traceroute.
4. Click **Start** to execute the operation. **Start** then changes to **Stop**. Results from any previous operation are cleared from the window.
5. If the extranet server requires authentication, you may be prompted with a server certificate or required to enter a username and password. (For more information about server certificates and username/password authentication, see "Manage Authentication Modules" in the *Administrator's Guide*.)
6. Once the connection to the host has been made, the information returned from the server will be displayed in the **Results** window.

To stop ping or traceroute

- Click **Stop**.

  This stops the operation and changes **Stop** to **Start**. The results of the operation remain displayed in the **S5 Ping** dialog box.

To exit S5 Ping

- Click **Exit**.

  This clears the results and closes the **S5 Ping** dialog box.

SECURE EXTRANET EXPLORER

Secure Extranet Explorer (SEE) allows you to view your Extranet Neighborhood, which is accessed through the **Extranet Neighborhood** icon on your desktop. The Extranet Neighborhood user interface resembles that of Network Neighborhood. However, while Network Neighborhood displays all computers on your local network, Extranet Neighborhood allows you to browse, copy, move, and delete files from remote computers via the Aventail Connect extranet connection. With Extranet Neighborhood, all interaction with the remote server can be secured. Network administrators determine which local and remote computers are available to users.

NOTE: *Some installations of Aventail Connect may not include SEE. Network administrators can decide whether or not to include SEE in a custom setup package.*

Extranet Neighborhood, a Windows Explorer shell extension, is a collection of Windows file servers and Windows NT domains. Network Neighborhood displays only those remote computers that the network administrator has specified. SEE requires a hosts file (SEEHosts) that determines which Windows file servers and NT domains are available. You can include a SEEHosts file with the Aventail Customizer tool. If users install a custom package that does not include a SEEHosts file, then the first time they open Extranet Neighborhood, SEE will create a SEEHosts file. For more information, see the "Customizer" section in the *Administrator's Guide*.

Extranet Neighborhood offers Aventail Connect users a secure alternative to traditional file-browsing methods. Users can securely access computers from the desktop through Extranet Neighborhood (see icon below), or through Windows Explorer.

Generally, you will use Extranet Neighborhood to connect to a remote network through Aventail Connect. For example, you will use Extranet Neighborhood when:

- you are inside the office, on the corporate network, and you connect through an Aventail ExtraNet Server to your company's remote site, or to another company's network.
- you are outside the office, and you connect your laptop through an Aventail ExtraNet Server to your internal company network, or to another company's network.

NOTE: To use Extranet Neighborhood with remote hosts, Aventail Connect must be running and configured correctly.

How Extranet Neighborhood Works

Typically, with Windows networking, the Microsoft Windows Explorer and Network Neighborhood browse files using NetBIOS (NBT), over TCP. Network Neighborhood does not use the standard WinSock programming interface. This prevents Aventail Connect from redirecting TCP connections. Since Aventail Connect redirects only WinSock calls, it cannot redirect NBT calls.

To deliver a secured version of standard Windows browsing, Aventail Connect redirects NBT calls to WinSock. This allows Aventail Connect to redirect this traffic based on a set of redirection rules, as defined in the Aventail Connect configuration file.

Extranet Neighborhood can use either hosts files or Windows Internet Naming Service (WINS) servers to map a computer's Internet (host) name to its Windows machine name. Without a hosts file or a WINS server, Extranet Neighborhood cannot associate a computer's Internet name with its Windows machine name.

Extranet Neighborhood includes a browsing mode, which allows you to view a dynamic list of available Windows hosts. Hosts files provide a static list of hosts.

There are two basic methods for configuring Extranet Neighborhood.

- **Listing WINS Servers:** List only WINS servers for the domain(s) in the hosts file. You do not need to list individual hosts within the domain.
- **Listing Individual Hosts:** List every individual host in the hosts file that will be accessible to users.

Listing WINS Servers

To use Extranet Neighborhood in the browsing mode, you must configure Extranet Neighborhood to use WINS, and you must identify the IP address (hostname) of the WINS server(s) and, possibly, the primary domain controller (PDC) for the domain. If you do not specify a WINS server, you will not be able to use Extranet Neighborhood in the browsing mode.

The PDC for the domain is required only if the destination network is not accessible by UDP. (For example, when using MultiProxy, the destination network is not UDP-accessible.) When Extranet Neighborhood is in browsing mode, it must be able to resolve the name of the host. If the destination network is UDP-accessible, then the WINS server is used to map a computer's Internet (host) name to its Windows machine name. If the destination network is not UDP-accessible, then Extranet Neighborhood uses the PDC and DNS to determine the host's address.

LISTING INDIVIDUAL HOSTS

To use Extranet Neighborhood in the static host list mode, you must define, in the hosts file, each individual host in the domain. This allows you to restrict access to designated hosts only. In the hosts file, you must specify the host's IP address or DNS name along with the Windows machine name. WINS and PDC are not used in this method.

INSTALLING EXTRANET NEIGHBORHOOD

When installed, Extranet Neighborhood appears on your desktop as an icon, and in Windows Explorer. You can open, move, copy, and delete files in Extranet Neighborhood just as you would in Network Neighborhood.

If you need to install Extranet Neighborhood, install it from the Aventail Connect CD. Or, if you downloaded your copy of Aventail Connect, run the downloaded executable package. When the **Installation Components and Sub-components** dialog box appears, select **Extranet Neighborhood** (located under **Components**). Continue with the installation process.

The default installation directory is `\Program Files\Aventail\Connect`.

NOTE: *Secure Extranet Explorer/Extranet Neighborhood is available only on Windows 95, Windows 98, and Windows NT 4.0 operating systems.*

CONFIGURING EXTRANET NEIGHBORHOOD

You can include a SEEHosts file with the Aventail Customizer tool. Only by installing a custom package will users have a local or remote hosts file automatically configured. If users install a custom package that does not include a SEEHosts file, the SEE Configuration wizard will run when users open Extranet Neighborhood for the first time. The SEE Configuration wizard walks you through the process of defining local or remote hosts files. Aventail recommends that you use the Customizer tool to distribute Extranet Neighborhood, bundled with a hosts file, in a custom setup package.

Extranet Neighborhood can automatically construct a hosts file from your local network or a remote network. Using the Search feature, Extranet Neighborhood can automatically "browse" available computers and build the local hosts file. The Search feature is available through the **Extranet Neighborhood Properties | Local** tab. Alternatively, you can enter the names of the available computers manually. The Search feature browses only those computers that are within your internal network. To search remote networks, you must manually enter the fully qualified hostname of each remote WINS server that is outside your Aventail ExtraNet Server. When using the Search feature, the same UDP restrictions described in "Listing WINS Servers" apply.

NOTE: *To use the Search feature, Aventail Connect must be running and configured correctly.*

Do not use the Search feature if you are using the WINS-browsing mode. The Search feature builds the local hosts file for all of the computers, which is not necessary with WINS. Use Search when creating a local hosts file using the "listing individual hosts" method.

NOTE: *When you click* ***Search****, you may see more than one domain in the resulting local hosts file. This is because Search includes trusted domains.*

To create a hosts file

Use this procedure if you have not yet created a hosts file.

1. Decide which method, listing WINS servers or listing all individual hosts, to use.
2. If no hosts file exists, launch Extranet Neighborhood (Extranet Neighborhood will prompt you automatically if you are running Extranet Neighborhood for the first time),

-OR-

Right-click the **Extranet Neighborhood** icon on your desktop and then click **Properties**.

3. Follow the on-screen instructions to create the hosts file.
4. To distribute the new hosts file, include the SEEHosts file in your custom setup package, if using the Customizer tool.

After creating the hosts file, users can browse only those domains and machines that the network administrator has included in that list of hosts. This list may be a local hosts file called "SEEHosts" and/or a remote host list, which is identified by `[share]\[path]\[filename]`.

NOTE: *To use the browsing mode, you must specify the domain's WINS server(s) in the local hosts file.*

CAUTION: *SEE cannot recognize share names that contain special characters (e.g., é) or multiple spaces (e.g.,* Aventail Custom Computer*). SEE also will not recognize hidden one-letter share names (e.g.,* C$ *or* D$*).*

SEE CONFIGURATION METHODS

There are numerous methods for configuring SEE. The three most common methods are described below.

Local Hosts File Method

With this method, the hosts file contains a list of all domains and servers in the local hosts file. Every host is listed.

There are two ways to configure SEE using this method.

- In the **Extranet Neighborhood Properties | Local** tab, manually add each domain and host to the local hosts file

-OR-

- On the **Local** tab, click **Search**, click **Search Local Network**, and then search any remote networks, if necessary. SEE automatically builds a list of all hosts. You may delete hosts from the local hosts file if you do not want users to view them.

NOTE: *To view your domains, double-click the* ***Extranet Neighborhood*** *icon on your desktop. If you make changes to the hosts file, you can reload the* ***Extranet Neighborhood domains*** *window by pressing the* F5 *key.*

Remote Hosts File Method

With this method, the local hosts file contains the path of the remote hosts file, and the remote hosts file contents are determined by which configuration method you use.

To use this method, first create the remote hosts file, and then create a local hosts file that points to the remote hosts file.

To configure SEE using the remote hosts file method

1. Create a local hosts file, using one of the methods listed above, and copy it to a central location. (This creates a remote hosts file; this file is not distributed with Aventail Connect.)
2. On the **Remote** tab, click **Add**, and then add a pointer to the remote hosts file that you created in Step 1. (This file is distributed with Aventail Connect.)

NOTE: *You can point to multiple remote hosts files on a single list.*

WINS Browsing Method

With this method, the hosts file contains a list of all domains, and the WINS servers for each domain. You do not need to list all of the computers.

To use this method, add each domain in the **Local** tab, specifying the primary WINS server and, if applicable, the secondary WINS server, and then select the **Make domain browsable** check box in the **Windows Domain** dialog box.

Choosing a Method

Each of the three methods has advantages and disadvantages. The table below lists pros and cons for each of the three methods.

Method	Advantages	Disadvantages
Local hosts file with individual computers	The administrator controls exactly which hosts the users can see. On slower connections, this method is fastest since you do not need to send a list of servers to the client.	The administrator must update the local hosts file if file servers are added to or removed from the domains.
Remote hosts file	• The administrator can edit the centrally stored hosts file whenever necessary. • If the hosts file is stored behind a firewall, SEE can go through an extranet server (using encryption and authentication) to reach it.	• Users are immediately prompted to enter authentication credentials upon opening SEE (because SEE must load the remote hosts file). • If a user loses network connectivity to the hosts file, SEE will not display the list of hosts/computers.
Local hosts file with WINS browsing	The administrator does not need to update the hosts file if new computers are added or removed.	• The administrator must update the local hosts file if domains are added or removed. • The administrator cannot control which computers appear in SEE; all computers in the NT domain are displayed. • On slower connections, this method is slower than other methods because a list of computers must be sent to the client.

You are not limited to using only one method for configuring SEE. You can use a combination of the various methods. For example:

- Use WINS browsing for some domains, and explicitly list hosts for other domains

-OR-

- Use multiple remote hosts files

-OR-

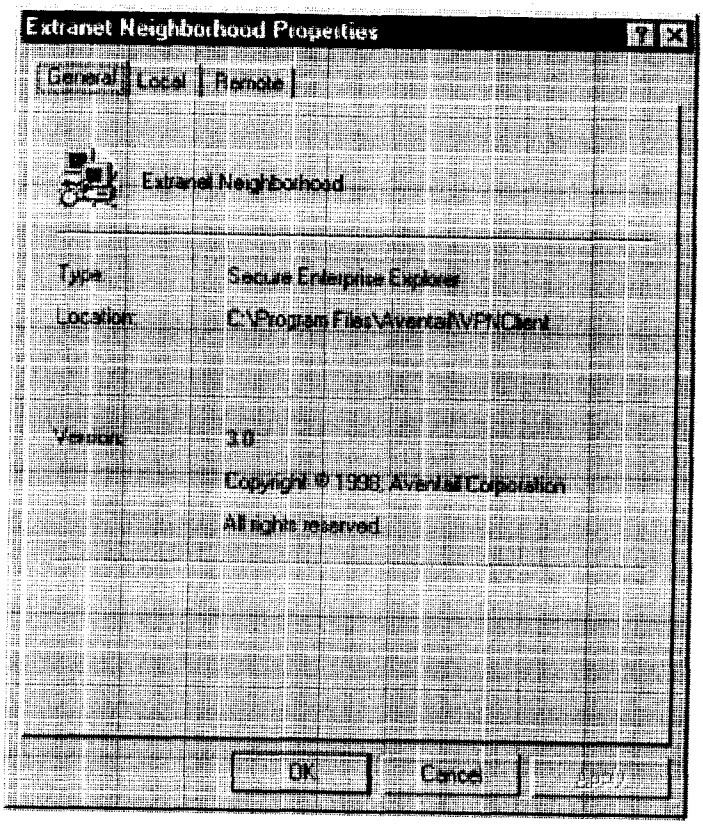
- Specify some computers in a local hosts file and others in a remote hosts file.

SEE PROPERTIES

To access information about the current configuration of SEE, or to make changes to that configuration, right-click the **Extranet Neighborhood** icon and click **Properties**, or click **View | Options** in any open **SEE** window. The **Extranet Neighborhood Properties** window will appear with the **General** tab selected.

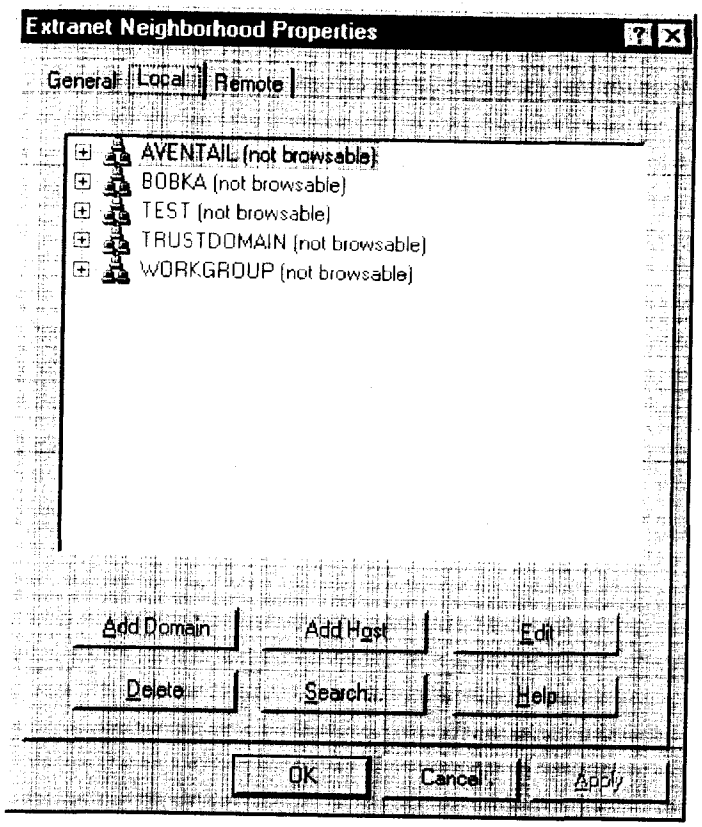
THE GENERAL TAB

The **General** tab displays information about the current configuration of SEE.



THE LOCAL TAB

The **Local** tab displays the computers that are listed in the local hosts file.



If you have specified a host in the local hosts file, you can add, edit, or remove computers or domains that appear in the **Local** tab. If you have specified hosts in the remote hosts file, they will not appear in this tab. To edit hosts in the remote hosts file, you must copy the file to your Aventail Connect directory, edit it, and then replace it in the remote hosts directory.

If you are using the WINS browsing mode, the individual computer names will not appear. Any hosts specified in remote hosts files, including WINS servers, will not appear in this tab.

The **Add Host** and **Add Domain** buttons allow you to add additional computers or domains in the **Add Host to Aventail** dialog box and the **Windows Domain** dialog box.

If no computers or domains appear in your **Local** tab, check the **Remote** tab. It is possible that your network administrator has configured Extranet Neighborhood with only a remote hosts file.

The Search feature can automatically browse available computers in local or remote domains and populate your local hosts file. Alternatively, you can enter the names of the hosts files manually.

NOTE: *To view your domains, double-click the* ***Extranet Neighborhood*** *icon on your desktop. To reload the hosts files in the* ***Extranet Neighborhood domains*** *window, press the* F5 *key.*

NOTE: *In the* **Local** *tab, "browsable" domains do not show individual computers in them.*

Hosts File Locking

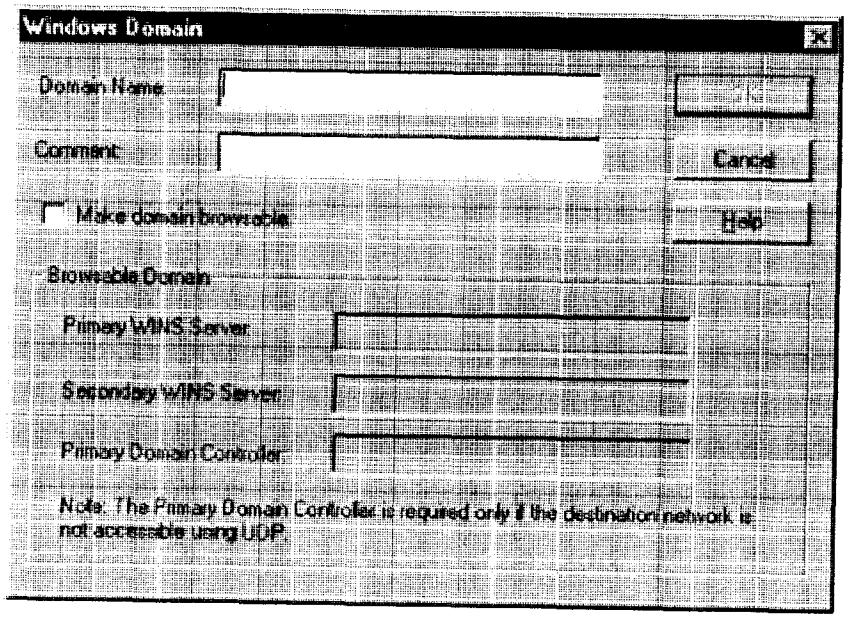
If the controls in this window are disabled (dimmed), then the hosts file has been "locked." The network administrator determines which, if any, hosts files are locked.

You can lock and unlock files from any **Extranet Neighborhood Properties** tab.

- To lock a file, use the **Ctrl+L** command.
- To unlock a file, use the **Ctrl+U** command.

Windows Domain Dialog Box

To open the **Windows Domain** dialog box, click **Add Domain** in the **Extranet Neighborhood Properties | Local** tab.



For each domain, you can either specify the WINS server names or specify each individual host that should appear in the domain. Listing WINS servers will result in a smaller, more manageable hosts file. You must add a domain before you can add hosts to that domain.

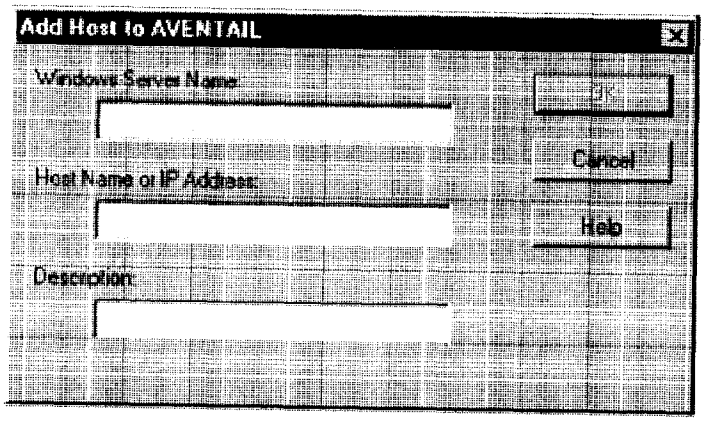
To make the specified domain "browsable," enter WINS server information in the **Primary WINS Server** box and, if desired, the **Secondary WINS Server** box. In both of these boxes, you can enter either the server's IP address or its fully qualified host name. You must also select the **Make domain browsable** check box. If you do not select the **Make domain browsable** check box, Extranet Neighborhood will display only those computers in the local or remote hosts file, even if you have specified a WINS server.

NOTE: *To use the browsing mode for a domain, you must specify the domain's WINS server(s) in the hosts file. You must specify the WINS server(s) only if you want to use the browsing mode.*

To view your domains, double-click the **Extranet Neighborhood** icon on your desktop. To reload the hosts files in this screen, press the F5 key.

Add Host to Aventail Dialog Box

To open the **Add Host to Aventail** dialog box, click **Add Host** on the **Extranet Neighborhood Properties | Local** tab.

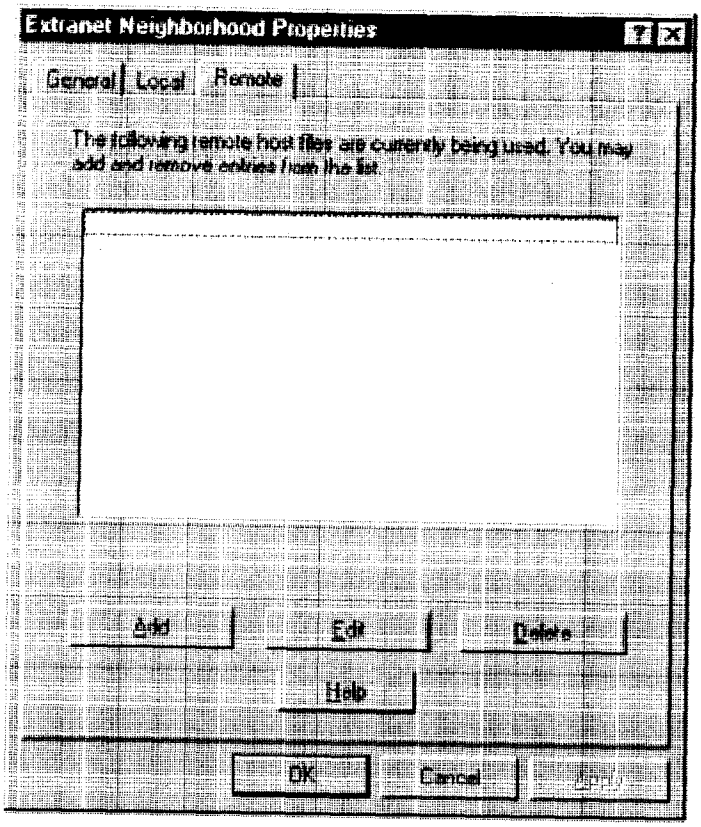


Aventail Connect automatically places hosts within the domain that is selected when you click **Add Host**. Select the correct domain before clicking **Add Host**. You must specify a domain before you can add hosts to that domain.

In the **Host name or IP address** box, be sure to enter the server's Internet address, not its Windows machine name.

THE REMOTE TAB

If the network administrator has configured Extranet Neighborhood to use a remote hosts file, this tab displays the information about the currently configured remote hosts file(s). Server name, host name or address, pathname, and username are all configurable through the **Remote** tab.



Remote hosts files are always used in conjunction with a local hosts file. When you add a remote hosts file to the list, Extranet Neighborhood adds the path to the local hosts file. Extranet Neighborhood always has a single local hosts file; this file can include references to multiple remote hosts files.

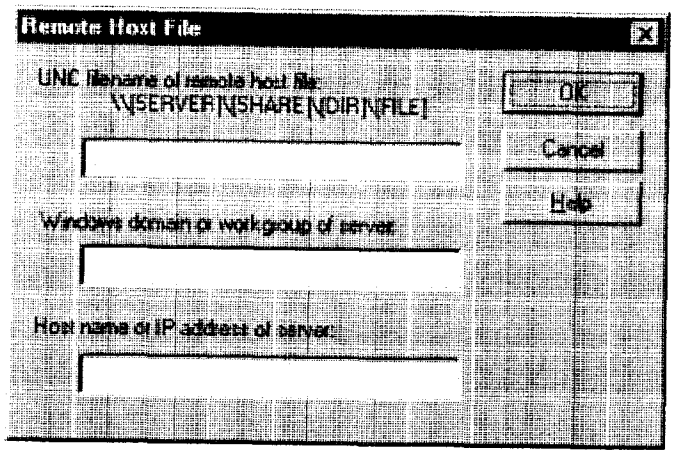
The most common configuration is one remote hosts file (with all domains and hosts in the remote hosts file) and one local hosts file that contains a pointer to the remote hosts file. If you want users to share a common hosts file, and if you want to simplify administration, use a remote hosts file.

To add entries to the list of remote hosts files, click **Add**. The **Remote Hosts File** dialog box appears, and you can type the names of the remote hosts file(s) you want to add.

NOTE: *To access remote hosts files, Aventail Connect must be running and configured correctly.*

Remote Hosts File Dialog Box

To open the **Remote Hosts File** dialog box, click **Add** on the **Remote** tab.



When entering the Universal Naming Convention (UNC) filename of the remote hosts file that you are adding, note that the [SERVER] name is the Windows machine name, not its IP address or hostname.

In the **Host name or IP address of Server** box, be sure to enter the server's Internet address, not its Windows machine name.

NOTE: *Extranet Neighborhood ignores any remote hosts files that it cannot access.*

Troubleshooting

Aventail Connect-related problems tend to fall into four categories: Installation, Network Connectivity, Configuration, and Application and TCP/IP Stack Interoperability.

AVENTAIL CONNECT INSTALLATION PROBLEMS

When the instructions in "Installing" in the *Administrator's Guide* are followed, Aventail Connect installation problems rarely occur. When they do occur, they are often the result of:

- **Toolbars, virus-checking utilities, or other Windows applications running during the installation**

  If any of these are running during a failed installation, close them, uninstall Aventail Connect, reboot, and then re-install Aventail Connect, ensuring that the toolbars, virus-checking utilities, or applications are not automatically restarted when the system reboots.

- **Insufficient RAM or free space on the volume to which Aventail Connect is being installed**

  If you suspect either of these as the cause of a failed installation, increase the available resources and retry the installation.

- **Corrupted Aventail Connect installation media, or corrupted or incomplete FTP of Aventail Connect self-extracting, executable installation file**

  If you suspect corrupted Aventail Connect installation diskettes as the cause of a failed installation, contact Aventail Technical Support (206.215.0078) for assistance in determining whether the files on the diskettes may have been corrupted and whether Aventail or your vendor must supply replacement diskettes.

  If you suspect a corrupted or incomplete FTP transfer of Aventail Connect installation files obtained over the Internet, retry the transfer, taking care to ensure that the FTP client is in binary mode and confirm that the transfer completes normally. Contact Aventail Technical Support to confirm that the byte size of the transferred installation file is correct.

- **Installation to a workstation on which Aventail Connect was running or from which a previous version of Aventail Connect was not completely uninstalled**

  If you suspect either of these circumstances as the cause of a failed installation, contact Aventail Technical Support.

- **Installation script errors**

  Aventail Connect is installed with InstallShield. If InstallShield reports errors during a failed installation, note the text of the error messages and the specific circumstances in which they occurred and contact Aventail Technical Support.

NETWORK CONNECTIVITY PROBLEMS

Before Aventail Connect can successfully redirect WinSock application connections:

1. The workstation on which Aventail Connect is installed must also have a properly installed, WinSock-compatible, TCP/IP stack running on it.

   This installation can be confirmed by successfully pinging the IP address of the workstation, from the workstation itself, using a WinSock ping application. If this test fails, the failure must be corrected before Aventail Connect can be tested and before Aventail Technical Support can provide assistance.

2. Basic TCP/IP network connectivity must exist between the client workstation on which Aventail Connect is installed and the extranet (SOCKS) server(s) to which it is configured to redirect connections.

   This connectivity can be confirmed by successfully pinging the extranet server(s) by IP address, from the client workstation. If this test fails, the failure must be corrected before Aventail Connect can be tested and before Aventail Technical Support can provide assistance.

3. Basic TCP/IP network connectivity must also exist between the extranet server(s) and the network host(s) to which the extranet server(s) are expected to proxy connections.

   This connectivity can be confirmed by successfully pinging the network host(s), by IP address, from the extranet server(s). If this test fails, the failure must be corrected before Aventail Connect can be tested and before Aventail Technical Support can provide assistance.

AVENTAIL CONNECT CONFIGURATION PROBLEMS

This section addresses troubleshooting of simple Aventail Connect configuration problems. Troubleshooting complex Aventail Connect configuration problems is beyond the scope of this section.

It is easiest to troubleshoot Aventail Connect configuration problems by creating and testing simple Aventail Connect configuration files, such as those that may be created with the Aventail Connect configuration wizard. However, all references to host and domain names must be removed from configuration files created with the wizard, before testing, to defer possible name resolution complications until the files can be demonstrated to work with IP addresses alone.

NOTE: *The IP address and SOCKS port number of the extranet (SOCKS) server(s) to which Aventail Connect must connect must be known before troubleshooting Aventail Connect configuration problems. Neither Aventail Connect, nor Aventail Technical Support, can discover the IP address or port number of the extranet server(s).*

When troubleshooting Aventail Connect configuration problems, confirm that the Aventail Connect configuration file that is currently selected in the **Configuration File** dialog box is the one intended for testing.

After selecting a configuration file to test, open the Aventail Connect Config Tool and:

1. Confirm that the extranet server has been correctly identified by IP address.

   Click the **Servers** tab, select the server alias and then click **Edit...**. Compare the IP address in the **Hostname or IP** box with that of the extranet server.

   If the extranet server is a SOCKS v5 server, click **SOCKS v4** in the "SOCKS Version" area of the **Servers** tab. Then click **Detect Version**. The selection will revert to **SOCKS v5**, indicating that Aventail Connect detected a SOCKS v5 server running at the IP address specified in the **Hostname or IP** box.

   If, on the other hand, the extranet server is a SOCKS v4 server, click **SOCKS v5** in the "SOCKS Version" area. Then click **Detect Version**. The selection will revert **SOCKS v4**, indicating that Aventail Connect detected a SOCKS v4 server running at the IP address specified in the **Hostname or IP** box.

   If **Detect Version** fails to detect an extranet server of either version, it is possible that no extranet server is running on the host identified in the **Hostname or IP** box. Contact your extranet server administrator to confirm that the extranet server is running at the address specified.

2. Confirm that all Aventail Connect authentication modules are enabled.

   Click the **Authentication** tab and confirm that the "traffic light" icons for all of the authentication Modules are green, indicating that the modules are enabled. Enabling all the modules configures Aventail Connect to attempt any form of authentication demanded by the extranet server or null (no) authentication. Note the form of authentication demanded by the extranet server and, if necessary, obtain the proper authentication credentials, such as an extranet server username and password, from the extranet server administrator.

3. Confirm that the network hosts to which the extranet server is expected to proxy connections are within a redirected destination.

   Click the **Destinations** tab, select the destination that includes the network host to which the extranet server is expected to proxy connections, and then click **Edit...**. Confirm that the definition of the Destination includes the network host.

   Next, click the **Redirection Rules** tab. Confirm that connections to the Destination are configured to be redirected by the extranet server.

After making any necessary changes to the Aventail Connect configuration, restart Aventail Connect and then restart any WinSock applications before testing the new configuration.

Application and TCP/IP Stack Interoperability Problems

Aventail Connect is intended to "automatically socksify" all "well-behaved" WinSock applications. Occasionally, you may find WinSock applications that Aventail Connect does not socksify, due to interoperability problems with the application.

Aventail Connect is also intended to run on all WinSock-compliant Microsoft Windows TCP/IP stacks. Aventail Connect does not alter or replace WinSock or any other core TCP/IP components (files) provided by the operating system. Occasionally, you may find WinSock stacks on which Aventail Connect does not run as expected, due to interoperability problems with the stack.

If you suspect an application or stack interoperability problem, report it to Aventail Technical Support. Aventail will make every reasonable effort to resolve interoperability problems.

Aventail Connect Trace Logging

Aventail Connect includes a Logging Tool for tracing Aventail Connect and WinSock activity. Aventail Connect traces are often useful in troubleshooting Aventail Connect network, extranet server, and WinSock application interoperability problems. Aventail Technical Support engineers may request that you perform a verbose trace, log it to a file, and e-mail it to them as an attachment.

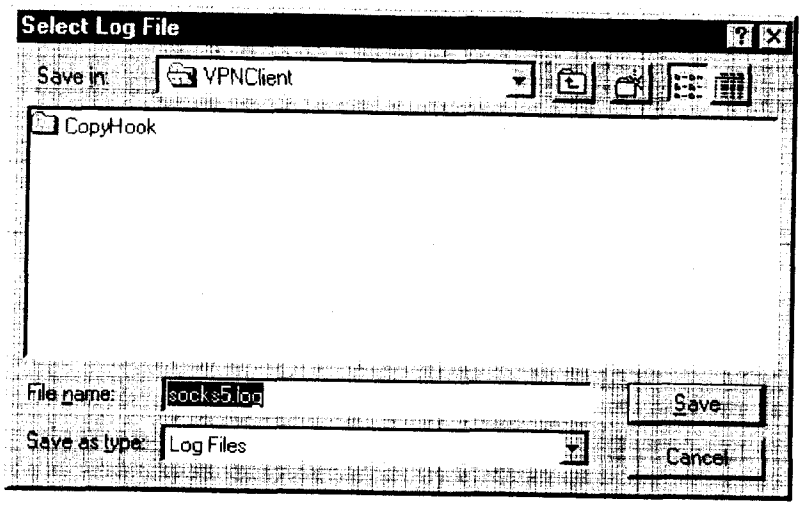
To run an Aventail Connect trace

1. Close any WinSock applications that are running on the workstation.
2. If Aventail Connect is running, close it and then restart it.
3. Start an Aventail Connect trace.

   In Windows 95, Windows 98, and Windows NT 4.0, right-click the minimized **Aventail Connect** icon in the system tray, and click **Logging Tool**. In Windows 3.1, Windows for Workgroups 3.11, and Windows NT 3.51, double-click the **Logging Tool** icon in the Aventail program group. The Aventail Connect **Logging Tool** window will open, as illustrated in Figure 1, below.
4. On the **Log** menu, confirm that the **Trace** command is checked. If it is not, click **Trace** to enable it.

To save an Aventail Connect trace to a file

1. On the **Log** menu, confirm that the **Log To File** command is checked. If it is not, click **Log To File** to enable it.
2. The **Select Log File** dialog box (shown below) appears. Enter a file name and click **Save**.



ERROR MESSAGES

Occasionally, you may see an error message while running Aventail Connect. The following table explains some of the more common Aventail Connect error messages.

Error Message	Meaning
Setup has determined that your computer does not have this support and needs the WinSock 2 patch, available from Microsoft.	SETUP: To install Aventail Connect 3.1, you must first install the Microsoft WinSock 2 upgrade.
The patch is available for download on the Microsoft Web site, at http://www.microsoft.com/Windows95/downloads/contents/wuadmintools/s_wunetworkingtools/W95Sockets2/default.asp.	SETUP: Location of the Microsoft WinSock 2 upgrade.

Error Message	Meaning
You must have administrator privileges to install.	SETUP: On Windows NT machines, you must have administrative privileges to install or uninstall Aventail Connect.
Setup has detected that a previous installation of (....) is present. Would you like to continue and upgrade to (....)? Pressing NO will leave your existing installation intact and will cause Setup to terminate.	SETUP: Retain the previous installation of Aventail Connect by pressing NO. Replace with the newer installation by pressing YES.
The package does not contain the necessary 3.1 files. Please contact your administrator.	SETUP: Setup cannot find the necessary Aventail Connect 3.1 files.
The package does not contain the necessary 2.6 files. Please contact your administrator.	SETUP: Setup cannot find the necessary Aventail Connect 2.6 files.
The file you have selected is not a valid Aventail setup file. Would you like to create it?	CUSTOMIZER: Create a new setup file, or retain a previous setup file.
Customizer must be run from a valid Customize directory. Your changes will not be saved.	CUSTOMIZER: Must run Customizer from a valid Customize directory.
The Connect executable does not have a valid Aventail digital signature.	The specified signature is not valid.
Connect cannot find your license file, aventail.alf.	Aventail Connect cannot find a valid Aventail license file, aventail.alf.
Connect cannot load because your license file does not contain a license.	The license file exists, but it contains no license.
This version of Connect does not support HTTP servers.	Aventail Connect 2.6 does not support HTTP servers.

Reporting Aventail Connect Problems

Report Aventail Connect problems to Aventail Technical Support by completing and submitting an Online Support form on the Support page of the Aventail Web site, **http://www.aventail.com**.

Glossary

ALIAS

User-friendly name for destination network or host computer.

AUTHENTICATION

A method for identifying a user in order to establish access to a system resource or network. Authentication information such as username/password is entered via prompts.

CERTIFICATE

A certificate is essentially an electronic "statement" which verifies that a certain RSA public key is associated with a particular name. Certificates are issued by a Certification Authority (CA).

CLIENT

A program or Internet service that sends commands to and receive information from a corresponding program known as a server. Most Internet services run as client/server programs.

CONFIGURATION FILE

A file of information containing traffic redirection rules used to determine if and how SOCKS redirection should occur.

CREDENTIALS

Credentials include the information (such as username/password) that you enter when establishing a connection to a SOCKS server requiring user authentication.

DOMAIN

Internet name for a network or computer system.

ENCRYPTION

A security procedure that converts data into a format which can be read only by the intended recipient computer.

EXTRANET

A network that is partially accessible to outsiders.

FIREWALL

Software or hardware barriers that control the flow of information to Private networks.

GATEWAY

A communications device/program that passes data between networks.

HACKER

A person who enjoys using computers and has a thorough understanding of how they work, as well as the networks they run on. Often used to mean "cracker," the correct term for someone who accesses computer systems without authorization.

HOST

A server connected to the Internet.

IETF

Internet Engineering Task Force: An open community of network designers, vendors, etc. who resolve protocol and architectural issues for the quickly evolving Internet.

INTERNET PROTOCOL (IP)

The basic data transfer protocol used for the Internet. Information such as the address of the sender and the recipient is inserted into an electronic "packet" which is then transmitted.

INTRANET

A network that is internal to a company or organization.

LAN

Local area network

LAYERED SERVICE PROVIDER (LSP)

A program that is installed just below WinSock 2, allowing two-way communication between the WinSock 2-compatible application and the underlying TCP/IP stack. An LSP can redirect and/or change data before sending the data to the operating system's TCP/IP stack for transport over the network.

LOG WINDOW

The window of the Logging Tool which shows alerts, messages, and warnings generated by Aventail Connect.

PING

A utility that determines if a remote host computer is up. ping sends data packets to the host. If the packets are not returned, the host is down.

PROTOCOL

Rules and procedures used to exchange information between networks and computer systems.

REDIRECTION RULES

Rules defined in the configuration file which specify how network requests are routed to SOCKS servers.

ROUTER

A device that transmits traffic between networks

SERVER

A networked computer that shares resources with other computers. Servers "serve up" information to clients.

SMB

Server Message Block. A message format used by DOS and Windows for sharing files, directories, and other resources.

SOCKS

SOCKS is a security protocol. It acts as a proxy mechanism that manages the flow and security of data traffic to and from your local area network or intranet.

SSL

Security Sockets Layer. An authentication and encryption protocol.

TRACEROUTE

A utility that traces the routing of data over the Internet to a specific computer. Traceroute sends a data packet and then lists the intermediate host computers that it traverses on it's way to the destination machine.

TRANSMISSION CONTROL PROTOCOL (TCP)

A means of sending data over the Internet with guaranteed delivery.

TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL (TCP/IP)

A suite of protocols the Internet uses to provide for services such as e-mail, ftp, and telnet.

USER DATAGRAM PROTOCOL (UDP)

A means of sending data over the Internet without guaranteed delivery. Also known as "connectionless" protocol, it is used for data such as RealAudio®.

UNIVERSAL NAMING CONVENTION (UNC)

A way of accessing a file or directory on another computer. For example: // host/share/directory/file ("share" refers to the alias used to make the resource available.)

VIRUS

A self-replicating code segment that can infect a computer or network, causing minor to major damage

VPN

Virtual Private Network: A secure channel used to transmit data over a public network

WINSOCK

Windows Sockets. A Windows component that connects a Windows PC to the Internet using TCP/IP.

WORKSTATION

Any computer connected to a network.

X.509

An ISO format standard for client and server certificates.

EXHIBIT K

TO MICHAEL FRATTO'S DECLARATION

AVENTAIL SHIPS DIRECTORY-ENABLED EXTRANET SOLUTION; AVENTAIL EXTRANET CENTER V3.1

AVAILABLE AT WWW.AVENTAIL.COM

(BUSINESS WIRE, AUGUST 9, 1999)

Aventail Ships Directory-enabled Extranet Solution; Aventail Extranet Center V3.1 Available At www.aventail.com.

Business Wire
August 9, 1999

SEATTLE--(BUSINESS WIRE)--Aug. 9, 1999--

Aventail Corporation, the leading provider of Extranet Management and Security (EMS) solutions, announced today that they have shipped the latest versions of its award-winning product, Aventail ExtraNet Center(tm).

This latest offering simplifies extranet user management by including broader support for Public Key Infrastructure (PKI) and Lightweight Directory Access Protocol (LDAP)-enabled directories as well as automatic client updating. With these enhancements, Aventail has greatly simplified how large multi-enterprise organizations deploy and manage world-class extranets.

"I was quite impressed with the latest version of Aventail ExtraNet Center and its ability to seamlessly integrate with various LDAP directories and PKI environments," stated Ken Aull, Technical Fellow for TRW. "Aventail ExtraNet Center is a perfect tool for any enterprise organization conducting business-to-business commerce and collaboration. Aventail's standards-based solution reassures corporations that their extranet will work now as well as in the future."

Simplifying User Management

Aventail ExtraNet Center's latest features enable deployments that are easily scalable for any number of extranet users. These new features include:

-- LDAP-enabled Authentication and Authorization: Aventail ExtraNet

Center v3.1 allows corporations implementing various LDAP

directories, including Netscape (NYSE:AOL) Directory Server, IBM

(NYSE:IBM) SecureWay Directory, and Lotus Domino, to authorize

users and groups from an authoritative directory. Aventail's LDAP

implementation complements its existing support for NDS and

Bindery, RADIUS, Windows NT Domain, UNIX Passwrd Files, and

Security Dynamics' ACE/Server. -- Increased Support for Emerging PKI Standards: Adding

broader PKI

support gives users more choices when using digital certificates.

These include the ability to acquire a certificate via a browser

(PKCS #12) and support for smart card and other device-based

authentication (PKCS #11) such as SPYRUS' Rosetta Smart Card. The

expanded suppourID, Hewlett-Packards' (NYSE:HWP) Authorization Server, and

x.509 certificates from VeriSign (Nasdaq:VRSN), Netscape

(Nasdaq:NSCP), GTE (Nasdaq:GTE), and Microsoft (Nasdaq:MSFT). -- Automated Configuration Updates: Aventail ExtraNet Center also

includes the ability to easily configure, distribute, and

automatically update client configuration files. Utilizing

Aventail Customizer(tm), administrators can easily configure and

distribute pre-packaged clients via e-mail, FTP, HTTP, or

application deployment products such as Microsoft's SMS. After

deployment, Aventail's AutoUpdate(tm) allows new configuration

files to be automatically installed without user intervention at

an interval set by the administrator. This easy-to-use

administrative tool reiterates Aventail's efforts in providing a

transparent client for business partners, suppliers, consultants,

and customers.

Pricing and Availability

Aventail ExtraNet Center is currently shipping on Windows NT and Solaris. Additional platform support will be available at the end of August.

Aventail ExtraNet Center is available through Aventail's worldwide sales team and Aventail Extranet Advantage VAR partners as well as leading security vendors such as Hewlett-Packard and BullSoft. Pricing begins at $10,000 depending on client requirements.

A Comprehensive Solution for Extranet Management and Security

Aventail ExtraNet Center is a client/server software solution that includes integrated encryption, authentication and authorization services using the popular IETF standards SSL and SOCKS v5.

Aventail ExtraNet Center has the ability to seamlessly integrate into any existing infrastructure, making it less costly to install and easier to implement and support than other extranet solutions. Aventail ExtraNet Center works with any IP-based application, including legacy host, mainframe, Java, CORBA-based, custom corporate, and client/server applications from vendors such as SAP (NYSE:SAP), BAAN (Nasdaq:BAANF), Oracle (Nasdaq:ORCL), and PeopleSoft (Nasdaq: PSFT). Aventail ExtraNet Center can also traverse any firewall, such as Check Point Software Ltd.'s (Nasdaq:CHKP) Firewall-1/VPN-1, AXENT Technologies' (Nasdaq:AXNT) Raptor Firewall, and IBM's (NYSE:IBM) eNetwork Firewall.

About Aventail Corporation

Founded in 1996, Aventail has quickly emerged as the leading provider of EMS solutions for the Global 2000. Aventail's solutions allow organizations to securely extend their enterprise resources to strategic partners, suppliers, customers, consultants and other key individuals over public IP networks.

Leading corporations are using Aventail's solutions to help them increase their competitive advantage, raise profits, and leverage investments in existing and future enterprise systems. Aventail's solutions are currently deployed at companies such as Aetna, Bear Stearns, Kodak, Hewlett-Packard, IBM, IKON, Marriott, and Xerox. With a strong reputation for providing highly secure and easy-to-manage software solutions, Aventail has received numerous industry awards from publications and industry analyst firms such as Giga Information Group, InfoWorld, Network Computing, LAN Times, BYTE Magazine, Software Digest, and Computer Reseller News.

Aventail Corporation is privately held and headquartered in Seattle, Washington. For more information on the company or to download a trial version of Aventail ExtraNet Center, please visit www.aventail.com, or contact the company directly at 206-215-1111, 877-AVENTAIL, or info@aventail.com. Information on Aventail can also be obtained through Yahoo (Nasdaq:YHOO), Infoseek (Nasdaq:SEEK), Lycos (Nasdaq:LCOS), and Excite (Nasdaq:XCIT).

Aventail is a registered trademark of Aventail Corporation. Aventail ExtraNet Center, Aventail Customizer and Aventail AutoUpdate are trademarks of Aventail Corporation. All other trademarks are the property of their respective owners.

EXHIBIT L

TO MICHAEL FRATTO'S DECLARATION

RFC 1928, SOCKS PROTOCOL VERSION 5,
MARCH 1996, AVAILABLE AT
HTTP://TOOLS.IETF.ORG/HTML/RFC1928

PROPOSED STANDARD

Network Working Group
Request for Comments: 1928
Category: Standards Track

M. Leech
Bell-Northern Research Ltd
M. Ganis
International Business Machines
Y. Lee
NEC Systems Laboratory
R. Kuris
Unify Corporation
D. Koblas
Independent Consultant
L. Jones
Hewlett-Packard Company
March 1996

SOCKS Protocol Version 5

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Acknowledgments

This memo describes a protocol that is an evolution of the previous version of the protocol, version 4 [1]. This new protocol stems from active discussions and prototype implementations. The key contributors are: Marcus Leech: Bell-Northern Research, David Koblas: Independent Consultant, Ying-Da Lee: NEC Systems Laboratory, LaMont Jones: Hewlett-Packard Company, Ron Kuris: Unify Corporation, Matt Ganis: International Business Machines.

1. Introduction

The use of network firewalls, systems that effectively isolate an organizations internal network structure from an exterior network, such as the INTERNET is becoming increasingly popular. These firewall systems typically act as application-layer gateways between networks, usually offering controlled TELNET, FTP, and SMTP access. With the emergence of more sophisticated application layer protocols designed to facilitate global information discovery, there exists a need to provide a general framework for these protocols to transparently and securely traverse a firewall.

There exists, also, a need for strong authentication of such traversal in as fine-grained a manner as is practical. This requirement stems from the realization that client-server relationships emerge between the networks of various organizations, and that such relationships need to be controlled and often strongly authenticated.

The protocol described here is designed to provide a framework for client-server applications in both the TCP and UDP domains to conveniently and securely use the services of a network firewall. The protocol is conceptually a "shim-layer" between the application layer and the transport layer, and as such does not provide network-layer gateway services, such as forwarding of ICMP messages.

2. Existing practice

There currently exists a protocol, SOCKS Version 4, that provides for unsecured firewall traversal for TCP-based client-server

applications, including TELNET, FTP and the popular information-discovery protocols such as HTTP, WAIS and GOPHER.

This new protocol extends the SOCKS Version 4 model to include UDP, and extends the framework to include provisions for generalized strong authentication schemes, and extends the addressing scheme to encompass domain-name and V6 IP addresses.

The implementation of the SOCKS protocol typically involves the recompilation or relinking of TCP-based client applications to use the appropriate encapsulation routines in the SOCKS library.

Note:

Unless otherwise noted, the decimal numbers appearing in packet-format diagrams represent the length of the corresponding field, in octets. Where a given octet must take on a specific value, the syntax X'hh' is used to denote the value of the single octet in that field. When the word 'Variable' is used, it indicates that the corresponding field has a variable length defined either by an associated (one or two octet) length field, or by a data type field.

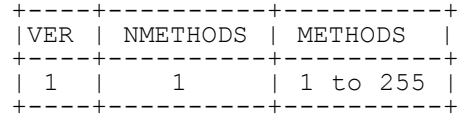
3. Procedure for TCP-based clients

When a TCP-based client wishes to establish a connection to an object that is reachable only via a firewall (such determination is left up to the implementation), it must open a TCP connection to the appropriate SOCKS port on the SOCKS server system. The SOCKS service is conventionally located on TCP port 1080. If the connection request succeeds, the client enters a negotiation for the

authentication method to be used, authenticates with the chosen method, then sends a relay request. The SOCKS server evaluates the request, and either establishes the appropriate connection or denies it.

Unless otherwise noted, the decimal numbers appearing in packet-format diagrams represent the length of the corresponding field, in octets. Where a given octet must take on a specific value, the syntax X'hh' is used to denote the value of the single octet in that field. When the word 'Variable' is used, it indicates that the corresponding field has a variable length defined either by an associated (one or two octet) length field, or by a data type field.

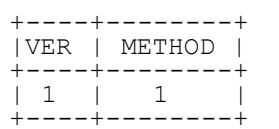
The client connects to the server, and sends a version identifier/method selection message:



VER	NMETHODS	METHODS
1	1	1 to 255

The VER field is set to X'05' for this version of the protocol. The NMETHODS field contains the number of method identifier octets that appear in the METHODS field.

The server selects from one of the methods given in METHODS, and sends a METHOD selection message:



VER	METHOD
1	1

If the selected METHOD is X'FF', none of the methods listed by the client are acceptable, and the client MUST close the connection.

The values currently defined for METHOD are:

- X'00' NO AUTHENTICATION REQUIRED
- X'01' GSSAPI
- X'02' USERNAME/PASSWORD

- X'03' to X'7F' IANA ASSIGNED
- X'80' to X'FE' RESERVED FOR PRIVATE METHODS
- X'FF' NO ACCEPTABLE METHODS

The client and server then enter a method-specific sub-negotiation.

Descriptions of the method-dependent sub-negotiations appear in separate memos.

Developers of new METHOD support for this protocol should contact IANA for a METHOD number. The ASSIGNED NUMBERS document should be referred to for a current list of METHOD numbers and their corresponding protocols.

Compliant implementations MUST support GSSAPI and SHOULD support USERNAME/PASSWORD authentication methods.

4. Requests

Once the method-dependent subnegotiation has completed, the client sends the request details. If the negotiated method includes encapsulation for purposes of integrity checking and/or confidentiality, these requests MUST be encapsulated in the method-dependent encapsulation.

The SOCKS request is formed as follows:

VER	CMD	RSV	ATYP	DST.ADDR	DST.PORT
1	1	X'00'	1	Variable	2

Where:

- VER protocol version: X'05'
- CMD
  - CONNECT X'01'
  - BIND X'02'
  - UDP ASSOCIATE X'03'
- RSV RESERVED
- ATYP address type of following address
  - IP V4 address: X'01'
  - DOMAINNAME: X'03'
  - IP V6 address: X'04'
- DST.ADDR desired destination address
- DST.PORT desired destination port in network octet order

The SOCKS server will typically evaluate the request based on source and destination addresses, and return one or more reply messages, as appropriate for the request type.

5. Addressing

In an address field (DST.ADDR, BND.ADDR), the ATYP field specifies the type of address contained within the field:

- X'01'

the address is a version-4 IP address, with a length of 4 octets

- X'03'

the address field contains a fully-qualified domain name. The first

octet of the address field contains the number of octets of name that follow, there is no terminating NUL octet.

- X'04'

the address is a version-6 IP address, with a length of 16 octets.

6. Replies

The SOCKS request information is sent by the client as soon as it has established a connection to the SOCKS server, and completed the authentication negotiations. The server evaluates the request, and returns a reply formed as follows:

```
+----+-----+-------+------+----------+----------+
|VER | REP |  RSV  | ATYP | BND.ADDR | BND.PORT |
+----+-----+-------+------+----------+----------+
| 1  |  1  | X'00' |  1   | Variable |    2     |
+----+-----+-------+------+----------+----------+
```

Where:

- VER protocol version: X'05'
- REP Reply field:
  - X'00' succeeded
  - X'01' general SOCKS server failure
  - X'02' connection not allowed by ruleset
  - X'03' Network unreachable
  - X'04' Host unreachable
  - X'05' Connection refused
  - X'06' TTL expired
  - X'07' Command not supported
  - X'08' Address type not supported
  - X'09' to X'FF' unassigned
- RSV RESERVED
- ATYP address type of following address

  - IP V4 address: X'01'
  - DOMAINNAME: X'03'
  - IP V6 address: X'04'
- BND.ADDR server bound address
- BND.PORT server bound port in network octet order

Fields marked RESERVED (RSV) must be set to X'00'.

If the chosen method includes encapsulation for purposes of authentication, integrity and/or confidentiality, the replies are encapsulated in the method-dependent encapsulation.

CONNECT

In the reply to a CONNECT, BND.PORT contains the port number that the server assigned to connect to the target host, while BND.ADDR contains the associated IP address. The supplied BND.ADDR is often different from the IP address that the client uses to reach the SOCKS server, since such servers are often multi-homed. It is expected that the SOCKS server will use DST.ADDR and DST.PORT, and the client-side source address and port in evaluating the CONNECT request.

BIND

The BIND request is used in protocols which require the client to accept connections from the server. FTP is a well-known example, which uses the primary client-to-server connection for commands and status reports, but may use a server-to-client connection for transferring data on demand (e.g. LS, GET, PUT).

It is expected that the client side of an application protocol will use the BIND request only to establish secondary connections after a primary connection is established using CONNECT. In is expected that a SOCKS server will use DST.ADDR and DST.PORT in evaluating the BIND request.

Two replies are sent from the SOCKS server to the client during a BIND operation. The first is sent after the server creates and binds a new socket. The BND.PORT field contains the port number that the SOCKS server assigned to listen for an incoming connection. The BND.ADDR field contains the associated IP address. The client will typically use these pieces of information to notify (via the primary or control connection) the application server of the rendezvous address. The second reply occurs only after the anticipated incoming connection succeeds or fails.

In the second reply, the BND.PORT and BND.ADDR fields contain the address and port number of the connecting host.

UDP ASSOCIATE

The UDP ASSOCIATE request is used to establish an association within the UDP relay process to handle UDP datagrams. The DST.ADDR and DST.PORT fields contain the address and port that the client expects to use to send UDP datagrams on for the association. The server MAY use this information to limit access to the association. If the client is not in possesion of the information at the time of the UDP ASSOCIATE, the client MUST use a port number and address of all zeros.

A UDP association terminates when the TCP connection that the UDP ASSOCIATE request arrived on terminates.

In the reply to a UDP ASSOCIATE request, the BND.PORT and BND.ADDR fields indicate the port number/address where the client MUST send UDP request messages to be relayed.

Reply Processing

When a reply (REP value other than X'00') indicates a failure, the SOCKS server MUST terminate the TCP connection shortly after sending the reply. This must be no more than 10 seconds after detecting the condition that caused a failure.

If the reply code (REP value of X'00') indicates a success, and the request was either a BIND or a CONNECT, the client may now start passing data. If the selected authentication method supports encapsulation for the purposes of integrity, authentication and/or confidentiality, the data are encapsulated using the method-dependent encapsulation. Similarly, when data arrives at the SOCKS server for the client, the server MUST encapsulate the data as appropriate for the authentication method in use.

7. Procedure for UDP-based clients

A UDP-based client MUST send its datagrams to the UDP relay server at the UDP port indicated by BND.PORT in the reply to the UDP ASSOCIATE request. If the selected authentication method provides encapsulation for the purposes of authenticity, integrity, and/or confidentiality, the datagram MUST be encapsulated using the appropriate encapsulation. Each UDP datagram carries a UDP request header with it:

RSV	FRAG	ATYP	DST.ADDR	DST.PORT	DATA
2	1	1	Variable	2	Variable

The fields in the UDP request header are:

- RSV Reserved X'0000'
- FRAG Current fragment number
- ATYP address type of following addresses:
  - IP V4 address: X'01'
  - DOMAINNAME: X'03'
  - IP V6 address: X'04'
- DST.ADDR desired destination address
- DST.PORT desired destination port
- DATA user data

When a UDP relay server decides to relay a UDP datagram, it does so silently, without any notification to the requesting client. Similarly, it will drop datagrams it cannot or will not relay. When a UDP relay server receives a reply datagram from a remote host, it MUST encapsulate that datagram using the above UDP request header, and any authentication-method-dependent encapsulation.

The UDP relay server MUST acquire from the SOCKS server the expected IP address of the client that will send datagrams to the BND.PORT given in the reply to UDP ASSOCIATE. It MUST drop any datagrams arriving from any source IP address other than the one recorded for the particular association.

The FRAG field indicates whether or not this datagram is one of a number of fragments. If implemented, the high-order bit indicates end-of-fragment sequence, while a value of X'00' indicates that this datagram is standalone. Values between 1 and 127 indicate the fragment position within a fragment sequence. Each receiver will have a REASSEMBLY QUEUE and a REASSEMBLY TIMER associated with these fragments. The reassembly queue must be reinitialized and the associated fragments abandoned whenever the REASSEMBLY TIMER expires, or a new datagram arrives carrying a FRAG field whose value is less than the highest FRAG value processed for this fragment sequence. The reassembly timer MUST be no less than 5 seconds. It is recommended that fragmentation be avoided by applications wherever possible.

Implementation of fragmentation is optional; an implementation that does not support fragmentation MUST drop any datagram whose FRAG field is other than X'00'.

The programming interface for a SOCKS-aware UDP MUST report an available buffer space for UDP datagrams that is smaller than the actual space provided by the operating system:

- if ATYP is X'01' - 10+method_dependent octets smaller
- if ATYP is X'03' - 262+method_dependent octets smaller
- if ATYP is X'04' - 20+method_dependent octets smaller

8. Security Considerations

This document describes a protocol for the application-layer traversal of IP network firewalls. The security of such traversal is highly dependent on the particular authentication and encapsulation methods provided in a particular implementation, and selected during negotiation between SOCKS client and SOCKS server.

Careful consideration should be given by the administrator to the selection of authentication methods.

9. References

[1] Koblas, D., "SOCKS", Proceedings: 1992 Usenix Security Symposium.

Author's Address

Marcus Leech
Bell-Northern Research Ltd
P.O. Box 3511, Stn. C,
Ottawa, ON
CANADA K1Y 4H7

Phone: (613) 763-9145
EMail: mleech@bnr.ca

EXHIBIT M

TO MICHAEL FRATTO'S DECLARATION

RFC1035

DOMAIN NAMES-IMPLEMENTATION AND SPECIFICATION

Updated by: 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2137, 2181, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966 STANDARD
Errata Exist

Network Working Group
Request for Comments: 1035
P. Mockapetris
ISI
November 1987

Obsoletes: RFCs 882, 883, 973

DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION

1. STATUS OF THIS MEMO

This RFC describes the details of the domain system and protocol, and assumes that the reader is familiar with the concepts discussed in a companion RFC, "Domain Names - Concepts and Facilities" [RFC-1034].

The domain system is a mixture of functions and data types which are an official protocol and functions and data types which are still experimental. Since the domain system is intentionally extensible, new data types and experimental behavior should always be expected in parts of the system beyond the official protocol. The official protocol parts include standard queries, responses and the Internet class RR data formats (e.g., host addresses). Since the previous RFC set, several definitions have changed, so some previous definitions are obsolete.

Experimental or obsolete features are clearly marked in these RFCs, and such information should be used with caution.

The reader is especially cautioned not to depend on the values which appear in examples to be current or complete, since their purpose is primarily pedagogical. Distribution of this memo is unlimited.

Table of Contents

2. INTRODUCTION

2.1. Overview

The goal of domain names is to provide a mechanism for naming resources in such a way that the names are usable in different hosts, networks, protocol families, internets, and administrative organizations.

From the user's point of view, domain names are useful as arguments to a local agent, called a resolver, which retrieves information associated with the domain name. Thus a user might ask for the host address or mail information associated with a particular domain name. To enable the user to request a particular type of information, an appropriate query type is passed to the resolver with the domain name. To the user, the domain tree is a single information space; the resolver is responsible for hiding the distribution of data among name servers from the user.

From the resolver's point of view, the database that makes up the domain space is distributed among various name servers. Different parts of the domain space are stored in different name servers, although a particular data item will be stored redundantly in two or more name servers. The resolver starts with knowledge of at least one name server. When the resolver processes a user query it asks a known name server for the information; in return, the resolver either receives the desired information or a referral to another name server. Using these referrals, resolvers learn the identities and contents of other name servers. Resolvers are responsible for dealing with the distribution of

the domain space and dealing with the effects of name server failure by consulting redundant databases in other servers.

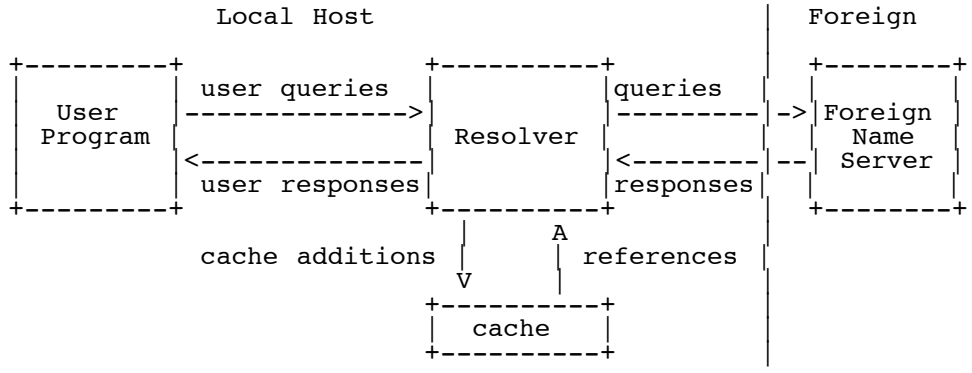
Name servers manage two kinds of data. The first kind of data held in sets called zones; each zone is the complete database for a particular "pruned" subtree of the domain space. This data is called authoritative. A name server periodically checks to make sure that its zones are up to date, and if not, obtains a new copy of updated zones

from master files stored locally or in another name server. The second kind of data is cached data which was acquired by a local resolver. This data may be incomplete, but improves the performance of the retrieval process when non-local data is repeatedly accessed. Cached data is eventually discarded by a timeout mechanism.

This functional structure isolates the problems of user interface, failure recovery, and distribution in the resolvers and isolates the database update and refresh problems in the name servers.

2.2. Common configurations

A host can participate in the domain name system in a number of ways, depending on whether the host runs programs that retrieve information from the domain system, name servers that answer queries from other hosts, or various combinations of both functions. The simplest, and perhaps most typical, configuration is shown below:

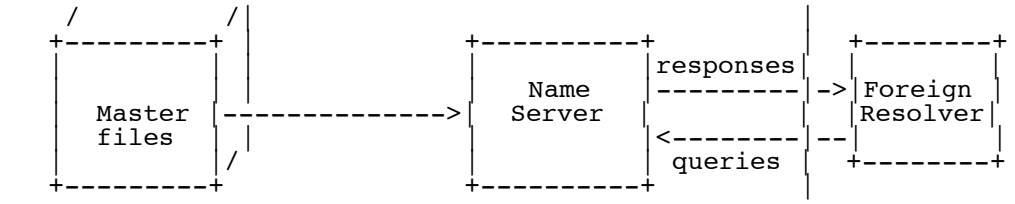


User programs interact with the domain name space through resolvers; the format of user queries and user responses is specific to the host and its operating system. User queries will typically be operating system calls, and the resolver and its cache will be part of the host operating system. Less capable hosts may choose to implement the resolver as a subroutine to be linked in with every program that needs its services. Resolvers answer user queries with information they acquire via queries to foreign name servers and the local cache.

Note that the resolver may have to make several queries to several different foreign name servers to answer a particular user query, and hence the resolution of a user query may involve several network accesses and an arbitrary amount of time. The queries to foreign name servers and the corresponding responses have a standard format described

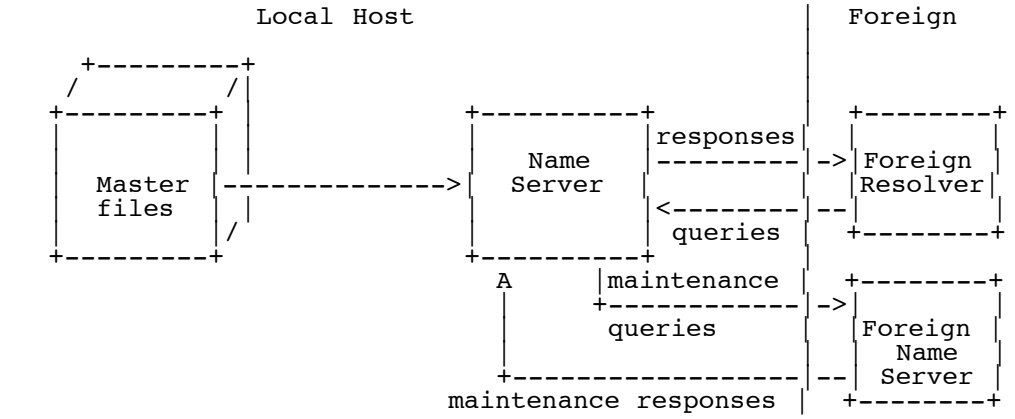
in this memo, and may be datagrams.

Depending on its capabilities, a name server could be a stand alone program on a dedicated machine or a process or processes on a large timeshared host. A simple configuration might be:



Here a primary name server acquires information about one or more zones by reading master files from its local file system, and answers queries about those zones that arrive from foreign resolvers.

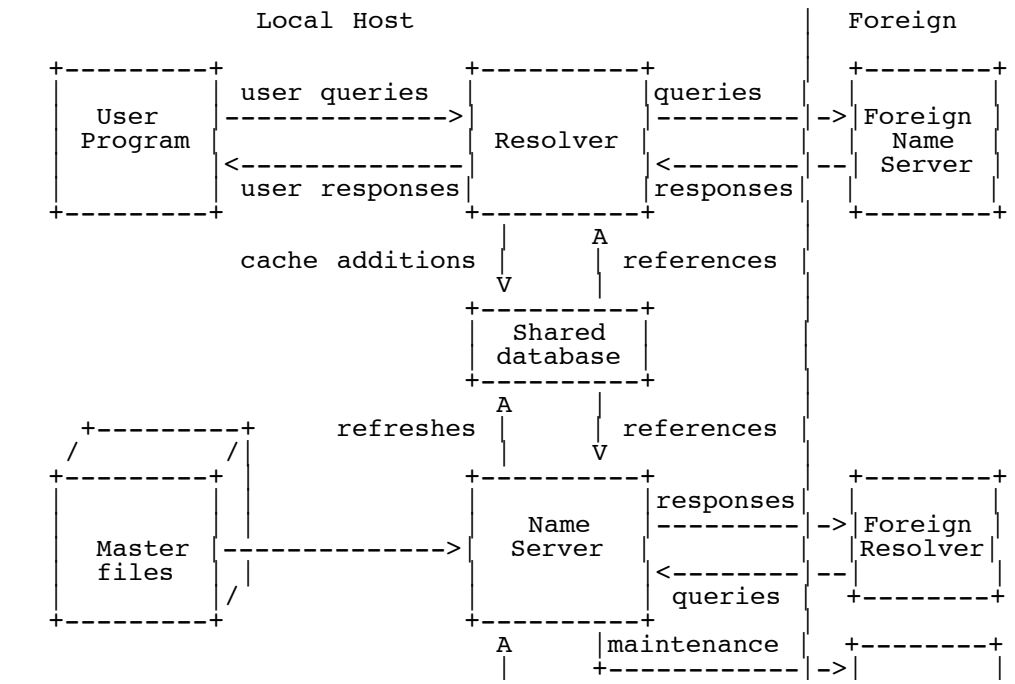
The DNS requires that all zones be redundantly supported by more than one name server. Designated secondary servers can acquire zones and check for updates from the primary server using the zone transfer protocol of the DNS. This configuration is shown below:

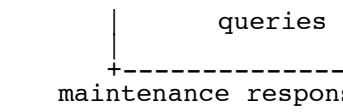


In this configuration, the name server periodically establishes a virtual circuit to a foreign name server to acquire a copy of a zone or to check that an existing copy has not changed. The messages sent for

these maintenance activities follow the same form as queries and responses, but the message sequences are somewhat different.

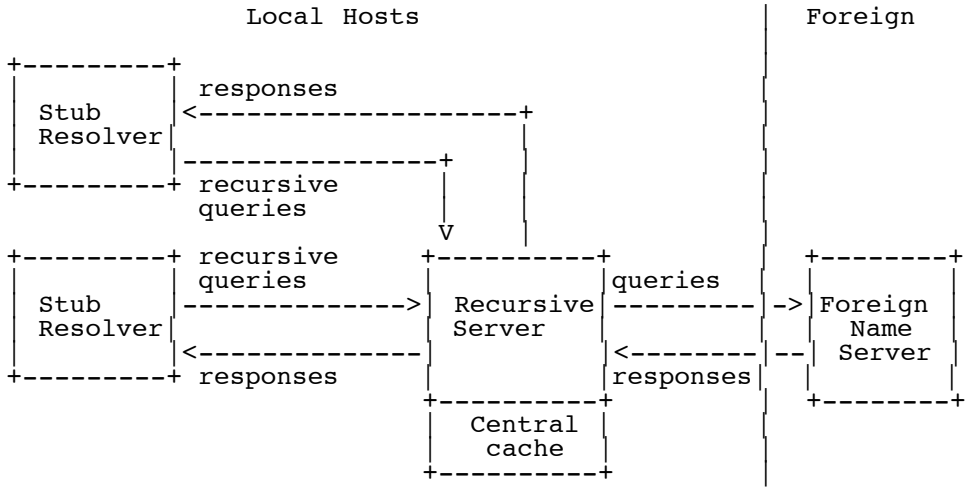
The information flow in a host that supports all aspects of the domain name system is shown below:





The shared database holds domain space data for the local name server and resolver. The contents of the shared database will typically be a mixture of authoritative data maintained by the periodic refresh operations of the name server and cached data from previous resolver requests. The structure of the domain data and the necessity for synchronization between name servers and resolvers imply the general characteristics of this database, but the actual format is up to the local implementor.

Information flow can also be tailored so that a group of hosts act together to optimize activities. Sometimes this is done to offload less capable hosts so that they do not have to implement a full resolver. This can be appropriate for PCs or hosts which want to minimize the amount of new network code which is required. This scheme can also allow a group of hosts can share a small number of caches rather than maintaining a large number of separate caches, on the premise that the centralized caches will have a higher hit ratio. In either case, resolvers are replaced with stub resolvers which act as front ends to resolvers located in a recursive server in one or more name servers known to perform that service:



In any case, note that domain components are always replicated for reliability whenever possible.

2.3. Conventions

The domain system has several conventions dealing with low-level, but fundamental, issues. While the implementor is free to violate these conventions WITHIN HIS OWN SYSTEM, he must observe these conventions in ALL behavior observed from other hosts.

2.3.1. Preferred name syntax

The DNS specifications attempt to be as general as possible in the rules for constructing domain names. The idea is that the name of any existing object can be expressed as a domain name with minimal changes.

However, when assigning a domain name for an object, the prudent user will select a name which satisfies both the rules of the domain system and any existing rules for the object, whether these rules are published

or implied by existing programs.

For example, when naming a mail domain, the user should satisfy both the rules of this memo and those in RFC-822. When creating a new host name, the old rules for HOSTS.TXT should be followed. This avoids problems when old software is converted to use domain names.

The following syntax will result in fewer problems with many

applications that use domain names (e.g., mail, TELNET).

```
<domain> ::= <subdomain> | " "

<subdomain> ::= <label> | <subdomain> "." <label>

<label> ::= <letter> [ [ <ldh-str> ] <let-dig> ]

<ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>

<let-dig-hyp> ::= <let-dig> | "-"

<let-dig> ::= <letter> | <digit>

<letter> ::= any one of the 52 alphabetic characters A through Z in
upper case and a through z in lower case

<digit> ::= any one of the ten digits 0 through 9
```

Note that while upper and lower case letters are allowed in domain names, no significance is attached to the case. That is, two names with the same spelling but different case are to be treated as if identical.

The labels must follow the rules for ARPANET host names. They must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, and hyphen. There are also some restrictions on the length. Labels must be 63 characters or less.

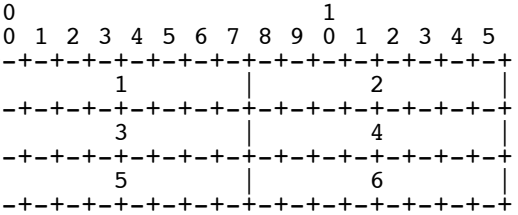
For example, the following strings identify hosts in the Internet:

A.ISI.EDU XX.LCS.MIT.EDU SRI-NIC.ARPA

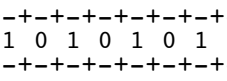
2.3.2. Data Transmission Order

The order of transmission of the header and data described in this document is resolved to the octet level. Whenever a diagram shows a

group of octets, the order of transmission of those octets is the normal order in which they are read in English. For example, in the following diagram, the octets are transmitted in the order they are numbered.



Whenever an octet represents a numeric quantity, the left most bit in the diagram is the high order or most significant bit. That is, the bit labeled 0 is the most significant bit. For example, the following diagram represents the value 170 (decimal).



Similarly, whenever a multi-octet field represents a numeric quantity the left most bit of the whole field is the most significant bit. When a multi-octet quantity is transmitted the most significant octet is transmitted first.

2.3.3. Character Case

For all parts of the DNS that are part of the official protocol, all comparisons between character strings (e.g., labels, domain names, etc.) are done in a case-insensitive manner. At present, this rule is in force throughout the domain system without exception. However, future additions beyond current usage may need to use the full binary octet capabilities in names, so attempts to store domain names in 7-bit ASCII or use of special bytes to terminate labels, etc., should be avoided.

When data enters the domain system, its original case should be preserved whenever possible. In certain circumstances this cannot be done. For example, if two RRs are stored in a database, one at x.y and one at X.Y, they are actually stored at the same place in the database, and hence only one casing would be preserved. The basic rule is that case can be discarded only when data is used to define structure in a database, and two names are identical when compared in a case insensitive manner.

Loss of case sensitive data must be minimized. Thus while data for x.y and X.Y may both be stored under a single location x.y or X.Y, data for a.x and B.X would never be stored under A.x, A.X, b.x, or b.X. In general, this preserves the case of the first label of a domain name, but forces standardization of interior node labels.

Systems administrators who enter data into the domain database should take care to represent the data they supply to the domain system in a case-consistent manner if their system is case-sensitive. The data distribution system in the domain system will ensure that consistent representations are preserved.

2.3.4. Size limits

Various objects and parameters in the DNS have size limits. They are listed below. Some could be easily changed, others are more fundamental.

labels	63 octets or less
names	255 octets or less
TTL	positive values of a signed 32 bit number.
UDP messages	512 octets or less

3. DOMAIN NAME SPACE AND RR DEFINITIONS

3.1. Name space definitions

Domain names in messages are expressed in terms of a sequence of labels. Each label is represented as a one octet length field followed by that number of octets. Since every domain name ends with the null label of the root, a domain name is terminated by a length byte of zero. The high order two bits of every length octet must be zero, and the remaining six bits of the length field limit the label to 63 octets or less.

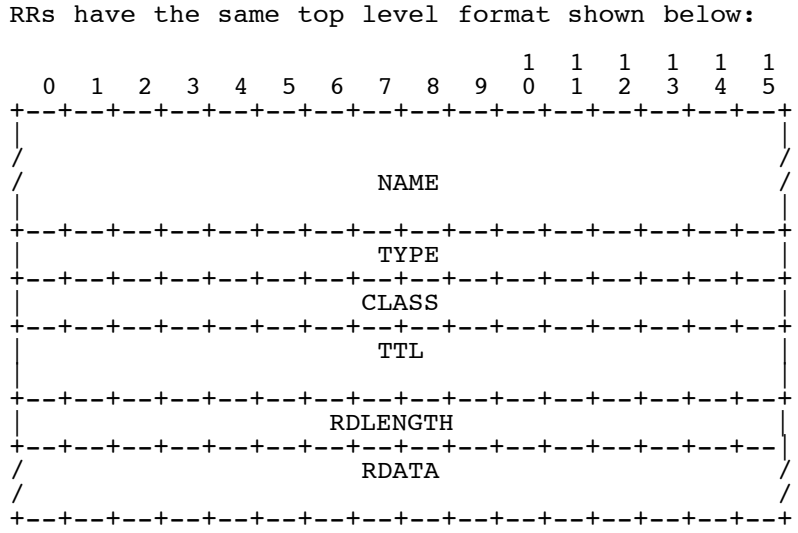
To simplify implementations, the total length of a domain name (i.e., label octets and label length octets) is restricted to 255 octets or less.

Although labels can contain any 8 bit values in octets that make up a label, it is strongly recommended that labels follow the preferred syntax described elsewhere in this memo, which is compatible with existing host naming conventions. Name servers and resolvers must compare labels in a case-insensitive manner (i.e., A=a), assuming ASCII with zero parity. Non-alphabetic codes must match exactly.

3.2. RR definitions

3.2.1. Format

All RRs have the same top level format shown below:



where:

NAME an owner name, i.e., the name of the node to which this resource record pertains.

TYPE two octets containing one of the RR TYPE codes.

CLASS two octets containing one of the RR CLASS codes.

TTL a 32 bit signed integer that specifies the time interval that the resource record may be cached before the source of the information should again be consulted. Zero values are interpreted to mean that the RR can only be used for the transaction in progress, and should not be cached. For example, SOA records are always distributed with a zero TTL to prohibit caching. Zero values can also be used for extremely volatile data.

RDLENGTH an unsigned 16 bit integer that specifies the length in octets of the RDATA field.

RDATA a variable length string of octets that describes the resource. The format of this information varies according to the TYPE and CLASS of the resource record.

3.2.2. TYPE values

TYPE fields are used in resource records. Note that these types are a subset of QTYPEs.

TYPE	value and meaning
A	**1 a host address**
NS	2 an authoritative name server
MD	3 a mail destination (Obsolete - use MX)
MF	4 a mail forwarder (Obsolete - use MX)
CNAME	5 the canonical name for an alias
SOA	6 marks the start of a zone of authority

MB	7 a mailbox domain name (EXPERIMENTAL)
MG	8 a mail group member (EXPERIMENTAL)
MR	9 a mail rename domain name (EXPERIMENTAL)
NULL	10 a null RR (EXPERIMENTAL)
WKS	11 a well known service description
PTR	12 a domain name pointer
HINFO	13 host information
MINFO	14 mailbox or mail list information
MX	15 mail exchange
TXT	16 text strings

3.2.3. QTYPE values

QTYPE fields appear in the question part of a query. QTYPES are a superset of TYPEs, hence all TYPEs are valid QTYPEs. In addition, the following QTYPEs are defined:

AXFR	252 A request for a transfer of an entire zone
MAILB	253 A request for mailbox-related records (MB, MG or MR)
MAILA	254 A request for mail agent RRs (Obsolete - see MX)
*	255 A request for all records

3.2.4. CLASS values

CLASS fields appear in resource records. The following CLASS mnemonics and values are defined:

IN	1 the Internet
CS	2 the CSNET class (Obsolete - used only for examples in some obsolete RFCs)
CH	3 the CHAOS class
HS	4 Hesiod [Dyer 87]

3.2.5. QCLASS values

QCLASS fields appear in the question section of a query. QCLASS values are a superset of CLASS values; every CLASS is a valid QCLASS. In addition to CLASS values, the following QCLASSes are defined:

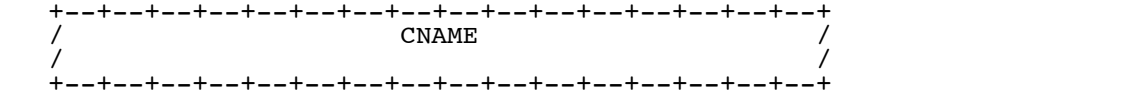
*	255 any class

3.3. Standard RRs

The following RR definitions are expected to occur, at least potentially, in all classes. In particular, NS, SOA, CNAME, and PTR will be used in all classes, and have the same format in all classes. Because their RDATA format is known, all domain names in the RDATA section of these RRs may be compressed.

<domain-name> is a domain name represented as a series of labels, and terminated by a label with zero length. <character-string> is a single length octet followed by that number of characters. <character-string> is treated as binary information, and can be up to 256 characters in length (including the length octet).

3.3.1. CNAME RDATA format

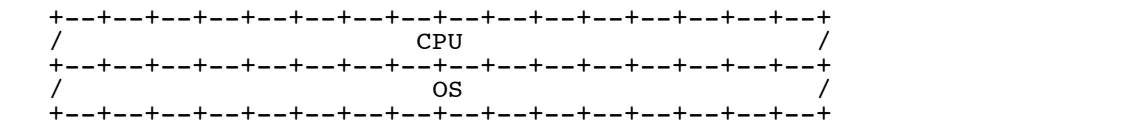


where:

CNAME A <domain-name> which specifies the canonical or primary name for the owner. The owner name is an alias.

CNAME RRs cause no additional section processing, but name servers may choose to restart the query at the canonical name in certain cases. See the description of name server logic in [RFC-1034] for details.

3.3.2. HINFO RDATA format



where:

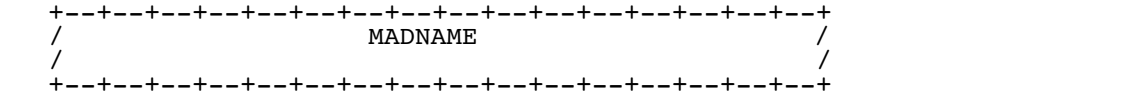
CPU A <character-string> which specifies the CPU type.

OS A <character-string> which specifies the operating system type.

Standard values for CPU and OS can be found in [RFC-1010].

HINFO records are used to acquire general information about a host. The main use is for protocols such as FTP that can use special procedures when talking between machines or operating systems of the same type.

3.3.3. MB RDATA format (EXPERIMENTAL)

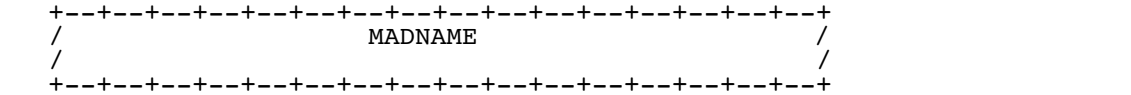


where:

MADNAME A <domain-name> which specifies a host which has the specified mailbox.

MB records cause additional section processing which looks up an A type RRs corresponding to MADNAME.

3.3.4. MD RDATA format (Obsolete)



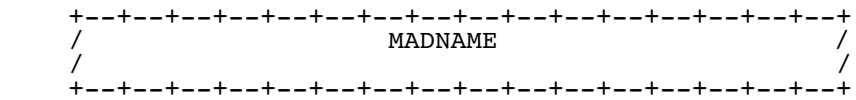
where:

MADNAME A <domain-name> which specifies a host which has a mail agent for the domain which should be able to deliver mail for the domain.

MD records cause additional section processing which looks up an A type record corresponding to MADNAME.

MD is obsolete. See the definition of MX and [RFC-974] for details of the new scheme. The recommended policy for dealing with MD RRs found in a master file is to reject them, or to convert them to MX RRs with a preference of 0.

3.3.5. MF RDATA format (Obsolete)



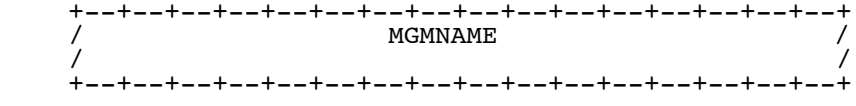
where:

MADNAME A <domain-name> which specifies a host which has a mail agent for the domain which will accept mail for forwarding to the domain.

MF records cause additional section processing which looks up an A type record corresponding to MADNAME.

MF is obsolete. See the definition of MX and [RFC-974] for details ofw the new scheme. The recommended policy for dealing with MD RRs found in a master file is to reject them, or to convert them to MX RRs with a preference of 10.

3.3.6. MG RDATA format (EXPERIMENTAL)

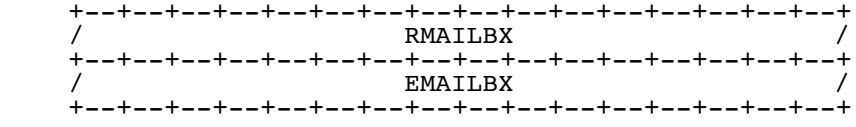


where:

MGMNAME A <domain-name> which specifies a mailbox which is a member of the mail group specified by the domain name.

MG records cause no additional section processing.

3.3.7. MINFO RDATA format (EXPERIMENTAL)



where:

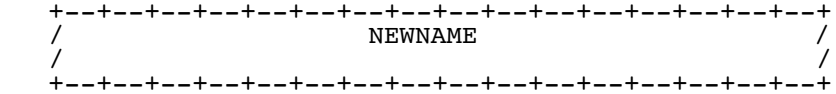
RMAILBX A <domain-name> which specifies a mailbox which is responsible for the mailing list or mailbox. If this domain name names the root, the owner of the MINFO RR is responsible for itself. Note that many existing mailing lists use a mailbox X-request for the RMAILBX field of mailing list X, e.g., Msgroup-request for Msgroup. This field provides a more general mechanism.

EMAILBX A <domain-name> which specifies a mailbox which is to receive error messages related to the mailing list or mailbox specified by the owner of the MINFO RR (similar to the ERRORS-TO: field which has been proposed). If this domain name names the root, errors should be returned to the sender of the message.

MINFO records cause no additional section processing. Although these

records can be associated with a simple mailbox, they are usually used with a mailing list.

3.3.8. MR RDATA format (EXPERIMENTAL)

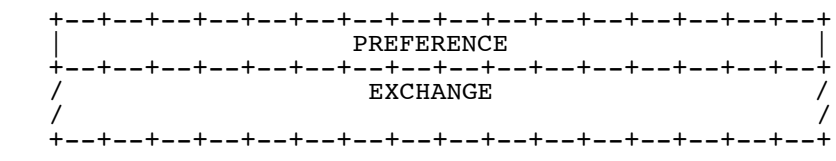


where:

NEWNAME A <domain-name> which specifies a mailbox which is the proper rename of the specified mailbox.

MR records cause no additional section processing. The main use for MR is as a forwarding entry for a user who has moved to a different mailbox.

3.3.9. MX RDATA format



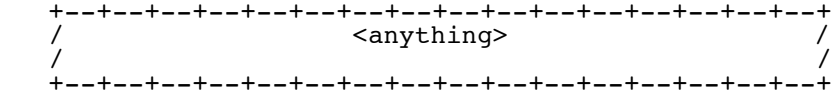
where:

PREFERENCE A 16 bit integer which specifies the preference given to this RR among others at the same owner. Lower values are preferred.

EXCHANGE A <domain-name> which specifies a host willing to act as a mail exchange for the owner name.

MX records cause type A additional section processing for the host specified by EXCHANGE. The use of MX RRs is explained in detail in [RFC-974].

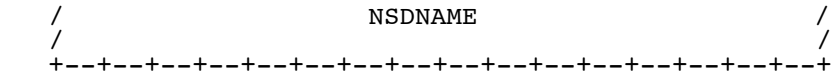
3.3.10. NULL RDATA format (EXPERIMENTAL)



Anything at all may be in the RDATA field so long as it is 65535 octets or less.

NULL records cause no additional section processing. NULL RRs are not allowed in master files. NULLs are used as placeholders in some experimental extensions of the DNS.

3.3.11. NS RDATA format



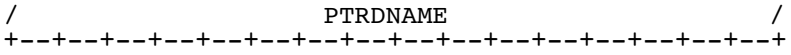
where:

NSDNAME A <domain-name> which specifies a host which should be authoritative for the specified class and domain.

NS records cause both the usual additional section processing to locate a type A record, and, when used in a referral, a special search of the zone in which they reside for glue information.

The NS RR states that the named host should be expected to have a zone starting at owner name of the specified class. Note that the class may not indicate the protocol family which should be used to communicate with the host, although it is typically a strong hint. For example, hosts which are name servers for either Internet (IN) or Hesiod (HS) class information are normally queried using IN class protocols.

3.3.12. PTR RDATA format

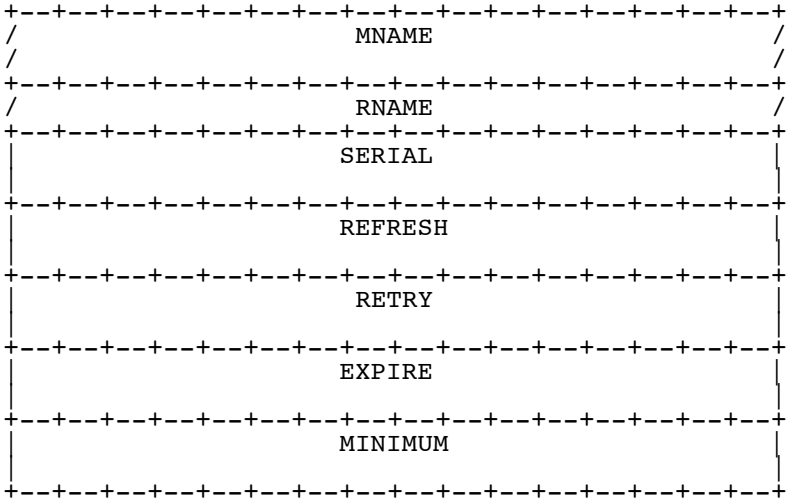


where:

PTRDNAME A <domain-name> which points to some location in the domain name space.

PTR records cause no additional section processing. These RRs are used in special domains to point to some other location in the domain space. These records are simple data, and don't imply any special processing similar to that performed by CNAME, which identifies aliases. See the description of the IN-ADDR.ARPA domain for an example.

3.3.13. SOA RDATA format



where:

MNAME The <domain-name> of the name server that was the original or primary source of data for this zone.

RNAME A <domain-name> which specifies the mailbox of the person responsible for this zone.

SERIAL The unsigned 32 bit version number of the original copy of the zone. Zone transfers preserve this value. This value wraps and should be compared using sequence space

arithmetic.

REFRESH A 32 bit time interval before the zone should be refreshed.

RETRY A 32 bit time interval that should elapse before a failed refresh should be retried.

EXPIRE A 32 bit time value that specifies the upper limit on the time interval that can elapse before the zone is no longer authoritative.

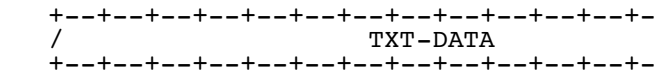
MINIMUM The unsigned 32 bit minimum TTL field that should be exported with any RR from this zone.

SOA records cause no additional section processing.

All times are in units of seconds.

Most of these fields are pertinent only for name server maintenance operations. However, MINIMUM is used in all query operations that retrieve RRs from a zone. Whenever a RR is sent in a response to a query, the TTL field is set to the maximum of the TTL field from the RR and the MINIMUM field in the appropriate SOA. Thus MINIMUM is a lower bound on the TTL field for all RRs in a zone. Note that this use of MINIMUM should occur when the RRs are copied into the response and not when the zone is loaded from a master file or via a zone transfer. The reason for this provison is to allow future dynamic update facilities to change the SOA RR with known semantics.

3.3.14. TXT RDATA format



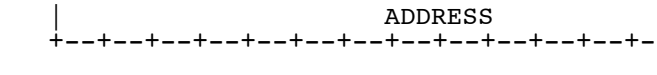
where:

TXT-DATA One or more <character-string>s.

TXT RRs are used to hold descriptive text. The semantics of the text depends on the domain where it is found.

3.4. Internet specific RRs

3.4.1. A RDATA format



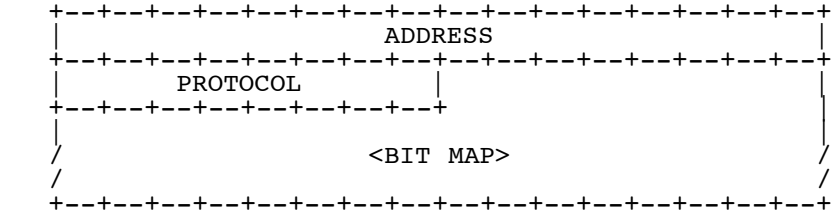
where:

ADDRESS A 32 bit Internet address.

Hosts that have multiple Internet addresses will have multiple A records.

A records cause no additional section processing. The RDATA section of an A line in a master file is an Internet address expressed as four decimal numbers separated by dots without any imbedded spaces (e.g., "10.2.0.52" or "192.0.5.6").

3.4.2. WKS RDATA format



where:

ADDRESS An 32 bit Internet address

PROTOCOL An 8 bit IP protocol number

<BIT MAP> A variable length bit map. The bit map must be a multiple of 8 bits long.

The WKS record is used to describe the well known services supported by a particular protocol on a particular internet address. The PROTOCOL field specifies an IP protocol number, and the bit map has one bit per port of the specified protocol. The first bit corresponds to port 0, the second to port 1, etc. If the bit map does not include a bit for a protocol of interest, that bit is assumed zero. The appropriate values and mnemonics for ports and protocols are specified in [RFC-1010].

For example, if PROTOCOL=TCP (6), the 26th bit corresponds to TCP port **25 (SMTP). If this bit is set, a SMTP server should be listening on TCP** port 25; if zero, SMTP service is not supported on the specified address.

The purpose of WKS RRs is to provide availability information for servers for TCP and UDP. If a server supports both TCP and UDP, or has multiple Internet addresses, then multiple WKS RRs are used.

WKS RRs cause no additional section processing.

In master files, both ports and protocols are expressed using mnemonics or decimal numbers.

3.5. IN-ADDR.ARPA domain

The Internet uses a special domain to support gateway location and Internet address to host mapping. Other classes may employ a similar strategy in other domains. The intent of this domain is to provide a guaranteed method to perform host address to host name mapping, and to facilitate queries to locate all gateways on a particular network in the Internet.

Note that both of these services are similar to functions that could be performed by inverse queries; the difference is that this part of the domain name space is structured according to address, and hence can guarantee that the appropriate data can be located without an exhaustive search of the domain space.

The domain begins at IN-ADDR.ARPA and has a substructure which follows the Internet addressing structure.

Domain names in the IN-ADDR.ARPA domain are defined to have up to four labels in addition to the IN-ADDR.ARPA suffix. Each label represents one octet of an Internet address, and is expressed as a character string for a decimal value in the range 0-255 (with leading zeros omitted except in the case of a zero octet which is represented by a single zero).

Host addresses are represented by domain names that have all four labels specified. Thus data for Internet address 10.2.0.52 is located at domain name 52.0.2.10.IN-ADDR.ARPA. The reversal, though awkward to read, allows zones to be delegated which are exactly one network of

address space. For example, 10.IN-ADDR.ARPA can be a zone containing data for the ARPANET, while 26.IN-ADDR.ARPA can be a separate zone for MILNET. Address nodes are used to hold pointers to primary host names in the normal domain space.

Network numbers correspond to some non-terminal nodes at various depths in the IN-ADDR.ARPA domain, since Internet network numbers are either 1, 2, or 3 octets. Network nodes are used to hold pointers to the primary host names of gateways attached to that network. Since a gateway is, by definition, on more than one network, it will typically have two or more network nodes which point at it. Gateways will also have host level pointers at their fully qualified addresses.

Both the gateway pointers at network nodes and the normal host pointers at full address nodes use the PTR RR to point back to the primary domain names of the corresponding hosts.

For example, the IN-ADDR.ARPA domain will contain information about the ISI gateway between net 10 and 26, an MIT gateway from net 10 to MIT's

net 18, and hosts A.ISI.EDU and MULTICS.MIT.EDU. Assuming that ISI gateway has addresses 10.2.0.22 and 26.0.0.103, and a name MILNET-GW.ISI.EDU, and the MIT gateway has addresses 10.0.0.77 and 18.10.0.4 and a name GW.LCS.MIT.EDU, the domain database would contain:

```
    10.IN-ADDR.ARPA.            PTR MILNET-GW.ISI.EDU.
    10.IN-ADDR.ARPA.            PTR GW.LCS.MIT.EDU.
    18.IN-ADDR.ARPA.            PTR GW.LCS.MIT.EDU.
    26.IN-ADDR.ARPA.            PTR MILNET-GW.ISI.EDU.
    22.0.2.10.IN-ADDR.ARPA.     PTR MILNET-GW.ISI.EDU.
    103.0.0.26.IN-ADDR.ARPA.    PTR MILNET-GW.ISI.EDU.
    77.0.0.10.IN-ADDR.ARPA.     PTR GW.LCS.MIT.EDU.
    4.0.10.18.IN-ADDR.ARPA.     PTR GW.LCS.MIT.EDU.
    103.0.3.26.IN-ADDR.ARPA.    PTR A.ISI.EDU.
    6.0.0.10.IN-ADDR.ARPA.      PTR MULTICS.MIT.EDU.
```

Thus a program which wanted to locate gateways on net 10 would originate a query of the form QTYPE=PTR, QCLASS=IN, QNAME=10.IN-ADDR.ARPA. It would receive two RRs in response:

```
    10.IN-ADDR.ARPA.            PTR MILNET-GW.ISI.EDU.
    10.IN-ADDR.ARPA.            PTR GW.LCS.MIT.EDU.
```

The program could then originate QTYPE=A, QCLASS=IN queries for MILNET-GW.ISI.EDU. and GW.LCS.MIT.EDU. to discover the Internet addresses of these gateways.

A resolver which wanted to find the host name corresponding to Internet host address 10.0.0.6 would pursue a query of the form QTYPE=PTR, QCLASS=IN, QNAME=6.0.0.10.IN-ADDR.ARPA, and would receive:

```
    6.0.0.10.IN-ADDR.ARPA.      PTR MULTICS.MIT.EDU.
```

Several cautions apply to the use of these services:

- Since the IN-ADDR.ARPA special domain and the normal domain for a particular host or gateway will be in different zones, the possibility exists that that the data may be inconsistent.

- Gateways will often have two names in separate domains, only one of which can be primary.

- Systems that use the domain database to initialize their routing tables must start with enough gateway information to guarantee that they can access the appropriate name server.

- The gateway data only reflects the existence of a gateway in a manner equivalent to the current HOSTS.TXT file. It doesn't replace the dynamic availability information from GGP or EGP.

3.6. Defining new types, classes, and special namespaces

The previously defined types and classes are the ones in use as of the date of this memo. New definitions should be expected. This section makes some recommendations to designers considering additions to the existing facilities. The mailing list NAMEDROPPERS@SRI-NIC.ARPA is the forum where general discussion of design issues takes place.

In general, a new type is appropriate when new information is to be added to the database about an existing object, or we need new data formats for some totally new object. Designers should attempt to define types and their RDATA formats that are generally applicable to all classes, and which avoid duplication of information. New classes are appropriate when the DNS is to be used for a new protocol, etc which requires new class-specific data formats, or when a copy of the existing name space is desired, but a separate management domain is necessary.

New types and classes need mnemonics for master files; the format of the master files requires that the mnemonics for type and class be disjoint.

TYPE and CLASS values must be a proper subset of QTYPEs and QCLASSes respectively.

The present system uses multiple RRs to represent multiple values of a type rather than storing multiple values in the RDATA section of a single RR. This is less efficient for most applications, but does keep RRs shorter. The multiple RRs assumption is incorporated in some experimental work on dynamic update methods.

The present system attempts to minimize the duplication of data in the database in order to insure consistency. Thus, in order to find the address of the host for a mail exchange, you map the mail domain name to a host name, then the host name to addresses, rather than a direct mapping to host address. This approach is preferred because it avoids the opportunity for inconsistency.

In defining a new type of data, multiple RR types should not be used to create an ordering between entries or express different formats for equivalent bindings, instead this information should be carried in the body of the RR and a single type used. This policy avoids problems with caching multiple types and defining QTYPEs to match multiple types.

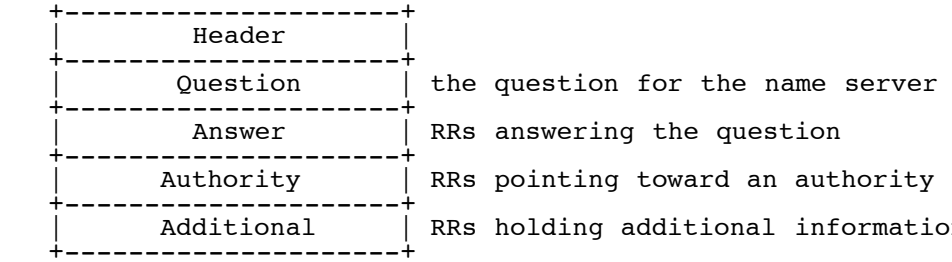
For example, the original form of mail exchange binding used two RR types one to represent a "closer" exchange (MD) and one to represent a "less close" exchange (MF). The difficulty is that the presence of one RR type in a cache doesn't convey any information about the other because the query which acquired the cached information might have used a QTYPE of MF, MD, or MAILA (which matched both). The redesigned

service used a single type (MX) with a "preference" value in the RDATA section which can order different RRs. However, if any MX RRs are found in the cache, then all should be there.

4. MESSAGES

4.1. Format

All communications inside of the domain protocol are carried in a single format called a message. The top level format of message is divided into 5 sections (some of which are empty in certain cases) shown below:

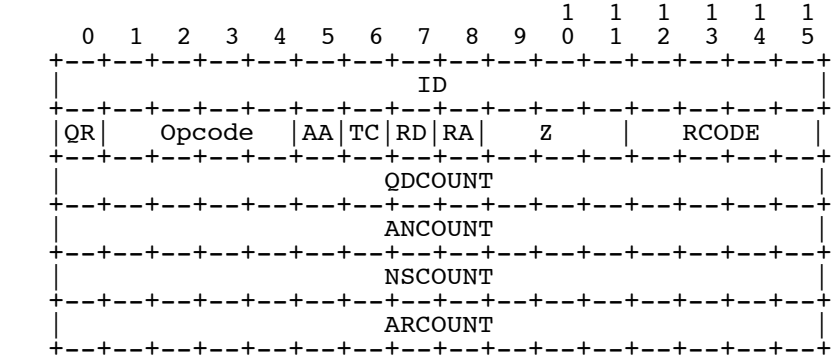


The header section is always present. The header includes fields that specify which of the remaining sections are present, and also specify whether the message is a query or a response, a standard query or some other opcode, etc.

The names of the sections after the header are derived from their use in standard queries. The question section contains fields that describe a question to a name server. These fields are a query type (QTYPE), a query class (QCLASS), and a query domain name (QNAME). The last three sections have the same format: a possibly empty list of concatenated resource records (RRs). The answer section contains RRs that answer the question; the authority section contains RRs that point toward an authoritative name server; the additional records section contains RRs which relate to the query, but are not strictly answers for the question.

4.1.1. Header section format

The header contains the following fields:



where:

ID A 16 bit identifier assigned by the program that generates any kind of query. This identifier is copied the corresponding reply and can be used by the requester to match up replies to outstanding queries.

QR A one bit field that specifies whether this message is a query (0), or a response (1).

OPCODE A four bit field that specifies kind of query in this message. This value is set by the originator of a query and copied into the response. The values are:

0	a standard query (QUERY)
1	an inverse query (IQUERY)
2	a server status request (STATUS)
3-15	reserved for future use

AA Authoritative Answer - this bit is valid in responses, and specifies that the responding name server is an authority for the domain name in question section.

Note that the contents of the answer section may have multiple owner names because of aliases. The AA bit

corresponds to the name which matches the query name, or the first owner name in the answer section.

TC TrunCation - specifies that this message was truncated due to length greater than that permitted on the transmission channel.

RD Recursion Desired - this bit may be set in a query and is copied into the response. If RD is set, it directs the name server to pursue the query recursively. Recursive query support is optional.

RA Recursion Available - this be is set or cleared in a response, and denotes whether recursive query support is available in the name server.

Z **Reserved for future use. Must be zero in all queries** and responses.

RCODE Response code - this 4 bit field is set as part of responses. The values have the following interpretation:

0	No error condition
1	Format error - The name server was unable to interpret the query.
2	Server failure - The name server was unable to process this query due to a problem with the name server.
3	Name Error - Meaningful only for responses from an authoritative name server, this code signifies that the domain name referenced in the query does not exist.
4	Not Implemented - The name server does not support the requested kind of query.
5	Refused - The name server refuses to perform the specified operation for policy reasons. For example, a name server may not wish to provide the information to the particular requester, or a name server may not wish to perform a particular operation (e.g., zone

transfer) for particular data.

6-15	Reserved for future use.

QDCOUNT an unsigned 16 bit integer specifying the number of entries in the question section.

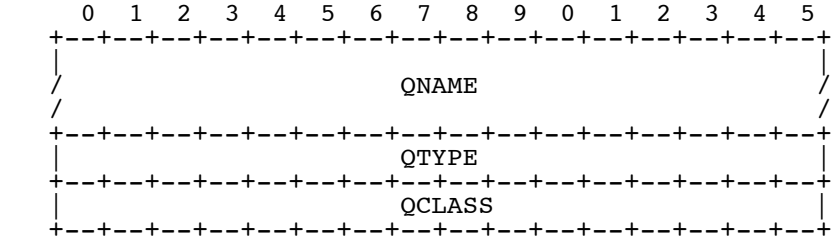
ANCOUNT an unsigned 16 bit integer specifying the number of resource records in the answer section.

NSCOUNT an unsigned 16 bit integer specifying the number of name server resource records in the authority records section.

ARCOUNT an unsigned 16 bit integer specifying the number of resource records in the additional records section.

4.1.2. Question section format

The question section is used to carry the "question" in most queries, i.e., the parameters that define what is being asked. The section contains QDCOUNT (usually 1) entries, each of the following format:



where:

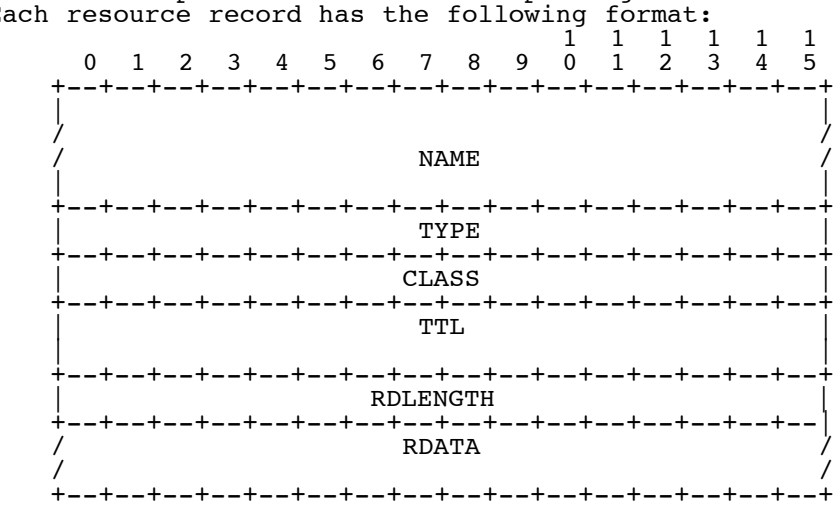
QNAME a domain name represented as a sequence of labels, where each label consists of a length octet followed by that number of octets. The domain name terminates with the zero length octet for the null label of the root. Note that this field may be an odd number of octets; no padding is used.

QTYPE a two octet code which specifies the type of the query. The values for this field include all codes valid for a TYPE field, together with some more general codes which can match more than one type of RR.

QCLASS a two octet code that specifies the class of the query. For example, the QCLASS field is IN for the Internet.

4.1.3. Resource record format

The answer, authority, and additional sections all share the same format: a variable number of resource records, where the number of records is specified in the corresponding count field in the header. Each resource record has the following format:



where:

NAME a domain name to which this resource record pertains.

TYPE two octets containing one of the RR type codes. This field specifies the meaning of the data in the RDATA field.

CLASS two octets which specify the class of the data in the RDATA field.

TTL a 32 bit unsigned integer that specifies the time

interval (in seconds) that the resource record may be cached before it should be discarded. Zero values are interpreted to mean that the RR can only be used for the transaction in progress, and should not be cached.

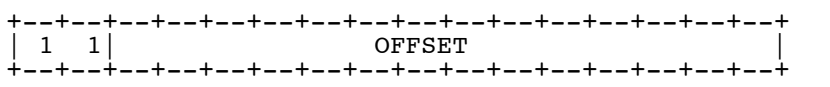
RDLENGTH an unsigned 16 bit integer that specifies the length in octets of the RDATA field.

RDATA a variable length string of octets that describes the resource. The format of this information varies according to the TYPE and CLASS of the resource record. For example, the if the TYPE is A and the CLASS is IN, the RDATA field is a 4 octet ARPA Internet address.

4.1.4. Message compression

In order to reduce the size of messages, the domain system utilizes a compression scheme which eliminates the repetition of domain names in a message. In this scheme, an entire domain name or a list of labels at the end of a domain name is replaced with a pointer to a prior occurance of the same name.

The pointer takes the form of a two octet sequence:



The first two bits are ones. This allows a pointer to be distinguished from a label, since the label must begin with two zero bits because labels are restricted to 63 octets or less. (The 10 and 01 combinations are reserved for future use.) The OFFSET field specifies an offset from the start of the message (i.e., the first octet of the ID field in the domain header). A zero offset specifies the first byte of the ID field, etc.

The compression scheme allows a domain name in a message to be represented as either:

- a sequence of labels ending in a zero octet
- a pointer
- a sequence of labels ending with a pointer

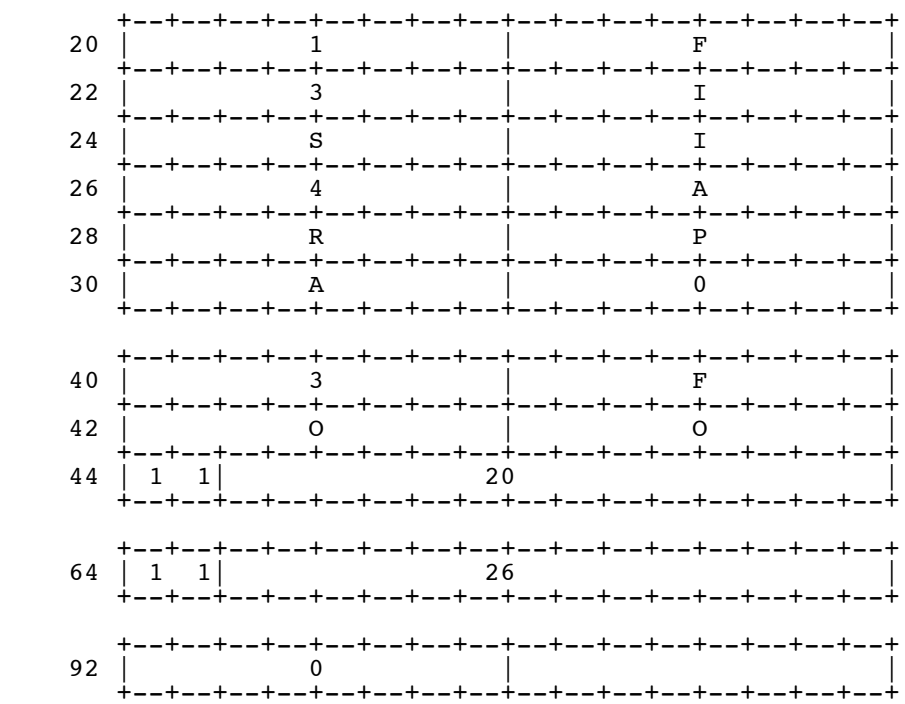
Pointers can only be used for occurances of a domain name where the format is not class specific. If this were not the case, a name server or resolver would be required to know the format of all RRs it handled. As yet, there are no such cases, but they may occur in future RDATA formats.

If a domain name is contained in a part of the message subject to a length field (such as the RDATA section of an RR), and compression is

used, the length of the compressed name is used in the length calculation, rather than the length of the expanded name.

Programs are free to avoid using pointers in messages they generate, although this will reduce datagram capacity, and may cause truncation. However all programs are required to understand arriving messages that contain pointers.

For example, a datagram might need to use the domain names F.ISI.ARPA, FOO.F.ISI.ARPA, ARPA, and the root. Ignoring the other fields of the message, these domain names might be represented as:



The domain name for F.ISI.ARPA is shown at offset 20. The domain name FOO.F.ISI.ARPA is shown at offset 40; this definition uses a pointer to concatenate a label for FOO to the previously defined F.ISI.ARPA. The domain name ARPA is defined at offset 64 using a pointer to the ARPA component of the name F.ISI.ARPA at 20; note that this pointer relies on ARPA being the last label in the string at 20. The root domain name is

defined by a single octet of zeros at 92; the root domain name has no labels.

4.2. Transport

The DNS assumes that messages will be transmitted as datagrams or in a byte stream carried by a virtual circuit. While virtual circuits can be used for any DNS activity, datagrams are preferred for queries due to their lower overhead and better performance. Zone refresh activities must use virtual circuits because of the need for reliable transfer.

The Internet supports name server access using TCP [RFC-793] on server port 53 (decimal) as well as datagram access using UDP [RFC-768] on UDP port 53 (decimal).

4.2.1. UDP usage

Messages sent using UDP user server port 53 (decimal).

Messages carried by UDP are restricted to 512 bytes (not counting the IP or UDP headers). Longer messages are truncated and the TC bit is set in the header.

UDP is not acceptable for zone transfers, but is the recommended method for standard queries in the Internet. Queries sent using UDP may be lost, and hence a retransmission strategy is required. Queries or their responses may be reordered by the network, or by processing in name servers, so resolvers should not depend on them being returned in order.

The optimal UDP retransmission policy will vary with performance of the Internet and the needs of the client, but the following are recommended:

- The client should try other servers and server addresses before repeating a query to a specific address of a server.

- The retransmission interval should be based on prior

statistics if possible. Too aggressive retransmission can easily slow responses for the community at large. Depending on how well connected the client is to its expected servers, the minimum retransmission interval should be 2-5 seconds.

More suggestions on server selection and retransmission policy can be found in the resolver section of this memo.

4.2.2. TCP usage

Messages sent over TCP connections use server port 53 (decimal). The message is prefixed with a two byte length field which gives the message

length, excluding the two byte length field. This length field allows the low-level processing to assemble a complete message before beginning to parse it.

Several connection management policies are recommended:

- The server should not block other activities waiting for TCP data.
- The server should support multiple connections.
- The server should assume that the client will initiate connection closing, and should delay closing its end of the connection until all outstanding client requests have been satisfied.
- If the server needs to close a dormant connection to reclaim resources, it should wait until the connection has been idle for a period on the order of two minutes. In particular, the server should allow the SOA and AXFR request sequence (which begins a refresh operation) to be made on a single connection. Since the server would be unable to answer queries anyway, a unilateral close or reset may be used instead of a graceful close.

5. MASTER FILES

Master files are text files that contain RRs in text form. Since the contents of a zone can be expressed in the form of a list of RRs a master file is most often used to define a zone, though it can be used to list a cache's contents. Hence, this section first discusses the format of RRs in a master file, and then the special considerations when a master file is used to create a zone in some name server.

5.1. Format

The format of these files is a sequence of entries. Entries are predominantly line-oriented, though parentheses can be used to continue a list of items across a line boundary, and text literals can contain CRLF within the text. Any combination of tabs and spaces act as a delimiter between the separate items that make up an entry. The end of any line in the master file can end with a comment. The comment starts with a ";" (semicolon).

The following entries are defined:

```
<blank>[<comment>]
```



```
$ORIGIN <domain-name> [<comment>]

$INCLUDE <file-name> [<domain-name>] [<comment>]

<domain-name><rr> [<comment>]
```

```
    <blank><rr> [<comment>]
```

Blank lines, with or without comments, are allowed anywhere in the file.

Two control entries are defined: $ORIGIN and $INCLUDE. $ORIGIN is followed by a domain name, and resets the current origin for relative domain names to the stated name. $INCLUDE inserts the named file into the current file, and may optionally specify a domain name that sets the relative domain name origin for the included file. $INCLUDE may also have a comment. Note that a $INCLUDE entry never changes the relative origin of the parent file, regardless of changes to the relative origin made within the included file.

The last two forms represent RRs. If an entry for an RR begins with a blank, then the RR is assumed to be owned by the last stated owner. If an RR entry begins with a <domain-name>, then the owner name is reset.

<rr> contents take one of the following forms:

```
    [<TTL>] [<class>] <type> <RDATA>

    [<class>] [<TTL>] <type> <RDATA>
```

The RR begins with optional TTL and class fields, followed by a type and RDATA field appropriate to the type and class. Class and type use the standard mnemonics, TTL is a decimal integer. Omitted class and TTL values are default to the last explicitly stated values. Since type and class mnemonics are disjoint, the parse is unique. (Note that this order is different from the order used in examples and the order used in the actual RRs; the given order allows easier parsing and defaulting.)

<domain-name>s make up a large share of the data in the master file. The labels in the domain name are expressed as character strings and separated by dots. Quoting conventions allow arbitrary characters to be stored in domain names. Domain names that end in a dot are called absolute, and are taken as complete. Domain names which do not end in a dot are called relative; the actual domain name is the concatenation of the relative part with an origin specified in a $ORIGIN, $INCLUDE, or as an argument to the master file loading routine. A relative name is an error when no origin is available.

<character-string> is expressed in one or two ways: as a contiguous set of characters without interior spaces, or as a string beginning with a " and ending with a ". Inside a " delimited string any character can occur, except for a " itself, which must be quoted using \ (back slash).

Because these files are text files several special encodings are necessary to allow arbitrary data to be loaded. In particular:

	of the root.
@	A free standing @ is used to denote the current origin.
\X	where X is any character other than a digit (0-9), is used to quote that character so that its special meaning does not apply. For example, "\." can be used to place a dot character in a label.
\DDD	where each D is a digit is the octet corresponding to the decimal number described by DDD. The resulting octet is assumed to be text and is not checked for special meaning.
( )	Parentheses are used to group data that crosses a line boundary. In effect, line terminations are not recognized within parentheses.
;	Semicolon is used to start a comment; the remainder of the line is ignored.

5.2. Use of master files to define zones

When a master file is used to load a zone, the operation should be suppressed if any errors are encountered in the master file. The rationale for this is that a single error can have widespread consequences. For example, suppose that the RRs defining a delegation have syntax errors; then the server will return authoritative name errors for all names in the subzone (except in the case where the subzone is also present on the server).

Several other validity checks that should be performed in addition to insuring that the file is syntactically correct:

1. All RRs in the file should have the same class.
2. Exactly one SOA RR should be present at the top of the zone.
3. If delegations are present and glue information is required, it should be present.

4. Information present outside of the authoritative nodes in the zone should be glue information, rather than the result of an origin or similar error.

5.3. Master file example

The following is an example file which might be used to define the ISI.EDU zone.and is loaded with an origin of ISI.EDU:

```
@   IN  SOA     VENERA      Action\.domains (
                                20     ; SERIAL
                                7200   ; REFRESH
                                600    ; RETRY
                                3600000; EXPIRE
                                60)    ; MINIMUM

        NS      A.ISI.EDU.
        NS      VENERA
        NS      VAXA
        MX      10      VENERA
        MX      20      VAXA

A       A       26.3.0.103

VENERA  A       10.1.0.52
        A       128.9.0.32

VAXA    A       10.2.0.27
        A       128.9.0.33


$INCLUDE <SUBSYS>ISI-MAILBOXES.TXT
```

Where the file <SUBSYS>ISI-MAILBOXES.TXT is:

```
    MOE     MB      A.ISI.EDU.
    LARRY   MB      A.ISI.EDU.
    CURLEY  MB      A.ISI.EDU.
    STOOGES MG      MOE
            MG      LARRY
            MG      CURLEY
```

Note the use of the \ character in the SOA RR to specify the responsible person mailbox "Action.domains@E.ISI.EDU".

6. NAME SERVER IMPLEMENTATION

6.1. Architecture

The optimal structure for the name server will depend on the host operating system and whether the name server is integrated with resolver operations, either by supporting recursive service, or by sharing its database with a resolver. This section discusses implementation considerations for a name server which shares a database with a resolver, but most of these concerns are present in any name server.

6.1.1. Control

A name server must employ multiple concurrent activities, whether they are implemented as separate tasks in the host's OS or multiplexing inside a single name server program. It is simply not acceptable for a name server to block the service of UDP requests while it waits for TCP data for refreshing or query activities. Similarly, a name server should not attempt to provide recursive service without processing such requests in parallel, though it may choose to serialize requests from a single client, or to regard identical requests from the same client as duplicates. A name server should not substantially delay requests while it reloads a zone from master files or while it incorporates a newly refreshed zone into its database.

6.1.2. Database

While name server implementations are free to use any internal data structures they choose, the suggested structure consists of three major parts:

- A "catalog" data structure which lists the zones available to this server, and a "pointer" to the zone data structure. The main purpose of this structure is to find the nearest ancestor zone, if any, for arriving standard queries.

- Separate data structures for each of the zones held by the name server.

- A data structure for cached data. (or perhaps separate caches for different classes)

All of these data structures can be implemented an identical tree structure format, with different data chained off the nodes in different parts: in the catalog the data is pointers to zones, while in the zone and cache data structures, the data will be RRs. In designing the tree framework the designer should recognize that query processing will need to traverse the tree using case-insensitive label comparisons; and that

in real data, a few nodes have a very high branching factor (100-1000 or more), but the vast majority have a very low branching factor (0-1).

One way to solve the case problem is to store the labels for each node in two pieces: a standardized-case representation of the label where all ASCII characters are in a single case, together with a bit mask that denotes which characters are actually of a different case. The branching factor diversity can be handled using a simple linked list for a node until the branching factor exceeds some threshold, and transitioning to a hash structure after the threshold is exceeded. In any case, hash structures used to store tree sections must insure that hash functions and procedures preserve the casing conventions of the DNS.

The use of separate structures for the different parts of the database is motivated by several factors:

- The catalog structure can be an almost static structure that need change only when the system administrator changes the zones supported by the server. This structure can also be used to store parameters used to control refreshing activities.

- The individual data structures for zones allow a zone to be replaced simply by changing a pointer in the catalog. Zone refresh operations can build a new structure and, when complete, splice it into the database via a simple pointer replacement. It is very important that when a zone is refreshed, queries should not use old and new data simultaneously.

- With the proper search procedures, authoritative data in zones will always "hide", and hence take precedence over, cached data.

- Errors in zone definitions that cause overlapping zones, etc., may cause erroneous responses to queries, but problem determination is simplified, and the contents of one "bad" zone can't corrupt another.

- Since the cache is most frequently updated, it is most vulnerable to corruption during system restarts. It can also become full of expired RR data. In either case, it can easily be discarded without disturbing zone data.

A major aspect of database design is selecting a structure which allows the name server to deal with crashes of the name server's host. State information which a name server should save across system crashes

includes the catalog structure (including the state of refreshing for each zone) and the zone data itself.

6.1.3. Time

Both the TTL data for RRs and the timing data for refreshing activities depends on 32 bit timers in units of seconds. Inside the database, refresh timers and TTLs for cached data conceptually "count down", while data in the zone stays with constant TTLs.

A recommended implementation strategy is to store time in two ways: as a relative increment and as an absolute time. One way to do this is to use positive 32 bit numbers for one type and negative numbers for the other. The RRs in zones use relative times; the refresh timers and cache data use absolute times. Absolute numbers are taken with respect to some known origin and converted to relative values when placed in the response to a query. When an absolute TTL is negative after conversion to relative, then the data is expired and should be ignored.

6.2. Standard query processing

The major algorithm for standard query processing is presented in [RFC-1034].

When processing queries with QCLASS=*, or some other QCLASS which matches multiple classes, the response should never be authoritative unless the server can guarantee that the response covers all classes.

When composing a response, RRs which are to be inserted in the additional section, but duplicate RRs in the answer or authority sections, may be omitted from the additional section.

When a response is so long that truncation is required, the truncation should start at the end of the response and work forward in the datagram. Thus if there is any data for the authority section, the answer section is guaranteed to be unique.

The MINIMUM value in the SOA should be used to set a floor on the TTL of data distributed from a zone. This floor function should be done when the data is copied into a response. This will allow future dynamic update protocols to change the SOA MINIMUM field without ambiguous semantics.

6.3. Zone refresh and reload processing

In spite of a server's best efforts, it may be unable to load zone data from a master file due to syntax errors, etc., or be unable to refresh a zone within the its expiration parameter. In this case, the name server

should answer queries as if it were not supposed to possess the zone.

If a master is sending a zone out via AXFR, and a new version is created during the transfer, the master should continue to send the old version if possible. In any case, it should never send part of one version and part of another. If completion is not possible, the master should reset the connection on which the zone transfer is taking place.

6.4. Inverse queries (Optional)

Inverse queries are an optional part of the DNS. Name servers are not required to support any form of inverse queries. If a name server receives an inverse query that it does not support, it returns an error response with the "Not Implemented" error set in the header. While inverse query support is optional, all name servers must be at least able to return the error response.

6.4.1. The contents of inverse queries and responses

Inverse queries reverse the mappings performed by standard query operations; while a standard query maps a domain name to a resource, an inverse query maps a resource to a domain name. For example, a standard query might bind a domain name to a host address; the corresponding inverse query binds the host address to a domain name.

Inverse queries take the form of a single RR in the answer section of the message, with an empty question section. The owner name of the query RR and its TTL are not significant. The response carries questions in the question section which identify all names possessing the query RR WHICH THE NAME SERVER KNOWS. Since no name server knows about all of the domain name space, the response can never be assumed to be complete. Thus inverse queries are primarily useful for database management and debugging activities. Inverse queries are NOT an acceptable method of mapping host addresses to host names; use the IN-ADDR.ARPA domain instead.

Where possible, name servers should provide case-insensitive comparisons for inverse queries. Thus an inverse query asking for an MX RR of "Venera.isi.edu" should get the same response as a query for "VENERA.ISI.EDU"; an inverse query for HINFO RR "IBM-PC UNIX" should produce the same result as an inverse query for "IBM-pc unix". However, this cannot be guaranteed because name servers may possess RRs that contain character strings but the name server does not know that the data is character.

When a name server processes an inverse query, it either returns:

1. zero, one, or multiple domain names for the specified resource as QNAMEs in the question section

2. an error code indicating that the name server doesn't support inverse mapping of the specified resource type.

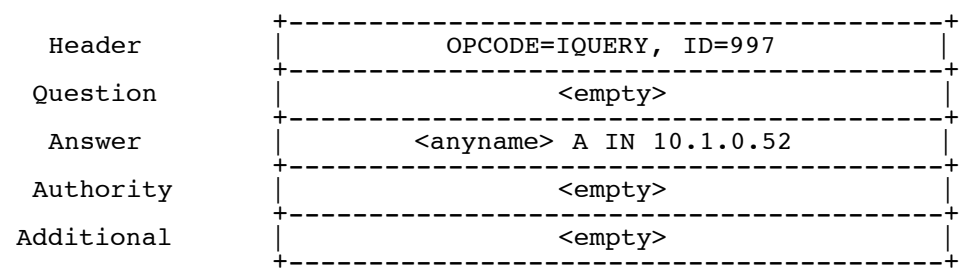
When the response to an inverse query contains one or more QNAMEs, the owner name and TTL of the RR in the answer section which defines the inverse query is modified to exactly match an RR found at the first QNAME.

RRs returned in the inverse queries cannot be cached using the same mechanism as is used for the replies to standard queries. One reason for this is that a name might have multiple RRs of the same type, and only one would appear. For example, an inverse query for a single address of a multiply homed host might create the impression that only one address existed.

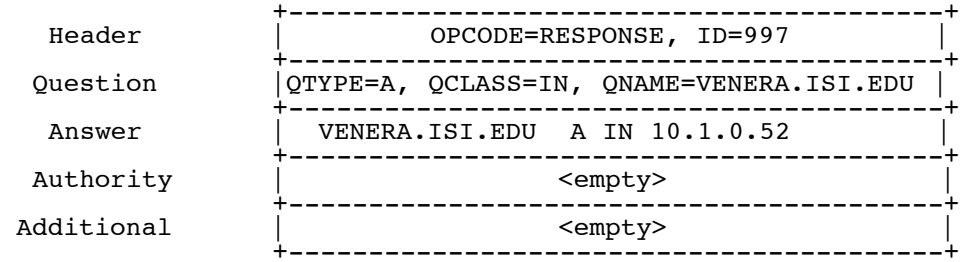
6.4.2. Inverse query and response example

The overall structure of an inverse query for retrieving the domain name that corresponds to

Internet address 10.1.0.52 is shown below:



This query asks for a question whose answer is the Internet style address 10.1.0.52. Since the owner name is not known, any domain name can be used as a placeholder (and is ignored). A single octet of zero, signifying the root, is usually used because it minimizes the length of the message. The TTL of the RR is not significant. The response to this query might be:



Note that the QTYPE in a response to an inverse query is the same as the TYPE field in the answer section of the inverse query. Responses to inverse queries may contain multiple questions when the inverse is not unique. If the question section in the response is not empty, then the RR in the answer section is modified to correspond to be an exact copy of an RR at the first QNAME.

6.4.3. Inverse query processing

Name servers that support inverse queries can support these operations through exhaustive searches of their databases, but this becomes impractical as the size of the database increases. An alternative approach is to invert the database according to the search key.

For name servers that support multiple zones and a large amount of data, the recommended approach is separate inversions for each zone. When a particular zone is changed during a refresh, only its inversions need to be redone.

Support for transfer of this type of inversion may be included in future versions of the domain system, but is not supported in this version.

6.5. Completion queries and responses

The optional completion services described in RFC-882 and RFC-883 have been deleted. Redesigned services may become available in the future.

7. RESOLVER IMPLEMENTATION

The top levels of the recommended resolver algorithm are discussed in [RFC-1034]. This section discusses implementation details assuming the database structure suggested in the name server implementation section of this memo.

7.1. Transforming a user request into a query

The first step a resolver takes is to transform the client's request, stated in a format suitable to the local OS, into a search specification for RRs at a specific name which match a specific QTYPE and QCLASS. Where possible, the QTYPE and QCLASS should correspond to a single type and a single class, because this makes the use of cached data much simpler. The reason for this is that the presence of data of one type in a cache doesn't confirm the existence or non-existence of data of other types, hence the only way to be sure is to consult an authoritative source. If QCLASS=* is used, then authoritative answers won't be available.

Since a resolver must be able to multiplex multiple requests if it is to perform its function efficiently, each pending request is usually represented in some block of state information. This state block will typically contain:

- A timestamp indicating the time the request began. The timestamp is used to decide whether RRs in the database can be used or are out of date. This timestamp uses the absolute time format previously discussed for RR storage in zones and caches. Note that when an RRs TTL indicates a relative time, the RR must be timely, since it is part of a zone. When the RR has an absolute time, it is part of a cache, and the TTL of the RR is compared against the timestamp for the start of the request.

  Note that using the timestamp is superior to using a current time, since it allows RRs with TTLs of zero to be entered in the cache in the usual manner, but still used by the current request, even after intervals of many seconds due to system load, query retransmission timeouts, etc.

- Some sort of parameters to limit the amount of work which will be performed for this request.

  The amount of work which a resolver will do in response to a client request must be limited to guard against errors in the database, such as circular CNAME references, and operational problems, such as network partition which prevents the

  resolver from accessing the name servers it needs. While local limits on the number of times a resolver will retransmit a particular query to a particular name server address are essential, the resolver should have a global per-request counter to limit work on a single request. The counter should be set to some initial value and decremented whenever the resolver performs any action (retransmission timeout, retransmission, etc.) If the counter passes zero, the request is terminated with a temporary error.

  Note that if the resolver structure allows one request to

start others in parallel, such as when the need to access a name server for one request causes a parallel resolve for the name server's addresses, the spawned request should be started with a lower counter. This prevents circular references in the database from starting a chain reaction of resolver activity.

- The SLIST data structure discussed in [RFC-1034].

  This structure keeps track of the state of a request if it must wait for answers from foreign name servers.

7.2. Sending the queries

As described in [RFC-1034], the basic task of the resolver is to formulate a query which will answer the client's request and direct that query to name servers which can provide the information. The resolver will usually only have very strong hints about which servers to ask, in the form of NS RRs, and may have to revise the query, in response to CNAMEs, or revise the set of name servers the resolver is asking, in response to delegation responses which point the resolver to name servers closer to the desired information. In addition to the information requested by the client, the resolver may have to call upon its own services to determine the address of name servers it wishes to contact.

In any case, the model used in this memo assumes that the resolver is multiplexing attention between multiple requests, some from the client, and some internally generated. Each request is represented by some state information, and the desired behavior is that the resolver transmit queries to name servers in a way that maximizes the probability that the request is answered, minimizes the time that the request takes, and avoids excessive transmissions. The key algorithm uses the state information of the request to select the next name server address to query, and also computes a timeout which will cause the next action should a response not arrive. The next action will usually be a transmission to some other server, but may be a temporary error to the

client.

The resolver always starts with a list of server names to query (SLIST). This list will be all NS RRs which correspond to the nearest ancestor zone that the resolver knows about. To avoid startup problems, the resolver should have a set of default servers which it will ask should it have no current NS RRs which are appropriate. The resolver then adds to SLIST all of the known addresses for the name servers, and may start parallel requests to acquire the addresses of the servers when the resolver has the name, but no addresses, for the name servers.

To complete initialization of SLIST, the resolver attaches whatever history information it has to the each address in SLIST. This will usually consist of some sort of weighted averages for the response time of the address, and the batting average of the address (i.e., how often the address responded at all to the request). Note that this information should be kept on a per address basis, rather than on a per name server basis, because the response time and batting average of a particular server may vary considerably from address to address. Note also that this information is actually specific to a resolver address / server address pair, so a resolver with multiple addresses may wish to keep separate histories for each of its addresses. Part of this step must deal with addresses which have no such history; in this case an expected round trip time of 5-10 seconds should be the worst case, with lower estimates for the same local network, etc.

Note that whenever a delegation is followed, the resolver algorithm reinitializes SLIST.

The information establishes a partial ranking of the available name server addresses. Each time an address is chosen and the state should be altered to prevent its selection again until all other addresses have been tried. The timeout for each transmission should be 50-100% greater than the average predicted value to allow for variance in response.

Some fine points:

- The resolver may encounter a situation where no addresses are available for any of the name servers named in SLIST, and where the servers in the list are precisely those which would normally be used to look up their own addresses. This situation typically occurs when the glue address RRs have a smaller TTL than the NS RRs marking delegation, or when the resolver caches the result of a NS search. The resolver should detect this condition and restart the search at the next ancestor zone, or alternatively at the root.

- If a resolver gets a server error or other bizarre response from a name server, it should remove it from SLIST, and may wish to schedule an immediate transmission to the next candidate server address.

7.3. Processing responses

The first step in processing arriving response datagrams is to parse the response. This procedure should include:

- Check the header for reasonableness. Discard datagrams which are queries when responses are expected.

- Parse the sections of the message, and insure that all RRs are correctly formatted.

- As an optional step, check the TTLs of arriving data looking for RRs with excessively long TTLs. If a RR has an excessively long TTL, say greater than 1 week, either discard the whole response, or limit all TTLs in the response to 1 week.

The next step is to match the response to a current resolver request. The recommended strategy is to do a preliminary matching using the ID field in the domain header, and then to verify that the question section corresponds to the information currently desired. This requires that the transmission algorithm devote several bits of the domain ID field to a request identifier of some sort. This step has several fine points:

- Some name servers send their responses from different addresses than the one used to receive the query. That is, a resolver cannot rely that a response will come from the same address which it sent the corresponding query to. This name server bug is typically encountered in UNIX systems.

- If the resolver retransmits a particular request to a name server it should be able to use a response from any of the transmissions. However, if it is using the response to sample the round trip time to access the name server, it must be able to determine which transmission matches the response (and keep transmission times for each outgoing message), or only calculate round trip times based on initial transmissions.

- A name server will occasionally not have a current copy of a zone which it should have according to some NS RRs. The resolver should simply remove the name server from the current SLIST, and continue.

7.4. Using the cache

In general, we expect a resolver to cache all data which it receives in responses since it may be useful in answering future client requests. However, there are several types of data which should not be cached:

- When several RRs of the same type are available for a particular owner name, the resolver should either cache them all or none at all. When a response is truncated, and a resolver doesn't know whether it has a complete set, it should not cache a possibly partial set of RRs.

- Cached data should never be used in preference to authoritative data, so if caching would cause this to happen the data should not be cached.

- The results of an inverse query should not be cached.

- The results of standard queries where the QNAME contains "*" labels if the data might be used to construct wildcards. The reason is that the cache does not necessarily contain existing RRs or zone boundary information which is necessary to restrict the application of the wildcard RRs.

- RR data in responses of dubious reliability. When a resolver receives unsolicited responses or RR data other than that requested, it should discard it without caching it. The basic implication is that all sanity checks on a packet should be performed before any of it is cached.

In a similar vein, when a resolver has a set of RRs for some name in a response, and wants to cache the RRs, it should check its cache for already existing RRs. Depending on the circumstances, either the data in the response or the cache is preferred, but the two should never be combined. If the data in the response is from authoritative data in the answer section, it is always preferred.

8. MAIL SUPPORT

The domain system defines a standard for mapping mailboxes into domain names, and two methods for using the mailbox information to derive mail routing information. The first method is called mail exchange binding and the other method is mailbox binding. The mailbox encoding standard and mail exchange binding are part of the DNS official protocol, and are the recommended method for mail routing in the Internet. Mailbox binding is an experimental feature which is still under development and subject to change.

The mailbox encoding standard assumes a mailbox name of the form "<local-part>@<mail-domain>". While the syntax allowed in each of these sections varies substantially between the various mail internets, the preferred syntax for the ARPA Internet is given in [RFC-822].

The DNS encodes the <local-part> as a single label, and encodes the <mail-domain> as a domain name. The single label from the <local-part> is prefaced to the domain name from <mail-domain> to form the domain name corresponding to the mailbox. Thus the mailbox HOSTMASTER@SRI-NIC.ARPA is mapped into the domain name HOSTMASTER.SRI-NIC.ARPA. If the <local-part> contains dots or other special characters, its representation in a master file will require the use of backslash quoting to ensure that the domain name is properly encoded. For example, the mailbox Action.domains@ISI.EDU would be represented as Action\.domains.ISI.EDU.

8.1. Mail exchange binding

Mail exchange binding uses the <mail-domain> part of a mailbox specification to determine where mail should be sent. The <local-part> is not even consulted. [RFC-974] specifies this method in detail, and should be consulted before attempting to use mail exchange support.

One of the advantages of this method is that it decouples mail destination naming from the hosts used to support mail service, at the cost of another layer of indirection in the lookup function. However, the addition layer should eliminate the need for complicated "%", "!", etc encodings in <local-part>.

The essence of the method is that the <mail-domain> is used as a domain

name to locate type MX RRs which list hosts willing to accept mail for <mail-domain>, together with preference values which rank the hosts according to an order specified by the administrators for <mail-domain>.

In this memo, the <mail-domain> ISI.EDU is used in examples, together with the hosts VENERA.ISI.EDU and VAXA.ISI.EDU as mail exchanges for ISI.EDU. If a mailer had a message for Mockapetris@ISI.EDU, it would route it by looking up MX RRs for ISI.EDU. The MX RRs at ISI.EDU name VENERA.ISI.EDU and VAXA.ISI.EDU, and type A queries can find the host addresses.

8.2. Mailbox binding (Experimental)

In mailbox binding, the mailer uses the entire mail destination specification to construct a domain name. The encoded domain name for the mailbox is used as the QNAME field in a QTYPE=MAILB query.

Several outcomes are possible for this query:

1. The query can return a name error indicating that the mailbox does not exist as a domain name.

   In the long term, this would indicate that the specified mailbox doesn't exist. However, until the use of mailbox binding is universal, this error condition should be interpreted to mean that the organization identified by the global part does not support mailbox binding. The appropriate procedure is to revert to exchange binding at this point.

2. The query can return a Mail Rename (MR) RR.

   The MR RR carries new mailbox specification in its RDATA field. The mailer should replace the old mailbox with the new one and retry the operation.

3. The query can return a MB RR.

   The MB RR carries a domain name for a host in its RDATA field. The mailer should deliver the message to that host via whatever protocol is applicable, e.g., b,SMTP.

4. The query can return one or more Mail Group (MG) RRs.

   This condition means that the mailbox was actually a mailing list or mail group, rather than a single mailbox. Each MG RR has a RDATA field that identifies a mailbox that is a member of the group. The mailer should deliver a copy of the message to each member.

5. The query can return a MB RR as well as one or more MG RRs.

   This condition means the the mailbox was actually a mailing list. The mailer can either deliver the message to the host specified by the MB RR, which will in turn do the delivery to all members, or the mailer can use the MG RRs to do the expansion itself.

In any of these cases, the response may include a Mail Information (MINFO) RR. This RR is usually associated with a mail group, but is legal with a MB. The MINFO RR identifies two mailboxes. One of these identifies a responsible person for the original mailbox name. This mailbox should be used for requests to be added to a mail group, etc. The second mailbox name in the MINFO RR identifies a mailbox that should receive error messages for mail failures. This is particularly appropriate for mailing lists when errors in member names should be reported to a person other than the one who sends a message to the list.

New fields may be added to this RR in the future.

9. REFERENCES and BIBLIOGRAPHY

[Dyer 87] S. Dyer, F. Hsu, "Hesiod", Project Athena Technical Plan - Name Service, April 1987, version 1.9.

Describes the fundamentals of the Hesiod name service.

[IEN-116] J. Postel, "Internet Name Server", IEN-116, USC/Information Sciences Institute, August 1979.

A name service obsoleted by the Domain Name System, but still in use.

[Quarterman 86] J. Quarterman, and J. Hoskins, "Notable Computer Networks", Communications of the ACM, October 1986, volume 29, number 10.

[RFC-742] K. Harrenstien, "NAME/FINGER", RFC-742, Network Information Center, SRI International, December 1977.

[RFC-768] J. Postel, "User Datagram Protocol", RFC-768, USC/Information Sciences Institute, August 1980.

[RFC-793] J. Postel, "Transmission Control Protocol", RFC-793, USC/Information Sciences Institute, September 1981.

[RFC-799] D. Mills, "Internet Name Domains", RFC-799, COMSAT, September 1981.

Suggests introduction of a hierarchy in place of a flat name space for the Internet.

[RFC-805] J. Postel, "Computer Mail Meeting Notes", RFC-805, USC/Information Sciences Institute, February 1982.

[RFC-810] E. Feinler, K. Harrenstien, Z. Su, and V. White, "DOD Internet Host Table Specification", RFC-810, Network Information Center, SRI International, March 1982.

Obsolete. See RFC-952.

[RFC-811] K. Harrenstien, V. White, and E. Feinler, "Hostnames Server", RFC-811, Network Information Center, SRI International, March 1982.

Obsolete. See RFC-953.

[RFC-812] K. Harrenstien, and V. White, "NICNAME/WHOIS", RFC-812, Network Information Center, SRI International, March 1982.

[RFC-819] Z. Su, and J. Postel, "The Domain Naming Convention for Internet User Applications", RFC-819, Network Information Center, SRI International, August 1982.

Early thoughts on the design of the domain system. Current implementation is completely different.

[RFC-821] J. Postel, "Simple Mail Transfer Protocol", RFC-821, USC/Information Sciences Institute, August 1980.

[RFC-830] Z. Su, "A Distributed System for Internet Name Service", RFC-830, Network Information Center, SRI International, October 1982.

Early thoughts on the design of the domain system. Current implementation is completely different.

[RFC-882] P. Mockapetris, "Domain names - Concepts and

Facilities," RFC-882, USC/Information Sciences Institute, November 1983.

Superceeded by this memo.

[RFC-883] P. Mockapetris, "Domain names - Implementation and Specification," RFC-883, USC/Information Sciences Institute, November 1983.

Superceeded by this memo.

[RFC-920] J. Postel and J. Reynolds, "Domain Requirements", RFC-920, USC/Information Sciences Institute, October 1984.

Explains the naming scheme for top level domains.

[RFC-952] K. Harrenstien, M. Stahl, E. Feinler, "DoD Internet Host Table Specification", RFC-952, SRI, October 1985.

Specifies the format of HOSTS.TXT, the host/address table replaced by the DNS.

[RFC-953] K. Harrenstien, M. Stahl, E. Feinler, "HOSTNAME Server", RFC-953, SRI, October 1985.

This RFC contains the official specification of the hostname server protocol, which is obsoleted by the DNS. This TCP based protocol accesses information stored in the RFC-952 format, and is used to obtain copies of the host table.

[RFC-973] P. Mockapetris, "Domain System Changes and Observations", RFC-973, USC/Information Sciences Institute, January 1986.

Describes changes to RFC-882 and RFC-883 and reasons for them.

[RFC-974] C. Partridge, "Mail routing and the domain system", RFC-974, CSNET CIC BBN Labs, January 1986.

Describes the transition from HOSTS.TXT based mail addressing to the more powerful MX system used with the domain system.

[RFC-1001] NetBIOS Working Group, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and Methods", RFC-1001, March 1987.

This RFC and RFC-1002 are a preliminary design for NETBIOS on top of TCP/IP which proposes to base NetBIOS name service on top of the DNS.

[RFC-1002] NetBIOS Working Group, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Detailed Specifications", RFC-1002, March 1987.

[RFC-1010] J. Reynolds, and J. Postel, "Assigned Numbers", RFC-1010, USC/Information Sciences Institute, May 1987.

Contains socket numbers and mnemonics for host names, operating systems, etc.

[RFC-1031] W. Lazear, "MILNET Name Domain Transition", RFC-1031, November 1987.

Describes a plan for converting the MILNET to the DNS.

[RFC-1032] M. Stahl, "Establishing a Domain - Guidelines for Administrators", RFC-1032, November 1987.

Describes the registration policies used by the NIC to administer the top level domains and delegate subzones.

[RFC-1033] M. Lottor, "Domain Administrators Operations Guide", RFC-1033, November 1987.

A cookbook for domain administrators.

[Solomon 82] M. Solomon, L. Landweber, and D. Neuhengen, "The CSNET Name Server", Computer Networks, vol 6, nr 3, July 1982.

Describes a name service for CSNET which is independent from the DNS and DNS use in the CSNET.

Index

EXHIBIT N

TO MICHAEL FRATTO'S DECLARATION

REED, "PROXIES FOR ANONYMOUS ROUTING" (1996)

Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. *Proxies for Anonymous Routing*, 12th Annual Computer Security Applications Conference, San Diego, CA, December 9-13, 1996.

Proxies for Anonymous Routing

Michael G. Reed, Paul F. Syverson, and David M. Goldschlag

Naval Research Laboratory

Center for High Assurance Computer Systems

Washington, DC 20375-5337

Phone: +1 202.767.2389 (voice)

Fax: +1 202.404.7942 (fax)

e-mail: {reed, syverson, goldschlag}@itd.nrl.navy.mil

Abstract

Using traffic analysis, it is possible to infer who is talking to whom over a public network. This paper describes a flexible communications infrastructure, onion routing, which is resistant to traffic analysis. Onion routing lives just beneath the application layer, and is designed to interface with a wide variety of unmodified Internet services by means of proxies. Onion routing has been implemented on Sun Solaris 2.4; in addition, proxies for World Wide Web browsing (HTTP), remote logins (RLOGIN), e-mail (SMTP), and file transfers (FTP) have been implemented.

Onion routing provides application independent, real-time, and bi-directional anonymous connections that are resistant to both eavesdropping and traffic analysis. Applications making use of onion routing's anonymous connections may (and usually should) identify their users over the anonymous connection. User anonymity may be layered on top of the anonymous connections by removing identifying information from the data stream. Our goal here is anonymous connections, not anonymous communication. The use of a packet switched public network should not automatically reveal who is talking to whom. This is the traffic analysis that onion routing complicates.

1. Introduction

1.1 The Problem

Using traffic analysis, it is possible to infer who is talking to whom over a public network (Figure 1). For example, in a packet switched network [11], packets have a header used for routing, and a payload that carries the data. The header, which must be visible to the network (and to observers of the network), reveals the source and destination of the packet. Even if the header were obscured in some way, the packet could still be tracked as it moves through the network. Encrypting the payload is similarly ineffective, because the goal of traffic analysis is to identify who is talking to whom and not (to identify directly) the content of that conversation.

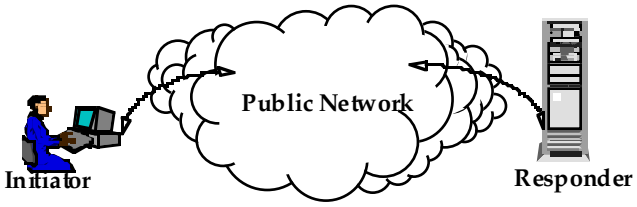


Figure 1. Communication over a Public Network

The efficiencies of the public Internet are strong motivation for companies to use it instead of private intranets. However, these companies may want to protect their interests. For example, a researcher using the World Wide Web (Web) may expect his particular focus to remain private, and inter-company collaborations should be confidential. Individuals may wish to protect their

privacy as well. For example, the sending of e-mail should keep the identities of the sender and recipient hidden from observers. Also, a person shopping online may not want his visits tracked. Certainly someone spending anonymous e-cash would expect that the source of the e-cash be untraceable.

The use of a packet switched public network should not require revealing who is talking to whom. This paper presents a flexible communications infrastructure, *onion routing*, which is resistant to traffic analysis.

1.2 Objective

Onion routing is an infrastructure that

- complicates traffic analysis,
- separates identification from routing,
- supports many different applications.

Without dedicated links between every node and full utilization of each link, traffic analysis can, in principle, always be effective. But traffic analysis can be made more costly. Onion routing accomplishes this goal by separating identification from routing. Onion routing provides *anonymous connections* that are resistant to both eavesdropping and traffic analysis. Instead of containing source and destination information, packets moving along an anonymous connection contain only next hop and previous hop information. These anonymous connections can replace socket connections. Since socket connections are commonly used to support applications running over the Internet (like Web browsers, remote login, and e-mail) onion routing's anonymous connections can support a wide variety of unmodified applications using *proxies* that interface between applications and the onion routing network.

1.3 Overview of the Solution

Onion routing works in the following way: An application, instead of making a (socket) connection directly to a destination machine, makes a connection to an *onion routing proxy* on some remote machine. That onion routing proxy builds an anonymous connection through several other *onion routers* to the destination. Each onion router can only identify adjacent onion routers along the route. When the connection is broken, even this limited information about the connection is cleared at each onion router. Data passed along the anonymous connection appears different *at* and *to* each onion router, so data cannot be tracked en route and compromised onion routers cannot cooperate. An onion routing network can exist in several configurations that permit efficient usage by both large institutions and individuals.

1.4 Traffic Analysis

Traffic analysis makes inferences from three sources of information:

- Routing information
- Coincidences
- Load

Routing information is available in many forms: packet headers, phone touch-tones, and envelope addresses. This is the most obvious source that needs protecting. Coincidences, like similar traffic entering or leaving a node, or connections opening or closing at roughly the same time, are more difficult to hide. Finally, the very presence of communication over some link may reveal sensitive information. But load is very difficult to obscure if one is unwilling to use a constant amount of capacity all the time.

1.5 Organization of Paper

This paper is organized in the following way: Section 2 presents background information. Section 3 presents our goals and threat model. Section 4 presents our solution, and sections 5 and 6 provide more details. Section 7 describes the implemented prototype. Section 8 discusses vulnerabilities, costs, and variants of onion routing. Section 9 presents some concluding remarks.

2. Background

Chaum [1,2] defines a mechanism for routing data through intermediate nodes, called *mixes*. These intermediate nodes may reorder, delay, and pad traffic to complicate traffic analysis. Our onion routers are based upon mixes.

Anonymous Remailers [4,6] use mixes to provide anonymous e-mail services and also invent an address through which mail can be forwarded back to the original sender. Remailers work in a store-and-forward manner at the mail application layer by stripping off headers at each mix and forwarding the mail message to the next mix. Some remailers provide confirmation of delivery.

In [8,9], mixes are used to provide untraceable communication in an ISDN network. In the described phone system, each telephone line is assigned to a particular local switch (i.e., local exchange), and switches

are interconnected by a (long distance) network. Anonymous calls in ISDN rely upon an anonymous connection within each switch between the caller and the long distance network, which is obtained by routing calls through a predefined series of mixes. The long distance endpoints of the connection are then mated to complete the call. (Notice that observers can tell which local switches are connected.) This approach relies upon two unique features of ISDN switches. Since each phone line has a subset of the switch's total capacity pre-allocated to it, there is no (real) cost associated with keeping a phone line active all the time, either by making calls to itself, to other phone lines on the same switch, or to the long distance network. Keeping phone lines active complicates traffic analysis because an observer cannot track coincidences.

Also, since each phone line has a control circuit connection to the switch, the switch can broadcast messages to each line using these control circuits. So, within a switch a truly anonymous connection can be established: a phone line makes an anonymous connection to some mix. That mix broadcasts a token identifying itself and the connection. A recipient of that token can make another anonymous connection to the specified mix, which mates the two connections to complete the call.

Our goal of anonymous connections over the Internet differs from anonymous remailers and anonymous ISDN. Unlike anonymous remailers, anonymous connections are application independent and are meant to be used by a wide variety of Internet applications. The data carried by anonymous connections is varied, with real-time constraints often more severe than mail, but usually somewhat looser than voice. Both Web and ISDN connections are bi-directional, but, unlike ISDN, Web connections are likely to be small requests followed by short bursts of returned data. In a local switch, capacity is pre-allocated to each phone line, and broadcasting is efficient. But broadcasting over the Internet is not free, and defining broadcast domains is not trivial. Most importantly, the network topology of the Internet is more akin to the network topology of the long distance network between switches, where capacity is a shared resource. In anonymous ISDN, the mixes hide communication within the local switch, but connections between switches are not hidden. This implies that all calls between two businesses, each large enough to use an entire switch, reveal which businesses are communicating. In onion routing, because of the topology of the Internet, mixing has to be dispersed throughout the Internet, so hiding is greatly improved.

3. Objectives

3.1 Applications

Onion routing's anonymous connections are designed to replace TCP/IP socket connections [3] and to be able to work with unmodified applications. A socket connection is a reliable bi-directional connection carrying a stream of data between two machines. Socket connections provide the abstraction that shields an application from the unreliable and unordered communication that is provided by lower levels of the IP stack.

Many applications use socket connections:

- Web requests (HTTP)
- Remote logins (RLOGIN)
- e-mail (SMTP)
- File transfer (FTP)
- Internet Relay Chat (IRC)
- Encrypted IP Tunnel

These applications can connect to onion routing's anonymous connections using *proxies*. A proxy [11] is usually a relay between an initiating and responding application. In onion routing, anonymous connections are terminated by application specific proxies that relay information between the connection and the unmodified applications. Many applications are already *proxy aware* because proxies are commonly used to communicate through firewalls. For example, a Web browser on a network with a firewall will reach sites outside the firewall through an HTTP proxy on the firewall machine. In that way, direct connections are never made between internal and external machines.

3.2 Threat Model: Active and Passive Attacks

Onion routing's design is very conservative since it assumes that the public network is very vulnerable. In particular, we assume that:

- All traffic is visible.
- All traffic can be modified.
- Onion routers may be compromised.
- Compromised onion routers may cooperate.

In addition, a sophisticated adversary may be able to detect timing coincidences such as the near simultaneous opening of connections. Timing coincidences are very

difficult to overcome, especially when real-time communication is important. But, if connections are routed over an unpredictable path in a busy network, this sort of attack is also very expensive.

The first four vulnerabilities, however, directly motivate certain design decisions in onion routing. Because traffic is visible, the headers and payloads of all traffic are essentially link encrypted between onion routers so the same data looks different when traveling between routers. Because traffic can be modified, stream ciphers [10] are used for encryption. Inserting, deleting, or modifying traffic en route will disrupt the stream and produce random bits downstream. Because onion routers may be compromised, anonymous connections span several onion routers, even though a single "perfect" mix is adequate to provide privacy. Because compromised onion routers may cooperate, data is encrypted in a layered fashion so it appears different to each onion router, not only between onion routers.

4. The Solution: Onion Routing

Onion routing has two parts: A network infrastructure that carries anonymous connections, and a proxy interfaces that mate these connections to unmodified applications.

4.1 Onion Routing: Network Infrastructure

The public network contains a set of onion routers. Each onion router has a single (socket) connection to each of a small set of neighboring onion routers. Onion routers only talk to their neighbors. Neighboring onion routers are neighbors for onion routing only. That is, communication between two neighboring onion routers is carried over a socket connection, and packets are routed (perhaps dynamically) through many hops by the IP protocol.

An anonymous *connection* is routed through a sequence of neighboring onion routers. Common segments of these routes are multiplexed over the single connection between neighbors. An onion router's obligation is to pass data from one connection to another after applying the appropriate cryptographic operations.

An anonymous connection from an initiator to a responder through four onion routers is illustrated in Figure 2.

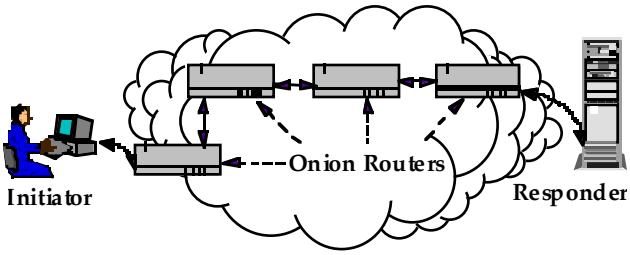


Figure 2. Onion Routing Network Infrastructure

4.2 Onion Routing: Proxy Interface

How are anonymous connections used? Proxies interface between applications and the network infrastructure. When a proxy is used on a firewall, it relays traffic between the protected site and the rest of the world. In onion routing, a proxy's functions are split into two: one part links the initiator to the anonymous connection and the other part links the anonymous connection to the responder. In this way, the initiating and responding applications need not be modified (although they do have to be able to use proxies).

Imagine an initiator sitting at her workstation using a Web browser. When she "clicks" on a URL link, the browser sends an HTTP request for that URL to some *onion routing proxy* instead of directly to the responder. In Figure 3, this is the onion routing proxy named W. W looks at the request and chooses a route through several other onion routers (e.g., W-X-Y-Z). W then sends an *onion* (see section 5.1) along that route; the onion is an instruction to those onion routers to construct an anonymous connection.

The last onion router in the route (Z) also functions as an onion routing proxy for the responder. Z passes data from the anonymous connection to the responder, and passes data from the responder back to the connection.

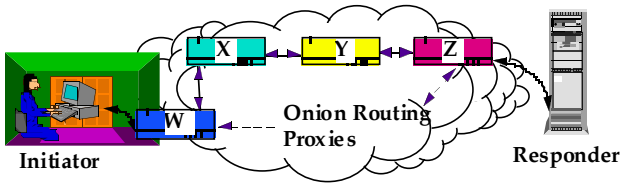


Figure 3: Onion Routing Proxy Interface

Instead of a single socket connection between an initiator and a responder, onion routing requires a socket connection between the initiator and his proxy, an anonymous connection between the initiator's proxy and the responder's proxy, and a socket connection between the responder's proxy and the responder. However, the three connections function as if they were a single (bi-

directional and real-time) socket connection between the initiator and responder.

There are many configurations of an onion routing network. In one basic configuration, a site that is concerned about traffic analysis should control an onion routing proxy in order to protect communication between that proxy and its users. That onion routing proxy must also function as an intermediate onion router in other anonymous connections. If it is not used in this way, observers can monitor the load coming from onion routing proxy and trace it back to the sensitive site. However, if the onion routing proxy is also a busy intermediate onion router, observers cannot tell whether the sensitive site is consuming, producing, or relaying traffic.

Individuals may access an onion router through their Internet Services Provider (ISP), if the ISP controls an onion routing proxy. An individual could also make an encrypted connection to some public domain onion routing proxy. Finally, a user could run an onion routing proxy on his workstation, and route anonymous connections through other onion routers.

5. Using Onion Routing

After the initiator contacts his proxy, onion routing follows four stages:

1. Define the route.
2. Construct the anonymous connection.
3. Move data through the anonymous connection.
4. Destroy the anonymous connection.

The next four sections describe these stages in more detail. (The extra details in each *Details* subsection are independent of the rest of the paper.)

5.1 Defining the Route

Consider Figure 3. The initiator's proxy, W, chooses to make an anonymous connection through (W-X-Y-Z). Therefore, W constructs a layered data structure called an *onion* (Figure 4):

Each layer of the onion is intended for a particular onion router and contains the identity of the next onion router in the anonymous connection, and the key that should be used when communicating with the previous onion router in the connection. The final layer of the onion is intended for Z. Since Z is the last onion router in the connection, its layer only contains a key.

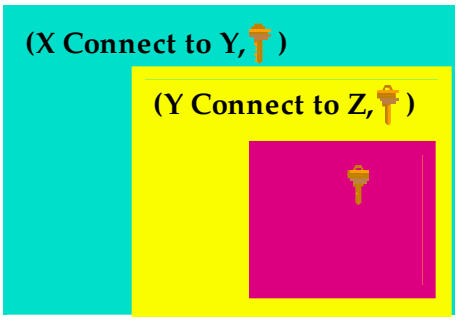


Figure 4: An Onion

Using public key cryptography [10], the onion is constructed so only the intended recipient can peel off the outermost layer, thereby revealing both his layer and the onion embedded inside. No recipient knows who created the onion. So, onion routers can identify only whom they received an onion from and to whom they are obliged to send the embedded onion. And, no recipient can determine what the other onions embedded in an onion look like.

The onion routing proxy that creates an onion keeps a copy of the keys in the onion until the anonymous connection is destroyed. We will see how these keys are used in sections 5.3 and 5.4.

5.1.1 Onion Details

The onion routing proxy routes the anonymous connection through neighboring onion routers. Therefore, it must know the topology of the onion routing network.

The size of an onion limits the length of a route. To prevent observers from inferring the length of a route, onions are padded to some fixed size. This padding becomes part of and is indistinguishable from the already embedded onion.

The key at each layer of the onion is used for bi-directional communication between an onion router and the previous onion router. Therefore, the key really specifies two stream ciphers, one for forward communication (in the direction the onion travels) and the other for backward communication (in the opposite direction).

Each layer of an onion also contains an expiration time. An onion router is to ignore an expired onion and is to ignore replayed onions. Therefore, onion routers must keep track of onions during their lifetimes.

For efficiency, the entire onion is not encrypted using a public key cryptosystem. Instead some prefix (corresponding to the block size of the public key

cryptosystem) of the onion is encrypted using public key cryptography, and the rest of the onion is encrypted using an efficient stream cipher initialized with a key specified in the prefix [5,7,10].

5.2 Constructing the Anonymous Connection

After constructing the onion, W sends the onion to the first onion router in the anonymous connection. The onion moves between onion routers (Figure 5):

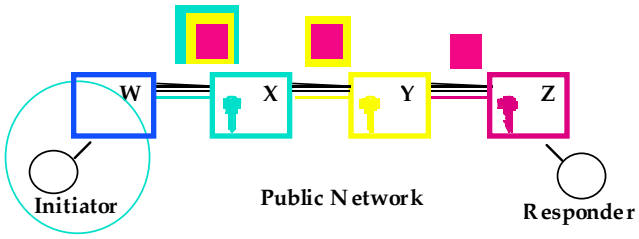


Figure 5: Use of an Onion

Each layer of the onion is intended for a particular onion router, and can be peeled off only by that onion router. The first layer of this onion is intended for X. When X peels off that layer, it obtains a key that it will use when communicating with W (from whom X received the onion), and notes that future traffic from that connection should be forwarded to Y. X also forwards the embedded onion to Y. In a similar way, Y peels off its layer of the onion, revealing the key that it should use to communicate with X (from whom Y received the onion), and notes that future traffic from that connection should be forwarded to Z. Z peels off its layer, revealing only the key that it will use to communicate with Y, and notes that it is the last onion router in the anonymous connection. The first data that Z receives along that anonymous connection will identify the intended responder.

5.2.1 Anonymous Connection Construction Details

To keep onion size constant, each onion router is obliged to add padding to the onion corresponding to the fixed size layer that was removed. Onion routers cannot distinguish padding from embedded onions. If an onion router fails to pad an onion, however, the next onion router will notice that the onion it received is too small and will not process the onion. Because of the padding, even onion routers themselves cannot tell how much of an anonymous connection has been constructed.

Remember that all communication between neighboring onion routers is multiplexed in the data stream of a single socket connection. Therefore, all data travels in a series of fixed size cells. Each cell has a header that identifies the anonymous connection it is assigned to, as well as the type of payload it carries. For example, cells carrying onions will be labeled as `onion` cells, and will also contain the identifier of the new anonymous connection that is to be multiplexed over that socket connection. Notice that this identifier is chosen by the onion router relaying the onion, and in each socket connection carrying a segment of an anonymous connection, the anonymous connection may have a different identifier. Each onion router maintains a table that maps between the identifiers of incoming connections and outgoing connections, and the cryptographic keys that are to be applied to data moving along an anonymous connection.

Cells traveling over a socket connection between onion routers are link encrypted in a peculiar way: headers and payloads are encrypted separately, for efficiency. For example, headers are encrypted with some stream cipher negotiated between the neighboring onion routers. The payload of a cell of type `onion` need not be encrypted, since the onion was already encrypted for the next onion router by the onion routing proxy that created the onion.

Because of the link encryption, observers monitoring the data stream between onion routers cannot read cell headers. Therefore, observers cannot distinguish between onions and other types of cells.

5.3 Moving Data Forward

The anonymous connection moves data from the initiator's proxy to the responder's proxy and vice versa. In the forward direction, the initiator sends plaintext to his onion routing proxy. The onion routing proxy repeatedly *crypts*[1] the data using the inverse of the keys[2] specified in the onion, applying the keys innermost first. Each onion router along the route removes one layer of cryption. The responder's proxy forwards the plaintext to the responder.

This is illustrated in Figure 6.

[1] We define the verb *crypt* to mean the application of a cryptographic operation, be it encryption or decryption, where the two are logically interchangeable. For example, in a stream cipher using Output Feedback Mode (e.g., DES OFB), encryption and decryption are the same operation.

[2] Each key really specifies a stream cipher. The inverse of a key, therefore, is the inverse of the corresponding stream cipher.

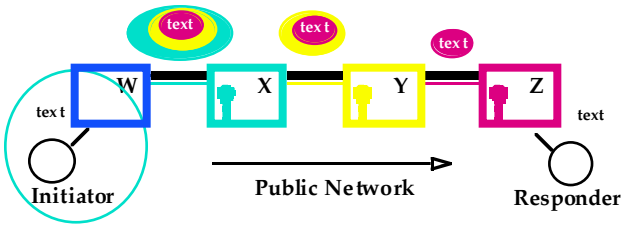


Figure 6: Moving Data Forward

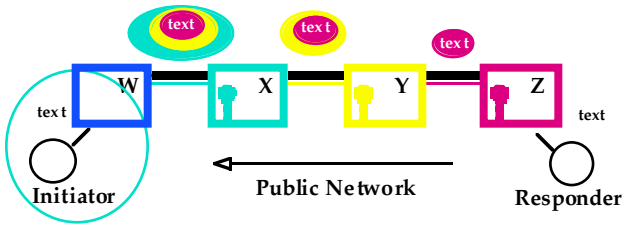


Figure 7: Moving Data Backward

The purpose of the pre-cryptions is to make the data look different as it travels through the anonymous connections, both to outside observers and to the onion routers. Notice that observers cannot match data along the route, and onion routers cannot predict what data will look like later.

Notice that each onion router does one cryption, while the initiator's onion routing proxy does one pre-cryption for each subsequent onion router in the connection.

5.3.1 Moving Data Forward Details

Data moving in the direction that the onion was sent is defined to be moving in the forward direction. Data moving in the reverse direction is defined to be moving backward. This distinction is important when discussing *reply onions* (section 6).

As with onions, data is carried in cells through the multiplexed socket connections. The cells have type `data` and are labeled with the identifier of the associated anonymous connection. Although the headers of `data` cells are link encrypted between onion routers, the payloads of `data` cells are not link encrypted, as the cryption operation done at each onion router is sufficient.

When a `data` cell arrives, the onion router looks up the cell's identifier in its tables and finds the corresponding outbound identifier. The appropriate cryptographic operation is applied and the crypted payload is formed and sent along the outbound connection.

As with the link encryption of the headers, the payloads of `data` cells are encrypted using stream ciphers.

5.4 Moving Data Backward

Moving data backward is just the reverse of sending data forward. The responder's onion routing proxy receives plaintext from the responder. It and each subsequent onion router adds one layer of cryption and sends the data to the next onion router. The initiator's onion routing proxy removes the layers of cryption by applying the inverse of the keys in the onion outermost first. The resulting plaintext is forwarded to the initiator.

This is illustrated in Figure 7.

5.4.1 Moving Data Backward Details

As with forward data, the initiator's onion routing proxy handles the bulk of the cryption burden.

5.5 Destroying the Anonymous Connection

Just as socket connections are torn down, anonymous connections need to be destroyed when the connection is broken. An onion router that decides to tear down a connection sends a destroy message forward and backward along the anonymous connection. It also cleans up its own tables. An onion router that receives a destroy message is obliged to clean up its own table and relay the message in the same direction.

This is illustrated in Figure 8.

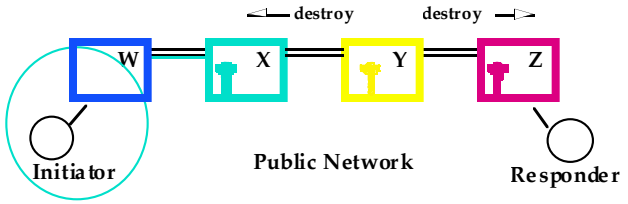


Figure 8: Destroying an Anonymous Connection

Notice that the multiplexed socket connections between neighboring onion routers remain active.

5.5.1 Destroy Details

Destroy messages are sent in cells of type `destroy`. The header identifies the anonymous connection that is to be destroyed. The payload is random and changes at each onion router.

6. Reply Onions

The (forward) onion described in section 5 is used by the initiator's onion routing proxy to construct an anonymous connection to some responder's onion routing proxy. What happens if an initiator expects a later reply from the responder? An obvious solution is to keep the anonymous connection open. This may not always be practical. Another solution is a *reply onion*.

An initiator's onion routing proxy can create a reply onion that defines a route back to him. For example,

Figure 9 illustrates a reply onion that will construct an anonymous connection back to W from Z through onion routers Y and X:

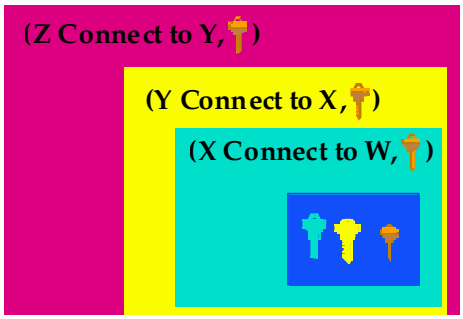


Figure 9: A Reply Onion

The reply onion is sent by the responder to onion routing proxy Z, who peels off its first layer, and sends the embedded reply onion on to onion router Y after extracting the key that Z will use when communicating with Y. Onion routers Y and X do the same operation. Onion routing proxy W receives a reply onion with a sequence of keys.

The anonymous connection established by this reply onion is illustrated in Figure 6, and is identical to the anonymous connection established by the (forward) onion illustrated in Figure 4. Once the anonymous connection is established, each onion router has the same role it has in a forward connection from the initiator to the responder: That is, the initiator's onion routing proxy repeatedly pre-crypts data and other onion routers crypt only once.

Reply onion's can also be used to allow anonymous replies back to some initiator. The initiator may publish a reply onion, which can be picked up and used by any responder. The responder forwards the onion to the designated onion routing proxy and an anonymous connection back to the initiator will be constructed.

6.1 Reply Onion Details

As with forward onions, reply onions contain expiration times to prevent replays. This means that published reply onions can only be used once. If an initiator expects several replies, he should publish many reply onions.

During connection construction, both the responder's and initiator's onion routing proxies know that they have a reply onion. Once the connection is established, however, this distinction is irrelevant. Intermediate onion routers can never distinguish between forward and reply onions. In fact, the only difference between anonymous connections formed by forward onions and those formed by reply onions is that the sets of keys used to crypt data in each direction are swapped: forward keys are used as backward keys and vice versa.

A reply onion may also be created by a third party to define an anonymous connection back to some initiator. Third party reply onions are unusual because both the third party and the initiator know all the onion keys.

7. Implementation

Onion routing has been implemented on Sun Solaris 2.4. Onion routing proxies for Web browsing (HTTP), RLOGIN, e-mail (SMTP), and FTP have been implemented also. Furthermore, versions of these proxies that anonymize the data stream have been implemented. These proxies allow anonymous communication that is resistant to both eavesdropping and traffic analysis.

An extension to this prototype must handle changes to the topology of the onion routing network. This includes, for example, new onion routers, different neighbors, and distribution of onion routers' public keys.

8. Discussion

To be effective, onion routing must be widely deployed and there must be significant use of all the onion routers. Furthermore, onion routing proxies must also be intermediate onion routers. Otherwise, it is easy to infer that traffic to and from a particular onion routing proxy is really to and from the sensitive site that controls the proxy.

8.1 Vulnerabilities

Onion routing is not invulnerable to traffic analysis attacks. With enough data, it is still possible to analyze usage patterns and make educated guesses about the routing of messages. Also, since our first application (Web requests) requires real-time communication, it may be possible to detect the near simultaneous opening of socket connections on the initiator's and responders' onion routing proxies, thereby revealing who is requesting what information. (Of course, even this attack is impossible if the initiator's onion routing proxy is controlled by his sensitive site.) However, these sorts of attacks require the collection and analysis of huge amounts of data by external observers.

One way to further complicate this sort of analysis is to pass dummy traffic through the network to make the traffic level fairly constant. There is an obvious tradeoff here between security and cost: Adding dummy traffic undermines the efficiencies of the Internet as a shared resource. It is difficult to calculate the value of this tradeoff. If traffic is very bursty and response time is important, smoothing out network traffic requires wasting capacity. If, however, traffic is relatively constant, additional smoothing may not be necessary. From a practical point of view, the Internet may not provide the control necessary to smooth out traffic: unlike ATM, users do not own capacity on shared connections. The important observation, however, is that onion routing provides an architecture within which these tradeoffs can be made and explored.

Other attacks depend upon compromised onion routers. If the initiator's onion routing proxy is compromised, then all information is revealed. In general, it is sufficient for a single onion router to be uncompromised to complicate traffic analysis.

Any compromised onion router can still destroy connections or stop forwarding messages, resulting in denial of service attacks. Although this appears to be akin to the denial of service problem in IP source routing, where the unreachability of any part of the route causes packet loss, the situation is closer to loose source routing where packets may be routed arbitrarily between the prespecified routers. Furthermore, in onion routing, if the connection is broken, the rest of the onion routers are informed via destroy messages.

Onion routing uses expiration times to prevent replay attacks. It is curious that, unlike other services that depend upon a common clock, the vulnerability due to poor synchronization here is a denial of service attack, instead of a replay attack. If an onion router's clock is too fast, otherwise timely onions will appear to have already expired. Also, since expiration times define the window during which onion routers must store used onions, an onion router with a slow clock will end up storing more information.

The data stream cannot be replayed, as stream ciphers are used for encryption. If the data stream is changed in any way, synchrony will be lost and the data stream will become irreversibly corrupted. Since TCP/IP socket connections are used to carry the data stream, we expect error free data delivery.

8.2 Cryptographic Overhead

In onion routing, the cryptographic overhead on intermediate onion routers is less than the burden of link encryption on routers. In link encryption, each packet is encrypted by each sender and decrypted by each recipient. In onion routing, only one cryptographic operation is applied between every two onion routers. This is because the initiator's onion routing proxy repeatedly pre-crypts data.

The total number of encryptions remains the same, however. It is interesting to note that shifting the encryption burden provides (for free):

- Link encryption.
- End to end encryption.
- Data hiding: the same data looks different to each onion router.

8.3 Infrastructure Variations

An interesting application of onion routing is a variation of IRC (Internet Relay Chat). Two sites can build anonymous connections through onion routers they each trust to meet at a designated onion routing proxy. That proxy mates the two connections. Privacy is guaranteed, and neither party needs to trust the other to hide his participation from outside observers.

Since connection setup is relatively expensive, it may be useful to delink sockets and anonymous connections. For example, when using a Web browser to view a particular Web page, several socket connections may be established to retrieve various parts of the document. There is no reason that those socket connections could not all use (either serially or in parallel) the same anonymous connection.

It is interesting to consider protocol encapsulation. Onions can be carried over the anonymous connections. This would enable extending connections and may enable using parts of connections or linking together parts of connections. This process allows the length of anonymous connection routes to be extended indefinitely, and permits the size of the onion routing network to grow arbitrarily.

Since real-time connections are inherently more vulnerable to traffic analysis than less time critical applications, it makes sense to tag connections with various service guarantees. A real-time connection will be less resistant to traffic analysis than a slow connection because

intermediate onion routers have less flexibility with buffering its data stream.

Onion routing networks can exist in many configurations to accommodate the requirements of large institutions, ISPs, and individuals through a combination of institution or ISP controlled onion routing proxies, public domain onion routing proxies, and public domain onion routers. The combination of many sources of traffic enables the network to further complicate traffic analysis.

8.4 Lower Levels of the Stack

Can onion routing be implemented at lower levels of the communications stack? The obvious advantage is that this would eliminate the need for application specific onion routing proxies. The difficulty with pushing onion routing beneath the socket layer of the IP stack is that onion routing's connection setup is relatively expensive, and is impractical to use each time a packet is sent over a connectionless circuit.

Since ATM is connection based, however, it is perfectly reasonable to consider using onion routing's approach for connection setup to make anonymous ATM connections. In fact, in our prototype, we are modeling our cells based on ATM cells.

9. Conclusion

Onion routing provides real-time, bi-directional communication through anonymous connections that are resistant to both eavesdropping and traffic analysis. These anonymous connections can substitute for socket connections in a wide variety of unmodified Internet applications using proxies. Our prototype of onion routing includes proxies for Web browsers (HTTP), remote login, e-mail, and file transfer protocols as well as anonymizing versions of these protocols The anonymizing version of the e-mail proxy creates an anonymous connection between two *sendmail* daemons and removes identifying information from the headers of the mail message. This approach contrasts with Anonymous Remailers, where each remailer provides a single hop in a chain of mail forwarding. This highlights the difference between onion routing and other uses of Chaum mixes: Privacy and anonymity are moved beneath the application layer and made application independent.

Onion routing will only be effective in complicating traffic analysis if its infrastructure is widely deployed and widely used. This deployment is considerably simplified because applications need not be modified.

Our motivation here is not to provide anonymous communication, but to separate identification from routing. Authenticating information must be carried in the data stream. Applications can (and usually should) identify themselves to each other. But, the use of a public network should not automatically reveal the identities of communicating parties. The goal here is anonymous routing, not anonymity.

References

1. D. Chaum. *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Communications of the ACM, v. 24, n. 2, Feb. 1981, pages 84-88.

2. D. Chaum, *The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability*, Journal of Cryptology, 1/1, 1988, pages 65-75.

3. D. E. Comer. *Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture*, Prentice--Hall, Engelwood Cliffs, New Jersey, 1995.

4. L. Cottrell. *Mixmaster and Remailer Attacks*, http://obscura.obscura.com/~loki/remailer/remailer-essay.html

5. D. Goldschlag, M. Reed, and P. Syverson. *Hiding Routing Information*. Workshop on Information Hiding, Cambridge, UK, May, 1996.

6. C. Gulcu and G. Tsudik. *Mixing Email with* Babel, 1996 Symposium on Network and Distributed System Security, San Diego, February 1996.

7. A. Pfitzmann and B. Pfitzmann. *How to Break the Direct RSA-implementation of MIXes*, Advances in Cryptology--EUROCRYPT '89 Proceedings, Springer-Verlag, Berlin, 1990, pages 373-381.

8. A. Pfitzmann, B. Pfitzmann, and M. Waidner. *ISDN-Mixes: Untraceable Communication with Very Small Bandwidth Overhead*, GI/ITG Conference: Communication in Distributed Systems, Mannheim Feb, 1991, Informatik-Fachberichte 267, Springer-Verlag, Heidelberg 1991, pages 451-463.

9. A. Pfitzmann and M. Waidner. *Networks Without User Observability*, Computers & Security, 6/2 1987, pages 158-166.

10. B. Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C*, 2nd edition, John Wiley and Sons, 1996, (the red one).

11. W. R. Stevens. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols*, Addison--Wesley, Reading, Mass., 1996.

EXHIBIT O

TO MICHAEL FRATTO'S DECLARATION

GOLDSCHLAG, "HIDING ROUTING INFORMATION" (1996)

David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. "Hiding Routing Information," Workshop on Information Hiding, Cambridge, UK, May, 1996.

Hiding Routing Information

David M. Goldschlag, Michael G. Reed, and Paul F. Syverson

Naval Research Laboratory, Center For High Assurance Computer Systems, Washington, D.C. 20375-5337, USA, phone: +1 202.404.2389, fax: +1 202.404.7942, e-mail: {*last name*}@itd.nrl.navy.mil.

Abstract. This paper describes an architecture, *Onion Routing*, that limits a network's vulnerability to traffic analysis. The architecture provides anonymous socket connections by means of proxy servers. It provides real-time, bi-directional, anonymous communication for any protocol that can be adapted to use a proxy service. Specifically, the architecture provides for bi-directional communication even though no-one but the initiator's proxy server knows anything but previous and next hops in the communication chain. This implies that neither the respondent nor his proxy server nor any external observer need know the identity of the initiator or his proxy server. A prototype of *Onion Routing* has been implemented. This prototype works with HTTP (World Wide Web) proxies. In addition, an analogous proxy for TELNET has been implemented. Proxies for FTP and SMTP are under development.

1 Introduction

This paper presents an architecture that limits a network's vulnerability to traffic analysis. We call this approach *Onion Routing*, because it relies upon a layered object to direct the construction of an anonymous, bi-directional, real-time virtual circuit between two communicating parties, an *initiator* and *responder*. Because individual *routing nodes* in each circuit only know the identities of adjacent nodes (as in [1]), and because the nodes further encrypt multiplexed virtual circuits, studying traffic patterns does not yield much information about the paths of messages. This makes it difficult to use traffic analysis to determine who is communicating with whom.

Onion Routing provides an anonymous socket connection through a proxy server. Since proxies are a well defined interface at the application layer [12, 11], and many protocols have been adapted to work with proxy servers in order to accommodate firewalls, Onion Routing can be easily used by many applications. Our prototype works with HTTP (World Wide Web) proxies. In addition, a proxy for TELNET has been implemented.

Traffic analysis can be used to help deduce who is communicating with whom by analyzing traffic patterns instead of the data that is sent. For example, in most networks, it is relatively easy to determine which pairs of machines are communicating by watching the routing information that is part of each packet. Even if data is encrypted, routing information is still sent in the clear because routers need to know packets' destinations, in order to route them in the right

direction. Traffic analysis can also be done by watching particular data move through a network, by matching amounts of data, or by examining coincidences, such as connections opening and closing at about the same time.

Onion Routing hides routing information by making a data stream follow a path through several nodes en route to its destination. The path is defined by the first node, which is also a proxy for the service being requested (e.g., HTTP requests). Therefore, this Proxy/Routing Node is the most sensitive one, so sites that are concerned about traffic analysis should also manage a Proxy/Routing Node. We will see later that it is important that this Proxy/Routing Node also be used as an intermediate routing node in other virtual circuits. Although the compromise of all routing nodes compromises the hiding, one uncompromised routing node is sufficient to complicate traffic analysis. Figure 1 illustrates the topology of an Onion Routing network with five nodes, one of which (W) is the Proxy/Routing node for the initiator's site.

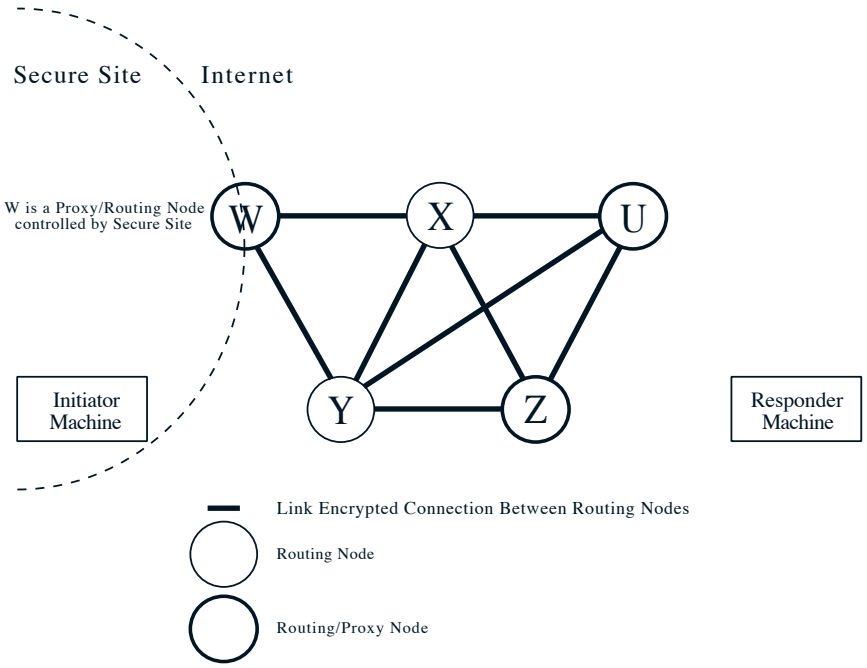


Fig. 1. Routing Topology.

The goal of Onion Routing is not to provide anonymous communication. Parties are free to (and usually should) identify themselves within a message. But the use of a public network should not automatically give away the identities and locations of the communicating parties. For example, imagine a researcher who uses the World Wide Web to collect data from a variety of sources. Although each

piece of information that he retrieves is publicly known, it may be possible for an outside observer to determine his sensitive interests by studying the patterns in his requests. Onion Routing makes it very difficult to match his HTTP requests to his site.

Anonymous re-mailers [5, 6] attempt to limit the feasibility of traffic analysis by providing an anonymous store and forward architecture. To prevent replay attacks, re-mailers keep a log of sent messages. These two characteristics make the anonymous re-mailer approach unsuitable for HTTP applications, as HTTP requests would both generate an enormous log and require bi-directional communication. Anonymous ISDN [8] has even more severe real-time and bi-directional requirements than HTTP, but, the architecture of an ISDN network is considerably different from the architecture of the Internet [4].

Onion Routing provides bi-directional communication, without requiring that the responder know the initiator's identity or location. Individual messages are not logged. In addition, Onion Routing is easily adapted to electronic mail. Messages can include *Reply Onions* that permit a later reply to the sender without knowing his address and without keeping the original virtual circuit open.

The rest of the paper is organized in the following way: Section 2 presents background information. Section 3 describes the *Onion*, the object that directs the construction of the virtual circuit. Section 4 describes the construction and use of these virtual circuits. Section 5 describes the vulnerabilities in the Onion Routing architecture. Section 6 presents some concluding remarks.

2 Background

Chaum [1] defines a layered object that routes data through intermediate nodes, called *mixes*. These intermediate nodes may reorder, delay, and pad traffic to complicate traffic analysis. Some work has been done using mixes in ATM networks [3].

Anonymous Remailers like [5, 6] use mixes to provide anonymous e-mail services and also to invent an address through which mail can be forwarded back to the original sender. Remailers work in a store and forward manner at the mail application layer, by stripping off headers at each mix, and forwarding the mail message to the next mix. These remailers provide confirmation of delivery.

In [8], mixes are used to provide untraceable communication in an ISDN network. In a phone system, each telephone line is assigned to a particular local switch (i.e., local exchange), and switches are interconnected by a (long distance) network. Anonymous calls in ISDN rely upon an anonymous connection within each switch between the caller and the long distance network, which is obtained by routing calls through a predefined series of mixes. The long distance endpoints of the connection are then mated to complete the call. (Notice that observers can tell which local switches are connected.) This approach relies upon two unique features of ISDN switches. Since each phone line has a subset of the switch's total capacity pre-allocated to it, there is no (real) cost associated with keeping

a phone line active all the time, either by making calls to itself, to other phone lines on the same switch, or to the long distance network. Keeping phone lines active complicates traffic analysis because an observer cannot track coincidences.

Also, since each phone line has a control circuit connection to the switch, the switch can broadcast messages to each line using these control circuits. So, within a switch a truly anonymous connection can be established: A phone line makes an anonymous connection to some mix. That mix broadcasts a token identifying itself and the connection. A recipient of that token can make another anonymous connection to the specified mix, which mates the two connections to complete the call.

Our goal of anonymous socket connections over the Internet differs from anonymous remailers and anonymous ISDN. The data is different, with real-time constraints more severe than mail, but somewhat looser than voice. Both HTTP and ISDN connections are bidirectional, but, unlike ISDN, HTTP connections are likely to be small requests followed by short bursts of returned data. In a local switch capacity is pre-allocated to each phone line, and broadcasting is efficient. But broadcasting over the Internet is not free, and defining broadcasts domains is not trivial. Most importantly, the network topology of the Internet is more akin to the network topology of the long distance network between switches, where capacity is a shared resource. In anonymous ISDN, the mixes hide communication within the local switch, but connections between switches are not hidden. This implies that all calls between two businesses, each large enough to use an entire switch, reveal which businesses are communicating. In Onion Routing, mixing is dispersed throughout the Internet, which improves hiding.

3 Onions

To begin a session between an initiator and a responder, the initiator's proxy identifies a series of routing nodes forming a route through the network and constructs an *onion* which encapsulates that route. Figure 2 illustrates an onion constructed by the initiator's Proxy/Routing Node W for an anonymous route to the responder's Proxy/Routing Node Z through intermediate routing nodes X and Y. The initiator's proxy then sends the onion along that route to establish a virtual circuit between himself and the responder's proxy.

The onion data structure is composed of layer upon layer of encryption wrapped around a payload. Leaving aside the shape of the payload at the very center, the basic structure of the onion is based on the route to the responder that is chosen by the initiator's proxy. Based on this route, the initiator's proxy encrypts first for the responder's proxy, then for the preceding node on the route, and so on back to the first routing node to whom he will send the onion. When the onion is received, each node knows who sent him the onion and to whom he should pass the onion. But, he knows nothing about the other nodes, nor about how many there are in the chain or his place in it (unless he is last). What a

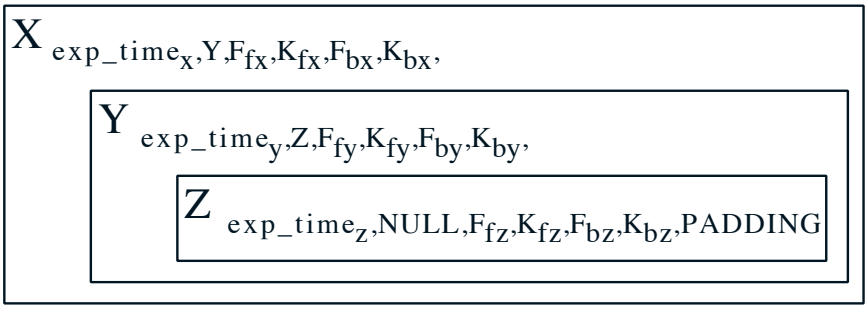


Fig. 2. A Forward Onion.

node P_x receives looks like this

$$\{exp\_time, next\_hop, F_f, K_f, F_b, K_b, payload\}_{PK_x}$$

Here PK_x is a public encryption key for routing node P_x, who is assumed to have the corresponding decryption key.[1] The decrypted message contains an expiration time for the onion, the next routing node to which the payload is to be sent, the payload, and two function/key pairs specifying the cryptographic operations and keys to be applied to data that will be sent along the virtual circuit. The forward pair (F_f, K_f) is applied to data moving in the forward direction (along the route that the onion is traveling) the backward pair (F_b, K_b) is applied to data moving in the opposite direction (along the onion's reverse route).[2] (If the receiving node is the responder's proxy, then the *next_hop* field is *null*.) For any intermediate routing node the payload will be another onion. The expiration time is used to detect replays, which pairs of compromised nodes could use to try to correlate messages. Each node holds a copy of the onion until *exp_time*. If he receives another copy of the same onion within that time he simply ignores it. And, if he receives an onion that has expired, he ignores that as well.

Notice that at each hop the onion shrinks as a layer is peeled off. To avoid compromised nodes inferring route information from this monotonically diminishing size, a random bit string the size of the peeled off layer is appended to the end of the *payload* before forwarding. No proxy except the last will know how much of the *payload* he receives is such padding because he won't know where

[1] Depending on certain assumptions about the fields in each onion layer, a naive RSA implementation of the simple public key encryption implied by our notation could be vulnerable to an attack as described in [7]. In our implementation, this potential vulnerability is illusory since the public key is only used to encrypt a secret key, and that secret key is used to encrypt the remainder of the message using an efficient symmetric algorithm. This also makes for a more efficient implementation than the simple, straightforward implementation using only public keys.

[2] Specifying two pairs of functions unifies the virtual circuits that are constructed by forward and reply onions. See section 3.3.

he is in the chain. He simply 'decrypts' the padding along with the rest of the onion. Even a constant size onion might be traced unless all onions are the same size, so we fix the size of the onion. To maintain this constant size to hide the length of the chain from the responder's proxy, the initiator's proxy will pad the central *payload* according to the size of the onion, i.e., the number of hops. So, when any onion arrives at the responder's proxy it will always have the same amount of padding, either added initially or en route.

3.1 Creating the circuit

The goal in sending the onion is to produce virtual circuits within link encrypted connections already running between routing nodes.[3] More details will be given in section 4. An onion occurs as the data field in one of the presently described 'messages'. Such messages contain a circuit identifier, a command (*create*, *destroy*, and *data*), and data. Any other command is considered an error, and the node who receives such a message ignores that message except to return a *destroy* command back through that virtual circuit. The *create* command accompanies an onion. When a node receives a create command along with an onion, he chooses a virtual circuit identifier and sends another *create* message containing this identifier to the next node and the onion (padded with his layer peeled off). He also stores the virtual circuit identifier he received and virtual circuit identifier he sent as a pair. Until the circuit is destroyed, whenever he receives data on the one connection he sends it off on the other. He applies the forward cryptographic function and key (obtained from the onion) to data moving in the forward direction (along the route the onion traveled) and the backward cryptographic function and key to data moving in the opposite direction (along the onion's reverse route). The virtual circuit established by the onion in figure 2 is illustrated in figure 3:

Data sent by the initiator over a virtual circuit is "pre-crypted"[4] repeatedly by his proxy by applying the inverse of all the forward cryptographic operations specified in the onion, innermost first. Therefore, these layers of cryptography will be peeled off as the data travels forward through the virtual circuit. Data sent by the responder is "crypted" once by his proxy and again by each previous node in the virtual circuit using the backward cryptographic operation specified at the corresponding layer of the onion. The initiator's proxy applies the inverse of the backward cryptographic operations specified in the onion, outermost first, to this stream, to obtain the plaintext.

3.2 Loose Routing

It is not necessary that the entire route be prespecified by the initiator's proxy. He can instruct various nodes along the route to choose their own route to the

[3] Onions could be used to carry data also, but since onions have to be tracked to prevent replay, this would introduce a large cost.

[4] We define the verb *crypt* to mean the application of a cryptographic operation, be it encryption or decryption, where the two are logically interchangeable.

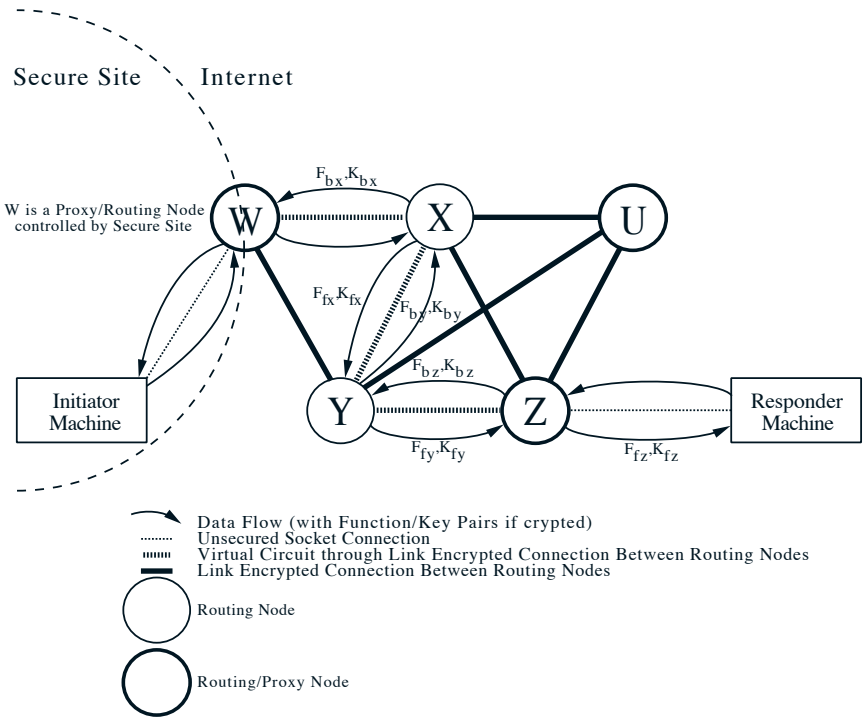


Fig. 3. A Virtual Circuit.

next prespecified node. This can be useful for security, adding more hops to the chain. It could also be used if the initiating proxy does not know a complete, connected route to the responder but believes that the node where any break occurs can construct a route to the next node. Or, loose routing can be used to handle connection changes that occur of which the initiator was unaware. Also, since onions are all of fixed size, there is a fixed maximum length to the route from the initiator's proxy to the responder's proxy. Loose routing allows us to increase the size of that maximum for the same fixed onion size. Why this is so should become clear presently.

It is also possible to iterate the loose routing process, allowing nodes on the added route to themselves add to the chain. Obviously, we need a mechanism to prevent the chain from lengthening indefinitely. This can be incorporated into the onion structure. An onion for a system that allows for loose routing is as follows:

$$\{exp\_time, next\_hop, max\_loosecount, F_f, K_f, F_b, K_b, payload\}_{PK_x}$$

If the node receiving this onion decides to loose-route the onion, he prepares a new onion with up to *max_loosecount* layers. The payload of this onion is

simply the onion he received with PK_x changed for the last (innermost) node he added to the chain. In other words, he behaves as an initiator's proxy except that his payload is itself already an onion. (This node behaves like an initiator's proxy with respect to data also, since he must repeatedly pre- and post- crypt data that moves along the diverted route.) To keep the onion a constant length he must truncate the payload by an amount commensurate with the layers he has added to the onion. The initiating proxy must anticipate the amount of padding (both present initially and any added and/or truncated en route) that will be on the central payload at the time loose routing occurs to allow for this truncation. Failure to pre-pad correctly or ignoring an onion's fixed size will result in a malformed onion later in the route. The total of the *max_loosecount* values occurring in the added layers plus the number of added layers must be less than or equal to the *max_loosecount* value that the adding node received.

3.3 Reply Onions

There are applications in which it would be useful for a responder to send back a reply after the original circuit is broken. This would allow answers (like e-mail replies) to be sent to queries that were not available at the time of the original connection. As we shall see presently, this also allows the responder as well as the initiator to remain hidden. The way we allow for these delayed replies is by sending a reply onion to accompany the reply. Like the forward onion, it reveals to each node en route only the next step to be taken. It has the same structure as the forward onion and is treated the same way by nodes en route. Intermediate nodes processing an onion cannot differentiate between forward and reply onions. Furthermore, the behavior of the original initiator and responder proxies are the same, once the circuit is formed.

The primary difference between a forward and a reply onion is the innermost payload. The payload of the forward onion can be effectively empty (containing only padding). The reply onion payload contains enough information to enable the initiator's proxy to reach the initiator and all the cryptographic function and key pairs that are to crypt data along the virtual circuit. The initiator's proxy retrieves the keys from the onion. Figure 4 illustrates a reply onion constructed by the initiator's Proxy/Routing Node W for an anonymous route back to him starting at the responder's Proxy/Routing Node Z through intermediate routing nodes Y and X:

There is no difference between virtual circuits established by reply onions and forward onions, except that in circuits established by reply onions intermediate routing nodes appear to think that forward points toward the initiator's proxy. But since the behavior of intermediate routing nodes is symmetric, this difference is irrelevant. The terminal Proxy/Routing nodes, however, have the same behavior in circuits established by forward and reply onions. Therefore, a figure of the virtual circuit formed by the reply onion illustrated in figure 4 would be identical to the virtual circuit illustrated in figure 3 even though the circuit was formed by the reply onion moving from the responder's proxy node to the

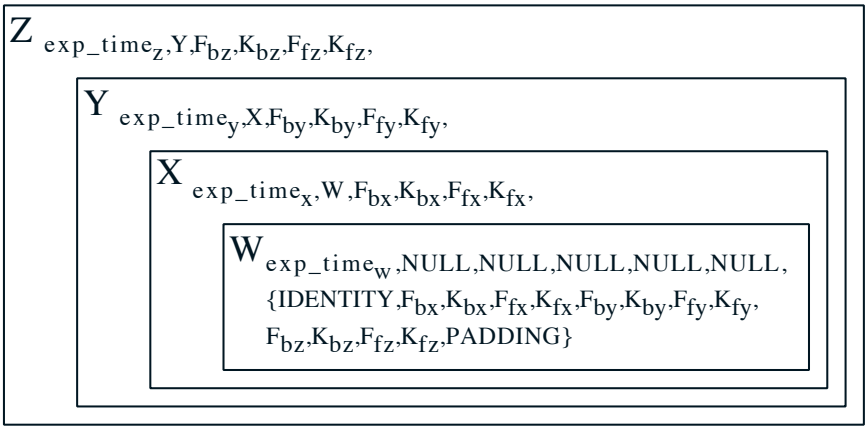


Fig. 4. A Reply Onion.

initiator's proxy node. Internally to the intermediate nodes, the forward cryptographic functions are applied to data moving in the direction that the circuit was established, and the backward cryptographic functions are applied to data moving in the opposite direction. The location of the terminal Proxy/Routing Nodes are in this sense reversed, with the initiator's proxy at the end of the circuit and the responder's proxy at the beginning of the circuit. However, the behavior of the initiator and responder proxies is identical to their behavior in the virtual circuit formed by a forward onion. This is the reason for having forward and backward function/key pairs at each layer of the onion.

Like a forward onion, a reply onion can only be used once. When a node receives an onion it is kept until it expires, and any onion received is compared to detect replay. If a replay is detected, it is treated as an error and ignored. Since reply onions can only be used once, if multiple replies are desired, multiple reply onions must be sent. Of course, they need not all follow the same return route; although they may. If replies are only likely to be forthcoming if they are anonymous, one or more reply onions can be broadcast. Anyone can then reply with an unused onion. If he can maintain anonymity from or in cooperation with the responder's proxy for that reply onion, then he can do so anonymously.

4 Implementation

The easiest way to build our system without requiring the complete redesign and deployment of new client and server software is to make use of existing proxy technologies. Historically, proxy technologies have been used to create tunnels through a firewall. The use of proxy technologies requires that the client applications be 'proxy aware'. The widespread deployment of firewalls on the Internet has created the demand for such proxy aware applications, which software manufacturers are rushing to meet.

In the firewall setting, a system administrator will set up a proxy server on the firewall machine which will be responsible for forwarding requests from the protected domain out onto the open Internet, and maintain a return path for the response to the request. A proxy server can be divided into two parts: the front end that receives and parses the request, and the back end that processes the request and returns the results back to the requester. Classically, the front and back ends are the same process running on one machine.

Under our system we will use a traditional proxy front end and back end, but, they will be separate processes on separate machines with a tunnel connecting them. In this manner, our Proxy/Routing Nodes will look no different to the client and server software than any other proxy server. A couple of assumptions will hold for the remainder of this paper: 1) Proxy/Routing Nodes and intermediate routing nodes know about each other in advance of their operation, and 2) public key certificates for each node have been securely distributed to all others prior to operation.

All nodes are connected by link encrypted connections which multiplex many virtual circuits between initiator and responder proxy nodes. These connections are link encrypted in an odd way (for efficiency). All messages moving through these connections are of fixed size and have two components, header and payload fields. Header fields contain the virtual circuit identifier and the command and are link encrypted using a stream cipher [10]. Since all payload fields will be encrypted via other mechanisms (public keys or onion keys), they need not be link encrypted.

There are three commands that nodes understand. The first is to *create* a virtual circuit. At each node, a virtual circuit has two connections. Data arriving on one is passed along on the other. The circuit is defined by the labels for these two connections. Creating a virtual circuit is the process of defining these labels for each node along the route. For the first Proxy/Routing Node, one connection is a link to the initiator, and the other is a link to the next routing node. The Proxy/Routing Node creates an onion defining the sequence of intermediate routing nodes to the responder's Proxy/Routing Node. It breaks the onion up into payload sized chunks and transmits these chunks in order to the next node with a control field containing both the label of the connection and a *create* command. Each subsequent node reassembles the onion and peels off a layer from the onion which reveals the next node in the route and two cryptographic function/key pairs. Before acting on the *create* command, the node checks whether the onion has expired or is a replay. To check for replay, the node consults a table of unexpired onions. If the onion is valid, it is inserted into the table, and the node then labels a new connection to the next node and passes the peeled and padded onion in a similar sequence of messages to the next node. It also updates a table containing the labels and cryptographic function/key pairs associated with the new virtual circuit. The appropriate (forward or backward) function/key pair should be used to crypt data moving along that circuit. The responder's Proxy/Routing Node, recognizing that the onion is empty, will partially update its tables. As with standard proxies the next *data* message along

this circuit will identify the responder.

The second command is *data*. The second role of the initiator's Proxy/Routing Node is to pass a stream of data from the initiator along the virtual circuit together with other control information for the responder's Proxy/Routing Node. To do this, he breaks the incoming stream into (at most) payload sized chunks, and repeatedly pre-crypts each chunk using the inverse of the cryptographic operations specified in the onion, innermost first. The function/key pairs that are applied, and the virtual circuit identifier of the connection to the next node are obtained from a table. The header field for each payload is the label of the connection and a *data* command. Each subsequent node looks at its table, obtaining the cryptographic function/key pair associated with the circuit (for the appropriate direction) and the virtual circuit identifier of the connection to the next node. It then peels off a layer of cryptography and forwards the peeled payload to the next node. Once the data reaches the responder's proxy, its final cryption will produce the plaintext that is to be processed or forwarded to the responder.

The *data* command can also be used to move data from the responder's Proxy/Routing Node to the initiator's Proxy/Routing Node. The responder's Proxy/Routing Node obtains the cryptographic function/key pair and the virtual circuit identifier for the next node from its tables, and crypts the stream. It breaks the crypted stream into payload sized chunks and forwards them to the next node with the appropriate control field. Each subsequent node further stream crypts each payload using the appropriate function/key associated with that virtual circuit. Once a messages arrives at the initiator's Proxy/Routing Node he looks at his table and applies the inverse of the backward cryptographic operations specified in the onion, outermost first, to this stream to obtain the plaintext. The plaintext is forwarded to the initiator.

The third command is *destroy* which is used to tear down a virtual circuit when it is no longer needed or in response to certain error conditions. Notice that *destroy* messages can be initiated by any node along a virtual circuit, and it is a node's obligation to forward the *destroy* messages in the appropriate directions. (A node initiating a *destroy* message in an active virtual circuit forwards it in both directions. A node that receives a *destroy* message passes it along in the same direction.) The payload of a *destroy* command is empty padding. Nonetheless, this payload is still crypted with the appropriate function/key pair. In addition to the *destroy* command, the control field contains the virtual circuit identifier of the recipient of the *destroy* command. Upon receipt of a *destroy* command a node deletes the table entries associated with that virtual circuit.

5 Vulnerabilities

Onion Routing is not invulnerable to traffic analysis attacks. With enough data, it is still possible to analyze usage patterns and make educated guesses about the routing of messages. Also, since our application requires real time communication, it may be possible to detect the near simultaneous opening of socket

connections on the first and last proxy servers revealing who is requesting what information. However, these sorts of attacks require the collection and analysis of huge amounts of data by external observers.

Other attacks depend upon compromised Proxy Servers and Routing Nodes. If the initiator's proxy is compromised then all information is revealed. In general it is sufficient for a single routing node to be uncompromised to complicate traffic analysis. However, a single compromised routing node can destroy connections or stop forwarding messages, resulting in denial of service attacks.

Onion Routing uses expiration times to prevent replay attacks. It is curious that, unlike timestamps, the vulnerability due to poorly synchronized clocks here is a denial of service attack, instead of a replay attack. If a node's clock is too fast, otherwise timely onions will appear to have already expired. Also, since expiration times define the window during which nodes must store used onions, a node with a slow clock will end up storing more information.

If the responder's proxy is compromised, and can determine when the unencrypted data stream has been corrupted, it is possible for compromised nodes earlier in the virtual circuit to corrupt the stream and ask which responder's proxy received uncorrupted data. By working with compromised nodes around a suspected initiator's proxy, one can identify the beginning of the virtual circuit. The difficulty with this attack is that once the data stream has been corrupted, it will remain corrupted (because we use a stream cipher), limiting further analysis.

In order for Onion Routing to be effective, there must be significant use of all the nodes, and Proxy Nodes must also be intermediate routing nodes. Choosing the appropriate balance between efficient use of network capacity and security is a hard problem both from a theoretical and practical standpoint. Theoretically, it is difficult to calculate the value of the tradeoff. For more security, network traffic must be relatively constant. This requires sending dummy traffic over a connection when traffic is light and buffering data when traffic is heavy. If traffic is very bursty and response time is important, smoothing out network traffic requires wasting capacity. If however, traffic is relatively constant, additional smoothing may not be necessary. From a practical point of view, the Internet may not provide the control necessary to smooth out traffic: unlike ATM, users do not own capacity on shared connections. The important observation, however, is that Onion Routing forms an architecture within which these tradeoffs can be made and explored.

6 Conclusion

Onion Routing is an architecture that hides routing information while providing real-time, bi-directional communication. Since it provides a virtual circuit that can replace a socket connection, Onion Routing can be used in any protocol that can be adapted to use a proxy service. Although our first use is in HTTP and TELNET, it is easy to imagine other applications. In e-mail, for example, Onion Routing would create an anonymous socket connection between two sendmail daemons. This contrasts with Anonymous Remailers, where each remailer pro-

vides a single hop in a chain of mail forwarding. In this sense, in Onion Routing, the rerouting of messages is independent of the type of message.

Other extensions are also possible and integrate nicely with the proxy approach to anonymity. For example, to create a completely anonymous conversation between two parties, each party would make an anonymous connection to some anonymity server, which mates connections sharing some token. This approach, similar to IRC servers, can also be used if the responder does not trust the initiator, especially with (broadcast) reply onions. The responder builds his own (trusted) connection to some anonymity server, and asks that anonymity server to build another connection to the initiator using a reply onion and to mate the two connections. Each party is therefore protected by a route that he determined.

In Onion Routing the encryption burden on connected intermediate nodes is less than the burden of link encryption on routers. In link encryption, each packet is encrypted by the sender and decrypted by the recipient. In Onion Routing the header and payload of each message are crypted separately: the header is encrypted and decrypted using the connection's key, and the payload is crypted (only by the recipient) using the appropriate function/key pair associated with the virtual circuit.

Our goal here is not to provide anonymous communication, but, to place identification where it belongs. The use of a public network should not automatically reveal the identities of communicating parties. If anonymous communication is undesirable, it is easy to imagine filters on the endpoint machines that restrict communication to signed messages.

Onion Routing will only be effective in complicating traffic analysis if its Proxy and Routing Nodes become widespread and widely used. There is an obvious tension between anonymity and law enforcement. If this tension is resolved in favor of law enforcement, it would be straightforward to integrate a key escrow system within the onion, which would make routing information available to the lawful authorities.

7 Acknowledgements

Discussions with many people helped develop the ideas in this paper. We would like to thank Ran Atkinson, Markus Jakobbsen, John McLean, Cathy Meadows, Andy Moore, Moni Naor, Holger Peterson, Birgit Pfitzmann, Michael Steiner, and the anonymous referees for their helpful suggestions.

References

1. D. Chaum. *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms,* Communications of the ACM, v. 24, n. 2, Feb. 1981, pages 84-88.
2. D. Chaum, *The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability,* Journal of Cryptology, 1/1, 1988, pages 65-75.

3. S. Chuang. *Security Management of ATM Networks,* Ph.D. thesis, in progress, Cambridge University.
4. D. E. Comer. *Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture,* Prentice–Hall, Engelwood Cliffs, New Jersey, 1995.
5. L. Cottrell. *Mixmaster and Remailer Attacks,* http://obscura.obscura.com/~loki/remailer/remailer-essay.html
6. C. Gulcu and G. Tsudik. *Mixing Email with* Babel, 1996 Symposium on Network and Distributed System Security, San Diego, February 1996.
7. A. Pfitzmann and B. Pfitzmann. *How to Break the Direct RSA-implementation of MIXes,* Advances in Cryptology–EUROCRYPT '89 Proceedings, Springer-Verlag, Berlin, 1990, pages 373-381.
8. A. Pfitzmann, B. Pfitzmann, and M. Waidner. *ISDN-Mixes: Untraceable Communication with Very Small Bandwidth Overhead,* GI/ITG Conference: Communication in Distributed Systems, Mannheim Feb, 1991, Informatik-Fachberichte 267, Springer-Verlag, Heildelberg 1991, pages 451-463.
9. A. Pfitzmann and M. Waidner. *Networks Without User Observability,* Computers & Security, 6/2 1987, pages 158-166.
10. B. Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C,* John Wiley and Sons, 1994.
11. W. R. Stevens. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols,* Addison–Wesley, Reading, Mass., 1996.
12. L. D. Stein. *How to Set up and Maintain a World Wide Web Site: The Guide for Information Providers,* Addison–Wesley, Reading, Mass., 1995.

This article was processed using the LaTeX macro package with LLNCS style

EXHIBIT P

TO MICHAEL FRATTO'S DECLARATION

U.S. PATENT 4,885,778

United States Patent [19]

Weiss

[11] **Patent Number: 4,885,778**

[45] **Date of Patent: Dec. 5, 1989**

[54] **METHOD AND APPARATUS FOR SYNCHRONIZING GENERATION OF SEPARATE, FREE RUNNING, TIME DEPENDENT EQUIPMENT**

[76] Inventor: **Kenneth P. Weiss,** 15 Dwight St., Boston, Mass. 02109

[21] Appl. No.: **802,579**

[22] Filed: **Nov. 27, 1985**

Related U.S. Application Data

[63] Continuation-in-part of Ser. No. 676,626, Nov. 30, 1984, Pat. No. 4,720,860.

[51] **Int. Cl.**[4] **H04L 9/00**

[52] **U.S. Cl.** **380/48**; 380/23; 380/25; 380/28; 235/382; 340/825.31; 340/825.34

[58] **Field of Search** 364/200, 900, 571; 235/382, 380; 375/110; 370/104, 103; 368/46, 47; 380/23–25, 28, 48; 178/22.08, 22.09, 22.17; 340/825.31, 825.34

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,764,742	10/1973	Abbott et al.	380/23
3,789,653	2/1974	Brejand	368/47
3,806,874	4/1974	Ehrat	178/22.08
3,886,451	5/1975	Chu et al.	364/571
3,900,867	8/1975	Wagner	342/45
3,995,111	11/1976	Tsuji et al.	370/104
4,104,694	8/1978	Hargrove	340/825.31
4,126,761	11/1978	Groupe et al.	380/48
4,145,568	3/1979	Ehrat	178/22.17
4,145,569	3/1979	Ehrat	178/22.17
4,185,166	1/1980	Kinch, Jr. et al.	380/43
4,193,073	3/1980	Kohnen	342/56
4,320,387	3/1982	Powell	340/825.34
4,326,098	4/1982	Bouricius et al.	380/25
4,494,211	1/1985	Schwartz	368/47
4,543,657	9/1985	Wilkinson	375/1
4,582,434	4/1986	Plangger et al.	368/47
4,589,066	5/1986	Lam et al.	364/200
4,599,489	7/1986	Cargile	380/4
4,609,777	9/1986	Cargile	380/4
4,636,583	1/1987	Bidell et al.	380/48
4,641,322	2/1987	Hasegawa	375/1
4,677,617	6/1987	O'Connor et al.	375/1 X
4,720,860	1/1988	Weiss	380/23

FOREIGN PATENT DOCUMENTS

0010496	4/1980	European Pat. Off. .
0140013	5/1985	European Pat. Off. .

OTHER PUBLICATIONS

IBM Tech. Discl. Bull.; (vol. 26; No. 7A; 12/83; pp. 3292–3293).

IBM Tech. Discl. Bull.; (vol. 28; No. 7A; 12/83; pp. 3286–3288).

Primary Examiner—Stephen C. Buczinski

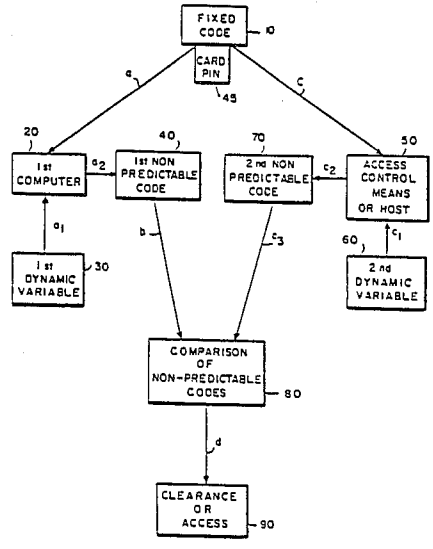
Assistant Examiner—Bernarr Earl Gregory

Attorney, Agent, or Firm—M. Lawrence Oliverio

[57] **ABSTRACT**

An apparatus and method for synchronizing the time definition of the dynamic variables by (a) calculating a first non-predictable code according to a secret predetermined algorithm, the algorithm generating the first non-predictable code on the basis of a first dynamic variable and a unique static variable; (b) automatically defining the first dynamic according to a first interval in which the static variable is input into the algorithm, the first interval of time having a predetermined duration; (c) calculating two or more second non-predictable codes according to the predetermined algorithm, the algorithm generating the second non-predictable codes on the basis of the two or more second dynamic variables and the unique static variable, (d) automatically defining the two or more second dynamic variables according to two or more cells of a second interval of time in which the static variable is input into the algorithm of the second computer, the second interval of time comprising a central cell of time having a predetermined duration and one or more cells of time bordering the central cell of time, each bordering cell of time having a predetermined duration; (e) comparing the first non-predictable code with the second non-predictable codes to determine a match, and (f) automatically synchronizing the clock mechanisms which define the first and second dynamic variables upon comparison and matching of the first non-predictable code with one of the second non-predictable codes.

26 Claims, 5 Drawing Sheets



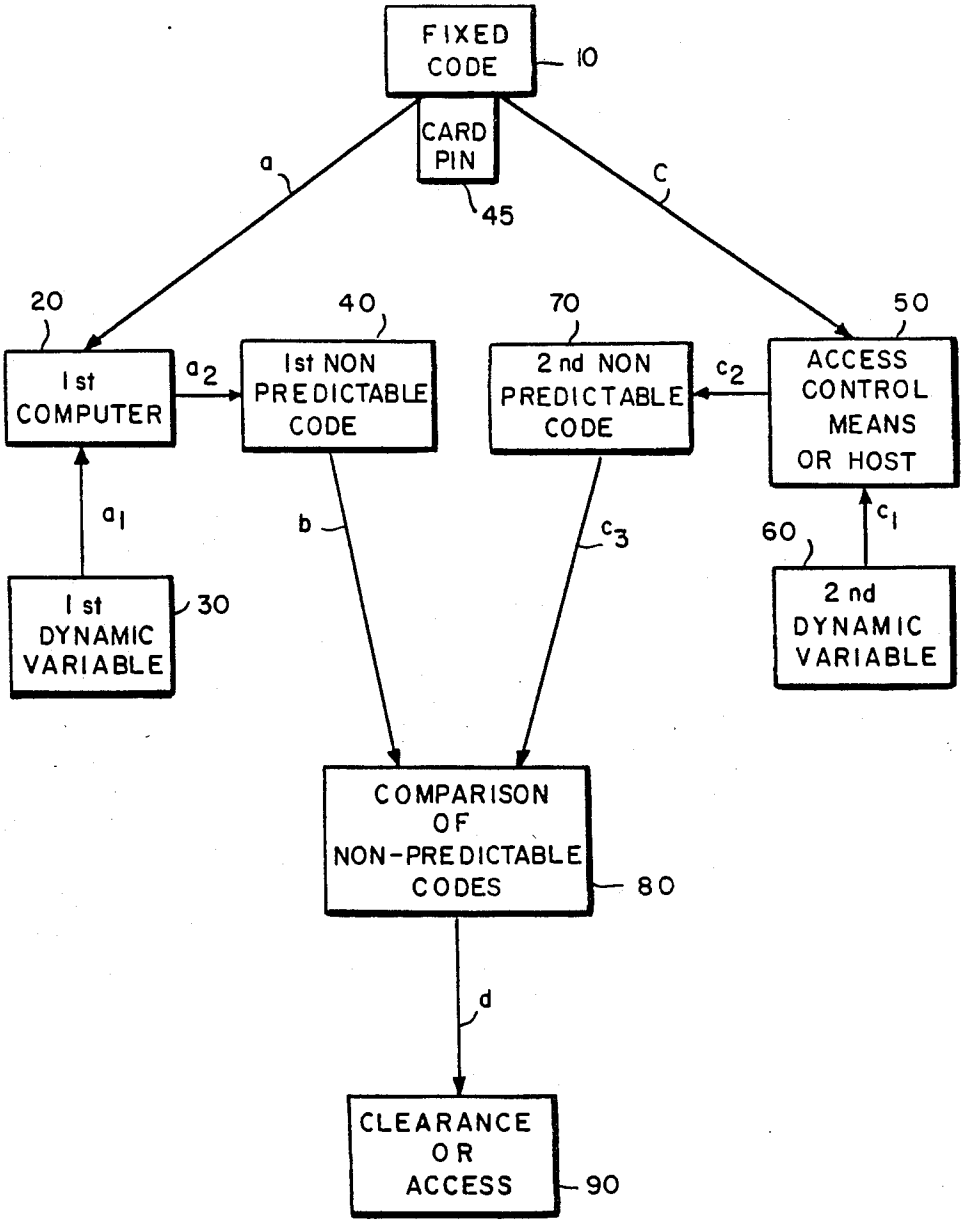

FIXED CODE
10
CARD PIN
45
a
c
20
40
70
50
1st COMPUTER
a2
1st NON PREDICTABLE CODE
2nd NON PREDICTABLE CODE
c2
ACCESS CONTROL MEANS OR HOST
a1
b
c3
c1
60
1st DYNAMIC VARIABLE
30
2nd DYNAMIC VARIABLE
COMPARISON OF NON-PREDICTABLE CODES
80
d
CLEARANCE OR ACCESS
90

FIG. 1

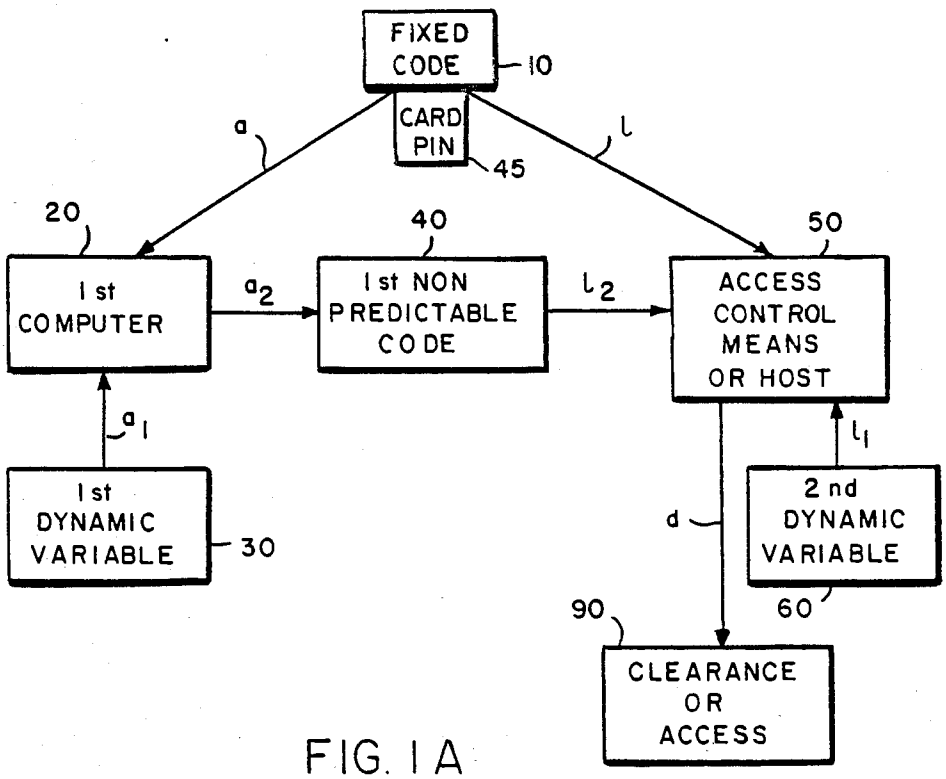

FIXED CODE
10
CARD PIN
45
a
l
20
40
50
1 st COMPUTER
a_2
1 st NON PREDICTABLE CODE
l_2
ACCESS CONTROL MEANS OR HOST
a_1
l_1
1 st DYNAMIC VARIABLE
30
d
2 nd DYNAMIC VARIABLE
60
90
CLEARANCE OR ACCESS

FIG. 1A

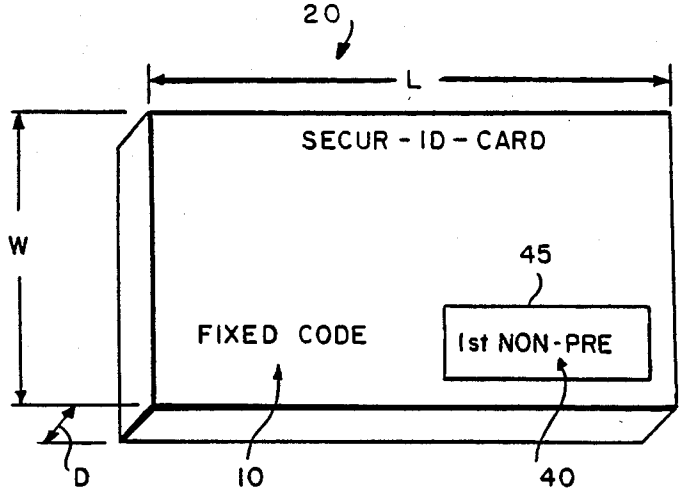

20
L
SECUR - ID - CARD
W
45
FIXED CODE
1st NON-PRE
D
10
40

FIG. 2

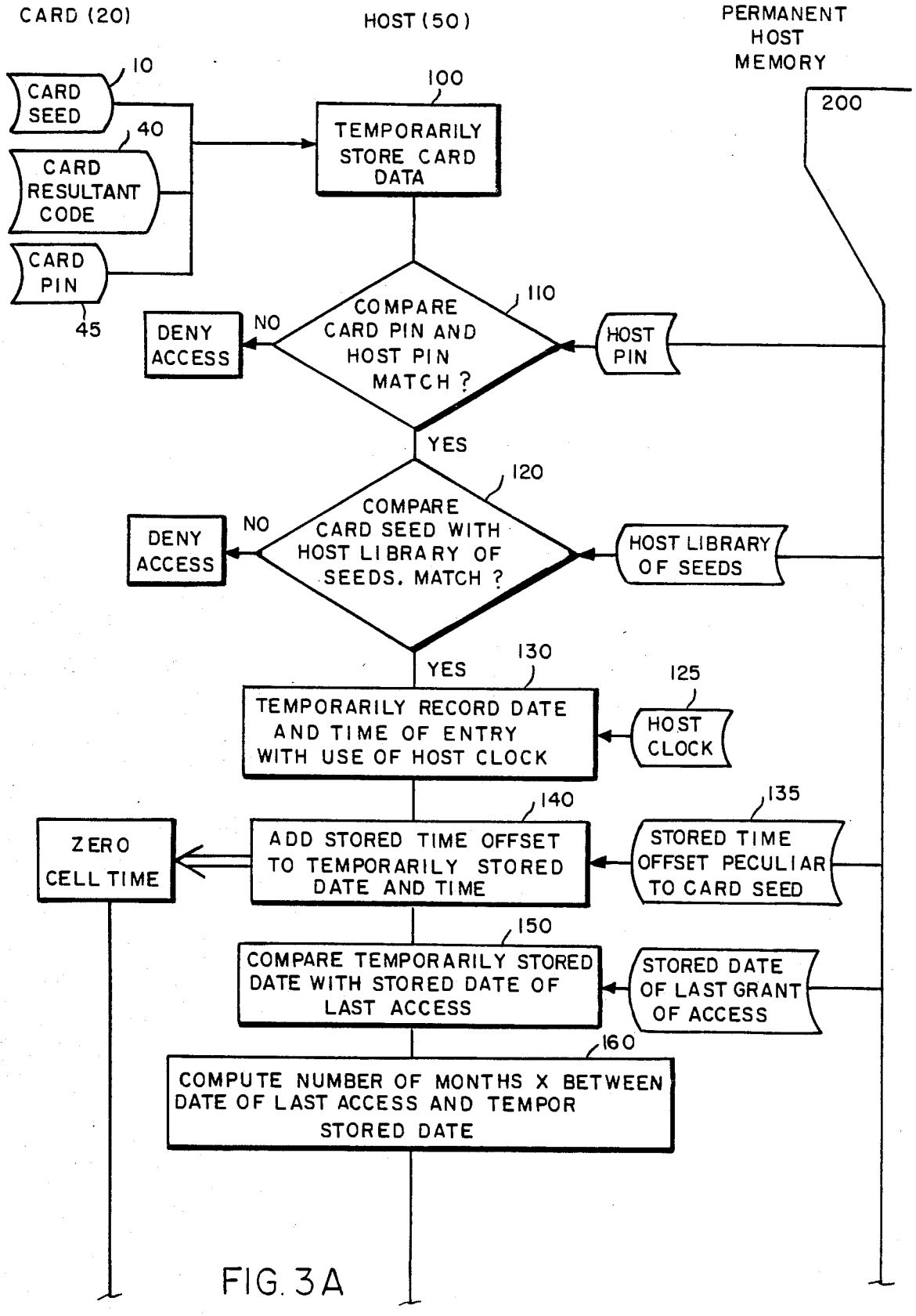

CARD (20)
HOST (50)
PERMANENT HOST MEMORY
10
CARD SEED
40
CARD RESULTANT CODE
CARD PIN
45
100
TEMPORARILY STORE CARD DATA
200
110
COMPARE CARD PIN AND HOST PIN MATCH ?
NO
DENY ACCESS
HOST PIN
YES
120
COMPARE CARD SEED WITH HOST LIBRARY OF SEEDS. MATCH ?
NO
DENY ACCESS
HOST LIBRARY OF SEEDS
YES
130
TEMPORARILY RECORD DATE AND TIME OF ENTRY WITH USE OF HOST CLOCK
125
HOST CLOCK
140
ADD STORED TIME OFFSET TO TEMPORARILY STORED DATE AND TIME
135
STORED TIME OFFSET PECULIAR TO CARD SEED
ZERO CELL TIME
150
COMPARE TEMPORARILY STORED DATE WITH STORED DATE OF LAST ACCESS
STORED DATE OF LAST GRANT OF ACCESS
160
COMPUTE NUMBER OF MONTHS X BETWEEN DATE OF LAST ACCESS AND TEMPOR STORED DATE

FIG. 3A

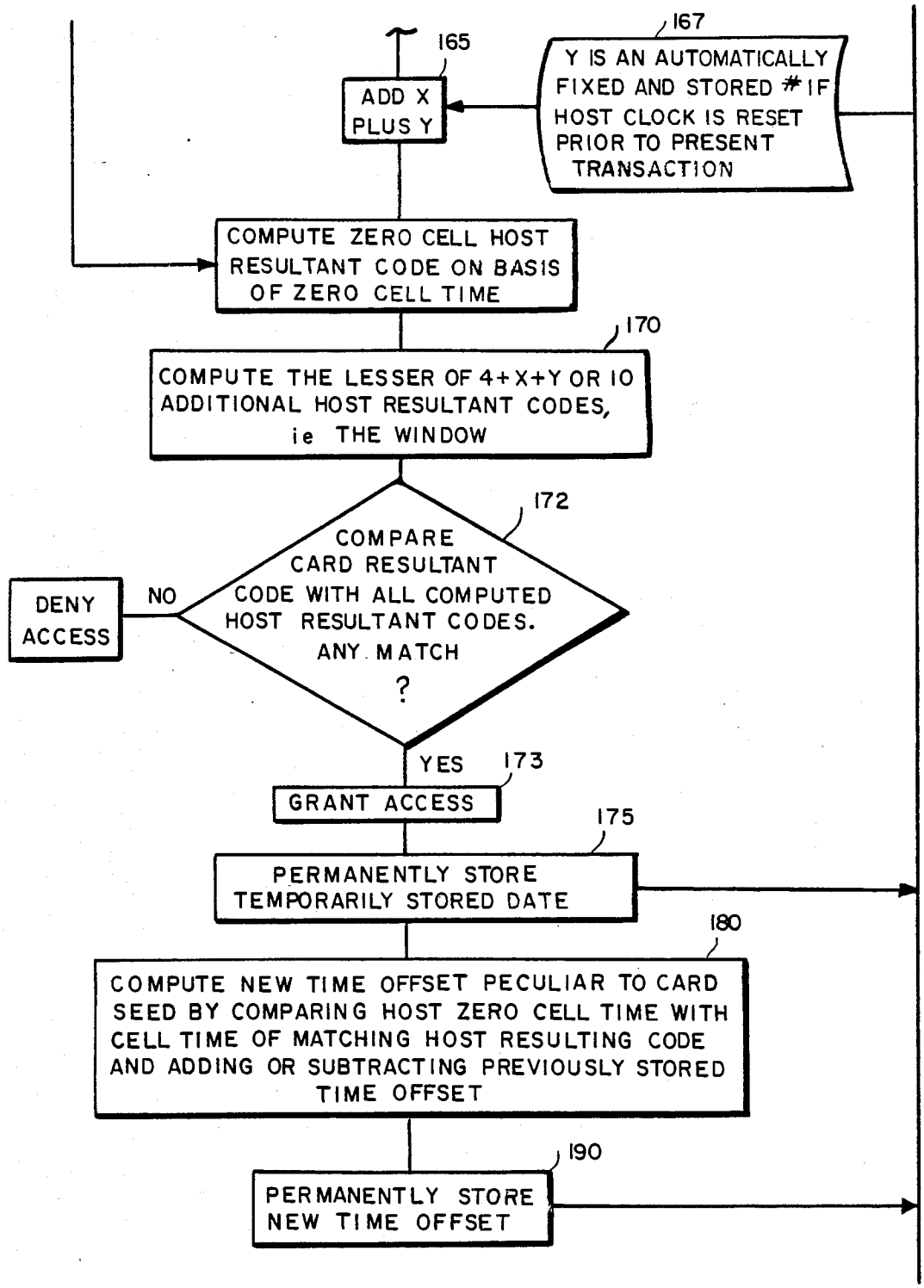

165
ADD X PLUS Y
167
Y IS AN AUTOMATICALLY FIXED AND STORED # IF HOST CLOCK IS RESET PRIOR TO PRESENT TRANSACTION
COMPUTE ZERO CELL HOST RESULTANT CODE ON BASIS OF ZERO CELL TIME
170
COMPUTE THE LESSER OF 4+X+Y OR 10 ADDITIONAL HOST RESULTANT CODES, ie THE WINDOW
172
COMPARE CARD RESULTANT CODE WITH ALL COMPUTED HOST RESULTANT CODES. ANY MATCH ?
NO
DENY ACCESS
YES
173
GRANT ACCESS
175
PERMANENTLY STORE TEMPORARILY STORED DATE
180
COMPUTE NEW TIME OFFSET PECULIAR TO CARD SEED BY COMPARING HOST ZERO CELL TIME WITH CELL TIME OF MATCHING HOST RESULTING CODE AND ADDING OR SUBTRACTING PREVIOUSLY STORED TIME OFFSET
190
PERMANENTLY STORE NEW TIME OFFSET

FIG. 3B

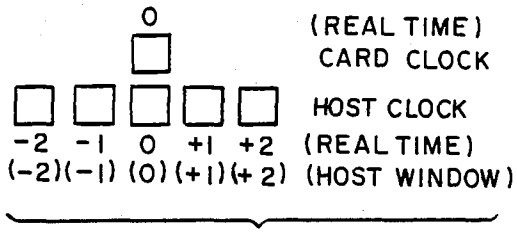


FIG.4

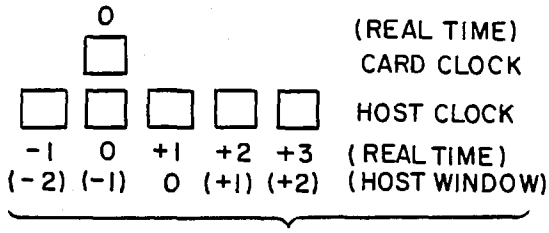


FIG.5

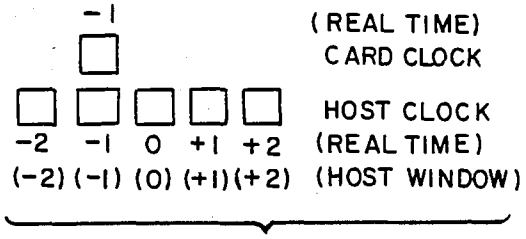


FIG.6

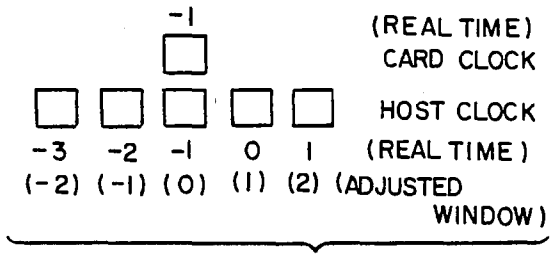


FIG. 7

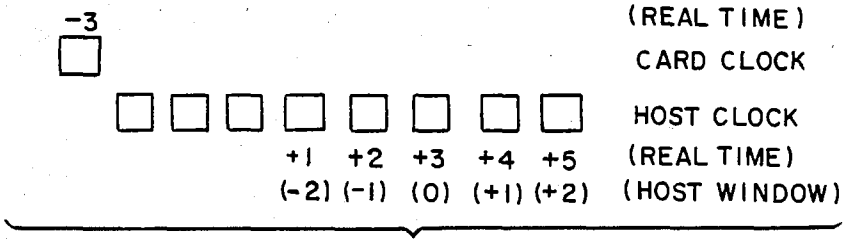


FIG.8

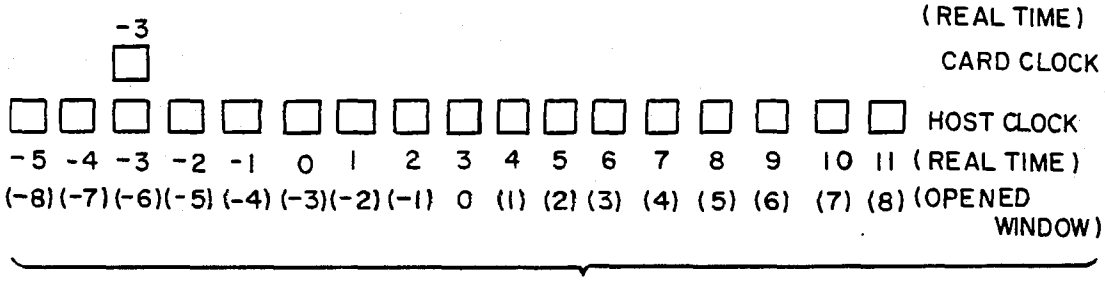


FIG.9

METHOD AND APPARATUS FOR SYNCHRONIZING GENERATION OF SEPARATE, FREE RUNNING, TIME DEPENDENT EQUIPMENT

CROSS-REFERENCE TO OTHER APPLICATION(S)

This is a continuation-in-part of U.S. Ser. No. 676,626 filed Nov. 30, 1984, Applicant: Kenneth Weiss, now U.S. Pat. No. 4,720,860, issued Jan. 19, 1988.

BACKGROUND OF THE INVENTION

The present invention relates to an apparatus and method for the electronic generation of variable, non-predictable codes and the validation and comparison of such codes for the purpose of positively identifying an authorized individual or user of an apparatus or system and thereafter giving clearance to carry out a privileged transaction or access to a protected system or facility.

There often arises a need to prevent all but selected authorized persons from being able to carry out some defined transaction (such as granting of credit) or to gain access to electronic equipment or other system, facility or data (hereinafter "clearance or access"). Prior methods for preventing unauthorized clearance or access typically involve devices which limit access to the subject data, facility, or transaction to those who possess a unique physical device, such as a key or who know a fixed or predictable (hereinafter "fixed") secret code. The problem inherent in relying on a fixed code or unique physical device as the means to gain such selective clearance or access is that would-be unauthorized users need only obtain possession of the fixed code or unique device to gain such clearance or access. Typical instances of fixed codes include card numbers, user numbers or passwords issued to customers of computer data retrieval services.

The principal object of the invention is to synchronize the generation of time-dependent non-predictable codes which are independently generated on the basis of date and time information which are generated on separate devices which over time may deviate out of time synchrony with each other. A further object of the invention is to provide a practical approach to generating identification codes which are unique to the user and which change periodically without user intervention but which provide a readily verifiable means of identification for providing clearance or access at any time.

SUMMARY OF THE INVENTION

The present invention eliminates the relatively easy access afforded to someone who copies or otherwise misappropriates a secret "fixed" code by periodically generating identification codes by using fixed codes, variable data, and a predetermined algorithm which is unknown in advance and unknowable outside the administration of the security system even to authorized users of the apparatus utilizing the fixed secret code. The predetermined algorithm constantly generates new unique and verifiable non-predictable codes, which are derived from the fixed data and at least one dynamic variable, such as the time of day (including the date) by the predetermined algorithm. The constant changes in the dynamic variables when processed by the algorithm results in the generation of constantly changing non-predictable codes.

In accordance with the invention, in a system for comparing and matching non-predictable codes generated by separate computers on the basis of dynamic variables defined by separate clock mechanisms according to time, there is provided an apparatus for synchronizing the time definition of the dynamic variables comprising: a first computer for calculating a first non-predictable code according to a predetermined algorithm, the algorithm generating the first non-predictable code on the basis of a first dynamic variable and a unique static variable; a first clock mechanism for automatically defining the first dynamic variable according to a first interval of time in which the static variable is input into the algorithm, the first interval of time having a first predetermined duration; a second computer for calculating two or more second non-predictable codes according to the predetermined algorithm, the algorithm generating the second non-predictable codes on the basis of the two or more second dynamic variables and the unique static variable; a second clock mechanism for automatically defining the two or more second dynamic variables according to two or more cells of a second interval of time in which the static variable is input into the algorithm of the second computer, the second interval of time comprising a central cell of time having a predetermined duration and one or more cells of time bordering the central cell of time, each bordering cell of time having a predetermined duration; a mechanism for comparing the first non-predictable code with the second non-predictable codes to determine a match; and, a mechanism for automatically synchronizing the first clock mechanism and the second clock mechanism upon comparison and matching of the first non-predictable code with one of the second non-predictable codes.

The central cell of time typically comprises the date and the minute in which the unique static variable is input into the second computer as defined by the second clock mechanism; and the bordering cells of time may comprise a cell of time comprising the date and the minute immediately preceding the central cell.

Preferably the mechanism for synchronizing comprises: a counting mechanism for counting the difference in time between a central cell of time and a bordering cell of time from which a matching second non-predictable code may be generated; a summing mechanism connected to the counting mechanism for summing successive differences in time counted by the counting mechanism; a storage mechanism connected to the summing mechanism for storing the output of the summing mechanism; and, a shifting mechanism connected to the storage mechanism for shifting a central cell and bordering cells of time by the output of the summing mechanism stored in the storage mechanism.

The bordering cells of time may comprise a selected number of cells of time immediately preceding the central cell and a selected number of cells of time immediately following the central cell; and the central and bordering cells of time are typically selected to be one minute in duration.

Preferably, the mechanism for synchronizing further comprises: a second storage mechanism connected to the comparison mechanism for storing the date of the most recent comparison and matching by the comparison mechanism; a second counting mechanism connected to the second storage mechanism for counting

the difference in time between the date stored and the date of present entry into the second computer; a dividing mechanism connected to the second counting mechanism for dividing the difference in time counted by the second counting mechanism by a selected value and prescribing the output as a first window opening number; a window opening mechanism connected to the dividing mechanism and the comparison mechanism for calculating as many extra second non-predictable codes on the basis of as many extra bordering cells of time immediately preceding and following the selected number of bordering cells as prescribed by the first window opening number.

Most preferably, the mechanism for synchronizing further comprises: a sensing mechanism connected to the second clock mechanism for sensing a re-setting of the second clock mechanism; a third storage mechanism connected to the sensing mechanism prescribing and storing the occurrence of a sensed re-setting of the second clock mechanism as a selected second window opening number; and, a second window opening mechanism connected to the third storage mechanism for calculating as many additional second non-predictable codes on the basis of an many additional bordering cells of time immediately preceding and following the extra bordering cells of time as prescribed by the second window opening number.

The first computer typically comprises a microprocessor wherein the algorithm is stored in volatile dynamic memory encapsulated with an energizing mechanism which when interrupted destroys all data including at least the algorithm and the static variable.

Most preferably, the algorithm of the second computer is stored in volatile dynamic memory encapsulated with an energizing mechanism which when interrupted destroys all data including at least the algorithm and the static variable.

In a method for comparing non-predictable codes generated by separate computers on the basis of dynamic variables defined by separate clock mechanisms according to time wherein the codes match when the dynamic variables match, there is also provided a method for synchronizing the time definition of the dynamic variables comprising the steps of: inputting a static variable into a first computer including a predetermined algorithm; employing the algorithm of the first computer to calculate a first non-predictable code on the basis of the static variable and a first dynamic variable defined by a first interval of time in which the step of inputting occurred according to a first clock mechanism; putting the static variable and the first non-predictable code into a second computer independently including the predetermined algorithm; using the algorithm of the second computer to independently calculate two or more second non-predictable codes on the basis of the static variable and two or more second dynamic variables defined by two or more cells of a second interval of time in which the step of putting occurred according to a second clock mechanism, the second interval of time comprising a central cell of time and one or more bordering cells of time; comparing the first non-predictable code with the second non-predictable codes to determine a match; and, synchronizing the first clock mechanism and the second clock mechanism upon comparison and matching of the first non-predictable code with one of the second non-predictable codes.

The step of synchronizing preferably comprises the steps of: counting the difference in time between a central cell of time and a bordering cell of time from which a matching second non-predictable code may be generated; summing successive differences in time counted during the step of counting; storing the summed successive differences in time; and, shifting the central and bordering cells of time by the summed successive differences in time.

Most preferably, the step of synchronizing further comprises the steps of: storing the date of the most recent comparison and determination of a match; counting the difference in time between the date stored and the date of present entry into the second computer; dividing the difference in dates counted by a selected value and prescribing the output as a first window opening number; and, calculating as many extra second non-predictable codes on the basis of as many extra bordering cells of time immediately preceding and following the selected number of bordering cells as prescribed by the first window opening number.

Most preferably, the step of synchronizing further comprises the steps of: sensing a re-setting of the second clock mechanism; prescribing and storing the occurrence of a sensed re-setting of the second clock mechanism as a second selected window opening number; and, calculating as many additional second non-predictable codes on the basis of as many additional bordering cells of time immediately preceding and following the extra bordering cells of time as prescribed by the second window opening number.

The volatile dynamic memory included in either or both of the first computer, the access control means, the host computer and the means for comparing preferably stores and maintains all programs such as the predetermined algorithm, system operating programs, code comparison and matching programs, and the like; and the volatile dynamic memory further preferably stores, maintains and makes available for use all data and results of operations such as fixed codes, resultant codes, dynamic variables and the like.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects, features and advantages will be apparent from the following detailed description of preferred embodiments thereof taken in conjunction with the accompanying drawings in which:

FIG. **1** is a block diagram of a basic apparatus and method according to the invention for generating and comparing non-predictable codes;

FIG. **1A** is a block diagram of a preferred apparatus and method for generating and comparing non-predictable codes where a means for comparing non-predictable codes is included in a calculator which generates a non-predictable code;

FIG. **2** is a front isometric view of a credit card sized calculator for calculating a first non-predictable code for use in gaining clearance or access according to the invention;

FIGS. **3A** and **3B** are a flow chart demonstrating a most preferred series of steps carried out by an apparatus according to the invention and/or in a method according to the invention; and,

FIGS. **4–9** are diagrammatic representations of series of resultant code cells separately generated by separate computers according to exemplary situations described herein; each diagram sets forth the relationship vis a vis real time between resultant codes generated on the basis of time as kept by separate clock mechanisms in the separate computers generating the resultant codes ac-

cording to the corresponding exemplary conditions described with reference to each figure.

DETAILED DESCRIPTION OF THE INVENTION

The following discussion describes the most preferred embodiments of the invention.

In accordance with the invention an authorized person is provided with a fixed secret code or card seed **10**, FIGS. **1**, **1A**, **2**, **3A**, typically a number, which is unique to that individual. In the case of a credit or bank/cash card **20**, FIG. **2**, that number **10** may be printed on the card itself such that if the authorized owner of the card forgets the number, it can be quickly retrieved by reference to the card or other premanently printed form of the fixed code **10**. Where the fixed code/card seed **10** is provided in permanent printed form on or in close connection with the apparatus of the invention there is also preferably provided an additional portion of the fixed code **10**, a so-called pin **45** (personal identification number), which the authorized user memorizes in order to further guard against misappropriation of the fixed code/card seed **10**. The fixed code/card seed **10** or pin **45** may alternatively be used to identify an authorized terminal which has been issued by the authority presiding over the granting of clearance or access.

Such a fixed and/or memorized code (commonly referred to as a pin **45**, FIG. **3A** or personal identification number) is input into an access control module ("ACM") or host computer **50**, FIGS. **1**, **1A**, **3** together with the unique static variable **10** and temporarily stored within the memory of the host or ACM, step **100**, FIG. **3A**.

Preferably once the card seed **10** and pin **45** are input into the host or ACM **50**, each is separately compared against a library of authorized card pins, step **110**, FIG. **3A**, and a library of authorized card seeds, step **120**, FIG. **3A**, stored in the host or ACM memory to determine whether there is a match. If either of the pin **45** or card seed **10** which the user inputs into the host or ACM does not produce a match, clearance or access is denied and the card user must start over in order to gain access or clearance.

In order to generate a non-predictable code **40**, FIGS. **1-3**, which will ultimately give the user clearance or access, the fixed code or seed **10** and/or pin **45** must be input into a predetermined algorithm which manipulates the seed **10** and/or pin **45** as a static variable. The algorithm is typically provided to the user in the form of a calculator **20**, FIG. **2**, which is loaded with a program for carrying out the predetermined algorithm. The calculator **20** preferably comprises an electronic computer and most preferably comprises a microprocessor having a sufficient amount of volatile dynamic memory to store and carry out the functions of the predetermined algorithm. The computer **20** is most preferably provided in a card **20**, FIG. **2**, having the appearance and approximate size of a credit card.

Such credit card sized computer **20**, FIG. **2**, also preferably includes a conventional liquid crystal display **44** for displaying the ultimate non-predictable code **40** generated by the algorithm (referred to in FIG. **3A** as "card resultant code"). The non-predictable code **40** thus generated may be visually observed by the user for eventual input into a host computer or ACM **50**, FIGS. **1**, **1A**. **3**. As shown in FIG. **2**, the preferred form of card computer **20** has a length L of about 3.3 inches, a width W of about 2.1 inches and a depth D of less than about 0.07 inches. In addition or as an alternative to providing microprocessor **20** with a liquid crystal display **45** for visual observation of the first non-predictable code **40**, computer **20** may include means for machine reading the first non-predictable (or card resultant) code **40** and/or pin **45** to the ACM or host **50**, or may include sound producing or other means for personally sensing the first non-predictable code **40**.

With reference to FIG. **3A**, after the card and host pins are compared and found to match, step **110**, the card seed **10** is typically compared against a library of card seeds stored in the host or ACM memory in order to determine whether there is a match, step **120**, FIG. **3A**. If the card seed **10** input into the host or ACM **50** does not match up with one of the seeds stored in the host library, access or clearance is denied, "no" step **120**, FIG. **3A**.

For purposes of initial explanation the discussion which follows with reference to FIGS. **1** and **1A** assumes an embodiment of the invention whereby a single resultant code **70** is generated by the host or ACM **50**. The most preferred embodiment of the invention wherein the clock mechanisms which generate the resultant codes **40** and **70**, are synchronized and wherein the host or ACM preferably generates a series of resultant, non-predictable codes, as opposed to a single code **70**, is described hereinafter with reference to FIGS. **4-9**.

In addition to using the seed **10** and/or pin **45** as static variables the predetermined algorithm is designed to utilize a second variable, a dynamic variable **30**, **60**, FIGS. **1**, **1A**, to calculate the non-predictable codes **40**, **70** which may ultimately give access or clearance **90** to the user. A dynamic variable may comprise any code, typically a number, which is defined and determined by the interval of time in which the card seed **10** and/or pin **45** is put into the algorithm of either the card computer **20** or the host or ACM **50**. A dynamic variable is most preferably defined by the date and the minute in which the static variable is input into the predetermined algorithm. A dynamic variable thus defined can be seen to change every minute. The dynamic variable could alternatively be defined according to any interval of time, e.g., 2 minutes, 5 minutes, 1 hour and the like. A dynamic variable thus defined would alternatively change every 1 minute, 2 minutes, 5 minutes, 1 hour or with the passage of any other predetermined interval of time.

With reference to FIG. **1** the most preferred means of establishing such a dynamic variable is via a time keeping means, such as an electronic digital clock, which by conventional means automatically inputs, steps a_1 or c_1, the date and specific interval of time (e.g., 1 minute, 2 minutes, 5 minutes, etc.) into the predetermined algorithm of the card **20** or host or ACM **50** in response to the input, step a or c, of the static variable **10** and/or pin **45**. The date and time thus generated by the time keeping means may itself be independently manipulated according to another predetermined algorithm prior to input into the first predetermined algorithm of the dynamic variable. The fact that the dynamic variable **30** or **60** being input into the predetermined algorithm constantly changes in absolute value with passage of successive intervals of time of predetermined duration means that the card code **40** or host or ACM code **70** generated according to the predetermined algorithm is also constantly changing with successive intervals of time and is thereby completely non-predictable.

The non-predictability of the codes **40**, **70**, FIG. **1**, generated in the manner described above may be en-

hanced by the fact that the predetermined algorithm (together with the static variable **10** and/or pin **45** and dynamic variable **30** input thereinto) are preferably stored in the calculator **20** (and/or host or ACM **50**) in volatile dynamic electronic memory which is encapsulated with an energizing means which destroys the algorithm, the card seed **10**, and the dynamic variable **30** (or **60**) when the electronic memory is invaded, interrupted or violated in any way. The predetermined algorithm thus stored in such volatile dynamic memory cannot be discovered by a would-be thief because the entire memory including the predetermined algorithm is destroyed upon invasion of the memory.

In a preferred embodiment of the invention therefor the card seed **10** is stored in such volatile dynamic memory and by conventional means is automatically input step a, FIGS. **1**, **1A**, into the algorithm of the first computer **20** at regular intervals of time. Such automatic inputting of the card seed **10** may thereby work in conjunction with the automatic definition and inputting of the first dynamic variable **30** into the predetermined algorithm of the first computer **20** to effect completely automatic generation of the first non-predictable or resultant code **40** at regular intervals of time.

The invention most preferably contemplates providing authorized personnel with a card computer **20**, FIG. **2**, only, but not with knowledge of the predetermined algorithm included in the computer **20**. Authorized personnel are, therefore, provided with a computer **20** capable of carrying out an algorithm which is unknown to such authorized personnel.

In the most preferred embodiment of the invention where the predetermined algorithm provided to authorized users is stored in a volatile dynamic memory encapsulated with an energizing means which destroys the algorithm upon invasion of the memory, the only means of gaining unauthorized clearance or access is to misappropriate possession of the original computer **20** itself and knowledge of the fixed code/card seed **10** (and knowledge of the card pin **45** if employed in conjunction with the invention).

The algorithm may alternatively be designed to manipulate more than one fixed code and/or more than one dynamic variable. Several means for inputting each fixed code and dynamic variable may be included in the calculator **20** provided to users and in the host or ACM **50**, FIG. **3A**. Each dynamic variable is preferably defined by the interval of time in which one or more of the fixed codes/card seeds are input into the algorithm.

It can be seen, therefore, that the predetermined algorithm can comprise any one of an infinite variety of algorithms. The only specific requirement for an algorithm to be suitable for use in the present invention is that such algorithm generate a non-predictable code on the basis of two classes of variables, static variables (the fixed codes) and dynamic variables such as described hereinabove. A non-predictable code C which is ultimately generated by the predetermined algorithm, f (x,y), may be expressed mathematically as:

$$f(x,y)=C$$

where x is a static variable/fixed code and y is a dynamic variable. Where several (n) static variables (x_1, x_2, . . . x_n) and several (n) dynamic variable (y_1, y_2, . . . y_n) are intended for use in generating non-predictable codes, a non-predictable code thus generated may be expressed mathematically as $f(x_1, x_2, \ldots x_n, y_1, y_2, \ldots y_n)=C$.

The specific form of the algorithm only assumes special importance as part of the invention, therefore, when the algorithm is capable of being discovered by would-be unauthorized users. In the most preferred embodiment of the invention where the algorithm is completely undiscoverable by virtue of its storage in a volatile dynamic electronic memory which destroys the algorithm upon attempted invasion of the encapsulated memory, the specific form of the algorithm comprises only an incidental part of the invention. The mere fact of the use of some algorithm to manipulate the fixed code and the dynamic variable does, however, comprise a necessary part of the invention insofar as such an algorithm generates the ultimately important non-predictable code.

As the term "fixed code" or "card seed" or "seed" is used herein such terms include within their meaning numbers, codes, or the like which are themselves manipulated or changed, mathematically or otherwise, in some non-dynamic manner prior to or during the generation of a second non-predictable code **40**, FIG. **3A**. The first **20** or second computer **50** may, for example, be provided with a static program/algorithm utilizing the fixed code or seed as a variable and generating a new fixed code or seed which is ultimately input as the fixed code or seed **10** variable in the secret algorithm which generates the non-predictable codes. For example, for purposes of added security, a fixed code or seed **10** may be first added to another number and the result thereof used as the fixed code or seed **10** used to generate the non-predictable codes. Thus, the term fixed code or seed includes within its meaning the result of any non-dynamic operation performed on any fixed code or seed. It can be seen, therefore, that essentially any algorithm or operation may be performed on the fixed code **10** to generate another fixed code or seed, the algorithm or operation most preferably comprising a static algorithm or operation, i.e., one not utilizing dynamic variables so as to generate a static result.

With reference to FIG. **1**, after a first non-predictable code **40** is generated as described above, such first non-predictable code **40** is compared **80** with the "second" non-predictable code **70** which is also generated by the user by putting, step c, the fixed code/card seed **10** (and the pin **45**, if employed) into the host or ACM **50** which contains the same predetermined algorithm used to generate the first non-predictable code **40**.

With reference to FIG. **1A**, (a schematic diagram which assumes the host or ACM **50** includes the predetermined algorithm and the mechanism for comparing and matching the non-predictable codes) the first non-predictable code **40** is put, step e_2, into the host or ACM **50** essentially immediately after the fixed secret code **10** is put into the host or ACM **50** (i.e., step e_2 is carried out essentially immediately after step e) in order to gain clearance or access **90**. If steps e and e_2 are not carried out within the same interval of time as steps a and a_1, were carried out, (i.e., the same interval of time on which code **40** is based), then the host or ACM will not generate a second dynamic variable **60** which will allow the predetermined algorithm of the host or ACM **50** to generate a second non-predictable code which matches the 1st non-predictable code **40**.

The necessity for carrying out steps e and e_2, FIG. **1A**, within the same minute or other selected interval of time ("cell") is obviated in a most preferred embodi-

ment of the invention. With reference to FIGS. 3–4, the card 20 generates a resultant code 40, on the basis of a cell of time in which the code 40 was generated as defined by the card clock. Assuming for the sake of explanation that the card clock and the host or ACM clock 125 are synchronized with each other and with real time and assuming the user inputs the correct card seed 10 and resultant code 40 into the host or ACM 50 within the same cell of time as the resultant code 40 was generated by the card 20 the host 50 is preferably provided with a program which generates a series or "window" of resultant codes (as opposed to a single non-predictable code 70, FIG. 1). [As used hereinafter, the term "cell" is, depending on the context, intended to refer to an interval of time of predetermined duration on which the generation of a resultant code is based or to the resultant code itself.] The various second non-predictable codes which comprise the "window" are calculated by the host or ACM 50 on the basis of the cell of time in which the user correctly entered the seed 10, code 40, and pin 45 into the host or ACM 50 as defined by the host clock 125, FIG. 3A, and one or more bordering cells of time, e.g., −2, −1, and +1, +2 as shown in FIG. 4. An ACM or host computer 50 program then compares the card resultant code 40 with all of the individual resultant codes computed as the window of host cells shown in FIG. 4 to determine whether there is a match between any of the host cells and the card code 40. In the example stated, the card code 40 will of course match up, step 172, FIG. 3B, with the zero cell based host code, FIG. 4 because the user input the seed 10, pin 45 and code 40 within the same cell of time as the card code 40 was generated.

[As used hereinafter, "input" or "inputting" or "entry" into the host or ACM 50 refers to input of the correct card seed 10, card resultant code 40 and card pin 45 into the host or ACM 50 and positive matching of the card seed 10, step 120, FIG. 3A, and card pin 45, step 110, with a host seed and host pin which are stored in the permanent memory in the host or ACM 50].

Assuming in the example stated above with reference to FIG. 4, however, that the user had input the card code 40 and seed 10 (and pin 45), FIG. 3A, one minute later than the card had generated the code 40, the host or ACM 50 will have generated a different window of codes as shown in FIG. 5; that is, the host will have generated a central cell corresponding to a +1 cell code (based on real time) as if the +1 cell code is the zero cell of the window of cell (as shown in parenthesis in FIG. 5) and further generate the predetermined number of bordering cell codes (e.g., real time −1, 0 and +2, +3 as shown in FIG. 5). Thus although the user inputs the card seed 10 and the card resultant code 40 into the host or ACM 50 one minute late, the host computer 50 still generates a matching cell code, the real time zero cell code which "borders" the central cell, i.e., the +1 central cell code as shown in parenthesis in, FIG. 5.

Provision of the host or ACM 50, FIGS. 3A, 3B–5 with a mechanism for generating a series or window of second non-predictable codes, as opposed to a single second code 70, FIG. 1, thereby allows a card user a selected amount of leeway of time (beyond the time length of the interval of time on which code 40 is based) in which to input a correct seed 10, pin 45 and card code 40 into the host or ACM 50 and still generate a matching host resultant code.

The examples stated above assumed that the card clock and the host clock 125, FIG. 3A, were both synchronized with real time. Assuming the card clock and the host clock remain synchronized at all times, it would only be necessary to provide the host or ACM 50 with a mechanism for generating a selected number of bordering cells which "precede" the central cell of the window, e.g., with reference to FIG. 5, the (−2), (−1), (0) cells In those applications where the card clock and the host clock are maintained in synchrony with each other at all times, the host or ACM clock 125 preferably defines only two dynamic time variables so as to generate a central cell code and a −1 host window cell code. Such embodiment allows the user to input to seed 10, pin 45 and code 40 one cell code late but only one cell code late for security enhancement.

In the more typical case, however, where the card clock and the host clock 125 may be out of synchrony with real time, e.g., where the card clock is running fast relative to the host clock, the generation of cells which "follow" the central cell of the host window may be required to generate a matching host resultant code.

With reference to FIGS. 3A, 3B, 6 the invention most preferably provides a mechanism for synchronizing the card and host clocks in the case where such independent clocks more typically run fast or slow relative to real time and/or relative to each other.

The following examples assume for purposes of explanation that the time equivalent length of all cell codes are one minute in duration. Assuming that the card clock is one minute slow and the host clock 125, FIG. 3A is correct relative to real time, the card will generate a resultant code 40 based on a real time of −1 minute (relative to the host clock 125) and, if the user inputs the card resultant code 40 (and the correct seed 10 and pin 45) into the host or ACM 50 within the same minute as the code 40 is generated, the host or ACM 50 will generate a window of resultant codes according to the series of cells shown in FIG. 6 (assuming the predetermined number of bordering cells is selected as 2 cells immediately preceding and 2 cells immediately following). Matching resultant codes, i.e., the card −1 cell code and the host −1 cell code, will thus have been generated.

Although the card clock was one minute slow in the example just described, the host computer is provided with a program mechanism which will automatically adjust (i.e., synchronize) the host clock time with the card clock time when the card user next enters a correct card seed 10 and card pin 45 (and code 40) into the host or ACM 50. The host accomplishes such synchronization by storing a difference in matching cell time in the permanent memory of the host, step 190, FIG. 3B; e.g., in the example just described, the last matching transaction, step 180, FIG. 3B fell in the −1 cell of the host "window" as shown in parenthesis in FIG. 6. Such cell time difference is referred to herein as the "time offset" which is stored in the permanent host memory, step 190, FIG. 3B. The time offset is the difference in time between the central cell and the bordering cell from which a matching second non-predictable code was generated.

Upon the next entry of the card user into the host 50 (assuming the card clock has not run any slower since the last entry and assuming the host clock has remained synchronized with real time and assuming the user next enters the host 50 within the same minute as the card generates resultant code 40), the host computer 50 will automatically algebraically add the store time offset, steps 135, 140, FIG. 3A, to the temporarily stored host

clock time, step **130**, and generate the series of relative real time host cell codes shown in FIG. **7** wherein the card code cell which is one minute slow in real time, is now treated in the host window as a zero cell (as shown in parenthesis in FIG. **7**), i.e., the central cell of the host window of cells, is adjusted to subtract one minute therefrom, via subtraction of the one-minute stored time offset **135**, FIG. **3A**. As shown in FIG. **7**, the bordering cells of the host window are similarly adjusted by the one-minute stored time offset. Further, in all future entries by the user into the host **50**, the temporarily stored time and date of entry, step **130**, FIG. **3A**, will be adjusted by the permanently recorded one-minute stored time offset.

As to the example described above with reference to FIG. **5** wherein the card and host clocks were assumed to be synchronized with real time and wherein the user entered the host one minute late, it is noted that even though the host clock was synchronized with real time, the host will nevertheless compute a time offset, step **180**, FIG. **3B**, to be stored, step **190**, and used in adjusting the temporarily stored time of entry, step **130**, FIG. **3A**, in future transactions by the user, because the matching cell of the host window, as shown in parenthesis FIG. **5**, was not the central cell code of the window (i.e., was not the real time +1 cell code) but rather was a bordering real time cell code, i.e., the bordering real time zero cell code.

Simply stated, therefore, a stored time offset will be computed step **180**, FIG. **3B**, and stored, step **190**, FIG. **3**, for use in adjusting the time of entry into the host in all future entries, step **140**, FIG. **3A**, whenever on a given entry, step **130**, FIG., **3**, a "bordering" cell code of the host window (as opposed to the central cell code) produces a match with the input card resultant code **40**.

In storing, step **190**, FIG. **3B**, a time offset which is computed, step **180**, during any given transaction, the presently computed time offset is algebraically added or summed to any time offsets previously computed and stored as a result of previous entries and grantings of access, step **173**.

Inasmuch as a clock mechanism, once beginning to run fast or slow, will continue to run fast or slow during all future uses of the system of the invention, the host or ACM **50** will add or subtract all time offsets recorded during successive uses of the system to the stored time offset(s) recorded and permanently stored from previous transactions, step **180**, FIG. **3B**. Most preferably, a newly computed time offset will not be permanently stored, step **190**, in the host or ACM memory **200**, unless and until access or clearance has already been granted, step **173**.

As described and shown in the examples of FIGS. **4–7** the host or ACM is typically programmed to compute four (4) cell codes bordering the central cell code (i.e., two cells immediately preceding and two cells immediately following the central cell) as the "window" within which the user is allowed to deviate in inputting the card seed **10**, the pin **45**, and, the card resultant code **40** into the host or ACM. Such bordering cells have been described as corresponding to codes corresponding to one-minute intervals. It is noted that the number and time equivalent length of the bordering cells may be increased or decreased as desired.

The absolute degree by which the card clock and the host or ACM clock **125** may run fast or slow relative to real time typically increases with the passage of time. For example, if the card clock is running slow by 30 seconds per month and the host clock is running fast at 30 seconds per month, the two clocks will run the time equivalent of one minute out of synchrony after one month, two minutes out of synchrony after two months, three minutes out of synchrony after three months, etc. If the authorized card user uses the card each month, the automatic synchronizing means described above with reference to FIGS. **4–7** will have adjusted the host or ACM time window upon each usage to account for such lack of synchrony with real time. If, however, the card user does not actually use the card for, for example, six months, the card clock and the host clock will be six minutes out of synchrony, and even if the user correctly uses the system by inputting the pin **45**, card seed **10** and card code **40**, FIG. **3A**, into the host or ACM within the same minute (or other selected time cell interval) as the pin **45**, the seed **10** and code **40** were generated by the card, the user would not be able to gain access or clearance (i.e., cause the host or ACM to generate a matching resultant code) in the typical situation where the "window" of bordering cell times is selected as two one-minute cells immediately preceding and two one-minute cells immediately following the central host cell. FIG. **8** depicts such an exemplary situation as just described, wherein it can be seen that the card clock, after six months of non-usage, generates a resultant code **40**, FIG. **3**, which is based on −3 minutes in real time, and the host clock, after six months of non-usage, causes the generation of the typically selected five cell window comprising cell codes corresponding to +1, +2, +3, +4, and +5 minutes in relative real time. In the typical case, therefore, where the selected window comprises four bordering cells, matching second non-predictable codes will not be generated under any circumstances after six months of non-use.

The invention most preferably provides a mechanism by which the host window of bordering cells is opened wider than the preselected window by an amount which varies with the length of time of non-use of the card. Such window opening is accomplished by storing the most recent date of comparison and matching, determining the difference in time between such date and the present date of entry into the second computer and calculating as many additional bordering cells as may be predetermined according to the difference in time between the dates.

Typically the window is opened by two one minute bordering cells per month of non-use (e.g., one cell immediately preceding and one cell immediately following the preselected window) but the number of cells by which the window is opened and the time equivalent length of each cell may be predetermined to comprise any other desired number and length.

Assuming the exemplary situation described above where the card clock and the host clock **125**, FIG. **3A**, are running slow and fast respectively by 30 seconds per month and the user has not used the card for six months, the host or ACM compares, step **150**, the temporarily stored date of the present entry, step **130**, with the permanently stored date of the last access, step **175**, and computes the number of months X, step **160**, between the date of last access and the date of present entry. In the present example six months of non-use is calculated step **160**, FIG. **3A**, and the window is opened by six additional one-minute bordering cells on either side of the preselected four cell window as shown in FIG. **9** to give an overall window of sixteen minutes. The card resultant code **40** based on −3 minutes in relative real

time thus matches, step **172**, FIG. **3A**, as shown in FIG. **9** with the −6 host bordering cell code (−3 in real time) and access or clearance is ultimately granted. As described above with reference to FIGS. **4–7**, because the matching host cell code is a bordering cell code of the host window and not the central host cell (i.e., the zero cell), a new stored time offset of −6 minutes will be computed (i.e., added to the permanently stored time offset), step **180**, FIG. **3B**, and stored, step **190**, and the host clock thereafter will adjust the zero cell of the host window (and accompanying bordering cells) each time the user of the card having the particular card seed **10** and pin **45** which was used in the present transaction uses the card to gain access in future transactions.

Lastly the invention further includes a failsafe window opening mechanism to provide for the contingency where the host or ACM **50** and its clock **125**, FIG. **3A**, may shut down between card usages. In the event of such a shut down, the host or ACM clock **125** must typically be reset and re-synchronized, and in the course of such re-setting an error may be made in the resynchronization. In order to insure that the card user may reasonably gain access in the event of such an error in re-setting the host clock **125**, the host or ACM **50** is preferably provided with a mechanism for sensing such a re-setting and for storing a predetermined window opening number upon each re-setting of the host or ACM **125**. Such window opening number is typically selected as six additional one-minute bordering cells(e.g., three additional cells immediately preceding and three additional cells immediately following the existing window) but may be selected as more or fewer cells of other selected length.

The re-setting window opening number is typically added, step **165**, FIG. **3B**, to the result of non-usage step **160** and the total additional number of cells comprising the window is computed, step **170**, FIG. **3B**. i.e., all bordering cells surrounding the central cell are computed including (a) the preselected window allowing for user delay in inputting and/or card and host clock asynchrony, (b) the non-usage window allowing for card and host clock asynchrony over long periods of time of non-usage and (c) the re-setting window opening number.

Assuming the exemplary situation described above with reference to FIG. **9**, if the host or ACM had shut down within the six month period of non-use, the host window as depicted in FIG. **9** would be further opened by an additional six bordering cells such that −11, −10, −9 and +9, +10, +11 host window cells would also have been computed, step **170**, FIG. **3B**, and made available for comparison and potential matching with card resultant code **40** in step **172**, FIG. **3B**. As described with reference to FIGS. **5–9**, where a new time offset is computed and stored, steps **180**, **190**, FIG. **3B**, as a result of a match found in a bordering cell of a window generated by virtue of non-usage and/or the preselected window, a new time offset will similarly be computed and stored, steps **180**, **190**, if a match is found in a bordering cell generated as a result of shutdown.

Unlike the non-usage window opening number, the re-set window opening number is typically stored in the permanent memory **200** of the host or ACM **50**, FIG. **3A**, such that once the host clock **125** is re-set, the selected window opening number is available in permanent memory **200** to open the window upon the next attempted entry by the user. Although the re-set window opening number is established and stored in permanent memory **200**, such re-set window opening number is preferably eventually closed down or eliminated for security enhancement after it is established upon successive attempted entries by a variety of card users that the host clock **125** was correctly reset or after the host clock **125** is otherwise re-synchronized with real time to correct any errors which may have occurred as a result of the re-setting. The use of the re-set window opening number is, therefore, preferably temporary.

In the practical application of the invention, many cards are issued to many users and each card includes its own card clock. Recognizing that the average of the times being kept by the individual clocks of a statistically significant sample of a variety of cards, will produce an accurate or very nearly accurate representation of real time, the invention most preferably includes a mechanism for permanently adjusting the time kept by the host clock **125**, FIG. **3A**, after the clock **125** has been re-set, to the average of the times of entry (after re-setting of the host clock **125**) of a selected number of different cards or card users. For example, assuming that host clock **125** has been reset, the next time of entry of the next five (or other selected number of) separate card users is averaged, the host clock **125** is permanently adjusted or re-synchronized to such an average time, and the re-set window opening number is thereafter eliminated from the permanent host memory **200**. Re-adjusting or re-synchronization of the host clock **125** to the averaged time of the card clocks is typically carried out by the host **50** by computing another master time offset which is algebraically added to the time offsets peculiar to each card **20**. The computation of such a master offset assumes that a selected number of separate cards **20** were able to gain access, step **173**, FIG. **3B** as a result of the re-set window opening or otherwise. The average of the time offsets computed as to the selected number of cards which enter the host **50** (after the host clock **125** is re-set) is preferably stored as a master time offset (i.e., as a re-synchronization of the host clock **125**), the re-set window opening number is then eliminated as to all future entries by card users, and the master time offset is used (in addition to permanently stored time offsets peculiar to each card) to adjust the card clock **125** in transactions as to all card entries thereafter.

As a practical matter, a limit is typically placed on the total number of bordering cells by which the window is opened regardless of the length of time of non-usage by the card user or the number of times the host or ACM **50** is reset as a result of re-setting of clock **125**. For security reasons, such a limit is typically selected as ten one-minute bordering cells - as stated in step **170**, FIG. **3B** the number of codes comprising the windown are the lesser of (a) 4 bordering cell codes, the preferred selected window, plus X, the number of months or other selected non-usage periods, plus Y, the shut down window opening number, or (b) 10, the maximum number of additional cell codes. Such a maximum window may, of course, be selected as more or less than 10 depending on the degree of security desired.

It is noted that FIGS. **3A**, **3B** depicts a preferred sequence of operations and not necessarily the only sequence. Steps **110** and **120** could, for example, be interchanged or, for example, the step of automatically inputting the re-set window opening number, step **167** could precede any of steps **140–160**.

The host or ACM **50**, FIGS. **1**, **1A**, **3A**, **3B**, typically includes one or more programs and sufficient memory

to carry out all of the steps shown in FIGS. 3A, 3B, although one or more of those functions may be carried out by a device separate from and communicating with or connected to the host or ACM **50**.

With respect to the computation, storage and retrieval of time offsets, the host or ACM **50** is provided with mechanisms for recognizing, storing, retrieving and computing time offsets which are peculiar to each card seed **10** and/or pin **45** and responsive to the input of the same into the host or ACM **50**.

FIG. **2** depicts the most preferred form of the calculator **20** which is provided to authorized users for generating the first non-predictable or card resultant code **40**. As shown in FIG. **2** the calcultor **20** is of substantially the same size as a conventional credit card and includes a conventional liquid crystal display **44** for displaying the code **40** to the user. The credit/card computer **20**, FIG. **2**, may bear the identity of the card seed/fixed code **10** printed on its face, and includes a digital clock means, an energizing means, a microprocessor and sufficient memory for storing the predetermined secret algorithm, a program for generating a dynamic variable if desired, and the card seed **10** and pin **45** if desired.

In an embodiment of the invention where the goal is to grant access to a physical facility, the ACM **50** may comprise a portable device such that it may be carried by a security guard stationed at a central access location leading to a guarded building or other facility. A security guard thus in possession of such an ACM would typically read the card seed **10** and the non-predictable code **40** appearing on the card **20**, FIG. **2**, of authorized person and input such codes **10**, **40** (in addition to the pin **45** - otherwise provided to the guard by the card bearer) into the portable ACM **50** to determine whether the card bearer is truly in possession of a card **20** which was issued by the authority establishing the secret predetermined algorithm.

As described herein protection of the secrecy of the predetermined algorithm is preferably accomplished in the calculators provided to authorized personnel by virtue of its storage in volatile dynamic memory and encapsulation with a volatile dynamic energizing means. With respect to the algorithm provided in the ACM secrecy may be maintained in a similar manner or other conventional manner, e.g., by physically guarding the ACM or requiring additional access/user codes to gain direct access. Where all programs, data and results of operation are stored in such volatile dynamic memory, the same are similarly protected against invasion.

Although the invention contemplates some form of communication of the result of operation **40** carried out on the card **20**, FIG. **2**, to the host or ACM **50** or any other electronic device, a talking between the computer **20** and the host **50** is not required or contemplated by the invention. Therefore, after the first computer **20** has calculated the first non-predictable code **40** and the code **40** has been input into the host **50**, no other information need be communicated back to the first computer **20** from the host **50** or another device in order to gain clearance or access.

Lastly it is noted that the fixed code or seed **10** and/or pin **45**, FIGS. **3A**, **3B**, may be employed to identify a computer terminal or other piece of equipment or device as opposed to a card **20**. For example, a terminal or a space satellite or other device may be provided with a computer **20** which is assigned a code or seed **10** and/or a pin **45** (and, of course, provided with the secret predetermined algorithm and a clock and conventional electronic mechanisms for computing the code **40** and inputting the code **10**, pin **45**, and resultant code **40** to the host or ACM **50**) in order to identify such terminal, satellite or the like in the same manner as a card computer **20** is identifiable as dscribed hereinabove.

It will not be apparent to those skilled in the art that other embodiments, improvements, details, and uses can be made consistent with the letter and spirit of the foregoing disclosure and within the scope of this patent, which is limited only by the following claims, construed in accordance with the patent law, including the doctrine of equivalents.

What is claimed is:

1. In a system for comparing and matching non-predictable codes generated by separate computers on the basis of dynamic variables defined by separate first and second clock means according to time, an apparatus for effectively synchronizing the first and second clock means comprising:

a first computer for calculating a first non-predictable code according to a predetermined algorithm, the algorithm generating the first non-predictable code on the basis of a first dynamic variable and a unique static variable;

said first clock means automatically defining the first dynamic variable according to a first interval of time in which the static variable is input into the algorithm, the first interval of time having a first predetermined duration;

a second computer for calculating two or more second non-predictable codes according to the predetermined algorithm, the algorithm generating the second non-predictable codes on the basis of two or more second dynamic variables and the unique static variable;

said second clock means automatically defining the two or more second dynamic variables according to two or more time cells for which the static variable is input into the algorithm of the second computer, the time cells comprising a central cell of time having a predetermined duration and one or more cells of time bordering the central cell of time, each bordering cell of time having a predetermined duration;

means for comparing the first non-predictable code with the second non-predictable codes to determine a match; and

means for automatically synchronizing the first clock means and the second clock means upon comparison and matching of the first non-predictable code with one of the second non-predictable codes.

2. The system of claim **1** wherein the central cell of time comprises the date and the minute in which the unique static variable is input into the second computer as defined by the second clock means.

3. The system of claim **2** wherein the bordering cells of time comprise a cell of time comprising the date and the minute immediately preceding the central cell.

4. The system of claim **1** wherein the means for synchronizing comprises:

counting means for counting the difference in time between a central cell of time and a bordering cell of time from which a matching second non-predictable code may be generated;

summing means connected to the counting means for summing successive differences in time counted by the counting means;

storage means connected to the summing means for storing the output of the summing means; and,

shifting means connected to the storage means for shifting a central cell and bordering cells of time by the sumined times stored in the storage means.

5. The system of claim **4** wherein the bordering cells of time comprise a selected number of cells of time immediately preceding the central cell and a selected number of cells of time immediately following the central cell.

6. The system of claim **5** wherein the central and bordering cells of time are selected to be one minute in duration.

7. The system of claim **5** wherein the means for synchronizing further comprises:

second storage means connected to the comparison means for storing the date of the most recent comparison and matching by the comparison means;

second counting means connected to the second storage means for counting the difference in time between the date stored and the date of present entry into the second computer;

dividing means connected to the second counting means for dividing the difference in time counted by the second counting means by a selected value and prescribing the output as a first window opening number;

window opening means connected to the dividing means and the comparison means for calculating as many extra second non-predictable codes on the basis of as many extra bordering cells of time immediately preceding and following the selected number of bordering cells as prescribed by the first window opening number.

8. The system of claim **7** wherein the means for synchronizing further comprises:

sensing means connected to the second clock means for sensing a re-setting of the second clock means;

third storagae means connected to the sensing means for prescribing and storing the occurrence of a sensed re-setting of the second clock means as a selected second window opening number; and

second window opening means connected to the third storage means for calculating as many additional second non-predictable codes on the basis of as many additional bordering cells of time immediately preceding and following the extra bordering cells of time as prescribed by the second window opening number.

9. The system of claim **8** wherein the first computer comprises a microprocessor wherein the algorithm is stored in volatile dynamic memory encapsulated with an energizing means which when interrupted destroys all data including at least the algorithm and the static variable.

10. The system of claim **9** wherein the first computer and the first clock means are incorporated into a card of about the same size as a credit card.

11. The system of claim **10** wherein the algorithm of the second computer is stored in volatile dynamic memory encapsulated with an energizing means which when interrupted destroys all data including at least the algorithm and the static variable.

12. In a method for comparing non-predictable codes generated by separate computers on the basis of dynamic variables defined by separate clock means according to time wherein the codes match when the dynamic variables match, a method for effectively synchronizing the separate clock means comprising the steps of:

inputting the static variable into a first computer including a predetermined algorithm;

employing the algorithm of the first computer to calculate a first non-predictable code on the basis of the static variable and a first dynamic variable defined by a first interval of time in which the step of inputting occurred according to a first clock means;

putting the static variable and the first non-predictable code into a second computer, the second computer independently including the predetermined algorithm;

using the algorithm of the second computer to independently calculate two or more second non-predictable codes on the basis of the static variable and two or more second dynamic variables defined by two or more cells of time in which the step of inputting occurs according to a second clock means, the cells of time comprising a central cell of time and one or more bordering cells of time;

comparing the first non-predictable code with the second non-predictable codes to determine a match; and

synchronizing the first clock means and the second clock means upon comparison and matching of the first non-predictable code with one of the second non-predictable codes.

13. The method according to claim **12** wherein the step of synchronizing comprises the steps of:

counting the difference in time between a central cell of time and a bordering cell of time from which a matching second non-predictable code may be generated;

summing successive differences in time counted during the step of counting;

storing the summed successive differences in time; and,

shifting the central and bordering cells of time by the summed successive differences in time.

14. The method according to claim **13** wherein the step of synchronizing further comprises the steps of:

storing the date of the most recent comparison and determination of a match;

counting the difference in time between the date stored and the date of present entry into the second computer;

dividing the difference counted during the step of counting the difference in dates by a selected value and prescribing the output as a first window opening number; and,

calculating as many extra second non-predictable codes on the basis of as many extra bordering cells of time immediately preceding and following the selected number of bordering cells as prescribed by the first window opening number.

15. The method according to claim **13** wherein the step of synchronizing further comprises the steps of:

sensing a re-setting of the second clock means;

prescribing and storing the occurrence of a sensed re-setting of the second clock means as a second selected window opening number; and,

calculating as many additional second non-predictable codes on the basis of as many additional bordering cells of time immediately preceding and following the extra bordering cells of time as prescribed by the second window opening number.

16. The method of claim 12 wherein the central and bordering cells of time are selected to be one minute in duration.

17. In an identification system utilizing a first computer means to generate a first non-predictable code sequence as a function of a first time dependent variable in accordance with a predetermined algorithm, a first clock means for generating a sequence of said first time dependent variables during successive time cells, each of which cells is of a predetermined time interval, means for applying the output of said first clock means to said first computer means, a second computer means to generate a second non-predictable code sequence as a function of a second time dependent variable in accordance with said predetermined algorithm, a second clock means for generating a sequence of said second time dependent variables during successive time cells, means for applying the output of said second clock means to said second computer means and means for comparing the first and second non-predictable codes at a selected time cell; a means forming part of said second computer means to compensate for loss of synchronism between the first and second clock means comprising:

means for computing the second non-predictable code for at least one additional time cell bordering said selected time cell;

first storing means for storing the second non-predictable codes computed during said selected time cell and during each of said at least one additional time cells;

wherein said means for comparing includes means for comparing the first non-predictable code for said selected time cell with each of the second non-predictable codes stored in said first storing means; and

means for establishing identification if the first non-predictable code matches any of the second non-predictable codes.

18. A means to compensate as claimed in claim 17 including means for determining the number of time cells between the time cell for the second non-predictable code which matches the first non-predictable code and said selected time cell;

summing means for summing successively determined numbers of time cells;

second storing means for storing the output of the summing means; and

means responsive to the summed number stored in said means for summing for shifting the time cells for which said second non-predictable codes are generated for storing in said first storing means.

19. A means to compensate as claimed in claim 18 wherein said first storing means normally stores the second non-predictable codes for said selected time cell, for a selected number of time cells immediately preceding said selected time cell and for a selected number of time cell immediately following said selected time cell; and

wherein said means for shifting shifts the time cell for which each of the stored second non-predictable codes are determined by the value of the summed number stored in said means for summing.

20. A means to compensate as claimed in claim 19 including means for storing the date of the last successful comparison by said means for comparing;

means for determining the time between the current date and the stored date of last successful comparison;

means for dividing the time determined by said means for determining by a selected value and designating the output as a first window opening number; and

first window opening means for utilizing said first window opening number to determine a number of extra second non-predictable codes to be determined and stored in said first storing means for time cells immediately preceding and following time intervals for the codes otherwise stored therein.

21. A means to compensate as claimed in claim 19 including means for sensing a resetting of the second clock means;

means responsive to the sensing of a resetting of the second clock means for storing a selected second window opening number; and

second window opening means for utilizing the second window opening number to determine the number of extra second non-predictable codes to be determined and stored in said first storing means for time cells immediately preceding and following the time intervals for the codes otherwise stored therein.

22. In a method for effecting identification which involves the steps of generating a first non-predictable code sequence as a function of a first time dependent variable in accordance with a predetermined algorithm by use of a first computer means, utilizing a first clock means to generate a sequence of said first time dependent variables during successive time cells, each of which cells is of a predetermined time interval, applying the output of the first clock means to the first computer means, utilizing a second computer means to generate a second non-predictable code sequence as a function of a second time dependent variable in accordance with said predetermined algorithm, utilizing a second clock means to generate a sequence of said second time dependent variables during seuccessive time cells applying the output of the second clock means to the second computer means and comparing the first and second non-predictable codes at a selected time cell, a method for utilizing the second computer means to compensate for loss of synchronism between the first and second clock means, comprising the steps of:

computing the second non-predictable code for at least one additional time cell bordering said selected time cell;

storing the second non-predictable codes computed for said selected time cell and for each of said at least one additional time cells;

comparing the first non-predictable code for said selected time cell with each of the stored second non-predictable codes; and

establishing identification if the first non-predictable code matches any of the stored second non-predictable codes.

23. A method for compensating as claimed in claim 22 including the steps of determining the number of time cells between the time cell for the second non-predictable code which matches the first non-predictable code and said selected time cell;

summing successively determined numbers of time cells;

storing the summed number of time cells; and

shifting the time cells for which said second non-predictable codes are generated for storing and comparison with said first non-predictable code by a value dependent on the summed number of time intervals.

24. A method for compensating as claimed in claim 23, wherein the second non-predictable codes stored are for the selected time cell, a selected number of time cells immediately preceding said selected time cell and a selected number of time cells immediately following said selected time cell; and

wherein during said shifting step, the time cell for which each of the stored second non-predictable codes is determined is shifted by the value of the number of time cells determined during said summing step.

25. A method for compensating as claimed in claim 24 including the steps of storing the date of the last successful comparison during said comparing step;

determining the time between the current date and the stored date of last successful comparison;

dividing the time determined during said determining step by a selected value and designating the output as a first window opening number; and

utilizing said first window opening number to determine a number of extra second non-predictable codes to be determined and stored for time cells immediately preceding and following the time cells for the codes otherwise stored for comparison with said first non-predictable code.

26. A method for compensating as claimed in claim 24 including the steps of sensing a resetting of the second clock means;

storing a selected second window opening number in response to the sensing of a resetting of the second clock means; and

utilizing the second window opening number to determine a number of extra second non-predictable codes to be determined and stored for time cells immediately preceding and following the time cells for the codes otherwise stored for comparison with the first non-predictable code.

* * * * *