Network (In)Security Through IP Packet Filtering

D. Brent Chapman

Great Circle Associates

Brent@GreatCircle.COM
+1 415 962 0841

1057 West Dana Street
Mountain View, CA 94041

ABSTRACT

Ever-increasing numbers of IP router products are offering packet filtering as a tool for improving network security. Used properly, packet filtering is a useful tool for the security-conscious network administrator, but its effective use requires a thorough understanding of its capabilities and weaknesses, and of the quirks of the particular protocols that filters are being applied to. This paper examines the utility of IP packet filtering as a network security measure, briefly contrasts IP packet filtering to alternative network security approaches such as application-level gateways, describes what packet filters might examine in each packet, and describes the characteristics of common application protocols as they relate to packet filtering. The paper then identifies and examines problems common to many current packet filtering implementations, shows how these problems can easily undermine the network administrator's intents and lead to a false sense of security, and proposes solutions to these problems. Finally, the paper concludes that packet filtering is currently a viable network security mechanism, but that its utility could be greatly improved with the extensions proposed in the paper.

1. Introduction

This paper considers packet filtering as a mechanism for implementing network security policies. The consideration is from the point of view of a site or network administrator (who is interested in providing the best possible service to their users while maintaining adequate security of their site or network, and who often has an "us versus them" attitude with regard to external organizations), which is not necessarily the same point of view that a service provider or router vendor (who is interested in providing network services or products to customers) might have. An assumption made throughout is that a site administrator is generally more interested in keeping outsiders out than in trying to police insiders, and that the goal is to keep outsiders from breaking in and insiders from accidentally exposing valuable data or services, not to prevent insiders from intentionally and maliciously subverting security measures. This paper does not consider military-grade "secure IP" implementations (those that implement the "IP security options" that may be specified in IP packet headers) and related issues; it is limited to what is commonly available for sale to the general public.

Packet filtering may be used as a mechanism to implement a wide variety of network security policies. The primary goal of these policies is generally to prevent unauthorized

Published in *Proceedings of the Third USENIX UNIX Security Symposium*; Baltimore, MD; September, 1992.

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

network access without hindering authorized network access; the definitions of "unauthorized access" and "authorized access" vary widely from one organization to another. A secondary goal is often that the mechanisms be transparent in terms of performance, user awareness, and application awareness of the security measures. Another secondary goal is often that the mechanisms used be simple to configure and maintain, thus increasing the likelihood that the policy will be correctly and completely implemented; in the words of Bill Cheswick of AT&T Bell Laboratories, "Complex security isn't". Packet filtering is a mechanism which can, to a greater or lesser extent, fulfill all these goals, but only through thorough understanding of its strengths and weaknesses and careful application of its capabilities.

Several factors complicate implementation of these policies using packet filtering, including asymmetric access requirements, differing requirements for various internal and external groups of machines, and the varying characteristics of the particular protocols, services, and implementations of these protocols and services that the filters are to be applied to. Asymmetric access requirements usually arise when an organization desires that its internal systems have more access to external systems than vice versa. Differing requirements arise when an organization desires that some groups of machines have different network access privileges than other groups of machines (for instance, the organization may feel that a particular subnet is more secure than standard, and thus can safely take advantage of expanded network access, or they may feel that a particular subnet is especially valuable, and thus its exposure to the external network should be as limited as possible). Alternatively, an organization may desire to allow more or less network access to some specific group of external machines than to the rest of the external world (for instance, a company might want to extend greater network access than usual to a key client with whom they are collaborating, and less network access than usual to a local university which is known to be the source of repeated cracker attacks). The characteristics of particular protocols, services, and implementations also greatly affect how effective filtering can be; this particular issue is discussed in detail below, in Section 3 and Appendix A.

Common alternatives to packet filtering for network security include securing each machine with network access and using application gateways. Allowing network access on an all-or-nothing basis (a very coarse form of packet filtering) then attempting to secure each machine that has network access is generally impractical; few sites have the resources to secure and then monitor every machine that needs even occasional network access. Application gateways, such as those used by AT&T [Ches90], Digital Equipment Corporation [Ranum92], and several other organizations, are also often impractical because they require internal hosts to run modified (and often custom-written or otherwise not commonly available) versions of applications (such as "ftp" and "telnet") in order to reach external hosts. If a suitably modified version of an application is not available for a given internal host (a modified TELNET client for a personal computer, for instance), that internal host's users are simply out of luck and are unable to reach the past the application gateway.

2. How Packet Filtering Works

2.1. What packet filters base their decisions on

Current IP packet filtering implementations all operate in the same basic fashion; they parse the headers of a packet and then apply rules from a simple rule base to determine whether to route or drop† the packet. Generally, the header fields that are available to the filter

† "Permit" and "deny" are used synonymously with "route" and "drop" throughout this paper. If a router decides to "permit" or "route" a packet, it is passed through to its destination as if filtering never occurred. If a router decides to "deny" or "drop" a packet, the packet is simply discarded, as if it never existed; depending on the filter-

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

are packet type (TCP, UDP, etc.), source IP address, destination IP address, and destination TCP/UDP port. For some reason, the source TCP/UDP port is often *not* one of the available fields; this is a significant deficiency discussed in detail in Section 4.2.

In addition to the information contained in the headers, many filtering implementations also allow the administrator to specify rules based on which router interface the packet is destined to go out on, and some allow rules based on which interface the packet came in on. Being able to specify filters on both inbound and outbound† interfaces allows you significant control over where the router appears in the filtering scheme (whether it is "inside" or "outside" your packet filtering "fence"), and is very convenient (if not essential) for useful filtering on routers with more than two interfaces. If certain packets can be dropped using inbound filters on a given interface, those packets don't have to be mentioned in the outbound filters on all the other interfaces; this simplifies the filtering specifications. Further, some filters that an administrator would like to be able to implement require knowledge of which interface a packet came in on; for instance, the administrator may wish to drop all packets coming inbound from the external interface that claim to be from an internal host, in order to guard against attacks from the outside world that use faked internal source addresses.

Some routers with very rudimentary packet filtering capabilities don't parse the headers, but instead require the administrator to specify byte ranges within the header to examine, and the patterns to look for in those ranges. This is almost useless, because it requires the administrator to have a very detailed understanding of the structure of an IP packet. It is totally unworkable for packets using IP option fields within the IP header, which cause the location of the beginning of the higher-level TCP or UDP headers to vary; this variation makes it very difficult for the administrator to find and examine the TCP or UDP port information.

2.2. How packet filtering rules are specified

Generally, the filtering rules are expressed as a table of conditions and actions that are applied in a certain order until a decision to route or drop the packet is reached. When a particular packet meets all the conditions specified in a given row of the table, the action specified in that row (whether to route or drop the packet) is carried out; in some filtering implementations [Mogul89], the action can also indicate whether or not to notify the sender that the packet has been dropped (through an ICMP message), and whether or not to log the packet and the action taken on it. Some systems apply the rules in the sequence specified by the administrator until they find a rule that applies [Mogul89][Cisco90], which determines whether to drop or route the packet. Others enforce a particular order of rule application based on the criteria in the rules, such as source and destination address, regardless of the order in which the rules were specified by the administrator. Some, for instance, apply filtering rules in the same order as

---

ing implementation (and sometimes on the filtering specification), the router might send an ICMP message (usually "host unreachable") back to the source of a packet that is dropped, or it might simply pretend it never received the packet.

† Throughout this paper, the terms "inbound" and "outbound" are usually used to refer to connections or packets from the point of view of the protected network as a whole, and sometimes used to refer to packets from the point of view of the filtering router (which is at the edge of the internal network, between the internal network and the external world), or to the router interfaces those packets will pass through. A packet might appear to be "inbound" to the filtering router on its way to the external world, but that packet is "outbound" from the internal network as a whole. An "outbound connection" is a connection initiated from a client on an internal machine to a server on an external machine; note that while the connection as a whole is outbound, it includes both outbound packets (those from the internal client to the external server) and inbound packets (those from the external server back to the internal client). Similarly, an "inbound connection" is a connection initiated from a client on an external machine to a server on an internal machine. The "inbound interface" for a packet is the interface on the filtering router that the packet appeared on, while the "outbound interface" is the interface the packet will go out on if it isn't denied by the application of the filtering rules.

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

routing table entries; that is, they apply rules referring to more specific addresses (such as rules pertaining to specific hosts) before rules with less specific addresses (such as rules pertaining to whole subnets and networks) [CHS91][Telebit92a]. The more complex the way in which the router reorders rules, the more difficult it is for the administrator to understand the rules and their application; routers which apply rules in the order specified by the administrator, without reordering the rules, are easier for an administrator to understand and configure, and therefore more likely to yield correct and complete filter sets.

2.3. A packet filtering example

For example, consider this scenario. The network administrator of a company with Class B network 123.45 wishes to disallow access from the Internet to his network in general (123.45.0.0/16)†. The administrator has a special subnet in his network (123.45.6.0/24) that is used in a collaborative project with a local university which has class B network 135.79; he wishes to permit access to the special subnet (123.45.6.0/24) from all subnets of the university (135.79.0.0/16). Finally, he wishes to deny access (except to the subnet that is open to the whole university) from a specific subnet (135.79.99.0/24) at the university, because the subnet is known to be insecure and a haven for crackers. For simplicity, we will consider only packets flowing from the university to the corporation; symmetric rules (reversing the SrcAddr and DstAddr in each of the rules below) would need to be added to deal with packets from the corporation to the university. Rule C is the "default" rule, which specifies what happens if none of the other rules apply.

Rule	SrcAddr	DstAddr	Action
A	135.79.0.0/16	123.45.6.0/24	permit
B	135.79.99.0/24	123.45.0.0/16	deny
C	0.0.0.0/0	0.0.0.0/0	deny

Consider these "sample" packets, their desired treatment under the policy outlined above, and their treatment depending on whether the rules above are applied in order "ABC" or "BAC".

Packet	SrcAddr	DstAddr	Desired Action	ABC action	BAC action
1	135.79.99.1	123.45.1.1	deny	deny (B)	deny (B)
2	135.79.99.1	123.45.6.1	permit	permit (A)	*deny (B)*
3	135.79.1.1	123.45.6.1	permit	permit (A)	permit (A)
4	135.79.1.1	123.45.1.1	deny	deny (C)	deny (C)

A router that applies the rules in the order ABC will achieve the desired results: packets from the "hacker haven" subnet at the university to the company network in general (such as packet 1 above) will be denied (by rule B), packets from the university "hacker haven" subnet at the university to the company's collaboration subnet (such as packet 2 above) will be

† Throughout this paper, the syntax "*a.b.c.d/y*" denotes "the address *a.b.c.d*, with the top *y* bits significant for comparison". In other words, 123.45.0.0/16 means that the top 16 bits (123.45) are significant for comparisons to other addresses. The address 123.45.6.7 thus matches 123.0.0.0/8, 123.45.0.0/16, and 123.45.6/24, but not 123.45.99.0/24. A pattern with 0 significant bits (such as 0.0.0.0/0) matches any address, while a pattern with 32 significant bits (such as 123.45.6.7/32) matches only that particular address (123.45.6.7). This syntax is a simpler form of expressing an address pattern than the traditional "address, wildcard mask" tuple, particularly when the boundary between the wildcarded and non-wildcarded bits doesn't fall on an 8-bit boundary (for instance, on a Cisco router, the pattern 123.0.0.0/8 would be represented as "123.0.0.0 0.255.255.255", 123.45.6.0/24 would be represented as "123.45.6.0 0.0.0.255", and 123.45.6.240/28 would be represented as "123.45.6.240 0.0.0.15"). This syntax was originated in the *KA9Q* networking package for PCs, and is used in the Telebit *NetBlazer* and other products.

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

permitted (by rule A), packets from the university's general network to the company's "open" subnet (such as packet 3 above) will be permitted (by rule A), and packets from the university's general network to the company's general network (such as packet 4 above) will be denied (by rule C).

If, however, the router reorders the rules by sorting them into order by number of significant bits in the source address then number of significant bits in the destination address, the same set of rules will be applied in the order BAC. If the rules are applied in the order BAC, packet 2 will be denied, when we want it to be permitted.

2.4. Packet filtering caveats

2.4.1. Complexity of packet filtering specifications

In fact, there's a subtle error in this example that illustrates how difficult it is to correctly set up filters using such low-level specifications. Rule B above, which appears to restrict access from the "hacker haven" net, is actually superfluous and unnecessary, and is the cause of the incorrect denial of packet 2 if the rules are applied in the order BAC. If you remove rule B, both types of routers (those that apply rules in the order specified, and those that reorder rules by number of significant bits in source or destination addresses) will process the rules in the order AC. When processed in that order, the result table becomes:

Packet	SrcAddr	DstAddr	Desired Action	AC action
1	135.79.99.1	123.45.1.1	deny	deny (C)
2	135.79.99.1	123.45.6.1	permit	permit (A)
3	135.79.1.1	123.45.6.1	permit	permit (A)
4	135.79.1.1	123.45.1.1	deny	deny (C)

There are two points here. First, correctly specifying filters is difficult. Second, reordering filtering rules makes correctly specifying filters even more difficult, by turning a filter set that works (even if it's in fact overspecified) if evaluated in the order given into a filter set that doesn't work.

Even though the example presented above is a relatively simple application of packet filtering, most administrators will probably read through it several times before they feel they understand what is going on. Consider that the more difficult the rules are to comprehend, the less likely the rules will be correct and complete. The way in which filtering rules must be specified and the order in which they are applied are key determinants of how useful and powerful a given router's filtering capabilities are. Most implementations require the administrator to specify filters in ways which make the filters easy for the router to parse and apply, but make them very difficult for the administrator to comprehend and consider.

2.4.2. Reliance on accurate IP source addresses

Most filtering implementations, of necessity, rely on the accuracy of IP source addresses to make filtering decisions. IP source addresses can easily be faked, however, as discussed in [Bellovin89], [Kent89], [Bellovin92a], and [Bellovin92b]. This is a particular case where being able to filter inbound packets is useful. If a packet that appears to be from one internal machine to another internal machine comes in over the link from the outside world, you should be mighty suspicious. If your router can be told to drop such packets using inbound filters on the external interface, your filtering specifications for internal interfaces can be made both much simpler and more secure.

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.