US005875296A

United States Patent [19]

Shi et al.

[11] Patent Number: 5,875,296

[45] Date of Patent: Feb. 23, 1999

[54] **DISTRIBUTED FILE SYSTEM WEB SERVER USER AUTHENTICATION WITH COOKIES**

[75] Inventors: **Shaw-Ben Shi**; **Michael Bradford Ault**, both of Austin; **Ernst Robert Plassmann**, Pflugerville; **Bruce Arland Rich**, Round Rock; **Mickella Ann Rosiles**, Austin; **Theodore Jack London Shrader**, Cedar Park, all of Tex.

[73] Assignee: **International Business Machines Corporation**, Armonk, N.Y.

[21] Appl. No.: **790,041**

[22] Filed: **Jan. 28, 1997**

[51] **Int. Cl.**[6] **G06F 11/00**

[52] **U.S. Cl.** **395/188.01**; 395/188.01; 395/200.54

[58] **Field of Search** 395/186, 187.01, 395/188.01, 200.59, 200.33, 200.54, 200.47, 200.48, 200.49; 380/4, 24, 49

[56] **References Cited**

U.S. PATENT DOCUMENTS

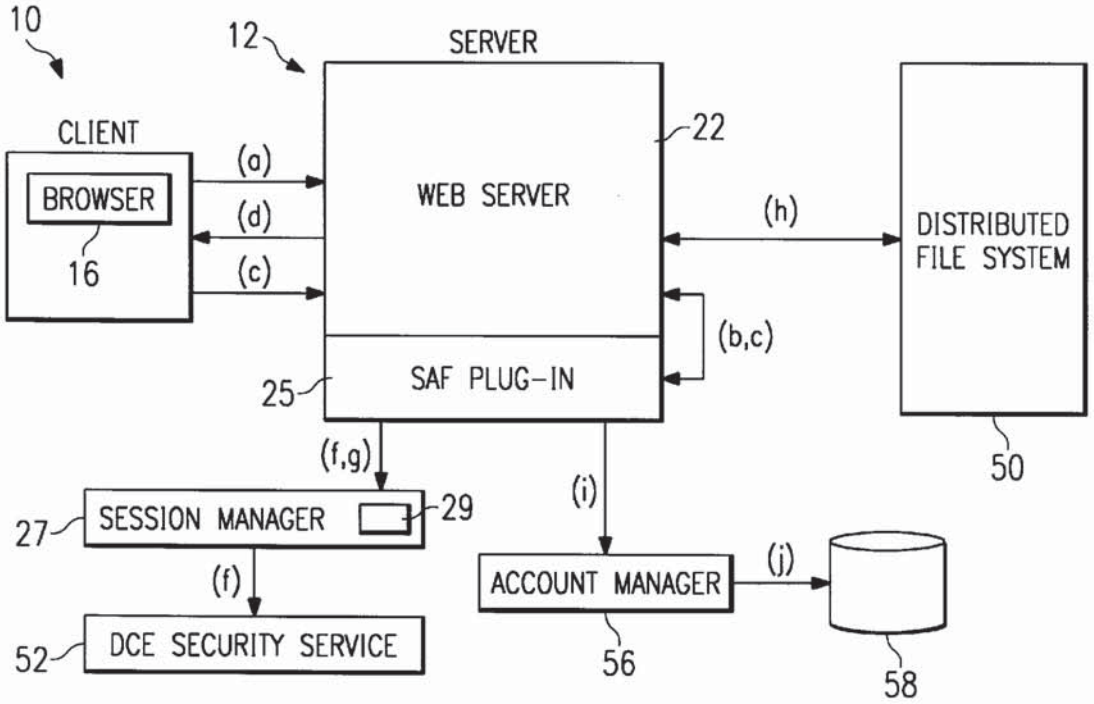
4,578,531	3/1986	Everhart et al.	380/21
5,187,790	2/1993	East et al. .	
5,491,752	2/1996	Kaufman et al.	380/30
5,497,463	3/1996	Stein et al.	200/200.33
5,530,852	6/1996	Meske, Jr. et al.	395/200.36
5,572,643	11/1996	Judson	395/200.48
5,644,711	7/1997	Murphy	395/188.01
5,678,041	10/1997	Baker et al.	395/188.01
5,708,780	1/1998	Levergood et al. .	
5,734,831	3/1998	Sanders	395/200.53
5,796,952	8/1998	Davis et al.	395/200.54

Primary Examiner—Robert W. Beausoliel, Jr.
Assistant Examiner—Pierre E. Elisca
Attorney, Agent, or Firm—Jeffrey S. LaBaw; David H. Judson

[57] **ABSTRACT**

A method of authenticating a Web client to a Web server connectable to a distributed file system of a distributed computing environment. The distributed computing environment includes a security service for returning a credential to a user authenticated to access the distributed file system. In response to receipt by the Web server of a user id and password from the Web client, a login protocol is executed with the security service. If the user can be authenticated, a credential is stored in a database of credentials associated with authenticated users. The Web server then returns to the Web client a persistent client state object having a unique identifier therein. This object, sometimes referred to as a cookie, is then used to enable the Web client to browse Web documents in the distributed file system. In particular, when the Web client desires to make a subsequest request to the distributed file system, the persistent client state object including the identifier is used in lieu of the user's id and password, which makes the session much more secure. In this operation, the cookie identifier is used as a pointer into the credential storage table, and the credential is then retrieved and used to facilitate multiple file accessess from the distributed file system. At the same time, the Web client may obtain access to Web server (as opposed to distributed file system) documents via conventional user id and password in an HTTP request.

20 Claims, 3 Drawing Sheets



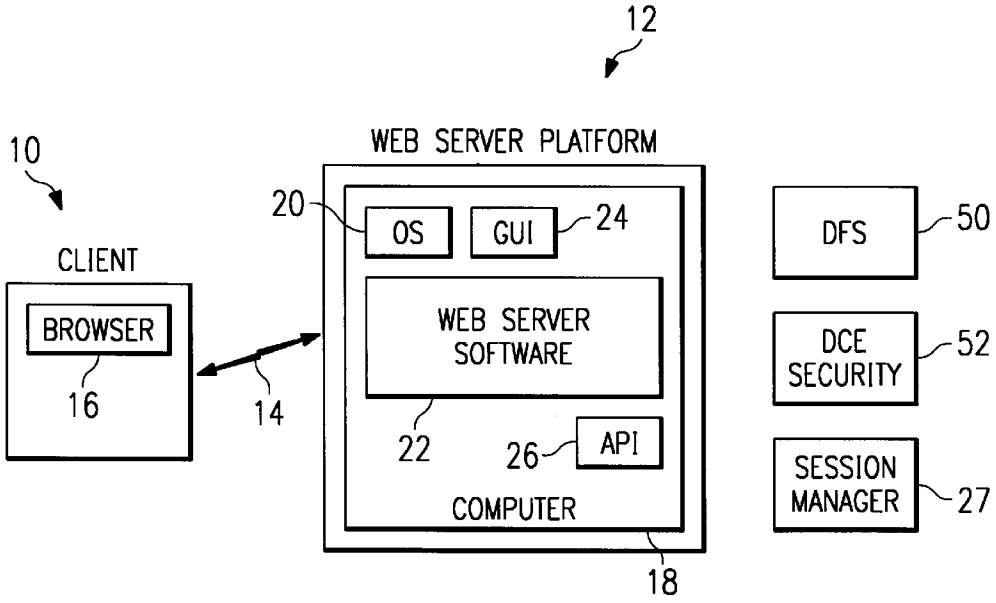


FIG. 1

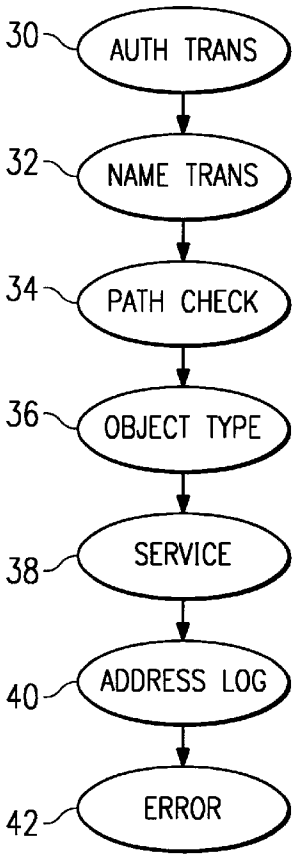


FIG. 2

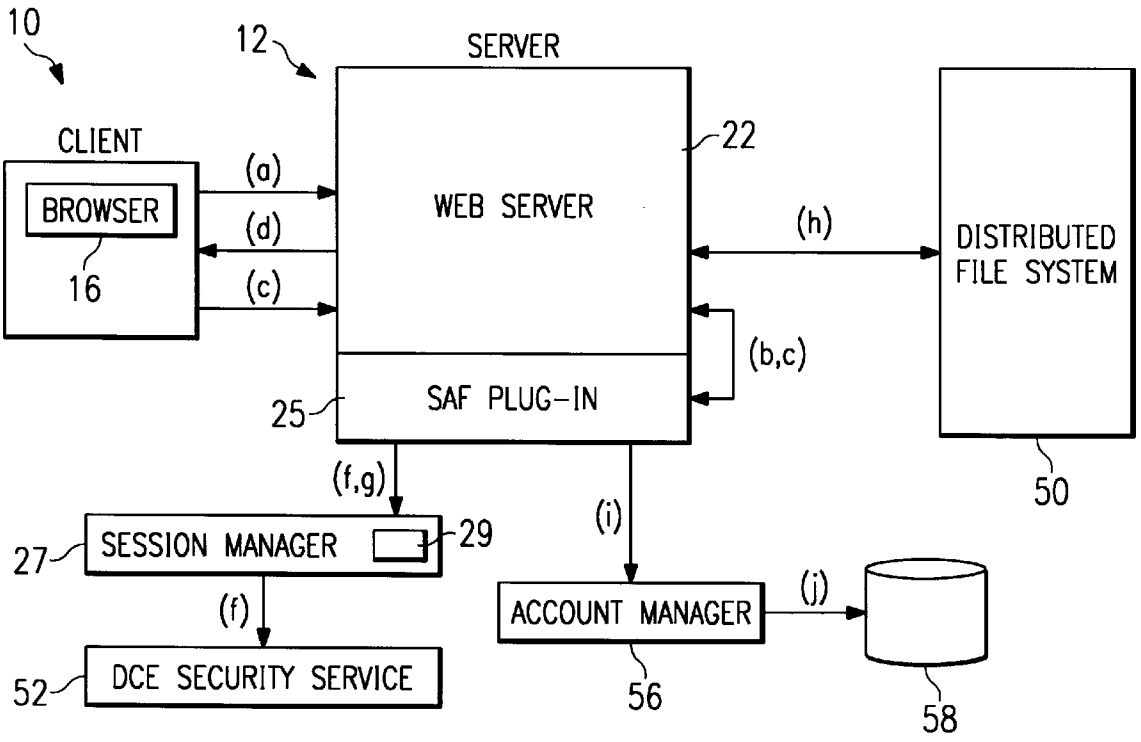


FIG. 3

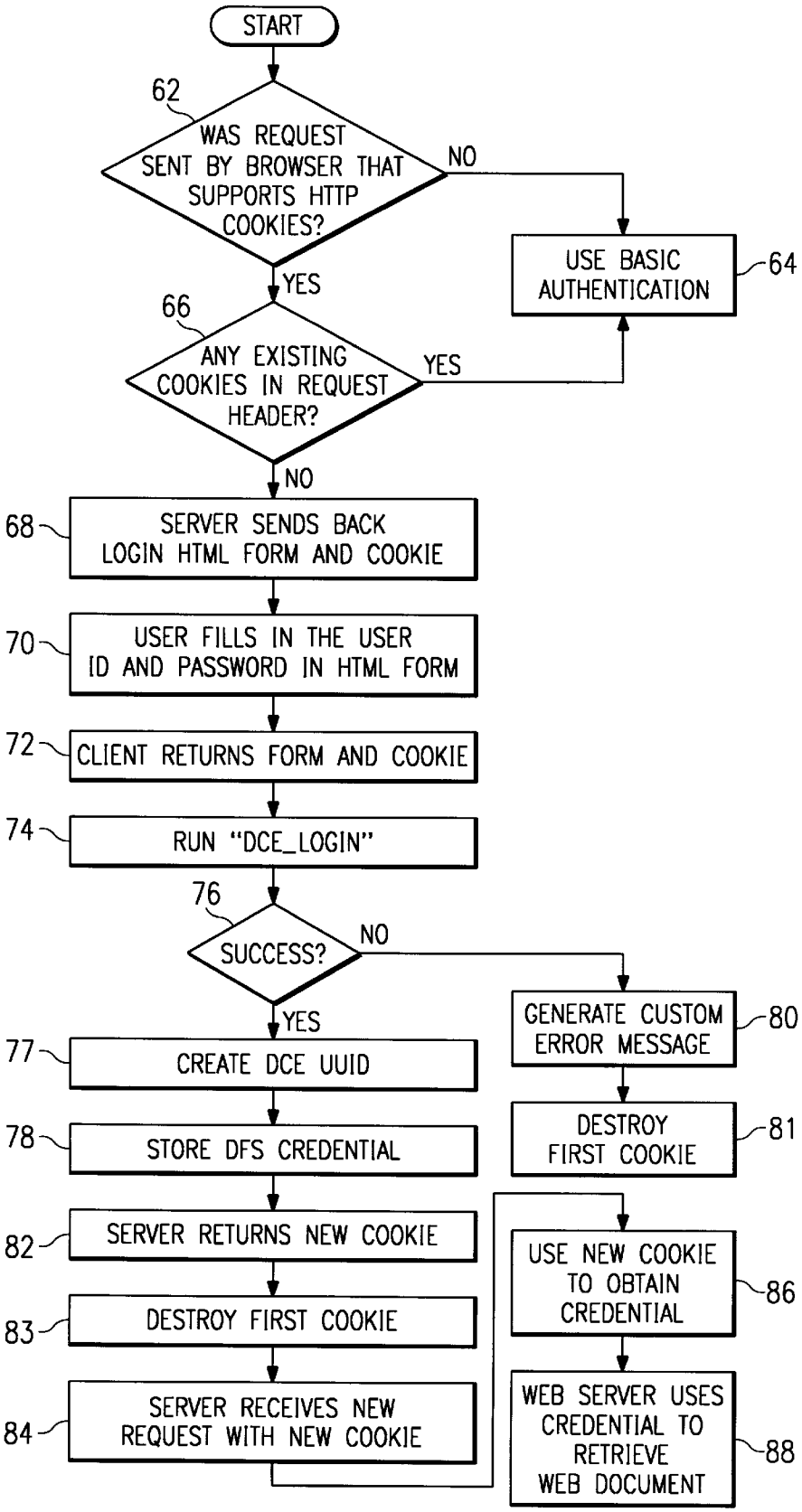


FIG. 4

DISTRIBUTED FILE SYSTEM WEB SERVER USER AUTHENTICATION WITH COOKIES

TECHNICAL FIELD

The present invention relates generally to Web transaction processing and more particularly to enabling access to Web documents stored in a secure distributed file system.

BACKGROUND OF THE INVENTION

The World Wide Web of the Internet is the most successful distributed application in the history of computing. In the Web environment, client machines effect transactions to Web servers use the Hypertext Transfer Protocol (HTTP), which is a known application protocol providing users access to files (e.g., text, graphics, images, sound, video, etc.) using a standard page description language known as Hypertext Markup Language (HTML). HTML provides basic document formatting and allows the developer to specify "links" to other servers and files. In the Internet paradigm, a network path to a server is identified by a so-called Uniform Resource Locator (URL) having a special syntax for defining a network connection. Use of an HTML-compatible browser (e.g., Netscape Navigator) at a client machine involves specification of a link via the URL. In response, the client makes a request to the server identified in the link and receives in return a document formatted according to HTML.

Many organizations use multiple computers interconnected into a distributed computing environment in which users access distributed resources and process applications. A known distributed computing environment, called DCE, has been implemented using software available from the Open Systems Foundation (OSF). As DCE environments become the enterprise solution of choice, many applications may be utilized to provide distributed services such as data sharing, printing services and database access. OSF DCE includes a distributed file system, called Distributed File Services (DFS), for use in these environments.

DFS provides many advantages over a standalone file server, such as higher availability of data and resources, the ability to share information throughout a very large-scale system, and protection of information by the robust DCE security mechanism. In particular, DFS makes files highly available through replication, making it possible to access a copy of a file if one of the machines where the file is located goes down. DFS also brings together all of the files stored in various file systems in a global namespace. Multiple servers can export their file system to this namespace. All DFS users, in the meantime, share this namespace, making all DFS files readily available from any DFS client machine.

It would be highly desirable to extend the functionality of existing standalone Web servers in the enterprise environment to take advantage of the scalability, file availability and security features of DFS (or other similar distributed file systems). As a by-product, users with an off-the-shelf browser would be able to easily access the Web information stored in the DFS namespace with no additional software on the client machine. Before this goal can be achieved, however, it is necessary to integratethe security mechanism provided by the Web Server with conventional DFS security. One of the alternatives is to use the Basic Authentication scheme (provided by the Web server) to obtain the userid and password for each HTTP request. However, using the known basic authentication scheme in the context of DFS has several problesm.

In particular, user ids and passwords are passed on every request. Thus, they are more likely to be attacked by

intruders even if passwords are protected by some encryption mechanism (for example, SSL). Secondly, it is difficult for the DFS and Web server security mechanisms to coexist. The browsers will memorize the userid and password sent to a specific server and the id and password will be attached to every HTTP request sent to that server. If a mechanism is provided for having the Web server access the distributed file system, the Web server will maintain both the documents stored on the server local directory (protected by Web server security) and DFS (protected by DFS security). From the browser's perspective, the Web server is a single server and will only remember one pair of userid and password for the Web server. If a user is browsing both DFS documents and Web server documents, he or she will be prompted for userid and password every time there is a switch from DFS document to Web server document, and vice versa. Finally, only limited error information can be returned to the user when DFS authentication fails.

These problems make the known basic authentication scheme ill-suited for integrating Web server and DFS security mechanisms.

The present invention solves this problem.

BRIEF SUMMARY OF THE INVENTION

It is thus a primary goal of the present invention to authenticate users accessing a distributed file system through an Internet World Wide Web server.

It is a further object of the invention to provide a distributed file system authentication scheme for Web browsing that only requires passing of a user id and password when the user initially logs in to the file system through a Web server. On subsequent requests, a secret handle stored in a "cookie" is passed from the Web browser to the Web server.

It is thus another object of the invention to use a persistent client state HTTP cookie authentication scheme to facilitate secure Web document access from a distributed file system.

It is yet another object of the invention to implement a cookie-based authentication scheme for DFS Web server applications that coexists with the basic authentication security scheme known in the art such that when a user switches from a DFS document to a Web server document, he or she will not be prompted for user id and password if already logged into DFS.

It is still another object of the invention to provide for customized error messages to be passed from the Web server to the browser instead of the error messages provided by the known basic authentication scheme.

It is a more general object of the invention to integrate the security mechanism provided by the Web Server with conventional DFS security. This will enable the functionality of existing standalone Web servers to be enhance in the enterprise environment to take advantage of the scalability, file availability and security features of DFS (or other similar distributed file systems). As a by-product, users with an off-the-shelf browser will be able to easily access the Web information stored in the DFS namespace with no additional software on the client machine.

These and other objects of the invention are provided in a method of authenticating a Web client to a Web server connectable to a distributed file system of a distributed computing environment. The distributed computing environment includes a security service for returning a credential to a user authenticated to access the distributed file system. In response to receipt by the Web server of a user id and password from the Web client, a login protocol is executed

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.