SAND95-0121C
Conf-9505162--1

Browsing the World Wide Web from Behind a Firewall

Randall W. Simons
Computer Security Department
Sandia National Laboratories
Albuquerque, NM 87185-0811

Abstract

The World Wide Web provides a unified method of access to various information services on the Internet via a variety of protocols. Mosaic and other browsers give users a graphical interface to the Web that is easier to use and more visually pleasing than any other common Internet information service today. The availability of information via the Web and the number of users accessing it have both grown rapidly in the last year. The interest and investment of commercial firms in this technology suggest that in the near future, access to the Web may become as necessary to doing business as a telephone.

This is problematical for organizations that use firewalls to protect their internal networks from the Internet. Allowing all the protocols and types of information found in the Web to pass their firewall will certainly increase the risk of attack by hackers on the Internet. But not allowing access to the Web could be even more dangerous, as frustrated users of the internal network are either unable to do their jobs, or find creative new ways to get around the firewall.

The solution to this dilemma adopted at Sandia National Laboratories is described. Discussion also covers risks of accessing the Web, design alternatives considered, and trade-offs used to find the proper balance between access and protection.

The Firewall Approach to Internet Connections

In computer security terminology, a firewall is a collection of components placed between two networks that only allows authorized traffic to pass [CHES94]. Many organizations use firewalls to enforce their security policy regarding access between their internal network and the Internet. Firewalls typically contain some means of packet filtering, such as a screening router, and gateways or proxies to handle specific protocols.

Sandia National Laboratories is located at two major sites, Albuquerque, New Mexico and Livermore, California. While this

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

paper discusses only the computing network at the New Mexico site, the California network is configured similarly.

Sandia maintains three separate computing environments at the New Mexico site to serve diverse security requirements [SAND94]:

- The secure environment supports processing of classified information, and is not directly connected to either the Internet or the other environments.
- The restricted environment supports processing of sensitive information, and is connected to the Internet through Sandia's corporate firewall. This firewall is developed and maintained by the Chief Information Officer's organization (CIO).
- The open environment is connected to the Internet with few safeguards or restrictions. Classified and sensitive information are not allowed in this environment, and security is the responsibility of the user organization.

In this paper, the "internal network" means the network supporting the restricted environment, and "Sandia's firewall" means the firewall supported by the CIO.

Sandia's firewall contains a screening router which filters packets passing through it based on source and destination address, and destination protocol and port. The Transport Control Protocol (TCP) uses port numbers to identify which service is to be connected to. Ports numbered below 1024 are privileged ports, and only the superuser (root) can open these ports. Higher numbered ports can be opened by any process on the server.

Sandia's screening router allows FTP and Telnet packets to pass if the session originated in the internal net. Before the modifications described in this paper, Gopher, WAIS, NNTP, and HTTP packets were not allowed to pass.

Sandia's firewall also includes several gateways. One allows the passage of electronic mail using the Simple Mail Transport Protocol (SMTP)[POST82], with some restrictions. The Xforward gateway allows X Protocol packets to pass between an X server (graphics display) on the inside and an X client (graphics-generating program) on the outside when properly authorized. A third gateway allows Telnet sessions to be initiated from the outside when authenticated with a SecurID card.

The Need for Access to the Web

The Web seems to have reached a critical mass, where more and more organizations find that the number of potential readers makes

it worth their effort to provide information on the Web. As the Web grows, more information is becoming available only via newer protocols such as Gopher [ANKL93], the Wide Area Information System (WAIS), and the HyperText Transfer Protocol (HTTP). It is becoming common practice to give Uniform Resource Locators (URLs) as references. A URL contains everything one needs to know to find a document on the Web, but URLs are useless without Web access.

New browsers like Mosaic make information usable by a wider audience. One no longer needs to be an Internet guru to find information on the Internet. Browsers make it easy to find information using either the newer protocols above, or older protocols like the File Transfer Protocol (FTP)[POST85] and the Network News Transfer Protocol (NNTP)[KANT86]. The browser is a single tool that provides access using all these protocols, with the details and differences hidden from the user. The use of separate viewers adds flexibility to support new formats without the user having to learn anything new.

The Web will probably continue to grow, and begin to support more commercial applications. Companies see its potential as an inexpensive way to direct marketing at a targeted audience, and even close the sale once there is secure support for passing credit card numbers. Customers look forward to the time when they can shop for anything, any time, from their desktop. Hypertext is a natural way to provide a quick overview, then let readers delve into more details if they are interested. The ability of a potential buyer to get just the information he wants with no need to fend off a hard sell should attract many customers.

As a national laboratory that claims computational and information sciences as one of its core competencies, Sandia cannot afford to have many of its users blocked from the Web. While users in the open environment could already access the Web (subject to their local security policy), the firewall blocked access for users in the restricted environment. A group was formed within the CIO to search for a way to allow access from the restricted environment to the Web while still maintaining adequate protection.

Threats from the Web

There are several potential and known ways in which the Web can be used by hackers to penetrate the security of machines running browsers.

One class of attacks has to do with the protocols used by the Web. Every protocol has the potential for security holes. Many holes have been discovered in protocols previously believed to be safe. There's no reason to believe that gopher, WAIS, and HTTP don't contain some undiscovered bugs. Unfortunately, we may not find out about them until some hacker makes use of them.

Another class of attacks involves placing malicious code on a server and attracting victims to the server to retrieve the code. A hacker could place such code either on a server to which he has legitimate access, or on a server that he has penetrated. In the case of an immediately obvious attack, word would spread quickly, and people would stop accessing the contaminated server. More damaging would be a covert or delayed attack that might not be noticed or connected with a particular server.

A bug in earlier versions of Mosaic caused it to execute any shell command found at the end of a Telnet URL retrieved from a WWW server. If a hacker could place such URLs on a server in places where people browsing the Web were likely to look, the hacker could execute any command he wanted on the browsing machine. While this bug was corrected quickly, there is no guarantee some users aren't still running the incorrect version.

Earlier versions of ghostscript (a widely used PostScript interpreter) implemented the PostScript command to delete or append to files. These PostScript commands have been disabled in later versions of ghostscript, but again, one can't be sure earlier versions aren't still being used. A hacker could use this to delete files or to modify files such as /etc/passwd or .rhosts to gain further access.

Mosaic and other browsers can be configured to accept and execute arbitrary shell scripts from WWW servers. Users must be educated that this is not safe and should perhaps be disallowed by security policy. Even so, education cannot be expected to reach everyone. This gives a hacker the potential to run any script he writes on the victim's machine.

Design Alternatives

Pass Everything Through the Firewall

One easy way to allow access to the Web through a firewall is to simply allow all the necessary protocols to pass. To use this approach, one must allow new outgoing TCP privileged ports for each of the protocols used in the Web. In addition, one must allow

all outgoing TCP ports above 1023, because many servers (especially gopher servers) use non-standard ports.

In general, each port represents a potential vulnerability. A firewall designer should minimize the port destinations allowed, passing only those that are believed both safe and necessary. In the case of Web access, we can do better than open all these ports from all machines in the internal network.

Telnet Through the Firewall

Restricted environment users can Telnet to the open environment and use a character-based browser. This requires that they have an account on a machine in the open environment where they can run a browser. Character-based browsers are unpopular because they can't display any of the graphics available on Web servers. In addition, because the browser is not running on the local machine, users cannot download files to a local disk, print on a local printer, or hear audio.

X-Protocol Through the Firewall

Restricted environment users can use the Xforward gateway [TREE93] to get limited access to the Web. This requires both that their machine can operate as an X terminal (PCs and Macs require expensive software), and that they have an account on a machine in the open environment where they can run a browser as an X client. This approach still has limitations. The user cannot download files to a local disk, print on a local printer, or hear audio.

Add a WWW Proxy to the Firewall

This is the solution we eventually chose, so it is described in the rest of the paper.

WWW Proxy to Reduce Risks

A WWW Proxy sits in the middle between a browser and the WWW servers it contacts. The browser sends a URL to the proxy using HTTP protocol. The proxy then acts just like a normal browser and retrieves the information from the server as specified in the URL, using whatever protocol is appropriate (FTP, NNTP, gopher, WAIS, HTTP). The proxy converts the information received from the server into HTTP format, and passes it back to the browser using HTTP protocol.

Because the proxy converts information to HTTP format, HTTP is the only new protocol that needs to be used by restricted environment machines. This avoids using gopher, WAIS, NNTP, and all ports above 1024 on all restricted environment machines. The number of potential avenues of attack has been significantly reduced.

Of course, all the new protocols have to be allowed to go to the proxy, but this machine is part of the firewall. Firewall components are carefully configured and maintained, and limit access to only those administrators and services that are required to do its job. For example, Sandia's WWW Proxy completely disables incoming telnet and FTP. All maintenance has to be performed at the machine itself, not across the network.

A proxy is a single point of control for auditing and selective shutdown if necessary. Administrators could selectively shut down access to specific WWW files or servers based on protocol, Content-Type, source or destination. We can expect new dangers will be found in accessing the Web, and having a single point of control makes it easy to react quickly when these dangers become known.

In addition to these security benefits, a proxy also provides a free bonus -- caching. Caching can speed access because the proxy doesn't need to retrieve information from a server if it already has it in its cache.

Because a proxy is a single point of control, it would be possible to have the proxy check the kind and version of browser being used. This information is typically provided as part of the header in the HTTP protocol. The proxy could refuse to honor requests from "unapproved" browsers, such as old versions of Mosaic that contain the Telnet URL bug. This would not be effective in catching problems with viewers, since viewer version information is not included in HTTP. For example, the proxy could not tell if a browser would use an old version of ghostscript that executes "delete file" commands. This approach was rejected as being more trouble than it was worth to Sandia. Different organizations with different security policies might find it useful.

Another approach would be to require users to register both browsers and viewers before letting them use the proxy. The proxy could have a list of registered users, and refuse to serve others. The registration process could be done completely manually, but would probably be an administrative nightmare. Registration could be automated by providing tests that would have to be passed by browsers and viewers, checking for known problems. Maintaining

the tests and helping users to pass them would still be labor-intensive. This method was rejected, both because of its cost and user-unfriendliness. Instead, it seems better to make sure users have easy access to good browsers and viewers, and educate them on the dangers of using old ones.

Proxy Modifications

A modified proxy can also address the threat of malicious code being brought in by a browser. A proxy can screen potentially dangerous data based on MIME Content-Type [BORE93], allowing only download to a local disk, not immediate execution. The user can still run a viewer and look at the file that was downloaded, thus triggering the attack, so this is not strong protection. But at least the user has to be aware of what kind of file he got and what viewer he ran. This is sort of like asking a DOS user who types "del *.*" if he's sure he wants to delete all files in the directory. He can still go ahead and do something stupid, but it is less likely he will do it unintentionally.

Note that one could instead choose to block dangerous Content-Types entirely. This idea was considered, but rejected. This is consistent with Sandia's earlier policy of allowing FTP to be initiated from the internal network to the outside. Dangerous files can be brought in using FTP, but some further action is required to actually trigger an attack.

Sandia's WWW proxy is based on the CERN httpd version 3.0. It has been modified to check the MIME Content-Type and only allow certain types to pass unchanged. Currently allowed types are:

text/html

text/plain

image/gif

image/jpeg

image/x-xbitmap

audio/basic

video/mpeg

video/quicktime

application/x-wais-source

application/octet-stream

Other types are converted to application/octet-stream. When the browser sees this Content-Type, the only action it can take is saving the file to local disk, since it cannot tell which viewer to use to execute or interpret the data. This prevents a hacker's attack from being launched with a single click of the mouse on a hypertext link.

The allowed types include only those which are both useful (i.e., widely used on the Web) and believed safe based on an analysis of the contents and viewers of that type.

Status

Sandia's WWW Proxy was opened for use by all restricted environment users in December 1994. There have been no known attacks on or through the proxy to date. Over 500 internal machines have used the proxy so far, retrieving about 2 gigabytes of information per week. These numbers are expected to increase.

There have been few complaints about the restrictions the proxy enforces. We evaluate requests to allow other Content-Types to pass unchanged as they come up. Quicktime was added this way, after a user requested it, and we evaluated the protocol and didn't find any apparent vulnerabilities.

Acknowledgments

The group that studied design alternatives for restricted environment access to the Web consisted of Arthurine Breckenridge, Matthew Finkelstein, Meeko Oishi, and the author. Thanks to Doug Brown and Jim Hutchins, who helped with the rationale behind the group's decisions.

Bibliography

[ANKL93] Anklesaria, Farhad, Mark McCahil, Paul Lindner, David Johnson, Danien Torrey, and Bob Alberti. The Internet Gopher Protocol. RFC 1436, March 1993.

[BORE93] Borenstein, Nathaniel and Ned Freed. MIME (Multipurpose Internet Mail Extensions). RFC 1521, September 1993

[CHES94] Cheswick, William R. and Steven M. Bellovin. Firewalls and Internet Security: Repelling the Wily Hacker, Addison-Wesley, 1994.

[KANT86] Kantor, Brian and Phil Lapsley. Network News Transfer Protocol. RFC 977, February 1986.

[POST82] Postel, Jon. Simple Mail Transfer Protocol. RFC 821, August 1982.

[POST85] Postel, Jon and joyce Reynolds. File Transfer Protocol. RFC 959, October 1985.

[SAND94] Sands, Paul D. Computer Security Desk Reference. SAND88-2231, Fourth Edition, Sandia National Laboratories.

[TREE93] Treese, Win and Alec Wolman. X Through the Firewall, and Other Application Relays. In USENIX Conference Proceedings, pages 87-99, Cincinnati, OH, June 1993.