;login:

THE MAGAZINE OF USENIX & SAGE

August 2002 volume 27 • number 4

inside:

SECURITY

Wool: Combating th Perils of Port 80 at the Firewall

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

combating the perils of port 80 at the firewall

by Avishai Wool

Dr. Avishai Wool is a co-founder and chief scientist of Lumeta Corporation, and an Assistant Professor at the Department of Electrical Engineering Systems, Tel Aviv University, Israel.

yash@acm.org

Last summer, the Code Red worm and its relatives hit Web servers all over the Internet. The worm spreads by requesting a Web page from a Web server running an un-patched version of Microsoft's Internet Information Server (IIS). The request is issued using the ubiquitous HTTP protocol and is sent to the server's default port 80. However, the requested page's URL is carefully and maliciously crafted to trigger a buffer overflow on the Web server, thus infecting the server with a copy of the worm. The newly infected server then turns around and does the same to other Web servers.

In the October 2001 "Inside Risks" column,[1] Somogyi and Schneier describe the worm and point out the general susceptibility of the Internet, and all who connect to it, to such worms. They argue that "http has become Internet-connected computers' lingua franca," yet popular Web servers have not been properly engineered to eradicate remotely exploitable vulnerabilities, so companies and customers are assuming increased risk in deploying and using the Web.

While the issues that Somogyi and Schneier raise are valid, the article points the finger only at software vendors such as Microsoft. The trouble with this approach is that it absolves other corporations and their network security staff of any responsibility. The article makes it sound as if the only way you can fight back and protect your network is to wait for, and install, the latest patches from Microsoft.

Installing security patches is important, and getting software vendors to improve the security of their products is indeed an excellent idea. It just requires time, effort, and money. In the meantime, though, there is something very simple that you can do today that will greatly decrease the ability of Code Red and its ilk to spread, and significantly reduce the risk to your internal network and to public Internet sites in general.

You need to ensure that your Web servers are properly quarantined by a firewall. The operative word here is "properly": practically all Web servers are placed behind firewalls, which are supposed to shield the servers from attacks. Unfortunately, these firewalls are not configured to do everything they could to combat HTTP-based worms such as Code Red. Evidence collected from firewall configurations run through the Lumeta Firewall Analyzer shows that HTTP traffic is often allowed through firewalls unhindered.[2] This policy is too liberal.

The point to remember is that a Web server is supposed to serve. It is passive. Under normal circumstances, a Web server waits for HTTP requests and serves the requested pages. A healthy Web server does not initiate requests to other Web servers. Only Web browsers actively request pages. However, once a Web server has been infected with a Code Red worm, it starts behaving like a Web browser – actively sending its maliciously crafted HTTP requests to other Web servers, either on your internal network or on the Internet. There is no reason to let your Web server initiate HTTP requests like this.

So here is the recipe. If you have a modern (stateful) firewall, you need two firewall rules to protect a Web server, in this order:

1. Allow the HTTP service from anywhere to YourWebServer.
2. Drop any service from YourWebServer to anywhere.

Rule 1 allows Web browsers on the Internet to request pages from your Web server and allows the server to serve the pages; a modern firewall can match the server's responses to the browsers' requests. This is what those state tables are for. (Technically, the firewall keeps track of the TCP three-way handshake so it can distinguish between the computer that initiated the HTTP session (the browser) and the computer that responds (the server).)

You probably already have something like rule 1 in your firewall's rule set. What you need to add is rule 2, which prevents your Web server from turning around and starting to actively request pages. Once the Web server is blocked from behaving like a Web browser, it will not be able to spread HTTP worms.

Now, actually, rule 2 prohibits the Web server from initiating any traffic. Taken literally, rule 2 may be too restrictive for the Web server to function, e.g., the Web server may need to initiate domain name queries. Also, Microsoft's "Windows Update" feature works by having the computer access Microsoft's own Web site using a Web browser that is embedded into Microsoft's operating systems. You'll need to add rules dealing with such exceptions before rule 2 – just make them specific only to those Web sites your server needs access to.

Note that the above recipe will not prevent your externally-visible Web server from getting infected in the first place. Get a patch from your software vendor for that. What the recipe will do is make sure your Web server is properly quarantined. It will prevent your Web server from infecting your internal networks, partners and clients. And, it will reduce the spread of the next HTTP worm that comes along, even before Microsoft issues a patch, and even if the next worm targets, say, Linux-based Apache Web servers.

So if you run a Web server, don't just passively wait for your software vendor to issue security patches. Take action. Review your firewall rules and make sure that your Web server is not allowed to behave like a browser. It's good for your network's security and it's important for the Internet as a whole.

Notes

1. S. Somogyi and B. Schneier, "Inside Risks: The Perils of Port 80," *Communications of the ACM*, vol. 44, no. 10, October 2001, 168.

2. A. Wool, "Architecting the Lumeta Firewall Analyzer," *USENIX Security Symposium*, Washington, D.C., August 2001, 85–97.