MANAGING TECHNOLOGY

• Cybertheft, Network Security, and the Library Without Walls

by Mark Cain

In December, officials at JSTOR, the non-profit agency that for almost a decade has been digitally archiving scholarly journals, reported that it had been the victim of a carefully coordinated theft attempt across the Internet. Apparently this thievery had been underway for some time:

> Toward the end of August we noticed that an IP address at a participating site was downloading a lot of articles—hundreds of complete issues. We denied access to JSTOR from that address and sent a note to our contacts at the site. At this point, we had no reason to think that this was anything other than ordinary "over-enthusiastic" use of the archive. A few days later, another address had a noticeably high number of article downloads, with hundreds of complete issues. So, again, we denied access from the second address and sent a message to our contacts there. Our first indication that something strange was afoot was in their reply. They had contacted the office to which the IP address in question belonged; no one there had been using JSTOR, and the machine that the IP address belonged to was an internal Web server and thus not a workstation from which people typically browsed the Web.[1]

The perpetrators were from another country, and before they were stopped they had succeeded in downloading 50,000 journal articles. While the extent of the theft was less than 5% of JSTOR's digital archives, the intent was clear: the systematic downloading of the entire database,[2] presumably to repackage and sell it.

JSTOR had a specific attack to thwart, which they did, but officials of that organization did not stop there. They did a great deal of research on the method through which the crime was effected. They went public with the story, informing the news media and their users about the security vulnerability that was exploited in an attempt to keep it from happening to other repositories of digital information. In fact, JSTOR has developed a Web site about the break-in.

How did it happen? Were the criminals world-class hackers who could have broken into any system they wanted, no matter what the security precautions? They could have been, but the method of attack was to use extremely common devices employed by many institutions, ironically, to provide security and enhanced network performance. I'm referring to proxy servers.[3]

Mark Cain is Chief Information Officer, Cincinnati State Technical and Community College, 3520 Central Parkway, Cincinnati, Ohio 45223 <mark.cain@cincinnatistate.edu>.

Some Network and Security Basics

To understand what proxy servers are and do—and what risks they can pose—requires some basic knowledge of networks, the Internet, and security. A short review is in order.

Networks are combinations of hardware and software that provide for the transport of information. Copper wire, optical fiber, and sometimes radio waves form a network's lines; switches or hubs split those lines up so they can be shared by multiple users. Servers dish up content. The Internet is comprised of pretty much the same items.

Information that travels over networks adheres to certain standards or protocols. There are many of these. HTTP, or hypertext transport protocol, runs the World Wide Web. File Transfer Protocol (FTP) lets users move files over great distances. Underneath these higher-order standards are some more fundamental network protocols. Local area networks may employ Appletalk (Macintosh), NetBUI (Windows), or some other communication standards, but if those networks are going to talk to the Internet, they must also support TCP/IP. (More and more networks just use TCP/IP; it's more efficient than running multiple standards.) TCP/IP is in fact a suite of protocols, but it has two main parts. The Transmission Control Protocol (TCP) disassembles information into smaller packets. These packets travel across networks and the Internet, in theory along many different paths depending upon which are the most efficient at any given moment, and then get reassembled on the other end.

The second part of TCP/IP is the Internet Protocol (IP). IP provides the addresses, so the packets know where to go. These addresses are either hard-coded, that is, a specific address typed into the software defining a computer's network properties, or dynamically and temporarily assigned to the computer by a server. This latter technique is called Dynamic Host Configuration Protocol, or DHCP. DHCP is what is typically used on networks because of the flexibility it provides. However, hard-coded or static IP addresses are still frequently used for networked devices, like switches and servers, including proxy servers. An IP address, whether hard-coded or temporarily assigned, must be unique, that is, there should be no two devices in the networked world with the same address. (This is a simplification but is basically correct.)

An IP address consists of four numbers separated by periods. The numbers between the periods can be anywhere from 0 to 255 and look something like this:

255.0.255.10

The IP address a machine has, or appears to have, bears directly on the resources it may access. Because of this, IP figures prominently in securing networked resources.

In a perfect world, network security wouldn't exist. It's a barrier and, generally speaking, has an inverse relationship with functionality. Librarians and other users of information systems usually find network security to be a nuisance. But just as we understand we should lock our cars when we leave them in parking lots, so we know that we must secure our networks.

A common method of doing so is to use firewalls. These are systems comprising hardware and/or software. They usually sit between an institution's network and the Internet, though they can also be used internally. For example, a firewall could be placed between a university's administrative system and the student side of its network. Firewalls employ one or more of four techniques: packet filter, application gateway, circuit-level gateway, and proxy server.[4]

A **packet filter** takes a look at the IP addresses of both the source and the target or destination of packets and, if appropriate, allows information to travel through specific holes or ports that have been opened on the firewall. Different Internet functions use different ports. FTP, for example, uses port 20; Web traffic (HTTP) travels through port 80. A packet filter on a firewall has a particular set of rules, its access control list, which governs what should be allowed through and what should be blocked. One rule, for example, might say that FTP traffic is permitted to be initiated from within the campus network (to go out) but not allowed to be initiated from off campus (to come in). By the way, networking types refer to the network space behind the firewall as the clean side and that on the other side (such as the Internet) as the dirty side.

Packet filters look only at the header of an information packet, but do no analysis of the content of that packet. Hackers can affix malicious data to a perfectly acceptable header and cause all sorts of damage. Yet the packet filter would let it through, because the packet on the surface appears legitimate.[5]

A firewall can employ an **application gateway.** This technique, used with an FTP server, for example, works well, but can slow things down.

A **circuit-level gateway** does its work when a connection is first established. After the gateway is convinced a connection is safe, it stops checking.[6]

The fourth possible technique used by a firewall is the **proxy server.** Proxy servers take a couple of extra steps beyond what a packet filter does. They actually look at the contents of information packets, again employing a set of rules. A packet that looks like an FTP session must, for example, contain the correct commands that you would expect to find in FTP.

Perhaps even more important, the proxy server functions as an intermediary between the requestor and the data source. The data from the source never travels directly to the end user, because the requesting client machine has set up the server as its proxy. The proxy server makes the information request on behalf of the requestor, retrieves the information and analyzes it, as described above. If everything checks out, the proxy server takes the valid data and inserts it into a new packet, which is sent to the requestor.[7]

Many college campuses use proxy servers for all Internet traffic, bypassing the proxies only for local Web addresses. Configuring a Web browser to use a proxy server is simple. On Internet Explorer 6, for example, one need only go into

Tools = >Internet Options = >Connections = >LAN Settings

and select "Use a proxy server for your LAN," then type in the address of that server.

Some proxy servers require this configuration; others, such as EZproxy, a product very popular with libraries, does not require configuration on the client end. It's all handled by the proxy server software itself.[8]

Proxy Servers and Remote User Authentication

Some commercial databases require a user name and password to access them. At best, this is an awkward method for authentication. First, it is anything but transparent to the user. Second, it requires that someone maintain user names and passwords, disseminate the information to the users, reset passwords when the users forget them or mistype them and lock up accounts. This is an incredible hassle. An alternative would be to have a single user name and password, sharing it out to everyone, but this isn't particularly secure.

A more elegant approach is to use a machine's IP address to validate the user with databases to which an institution legally has access. Remember that a device connected to the Internet must have a unique IP address. Every organization, whether a company (.com), non-profit (.org), higher education institution (.edu), government agency (.gov) and so forth, gets assigned a block or blocks of IP addresses. The IP addresses for Swarthmore look much like each other, but very different from those at Pepperdine. Swarthmore can subscribe to a database then provide that database vendor with a list of valid IP addresses for its campus. The DB owner can set up its systems to recognize those addresses and let in PCs with IP addresses that are on the list.

Validating a user by means of IP addresses works great for on-campus computers, but by itself not for remote users. A user trying to access a database from home could be connecting via almost any Internet Service Provider, each of which has its own block of IP addresses. The Pepperdine student connecting through a RoadRunner cable modem gets a RoadRunner IP address, which doesn't match the block of IP addresses for valid Pepperdine users. So the student can't get in.

Proxy servers work wonderfully well for remote access. A user can configure her Web browser to use a campus proxy server, so that no matter what ISP is used for Internet connectivity, Web queries will appear to come from the proxy server, which has a valid campus IP address. With a product like EZproxy, configuration isn't even necessary, because the server does it all. The proxy server gains access and transparently passes information back and forth between the user and the database. And she can do her research.

If these proxy servers are configured properly, they require the user first to validate against a campus list of valid users. This list could come from such sources as the campus network or the library's circulation database. The library database is particularly handy, because it can also take advantage of other features, such as the ability to block a user when s/he hasn't paid a library fine, for example. Yet when proxy servers are not configured properly, when user authen-

tication does not occur, the opportunities for abuse are enormous.

When JSTOR announced the security breach last December, its officials blamed **open proxy servers** on college campuses. It should be noted that any personal computer on a campus network can be set up as a Web server. For example, Microsoft used to have a completely free product called Personal Web Server software. It is also possible to find free proxy server software, as I just did two minutes ago via a Google search. According to Kevin Guthrie, president of JSTOR, "Anybody on a campus can set up a Web server and can either accidentally or for some other reason open up some other proxies."[9] The problem, says Guthrie is that "People have figured this out. They understand this. So what they do is they go out and search for these open proxies."[10] If the computer is left on, and if a hacker can discover this machine, he has an open door to whatever Web-enabled databases that machine can access.

This is how a portion of JSTOR's collection of digital materials was stolen. There are hundreds, perhaps thousands of open proxy servers on the nation's campuses. In January, *Library Journal* reported that Melissa Belvadi, a systems and services librarian at Maryville University in St. Louis, wondered how hard it would be to exploit this weakness. She searched Google for open proxies and quickly uncovered a site listing a number of them. Using some of the addresses from the list, she had no problem getting into databases licensed by other schools, databases to which her own institution did not have subscriptions.[11] I repeated her Google experiment and also found several lists of unsecured open proxies.

Open proxy servers are trouble. Aside from the illicit activity described above, they can be used for denial of service attacks. (A denial of service attack attempts to crash a network by overwhelming it with useless traffic.[12]) Open proxies can also be used to aid with spam mailings.

What Can Be Done

There are several strategies for addressing the problem of open proxy servers and the need for secure and effective authentication.

1. Return to the old days of requiring a user name and password for access to each database. I don't think anyone wants to do that.
2. Develop mechanisms for identifying open proxy servers on campuses and shut them down.
3. Configure the main campus firewall correctly so that it won't allow off-campus access to any open proxy servers that might exist on campus.
4. Configure all institutional computers to travel through a main campus proxy server for all Web traffic. There are a couple of problems with this approach. It puts a burden on one server and creates a single failure point, that is, if the proxy service stops or if the server crashes, all Internet connectivity stops cold.
5. Don't use proxy services for remote access to databases, but instead employ something like a virtual private network (VPN). Many schools, such as the University of Pittsburgh, are employing VPNs to provide a more secure off-site connection to the campus network. A virtual private network travels over the Internet, authenticates the user, and encrypts data. The remote machine becomes part of the network, gets assigned a campus IP address and has direct access to whatever resources a campus machine does, including library databases. In this scenario, there is no need for a proxy server, because the client already has a valid IP address. However, VPNs require installation of software on the client machine at home, which puts a support burden on campus IT. In addition, VPN encryption can slow the performance of the connection.[13] Finally, VPNs have their own security limitations.
6. Investigate other alternatives. The Shibboleth Project, for example, is developing architecture for sharing information resources among colleges and universities with the appropriate access controls. This is a more complicated security environment, involving personal attributes of users in higher education, trust relationships among campuses/networks, and so forth[14]

Something needs to change though. As JSTOR says on its Web site, "as long as IP authentication remains a primary authorization mechanism for resources, and as long as open proxy servers continue to proliferate, no technical solution can be 100% effective."[15] In the meantime, JSTOR intends to take action against any open proxies it identifies that are trying to access its resources. Other publishers, if they feel similarly threatened, are likely to follow suit. Responses would certainly include blocking access from unauthorized proxies, but other responses could be legal in nature.

Dangers Abound

Cybertheft is not new, though it is happening with increasing frequency and potentially devastating consequences. In March, someone hacked into systems at The University of Texas at Austin and stole the names and social security numbers of 52,000 individuals.[16] Fortunately, the individual was caught, apparently before he'd used the information to ill effect. I say fortunately, because with this kind of data, a good cyberthief could tap into financial accounts, stealing millions of dollars. In an age where information and money are electronic, the risks are very high.

Academic librarians are proponents of broad access to information, anytime and anyplace, for their user communities. Whether a current student or faculty member is on campus or in a cyber cafe in Paris should be irrelevant. Librarians strive to provide transparent systems, with a minimal amount of barriers between the user and the information she or he is seeking. Librarians also want to ensure user privacy and academic freedom. These are all laudable values.

But the publishing community has legitimate concerns as well. It took a lot to convince journal and index publishers to share their information resources electronically. A story like JSTOR's should be as scary to us as it is to these publishers, for if we don't provide a secure computing environment in which researchers can use the publishers' assets, those publishers may understandably want to pull back from the open access they have provided in the past. That would be a tragedy for scholarship inquiry. To protect our interests, we must protect theirs.

Notes and References

1. "Open Proxy Servers: Gateways to Unauthorized Use of Licensed Resources," *JSTORNEWS* 6 (December 2002) [Online]. http://www.jstor.org/news/2002.12/open-proxy.html.

2. Dan Carnevale, "Security Lapses Permit Theft from Database of Scholarly Journals," *The Chronicle of Higher Education* (January 2003): A29.
3. Ibid.
4. [Online]. www.webopedia.com/TERM/f/firewall. html.
5. Terry William Ogletree, "The First Line of Defense," *PC Magazine* (June 2001): 92.
6. [Online]. www.webopedia.com/TERM/f/firewall.html.
7. Ogletree, op. cit.
8. Gordon Bertrand, "Providing Access to Remote Patrons Can Be EZ," *Feliciter* (2002): 135.
9. Carnevale, op. cit.
10. Ibid.
11. "Open Proxy Servers Victimize JSTOR," *Library Journal* (January 2003): 20.
12. [Online]. http://www.webopedia.com/TERM/D/DoS_attack.html.
13. Florence Olsen, "Colleges Offer New Connections to Networks from Off Campus," *The Chronicle of Higher Education* 47 (August 2001): A31.
14. [Online]. http://shibboleth.internet2.edu.
15. [Online]. http://www.jstore.org/about/openproxies.html.
16. Vincent Kiernan, "U. of Texas Student Is Charged in Theft of Social Security Numbers," *The Chronicle of Higher Education* 49 (March 2003): A31.