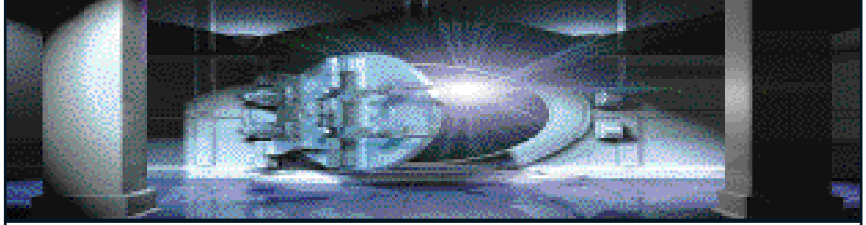

SECURITY PATROL

by Vincent LeVeque

Application Vendor Insecurities

QUESTION: Much of your advice about security may be good in theory, but, in our case, our application software vendor tells us how the AS/400 must be set up. If we don't follow its rules about the system security level (i.e., level 30 or level 40) , object authority, and the like, it tells us we will void our maintenance agreement and it will no longer support us. What can we do?

ANSWER: It never ceases to appall me that so many application software vendors force their customers into unsafe, wide-open system configurations. Many practices, such as setting the system value QSECURITY to level 40 or 50, have been part of IBM's recommendations for almost ten years. Other, more obvious security vulnerabilities include hardcoded user IDs and passwords and object code owned by a profile that also serves as the user group profile (meaning your users have ownership rights to your production environment!).

Now that I'm done ranting and raving, I will attempt to provide some useful advice for you and others. The first thing you should do is thoroughly review the vendor's security practices before purchasing the software. Wayne O. Evans, former author of this column, has an excellent checklist for starting this process. Check out his Web site at *www.woevans.com* for the security software checklist. Consider his questionnaire as a starting point and feel free to include additional points of concern to your environment. Even if you end up buying the software anyway, the fact that you have raised the issue with the vendor's sales force will get some attention.

Assuming you already have the software, consider dropping maintenance and maintaining the software in-house. If the vendor is unable to support something as basic as IBM's recommended security practices, chances are it does not support other essential features, either. Maintaining the software yourself may be a smart thing to do in this case. You not only will have control over your environment but also will be exerting economic pressure on the vendor to clean up its practices. Software maintenance is a big moneymaker for software vendors. A few large customers dropping lucrative maintenance contracts because of inadequate security may exert pressure to change these practices.

If neither of these are an option, consider adopting compensating controls around your AS/400 to make up for the lack of security in the vendor's mandated configuration. As an example, if the software has an embedded user ID and password, you should audit all use of this account, filtering out legitimate access and noting illegitimate efforts to log onto or submit jobs via the account. If the software requires that you run at QSECURITY level 30, you should ensure you audit for program failure and blocked instruction execution. Level 30 also requires tight control over job descriptions with user profiles specified. Remove all of these unless they are absolutely required and, of those that are required, ensure that the profile is "least privilege" (that it has the lowest authority necessary to get the job done).

Lastly, try to get the backing of your company and put some pressure on the vendor to change its practices. Doing you prior groundwork with top management in building a business case for security will help. Your CFO hopefully would understand the risk to company financial assets stemming from inadequately secured financial data and know his liability as a company officer for not properly protecting these assets. I've seen Fortune 500 companies with very large IT budgets convinced that they must maintain an unsecured AS/400 because of vendor pressure. I'm sure if pressure were applied in the other direction, these software vendors would very quickly change their practices to make a sale.

Don't Be Denied

QUESTION: Recently, some big-name Web sites were knocked out by hackers as part of a denial-of-service attack-Yahoo!, eBay, etc. How did this happen? Why didn't their firewalls stop these attacks? Am I at risk? Are AS/400-based Web sites at risk?

ANSWER: Actually, no one is sure exactly how these attacks were launched, though there is some reasonable evidence for the likelihood of certain well known hacker applications to have caused these problems. As of this writing (March 2000), the investigation is still going on, with no suspects publicly identified yet.

The main agents thought to be at work are programs called Trin00, Tribal Flood Network (TFN), and Stacheldraht (German for "Barbed Wire"). These programs are all freely available on the Internet and require a minimal skill level to use. The "architecture" of these attack tools consists of daemons controlled by a master program. To set up the attack, the hacker breaks into computers at third-party sites. These sites generally have nothing to do with the ultimate target; they are breached simply because it is easy to do so and because they have access to high-speed Internet links. Once the third-party machines are broken into, software referred to as *daemons* is installed. This software is designed to be inconspicuous, waiting quietly for some signal from the master program. The signal from the master program sets off the attack. It is typically a User Datagram Packet (UDP) on a high port, with the default Master-to-daemon port being 2744 for Trin00. A UDP in this range may slip by many firewalls but is immediately recognizable by the daemons software as a command to action. Upon receiving the signal, the daemons start attacking the targets.

The daemons deliver a flood of traffic to the target as their final blow. Several different methods are used, depending on the program. One particularly effective method uses what is called a *broadcast PING.* In TCP/IP, a PING request is used to tell if a particular host is online or offline. PING is a very useful utility for network diagnosis, so much so that many firewalls do not block it. In each TCP/IP subnet, there is a specific network address called the *broadcast address.* In effect, sending a packet to this address submits it to all the hosts on the subnet. One use of the broadcast address is to send a broadcast PING request to an entire subnet. Again, this can be used for entirely legitimate network management purposes. A broadcast PING will tell you exactly which systems on a subnet are responding and which aren't.

Suppose you were to forge the source, or "from," address of the PING request. The reply would go not to your own system but, instead, to the one whose address you forged. A subnet can contain a lot of hosts. An entire Internet "class C" network, for example, has up to 254 possible hosts. What would happen if you were to send a broadcast PING to several class C networks, each with the source address of your attack target? The attack target (your forged "from" address) would be bombarded with many hundreds or thousands of PING replies all at once. Such a volume of traffic could overwhelm the server's capacity or even the network itself. The massive number of PING replies would constitute a denial-of-service attack. This very attack has been around for at least three years and is commonly known as a *Smurf attack*. What's new is the automated remote coordination of a large group of these attacks all at once.

The weak points that the hacker will exploit are the systems that were compromised and on which the daemons were installed, not the ultimate targets of the attack. Installing the daemon software requires that the attacker obtain root access. In a UNIX system, "root" is the all-powerful system administrator account, similar in function to QSECOFR on an AS/400. Daemon software is known to be installable on Sun Solaris and Linux versions of UNIX. Recent information indicates that Windows systems may also be vulnerable. The possibility that an AS/400 would host the daemon software is remote.

While not a likely host for launching attacks, as an end target AS/400 is as vulnerable to denial-of-service attacks as any other system, no more and no less. The target of the attack is the network connection, not the Web server itself. Regardless of how secure or robust the server is, enough bogus traffic will prevent legitimate users from getting through. A firewall can't help much against a massive flood of traffic, either. If the firewall is configured to block the PING requests, it will protect internal systems from being flooded but will itself become unusable, as the amount of illegitimate ping attacks will overwhelm its ability to pass desired traffic.

What can you do to avoid being an unwitting accomplice to these attacks? For starters, you can make sure your systems are sufficiently secured to prevent agents and daemons from being installed on them as well. You certainly would not want a call from the FBI explaining that your computers were used to bring down a nationally known company's Web site. Ensure that your systems have broadcast PING disabled. The actual need for a broadcast PING service is very small these days and not worth the risk of it being hijacked by an attacker.

Some routers allow you to limit the bandwidth that certain traffic is allowed to use. Cisco routers now have a quality-of-service attribute that ensures that designated traffic will always get priority. This feature is designed for use with IP-based telephony services, where it is important that real-time telephony traffic always gets through quickly. It can also be used to ensure that essential Internet traffic of other types always get some priority, even if your system is flooded with attack traffic.

Several tools are available for scanning your systems for agent and daemon software. The National Infrastructure Protection Center (NIPC) has published a free tool called find_ddos. It is designed to work like a virus scanner, looking for traces of agent and daemon software on various systems. It is designed to work only with Sun UNIX systems and Linux, and only the binary executable is provided, without the source code. Also, Trend Micro (*www.antivirus.com*) has a free online scanning tool that will work with Windows 32-bit platforms.

Vincent LeVeque is a senior security engineer for Science Applications International Corporation (SAIC). He can be reached at *vleveque@earthlink.net.*

REFERENCES AND RELATED MATERIALS

- CERT Coordination Center—Denial of Service: *www.cert.org/tech_tips/denial_of_service.html*
- CERT Incident Note IN-99-04: *www.cert.org/incident_notes/IN-99-04.html*
- Results of the Distributed-Systems Intruder Tools Workshop: *www.cert/reports/dsit_workshop.pdf*
- The DoS Project's "trinoo" distributed denial of service attack tool: *www.staff.washington.edu/dittrich/misc/trinoo.analysis*