Are You Vulne

Of course you are. But implementing systematic vulnerability management will reduce your exposure

BY GREG SHIPLEY

The heart of any security effort is to become less vulnerable, but can enterprises achieve this goal? Consider that CERT (Computer Emergency Response Team) received 42,586 incident reports in the first quarter of 2003, compared with 21,756 for all of 2000. We attribute this sad state of affairs to the motley list of challenges plaguing our industry—in today's complex computing environments, vulnerabilities take many sizes, shapes and forms. They live in operating system bugs, flawed network designs, defect-laden business applications, desktop e-mail clients, Web browsers, media players and even security software. They affect voicemail systems, e-commerce applications and the very infrastructure that supports our networks. Product and configuration vulnerabilities let employees view information they shouldn't, provide a method for customers to access other customers' records, and enable a 17-year-old in Turkmenistan to break into that critical European Web server.

Software vulnerabilities can be as mainstream as gaping holes in Microsoft's Web server packages or as obscure as a bug-ridden custom application written by your marketing intern. Vulnerabilities even thrive outside of conventional IT forums: the administrative assistant who is susceptible to skilled social engineering, the proximity card system that is prone to "distance lifting" or the power generator attached to the data center that hasn't been started in a decade.

So how can an organization hope to stay safe, given such a wide range of potential attack vectors?

As with many information-security challenges, the solution lies partly with technology, partly with tactics and partly with strategy. Ratifying and enforcing policies that promote routine audits, timely patching, and implementing technologies that aid vulnerability assessment and configuration/patch management are starting points. But at the center of sound tactical vulnerability management are two basic concepts: identification and response. By leveraging tools and processes to identify vulnerabilities, and then responding

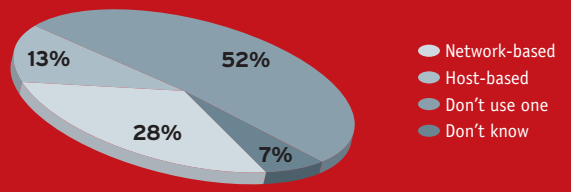
44 Executive Summary **44** E-Mail Poll Results **46** Critical Steps to Vulnerability Management **48** Slipping Under the Radar **51** VA Scanners Pinpoint Your Weak Spots **56** Vendors at a Glance

rable?

vul ner a bil i ty \vəl-n(ə-)rə-bi-lə-tē\
n **1:** susceptibility to injury or attack

E-MAIL POLL

What type of automated vulnerability-assessment tool does your organization use?



Source: NETWORK COMPUTING E-Mail Poll, 177 respondents

with plans to manage the associated risks, an organization can reduce its overall exposure.

Organizations that want to address their vulnerability at a strategic level need to move security principles beyond the traditional walls of infosec: Security must play a role in purchasing, design and implementation decisions—a major shift for most companies.

Identify, Then Respond

Before you can fix a vulnerability, you have to find it. This is easier said than done, but the key to narrowing your search is to realize that most technical vulnerabilities exist in one of two areas: design failures or implementation failures.

Examples of design failures include accidentally bringing third-party network connections into a network without implementing a firewalling mechanism, not including proper access controls between tiers in e-commerce applications and failing to implement cryptography to protect critical data sets.

Implementation failures may include forgetting to enable the ACLs (access-control lists) on a router, not patching a new Web server or forgetting to scrub user data in a Web form. Any of these vulnerabilities could expose sensitive information, allow unauthorized access or, in the case of worms and viruses, wreak digital carnage.

Design problems typically are harder to identify than implementation errors because few tools can replicate the abilities of a professional. This is why including security teams in the design life cycle is so critical—experienced humans can identify potential design failures quickly, avoiding costly long-term mitigation efforts. Implementation problems can also be costly, of course, but fortunately there are more tools and technology solutions that can reduce these risks.

Regardless of the type of vulnerability, the tactical process remains the same—identify, then respond. However, there may be multiple approaches to the response phase, some more proactive than others. An organization might choose to fix the problem directly with a software patch, or it might deploy a device to reduce the chances of exploitation. Some might even decide to do nothing and assume the level of risk associated with that particular vulnerability. Let's apply this concept to a real-world example:

» **Scenario:** A critical flaw has been found in Microsoft's Internet Explorer Web browser (not much of a stretch). This flaw lets attackers execute arbitrary code on a victim's (now) vulnerable desktop.

» **Evaluation Phase:** Identify whether vulnerable versions of IE reside on your network, possibly using a desktop-management system, an asset-tracking system or a vulnerability-assessment tool (network- or host-based).

» **Response Phase:** After finding vulnerable versions of IE, use a patch-management system to push out patches to hundreds of desktops. You might deploy a proxy server or smart caching system (see "Surf's Up" at *www.nwc.com/1402/1402sp1.html*) to filter hostile patterns and malicious code. Or you might take a dual approach, using a proxy to buy some time while scampering to get patches deployed.

For most organizations the drill is familiar—we've been patching Microsoft Outlook, IE and dozens more OSs and applications for years. What might be unfamiliar are some of the tools, like vulnerability-assessment suites, patch managers and integrity checkers, that can greatly reduce the overhead. With-

Executive Summary

VULNERABILITY MANAGEMENT

Vulnerabilities—flawed OSs, defective custom applications and poorly designed networks—constitute a clear danger to enterprises. A critical component of any vulnerability-management process is identifying those areas of potential risk. In this article, we provide a guide to rooting out and exterminating the bugs in your machines. Key are laying out both tactical and strategic lines of attack, making security a principal factor in product purchasing decisions, and choosing and deploying your tools wisely.

Speaking of tools, they say that any sufficiently advanced technology is indistinguishable from magic. Unfortunately, while we wish vulnerability-assessment scanners could turn the average security pro into a Houdini-like network defender, our tests of 11 VA products revealed this is not the case (see "VA Scanners Pinpoint Your Weak Spots," page 51). Although scanners from BindView Corp., Beyond Security, eEye Digital Security, Foundstone, Harris Corp., nCircle, Qualys, Rapid 7, SAINT, Tenable and Vigilante.com helped shore up our horribly insecure test network, none made us bulletproof. Still, don't discount a good VA scanner—our Editor's Choice, Foundstone Enterprise, identified a respectable number of vulnerabilities while providing detailed reporting and good management, and many of its rivals offered interesting features and unique protection models.

out the automation that these tools provide, most organizations don't stand a chance against the growing threats.

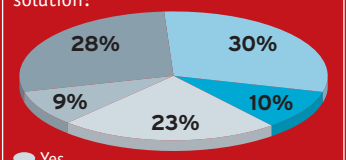
Tools of the Trade

Varying attack vectors. Staggering release rates. Relentless worms and other malicious code. If attacks are so brutal, can't security tools be just as comprehensive? Unfortunately for us, it's easier to attack than to defend, and even the best tools are no match for skilled foes. However, some toolsets can help defenders perform vulnerability-management tasks more effectively, and let's face it, the good guys need all the help they can get.

The identification effort can be aided by VA (vulnerability assessment) tools, such as network-based and host-based VA scanners and application-assessment suites. Network- and host-based VA tools both look to identify known OS vulnerabilities and common misconfigurations, and instruct users on ways to solve those problems. For example, the typical output of a network-based VA tool is a report of patches and configuration changes that need to be performed on the range of systems scanned (see "VA Scanners Pinpoint Your Weak Spots," page 51).

E-MAIL POLL

Does your organization have an automated vulnerability-assessment solution?



Source: NETWORK COMPUTING E-Mail Poll, 177 respondents

However, there are differences between host- and network-based models. For instance, network-based VA tools can operate without requiring that agents or software be deployed on the systems to be scanned. This is helpful in large organizations, where teams and administrative realms may be scattered across the globe. The downside is that these tools cannot delve as deeply as their host-based counterparts.

Although the host-based VA model does require agents, the advantage is that agents typically can probe systems and services that aren't usually available to a network scanner. The obvious disadvantage is that it's another agent, and another set of licenses, to manage. Host-based products also create problems for distributed administration teams, which often don't have access to systems outside their zones of control.

Application-assessment suites, like Cenzic's Hailstorm (see "Arming Your Top Security Guns" at *www.nwc.com/1408/1408f2.html*), @Stake's WebProxy and Sanctum's AppScan, are a little different from conventional host or network VA tools in that they are designed to evaluate both commercial and home-grown applications. These apps can arm skilled professionals with better tools to do their jobs, but operators need to be security savvy.

Once vulnerabilities are identified, patch-management tools and software-deployment systems can help with the response effort (see "PatchLink Helps Keep Windows Closed" at *www.nwc.com/1318/1318f3.html*). But sometimes even these systems aren't enough, as pesky users can muck up the patching works by reinstalling vulnerable applications, uninstalling patches and continually deploying potentially harmful software. Smart organizations will work to patch systems and keep them patched. Products that check for latest patch levels, antivirus images and general system health should be considered wherever possible (for examples, see "Got Discipline?" at *www.nwc.com/1410/1410f1.html*).

The Business Case

One of the most common sources of vulnerabilities are design and implementation flaws in off-the-shelf hard-

CRITICAL STEPS TO VULNERABILITY MANAGEMENT

» **Set a patching policy.** Although many organizations have robust policies, most policy frameworks don't require system administrators to keep up with current patches. Make patching mandatory, and define a timeframe within which critical patches must be deployed.

» **Implement a patch-management system.** Without automation, small organizations struggle with patching efforts, while large enterprises have little hope of staying current. Patch-automation tools and desktop-management systems are crucial elements in reducing risk.

» **Implement a vulnerability-assessment effort.** VA is necessary not only to identify existing vulnerabilities, but also to serve as the compliance-monitoring arm. Systems should be monitored to ensure that policies have been followed and timely patching has been performed.

» **Keep abreast of threats.** Subscribing to alert services, such as NETWORK COMPUTING's Security Alert Consensus newsletter at *portal.sans.org/nwc*), will help you understand and manage operating system and application vulnerabilities.

» **Integrate security into design and purchasing cycles.** Smart organizations will be proactive in deploying secure products. Do this right the first time and you will spend less time dealing with future security shortcomings, which will translate directly into cost savings.

ware and software. Last year, SAC (Security Alert Consensus, *www.sans.org/newsletters/sac*) reported about 1,000 new OS and application vulnerabilities—that's 83 new security vulnerabilities per month—probably a conservative estimate because SAC tends to focus on large threats to corporate and government computing environments. And SAC tells only part of the story. At the time of this writing, SecurityFocus had 7,679 entries in its vulnerability database (*www.securityfocus.com*), the National Institute of Standards and Technology's ICAT metabase (*icat.nist.gov/icat.cfm*) had 5,712 vulnerabilities listed, and the CVE (Common Vulnerabilities and Exposures, *cve.mitre.org*) effort had ratified 2,573 entries (see "Don't Panic, Plan," at *www.nwc.com/1408/1408f1.html*, for more details on what lurks beyond your borders).

One thing's for certain: The numbers aren't pretty, and they aren't getting any better.

Even if you've been spared so far, sooner or later a critical application or OS vulnerability will affect your organization. The costs, time and energy associated with the clean-up can be minimized if the proper tools and processes are in place.

The price you'll pay for not addressing attacks should also be of concern. Should an intruder leverage a given vulnerability, your organization could face data theft or destruction, prolonged outages, and humiliation and decreased client confidence should the incident go public. Each outcome has tangible and intangible dollar loss values. Those may be hard to put numbers around, but failing to include risk management in your vulnerability assessment plan will exact too high a price.

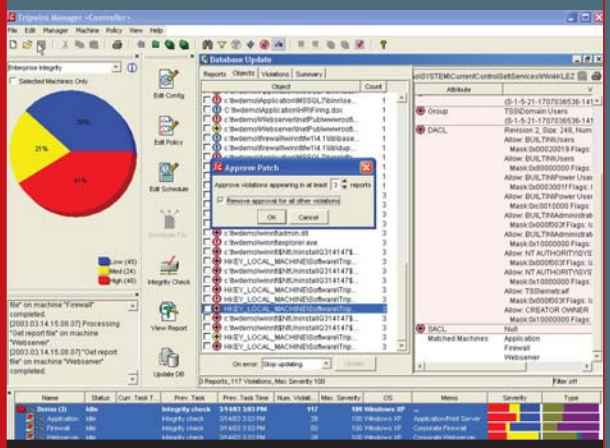
Organizations also face costs associated with automated, targetless attacks, such as those executed by worms, viruses and other malicious code. Worms have accounted for millions, if not billions, of dollars in damages and clean-up costs. What's disturbing is that every heavy-hitting worm we've faced leveraged a known OS or application vulnerability: Code Red used an IIS ISAPI buffer overflow. Nimda exploited an IIS Web traversal vulnerability. Slammer used the buffer overflow found in Microsoft SQL Server's resolution service six months earlier.

Had organizations patched their systems within three to four weeks after these vulnerabilities were announced, they would have been immune to these little buggers (see "Worm Sign," page 50). Unfortunately, most didn't.

SLIPPING UNDER THE RADAR

Application and operating system vulnerabilities pose obvious threats, but an often-overlooked problem is "pilot error"—mistakes made by technical staff during general day-to-day operations. Examples of pilot error include botched firewall entries, forgotten steps in router configurations and the inadvertent addition of a user to a privileged group. These mistakes are typically accidental, but the occasional intentional "internal bypass" is not all that uncommon.

One way to combat these less-visible threats to your security is to implement robust change-control procedures and audit the process using intelligent change-control monitoring software. Products such as Tripwire for Network Devices or Tripwire for Servers let organizations automate the process of "snap-shotting" system and device configurations and identifying configuration changes—changes that may include accidental or unauthorized modifications, even those made by malicious intruders (see screen below).



Tripwire compares current configurations to a previously generated database and flags possible problems with critical systems.

You May Ask Yourself ...

Regardless of whether you fear targeted attacks by humans or nontargeted threats, such as worms, every organization must ask a few basic questions periodically:

» Do we have tiered defenses?

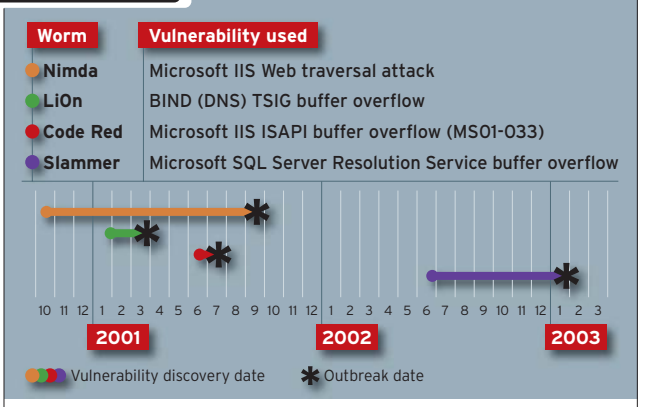
» Do we keep up-to-date with patches? Do we have a patch-deployment system in place that can distribute updates in a timely manner?

» Do we use an automated VA tool to identify potentially vulnerable systems?

According to our reader survey and the obviously abysmal state of the industry at large, we'd venture to say that the majority will answer "No" to most of these questions. This not only means increased risk, but increased costs, and it's where the business case comes into play: Like it or not, vulnerabilities cost money. Clean-up costs money. Lost work costs money. And not having a vulnerability-management plan in place will ultimately—you guessed it—cost money.

An interesting paper was released late last year discussing some of the vulnerability-management efforts

Worm Sign



at NASA (see *www.sans.org/top20/GISRA_NASA.pdf*). According to the study, NASA determined that the vast majority of its security incidents were related to a specific subset of total vulnerabilities. The agency concluded that it could reduce its risk profile by addressing this subset of known vulnerabilities. The result was an organizationwide vulnerability assessment and mitigation war, launched by NASA's CIO, that involved a few key components:

» Success was benchmarked using a vulnerability-to-host ratio.

» The performance of sites and teams was mapped directly to this ratio.

» Organizationwide ratios were measured from quarter to quarter.

Dramatic improvements were seen after just a few quarters, and NASA concluded that the effort cost about $30 per system annually. As this illustrates, organizations with a good handle on the management of their vulnerabilities can more cost-effectively deal with current threats.

Money Talks

Vulnerability-assessment software, asset-tracking systems and patch-management tools are not just beneficial. They're necessary. But these are still tactical solutions to a very strategic problem: the staggering number of vulnerabilities in mainstream OSs, services and applications. The answer—as any hardened security professional will tell you—is in the hands of developers. The world needs better software engineering, or we'll never escape today's patch-and-pray model.

But better software engineering is not going to come overnight, and it's certainly not something the enterprise customer can control. So what can be done? Start making security a key factor in product purchasing and deployment decisions, and seek out products with better security track records. ICAT is a good place for the motivated consumer to start, but those who subscribe to SAC, Bugtraq or VulnWatch will also be able to identify common offenders.

Philosophical debates aside, products that have better vulnerability track records, and thus have required less patching, not only reduce an organization's day-to-day risk, they also reduce the TCO (total cost of ownership) via lower support costs. For example, consider choosing Apache over IIS, Postfix instead of Sendmail, Tinydns instead of Bind, and Opera instead of IE. Obviously, these are not isolated decisions; security and patching requirements are just one point among many criteria. But it's a factor that is growing in importance.

If consumers begin factoring security track records into their purchasing decisions, it will send a clear message to vendors: Those that design and implement more secure products will be rewarded, and those that don't will be penalized. Think of how much easier and cheaper it would be to operate a computing environment that didn't require patching.

What Lies Ahead

Unfortunately, we suspect that the vulnerability landscape will get worse before it gets better. Over the past few years the number of discovered vulnerabilities in commercial products has risen dramatically.

Complicating matters, the management lines between application development and system, network and database administration continue to blur, particularly in regard to zones of security control. For example, administration and security of the OS (and the subsequent patching) still clearly falls under the jurisdiction of the system administrator, but his or her security efforts can be completely foiled by a single bad application; if an application developer places a vulnerable CGI form on a previously secure Web server, much of the system administrator's security controls may be bypassed. The network administrator can't be held responsible for insecure systems, and the security-conscious application developer can still be thwarted by a careless database administrator. The dependencies among administrative teams are growing ever more web-like, and these areas of authoritative haze will only be complicated by the adoption of new technologies, such as Web services.

Most organizations have two choices: Continue what they are doing and continue operating with large risk/exposure profiles, or invest in more mature vulnerability-management efforts. Those efforts must include both the tools and processes to quickly and effectively identify, and respond to, an ever-evolving set of threats. Forward-thinking organizations will not only build out better vulnerability-management systems, they will also become more security-conscious in their purchasing decisions. The safety of their data, and their businesses, depends on it.

GREG SHIPLEY is the CTO for Chicago-based security consultancy Neohapsis. Write to him at *gshipley@neohapsis.com*. Post a comment or question on this story at *www.nwc.com/go/ask.html*.