Chapter 9: Keeping the alligators out of the sewer

The backup administrator's guide to securing your network from hackers and viruses

When I was a kid, I heard a lot of stories about New York City and Chicago sewer workers who would encounter the occasional alligator lurking in the city sewer system, evoking the question, "How did an alligator get into the sewer system of such a big city?"

I knew—the Huber brothers, that's how. More than once, they'd return from a Florida vacation, pockets bulging with baby alligators. And as soon as Mrs. Huber wasn't looking, they'd toss the little monsters into the toilet or a local storm drain. When I asked them why in the world they were doing that, they replied coolly, "We wanna see how big they'll get and if they'll eat anybody."

The world is full of goofballs like the Brothers Huber. Somewhere, sometime, some goofball is going to hack your network "just because." They're going to hack it or send you a virus "just because" they want to see what happens or "just

because" they intend you harm. And if you don't believe it can happen to you, scan these news briefs for a little enlightenment:

> MSNBC's Bob Sullivan reported on a company that got hacked, sustaining an 80 percent probability that all of the credit card data in its system had been compromised. More than 2,000 unsuspecting clients may face inflated bills this month—because someone might have lifted their information.
>
> The *San Mateo County Times* announced that many Silicon Valley businesses would be under siege from a perilous virus and Trojan horse attack by—believe it or not—the Russian Mafia.
>
> Robert Lemos of ZDNet reported "Microsoft spreads virus—by accident." Unbeknownst to Microsoft, the Korean language versions of Visual Studio .NET sent to their South Korean developers harbored an unwelcome guest—the virulent Nimda worm.
>
> Robert Lemos (again) reported on a new type of attack: the JPEG worm, which is breaking new ground in virus attacks. This virus infects and attaches itself to images on the system as they're opened and viewed.

All this mayhem happened just last week—and I found these stories without really looking. I'm sure a lot more stories are out there, but I just can't bear to read them. So the question becomes, how much of this do you have to worry about, how can you protect yourself—and what does this have to do with backup? In short, how do you keep those pesky alligators out of the sewer?

Let's take the first one last: What does this have to do with backup? In a word, *EVERYTHING*. If you back up a file that's infected with a virus, you've now stored the virus, as well. When you restore the file, the virus gets restored, too. If someone hacks your company's website and defaces several pages, you'll be restoring those files—that is, unless you caught it before you backed them up again.

Try this, just for fun: Run an attack and virus test on your computer systems, just to see what will happen. You can run these tests without harming anything on your system—except for your false sense of security, which may never recover. But that's part of what this chapter is about: the process of testing for basic hacker vulnerabilities in your systems and then passing that information on to the security folks to so that you and they can implement some *real* security. Because at the end of the IT day, it's your job to restore lost data, no matter who lost it.

THE MAIN THING

The threat to your system? The loss of data and of service availability (like Web, e-mail, and databases) through corruption, theft, or erasure. You've got to protect against the loss or corruption of data due to hacking and virus attacks. You've got to restore your data from its most recent uncorrupted version. Your network hacker and anti-virus protection program must encompass these four elements:

1. Prevention through firewalls, anti-virus measures, regular, ongoing anti-hacking analysis, and policies that are taught and enforced.

2. An intrusion detection system that will monitor your computers and notify you when something happens.

3. A quick-reaction team and quick-reaction plan for the time you do get hit with a virus or hack attack. You have to be ready with a plan to quarantine, wipe clean, and restore any computer that is attacked by a virus or hacker.

4. An after-action routine that will allow you to examine what happened and the holes in your security plan when bad things do happen to your system, so that you can patch those holes against future attacks.

Special thanks…

This chapter was derived from a vast array of material, but I'd like to personally thank:

Peter Coffee, Timothy Dyck, Jim Rapoza, and Cameron Sturdevant of Ziff Davis, for writing *eWeek*'s 2001 series, "5 Steps to Enterprise Security," whence we extracted the security research site list[1].

Jean-Baptiste Hervet, of Lagoon Software, for help on defining what should be scanned and how often (as well as for his wonderful MacAnalysis program).

1. The entire series is much more comprehensive than the material we present here, and can be found at www.eweek.com/category2/0,3960,3647,00.asp.

Also, Merche Shannon and Wyatt Banks from NetIQ, for all their help with intruder analysis and intrusion detection systems.

Caveat emptor

Very much like the network troubleshooting chapter (see *Network corruption* on page 125), we aren't trying to turn you into a security specialist here. Reading and following the guidelines set by this chapter will give you the bare basics of what it takes to secure your system.

This is not the place to go into a play-by-play description of what ports to leave open in firewalls, SNMP devices, and key services.

This *is* the place for backup administrators to discover where they and their actions fall in the general realm of hacker and virus defense management.

In the chapter appendix, we've provided a list of various website resources for you to choose from to learn more about security (see *Security research sites* on page 257). We've also provided a "How hackers hack" segment (see *How does a hacker hack? What does he look for?* on page 246).

Finally, if you search Amazon.com for *security*, you'll find a plethora of books to help you. Macintosh users should turn to www.opendoor.com, for a book specifically about securing Macintosh systems: Alan B. Oppenheimer and Charles H. Whitaker's *Internet Security for Your Macintosh* (Peachpit Press, 2001).

That said, let's proceed. If you aren't the security administrator, go see him or her with this chapter and a box of hot Krispy Kremes with a nice, big cup of coffee, and beg for assistance in scanning—and then plugging—all the holes you find on your network. If you've just become the security administrator, here's a quick rundown on the process you need to run through: Scan, Protect, Assign, Fix, Verify.

"I'M GOING TO IGNORE THIS CHAPTER BECAUSE I HAVE A FIREWALL."

Uh, better not! Two basic types of firewalls are in service in computer networks today. The first is the software-based firewall usually loaded onto a server that connects on one port to the Internet, and on the other to the rest of the computer network as shown at the bottom-left of Figure 9-1. This could be as simple as a built-in firewall that comes with some servers or as complex as Check Point's Firewall-1.

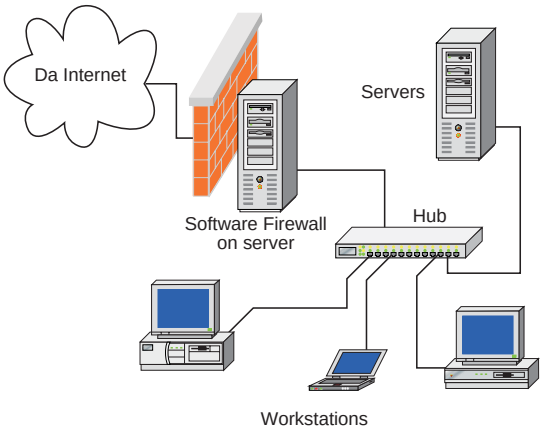


Figure 9-1. Software firewall

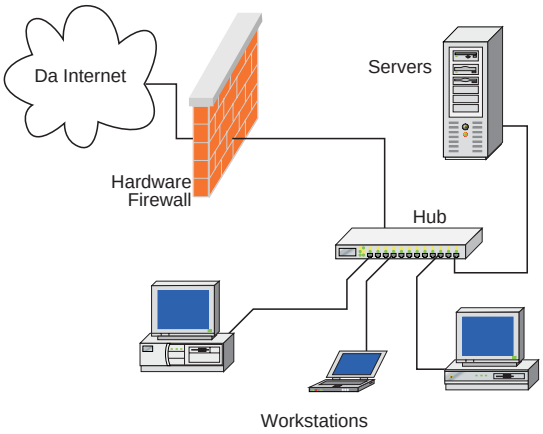


Figure 9-2. Hardware firewall

Then there's the hardware firewall that's a stand-alone device that separates the Internet from the rest of the network as shown at the top right of Figure 9-2. This could be as simple as the firewalls built into a cable router or as complex as a Cisco Pix.

Many people think that once they put up a firewall and create corporate security standards for what can go in and out, All Is Well—they're safe from those nasty things that go bump in the night. If think that way, the 49 percent of the hackers who come from *inside* the organization will love you for it. Ever wonder *why* those organizational security software makers like NetIQ, Foundstone, and others focus their efforts on scanning network *systems*? Because there are a whole lot of hackers who know how to get around the firewall, that's why. Just because you have a fire-

wall doesn't mean that it's going to stop anyone from hacking your system. No one's invulnerable—even Superman was laid low by those nasty green rocks.

In a real security test we conducted for a company in Silicon Valley, we were in their meeting room discussing why my security audit was "over the top," as they put it. While they pooh-poohed my "ludicrous" proposal, one of our staff members, dressed in coveralls, brought in a fake work order and proceeded to detach a database server from the network, jam it in a box, and lug it out the front door. Outside the building, he knocked on the window as if on cue, proudly displaying the company's server, strapped to a dolly. Although that beautiful bit of guerilla theatre didn't get us the gig, the company did get the point: Security doesn't stop at the firewall.

During another audit, this time of my own company, a prankster simply walked into a conference room, hooked up his laptop to one of the available network ports (DHCP was running, so he could get an address), and right in the middle of a conversation, he scanned the network and interrupted services. Since there was no "map" of where DHCP was giving out addresses (at the time, it was set up for all eight floors of the building), the security person couldn't find the prankster—though he was hacking away right under our noses (we caught him the next time around).

In an article by Sharon Gaudin of *Network World*[2], the dangers of insider attacks were presented by the U.S. Secret Service. James Savage, deputy special agent in charge of the Secret Service's financial crimes division, was quoted as saying, "The insider poses the greatest threat because they know where the most critical information is kept and how to bypass the safeguards on the system." He further added that "information data is the new currency of choice in the criminal community. Our dollars are best spent on prevention."

Back to the backup administrator's home front. You decide that you're running out of storage room, so you buy one of those easily-secured Network Attached Storage devices you hear about. You got it out of a catalog, from Dell, Quantum, Linksys, or Iomega—good stuff. You install it and set up passwords, maybe even tie it into your Windows domain controller for added security. You're set, right? Wrongo, Bucko. You might as well have gotten the model with the concentric-circles target painted on it. Curious? *Read the appendix to this chapter.*

2. "Study looks to define 'insider threat," *Network World*, 3/4/2002.

Testing (scanning) your network

You might want to use several methods to scan your systems for possible attacks. You can scan from the outside of your organization, testing to see how much gets through your firewall. You can scan from inside your organization, without the benefit of the firewall. I suggest that you run your tests from both directions—and take the inside attack just as seriously as the outside attack.

A host of available products conduct vulnerability assessment scans. Symantec's NetRecon (http://enterprisesecurity.symantec.com/), the open source–based Nessus (http://www.nessus.org/), and MacAnalysis (http://www.macanalysis.com/) are three very solid tools for small to medium-sized organizations.

The only two tools we recommend for larger organizations are FoundScan from Foundstone (http://www.foundstone.com/), and our company's favorite, Security Analyzer from NetIQ (http://www.netiq.com/).

Each of these packages scans either individual devices or your network, and prods the devices being scanned for open holes and potential vulnerabilities. Good stuff, these tools. But, as the immortal Jacqueline Susann taught us, once is not enough: You need to use these tools often, on a regular basis, to ensure that your software updates aren't cracking open any crevices for creepy-crawlies to slither through.

NetRecon

Symantec NetRecon utilizes a root-cause and path-analysis engine to illustrate the sequence of steps taken to uncover vulnerabilities. It tests the entire network infrastructure for security vulnerabilities and provides repair recommendations. It also learns as it scans, so if it cracks a password on one system, that password is then tried on others. Administrators can schedule repeating scans, as well. Management reports can be tailored for a range of audiences both technical and executive, and can be exported to a variety of formats including Microsoft Word, Excel, and HTML.

Shields Up!

The simplest scanner of all is a website dedicated to Internet security testing, featuring the web-based program, Shields Up! To use it, simply click their "Test My Shields" or "Probe My Ports" buttons, and the system scans your computer (this works only on the computer you're testing) for open holes in your system.

It's a great start, but doesn't cut the mustard for a workgroup or an organization[3].

Port	Service	Status	Security Implications
21	FTP	Closed	Your computer has responded that this port exists but is currently closed to connections.
23	Telnet	Closed	Your computer has responded that this port exists but is currently closed to connections.
25	SMTP	Closed	Your computer has responded that this port exists but is currently closed to connections.
79	Finger	Closed	Your computer has responded that this port exists but is currently closed to connections.
110	POP3	Closed	Your computer has responded that this port exists but is currently closed to connections.
113	IDENT	Closed	Your computer has responded that this port exists but is currently closed to connections.
135	RPC	Closed	Your computer has responded that this port exists but is currently closed to connections.
139	Net BIOS	Stealth!	There is NO EVIDENCE WHATSOEVER that a port (or even any computer) exists at this IP address!
143	IMAP	Closed	Your computer has responded that this port exists but is currently closed to connections.
443	HTTPS	Closed	Your computer has responded that this port exists but is currently closed to connections.
445	MSFT DS	Closed	Your computer has responded that this port exists but is currently closed to connections.
5000	UPnP	Closed	Your computer has responded that this port exists but is currently closed to connections.

Figure 9-3. Shields Up! test

Nessus

Nessus is an open source project, which means that more programmers are working on and making it better than any proprietary program. And in its open fashion, it has a plug-in architecture. Each security test is written as an external plug-in, so that you can easily add your own tests without having to read the code of the Nessus engine.

The Nessus Security Scanner includes NASL (Nessus Attack Scripting Language) a language designed to write security tests easily and quickly. The Nessus Security Scanner is made up of two parts: a server, which performs the attacks; and a client, which is the front end. You can run the server and the client on different systems

3. The site can be found at https://grc.com/x/ne.dll?bh0bkyd2.

so that you can create your reports on your personal computer while the server performs its attacks from the Unix mainframe upstairs.

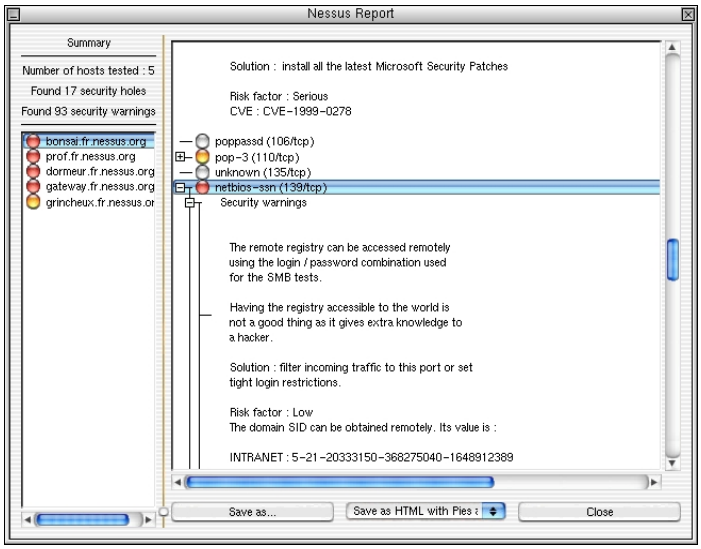


Figure 9-4. Nessus report

There are several clients: one for X11, one for Win32, and one written in Java. Nessus will not only tell you what's wrong on your network, but will, most of the time, give you the risk level of each problem found (from *Low* to *Very High*) and tell you how to prevent crackers from exploiting the security holes found.

The Unix client can export Nessus reports as ASCII text, LaTeX, HTML, "spiffy" HTML (with pies and graphs) and an easy-to-parse file format. And, given the power of your server, it can test a great many hosts at once.

MacAnalysis

The principal behind MacAnalysis is simple: It hacks your server (Unix, NT, Mac) as any experienced hacker would, informing you what it did and how to efficiently fix the vulnerabilities it exposed. MacAnalysis recognizes specific versions of specific daemons; for example, it knows the distinction between SendMail 8.8.2 and SendMail 8.8.3's vulnerabilities. Then, MacAnalysis offers a detailed description of the issues it discovers on your server.

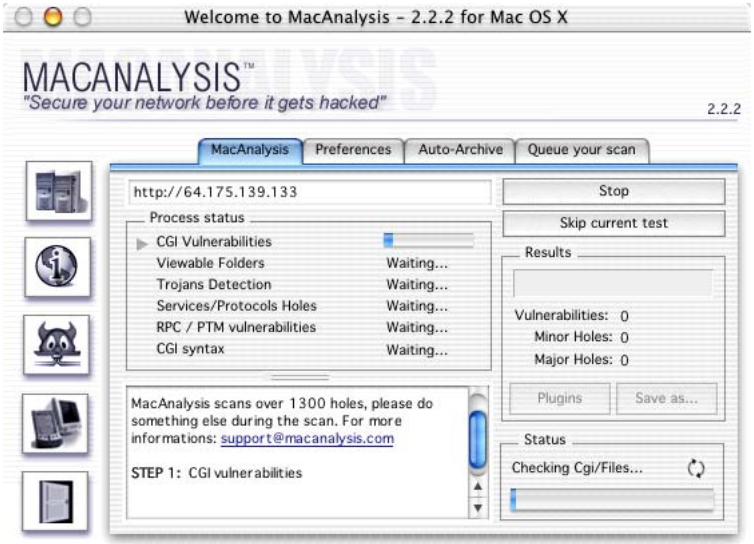


Figure 9-5. MacAnalysis

It doesn't provide path analysis like NetRecon, nor does it create fancy-schmancy graphs like Nessus. However, it does self-update its hack database and can run repeatedly scheduled tests on more than 1,300 holes to date. While it doesn't test multiple devices simultaneously, it lets you create a test schedule queue so that you can run multiple, scheduled tests. The tests we ran on our servers were done by MacAnalysis.

Corporate scanning with reporting

Corporate scanning systems should scan not only a single computer, they should run automatically and scan the entire network (adjusting for new nodes before each scan) on a regular basis. Since you're working with large-scale systems, they should include not only scanning, but problem reporting and courses of action planning, as well. And of course, with this kind of power and all-encompassing security testing you'll probably have to set aside a stand-alone computer just to run one of these babies. But trust me, they're worth it.

Security Analyzer

Another product in the same genre is Security Analyzer from NetIQ (the same folks who make WebTrends). While this isn't as heavy on business logic reporting capabilities as FoundScan, it is an *industrial-strength* product that does the job. NetIQ's Security Analyzer is a flexible, enterprise-scale vulnerability assessment product for Windows, Solaris, and Linux platforms. Security Analyzer scans computers in your network for vulnerabilities, providing reports that help you correct the problems it finds. Security Analyzer supplies detailed correction instructions, helping you close the door on attackers and prevent expensive outages. And it keeps itself up to date in its testing and corrective instructions. *As a technologist, I like this product immensely.*

Security Analyzer's security analysis report is broken down into a summary page, general statistics (how many hosts were scanned, a breakdown of high to low vulnerabilities found, etc.), host vulnerabilities, service vulnerabilities, the test policies put into play during the scan, and inventories of both the number and types of hosts found, as well as number and types of network services found.

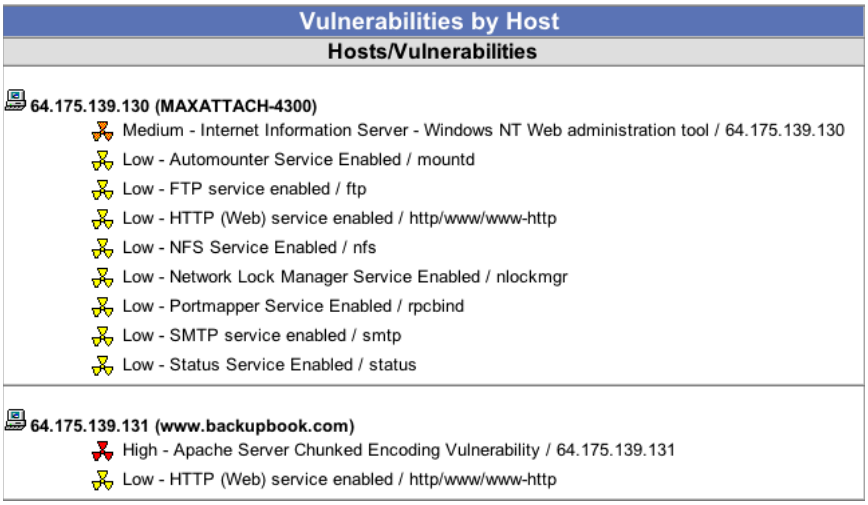


Figure 9-6. Vulnerabilities by host

The Host Vulnerabilities Report (Figure 9-6.) provides a visual breakdown of the high, medium, and low vulnerabilities *per device* that were found in the network

scan. This graph shows a total of eight computers spanning a range of vulnerabilities.

While all of this is nice and scary enough to separate your CFO from some bucks for upgrades and patches, the product's real strength becomes evident at the end of the vulnerability report, when it gives instructions on *what to do* to fix the problems found.

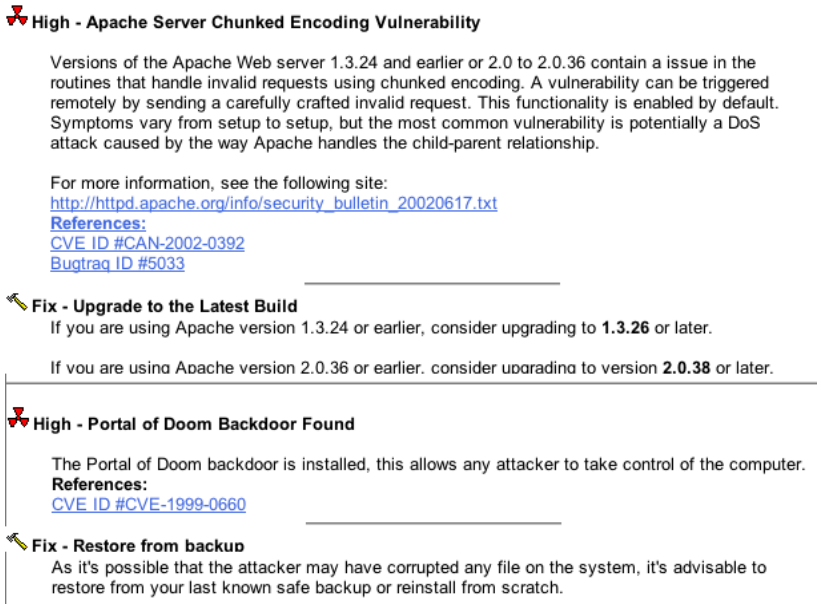


Figure 9-7. Fixit instructions

How smart is this "what to do" advice? I've spent hours talking to the engineers at NetIQ, ISS, Citadel, MacAnalysis, and other software vendors, and have come to realize that these guys are deadly serious about what they do. They're even more serious about the very best methods to handle the viruses, Trojan horses, and open security holes they find: They create a fix, attempt to hack the fix, and re-create and re-test until they get it right. The best move you can make is to follow their instructions directly and promptly.

Internet Security Systems Internet Scanner

Founded in 1994, Internet Security Systems (ISS) (Nasdaq: ISSX) is a world leader in software and services that protect critical online resources from attack and misuse. Their vulnerability assessment tool, Internet Scanner, performs scheduled or event-driven probes of network communication services, operating systems, routers, e-mail, web servers, firewalls, and applications to identify weaknesses that could be exploited by intruders to gain access to the network. Their SmartScan security data correlation detects interrelated network-based vulnerabilities, learns from vulnerabilities detected in previous scans, and builds on this knowledge to discover additional vulnerabilities that would otherwise go undetected. The additional FlexCheck capability provides additional coverage by allowing customers to write updates for custom applications.

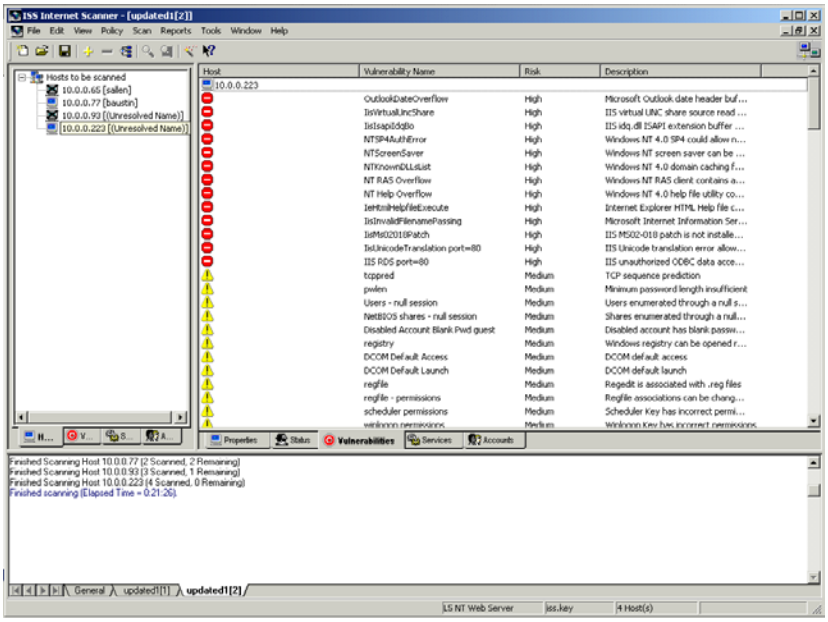


Figure 9-8. Internet Scanner

At this point, you can either remediate the problem manually, or you can turn to a remediation product—which we just happen to know about; read on.

QualysGuard Intranet Scanner

This product is the only hardware-based security scanner that we mention in our book. It's a pretty darn cool product. The device hooks into the network like any other computer and continuously monitors the goings-on, producing a web page front-end report that is clean and very usable.

Figure 9-9. QualysGuard Intranet Scanner

Just minutes after turning it on, connecting it to the network, and entering a username and password (for getting updates from the parent company), administrators can begin assessing their networks for vulnerabilities. If you're looking for an appliance to run your security tests, this is a great device.

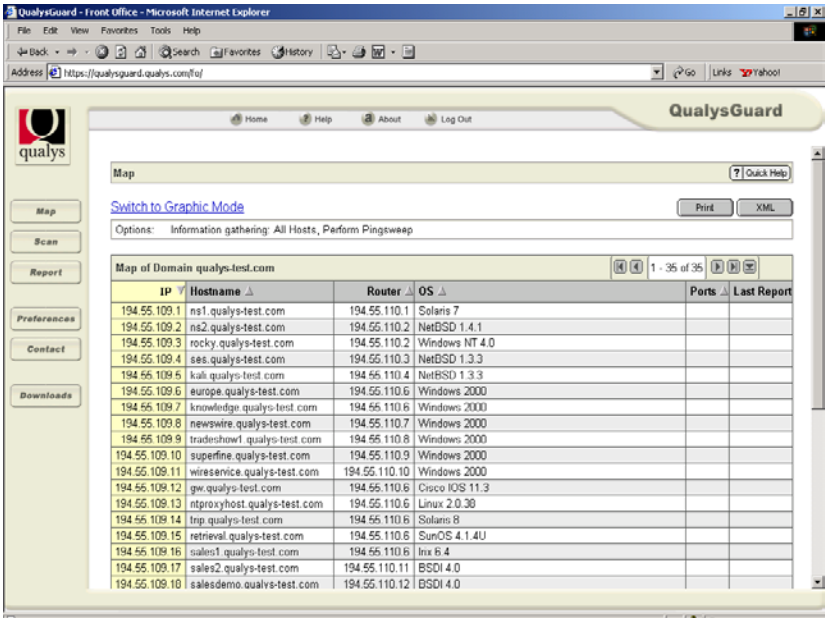


Figure 9-10. Intranet Scanner web front end

FoundScan

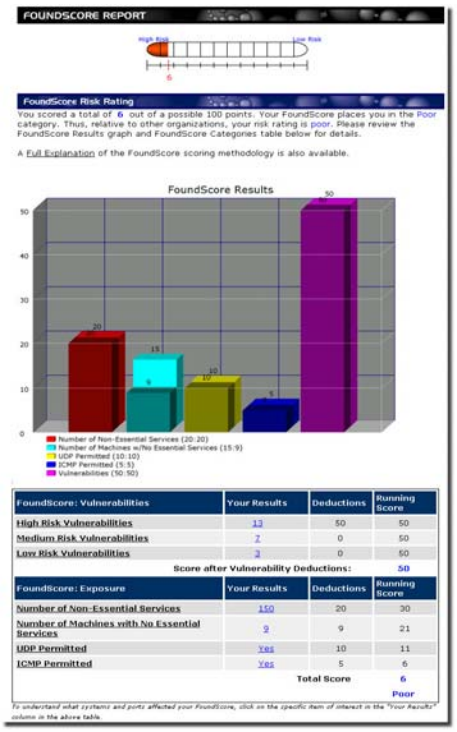


Figure 9-11. FoundScore

Foundstone's FoundScan application is one of the best scanning and reporting tools we've seen. We like FoundScan so much because of FoundScore, its reporting capabilities that rate an organization's vulnerability status from 0 to 100. These reports are powerful enough to get the job done, and easy enough to interpret so that you can present them to the CEO and CFO at your corporation's executive management meetings.

Trying to explain security issues to non-technical audiences can be frustrating. The *lingua franca* of CFOs is *not* patches, service packs, and upgrades—it's "What's the bottom line?" FoundScore levels this playing field nicely. Based on exposure risks (firewall policies and Internet architecture) and vulnerabilities discovered during its assessments, FoundScore is a quantitative gauge of the organization's success at securing their systems. Most usefully, the Long-Term Trend Report tracks an organization's FoundScore over the last 10 scans, graphically depicting improvements to the organization's risk profile so that the CFO can see the value of the security investment.Figure 9-11. shows a sample FoundScore report.

Immediate remediation

While I was giving a presentation in Orlando not so long ago, I met two very bright senior-level IT staff members, who asked me about immediate remediation products and whether or not I'd heard of them. Until that very moment, I had to admit that I hadn't. Soon after returning from the show, I was contacted by the folks at Citadel Security Software regarding Hercules—no, not the Greek hero; their vulnerability remediation product. I was impressed—so much so that I just *had* to call this chapter back from the editor to add this product to the list.

What does Hercules do, when it's spruced up those filthy Augean stables? Its labors, like those of its eponym, are simple, yet quite complex.

	Without Hercules		With Hercules
1.	Assign the vulnerability to a staff member	1.	Import Security Scan into Hercules
2.	Locate the machine/device		
3.	Research the vulnerability	2.	Review the vulnerability/remediation
4.	Research the remediation		
5.	Acquire & test remediation		
6.	Apply remediation to each system	3.	Click a button to remediate
7.	Hope for overtime or comp time	4.	Drink coffee and eat Krispy Kremes

Table 9-1. Manual remediation process vs. Hercules automated remediation process

Hercules is a classic case of swapping the cost of time with the cost of hardware and software. How much time is that? More than two hours a day, when scans are

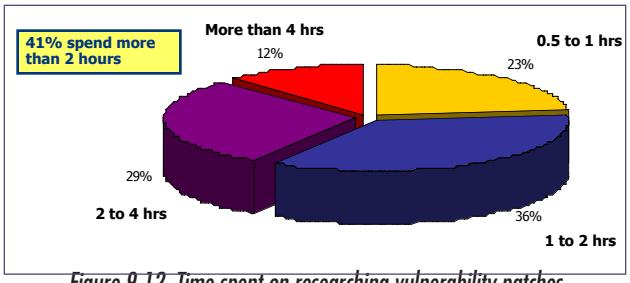


Figure 9-12. Time spent on researching vulnerability patches

happening for researching the patches alone, that's how long.Figure 9-12. shows a chart from a recent survey asking folks how much time they spend on researching vulnerability fixes (around one to four hours of research per patch). However, that's only a part of the manual-fix picture. Let's run a hypothetical scan of your system for vulnerabilities; say, 250 computers on the network and nine servers. The first time you run your scans (if you're a normal person and you've not been as security conscious as you should), you'll find a lot of patches and security updates that need changing. I'm not going to even try to guess how long that would take—probably as long as the original Hercules' 12 labors put together.

- Once you've cleaned up the system, and scan once a month, you find an av-

erage of three patches per server and one patch per computer. There are nine different patches for the servers, and six different patches among the 250 workstations.

- Research time for each patch is about two hours on average. That makes nine plus six patches times two hours, for a total of 30 hours of vulnerability research.
- With the network down for maintenance on a Friday night, the tech comes in and patches five devices simultaneously. At this rate, five patches can be done in half an hour, or one patch in six minutes. With three patches times nine servers and one patch times 250 workstations, that's 277 patches * 6 = 1,662 minutes, or 27.7 hours.
- (30 hrs. research + 27.7 hours updating/patching) = 57.7 hours * $30 an hour (that's for your cognitive IT professional) = $1,731 month in time costs.
- $1,731 a month * 12 months = $20,772 per year.

I'm sorry, but when I can have a computer assigned to do the legwork of securing other computers for me, I don't even think twice—that's where Citadel's Hercules comes in. Let's look at the process.

Importing a security scan

The first step is to run your security audit with your favorite tools, such as Nessus, etc. You then import your security scan into Citadel's Hercules. Hercules maps the information from the security scanner into its own format, examines its own templates and profiles, and then presents you with a window like the one shown inFigure 9-13.

The mapped vulnerabilities show up as a two-part window. The left windowpane reveals the list of computers that have vulnerabilities; the right windowpane shows the vulnerabilities that the system found on those computers, with some basic information such as the vulnerability's name, severity, and CVE reference number (CVE.mitre.org is an outstanding reference site and clearinghouse for vulnerability assessment).

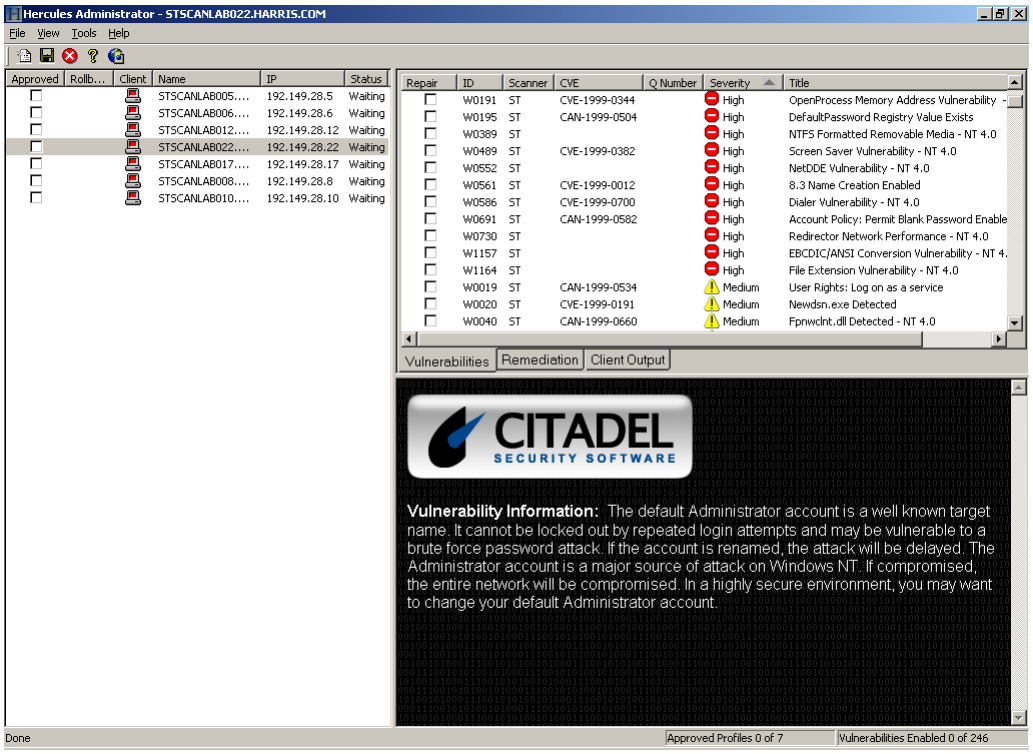


Figure 9-13. Mapped vulnerabilities window

In the Hercules system, every vulnerability has a description as well as a signature (think of this as the DNA of the vulnerability). The vulnerability's signature includes whether or not the patch can be uninstalled (which is very important if you find out that the patch causes more problems than the vulnerability), and where the patch can be found. It also includes the registry information for where the patch will be placed, along with other registry data and values.

Double-click on one of the vulnerabilities in the right pane of the mapped vulnerabilities window to get a dialog showing you the vulnerability's signature and giving you three distinct options. You may either chose to fix the vulnerability, ignore it, or (I really like this) roll back a previously fixed vulnerability. Some patches are best roll backed, because the patch turns out to be worse than the vulnerability. Knowing which *can* be rolled back and which *can't* be is a major plus—and doing it automatically is even better. Ever try sending out a tech back to each computer

that he installed a patch on and ask him to roll the device back to the way he found it? Yeah, like *that's* going to happen sometime soon. But even if it does, you'll be the new target on his dartboard. But back to the main thing: fixing vulnerabilities. Once you set the property to **Repair**, you go through the window again and instruct Hercules to repair all or any subset of devices that had this vulnerability.

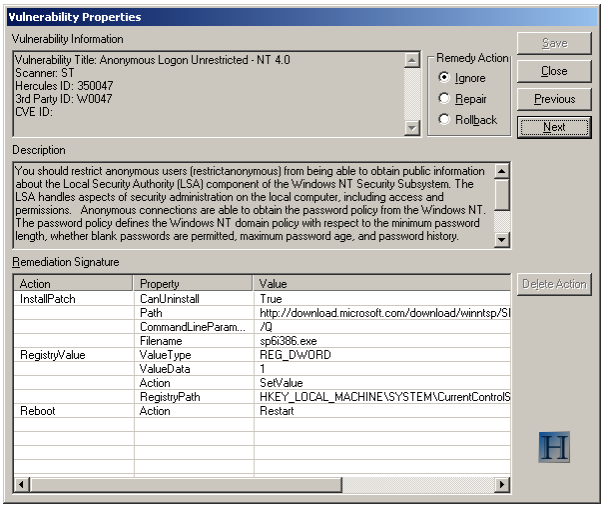


Figure 9-14. Vulnerability properties

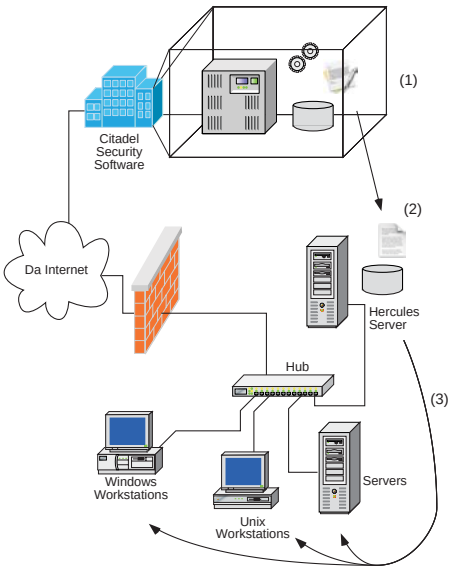
You can even create custom fix signatures so that you apply one set of rules for servers, another set for end users, and a third set for those kind folks in HR who scotched your last pay raise.

Reporting

I've gone through Hercules' reports, which are quite nice and can be customized. However, as a former executive committee member of a publicly traded company, I'm going to tell you a little secret: The best report you can ever present is the **before-and-after report**. You take your first Nessus scan, all 160 pages of it, and put it in a three-ring binder. You then use Hercules to fix all of your problems. Run another Nessus scan, and take that beautiful, blank single page with nothing on it and put *that* in a binder of the same size by itself. Then, ta-daa! You present the "before" and "after" binders as your full presentation to the audit committee,

while saying no more than "Here's where we *were*, and here's where we *are*." You'll get that next pay raise.

How Hercules works and what you'll need



The Hercules system works in three distinct steps, and you need to run it on a server class machine. If you have a server with enough processing power and disk available, use that. If not, add another machine to your network, and enough bandwidth to send the patches across (which really doesn't take much bandwidth at all).

Citadel's offices keep constant watch on the CVE list and others,continually updating their master database of all known common vulnerabilities and exposures.

In step one of the process, all incoming intelligence is scrubbed and then added to their database, where a combination of their proprietary signature creation software and their engineers build and then test the constantly updated list of vulnerability and patch signatures.

Figure 9-15. The Hercules process

In the second step, the local Hercules server resident on your network downloads those signatures into its database so that they'll be ready as soon as you are. Once you have the most up-to-date information in the server, you're ready to load in your scans and then patch your devices.

The third and final step, of course, is the patching of your devices while you're gulping Jamaican Blue Mountain coffee and hot Krispy Kremes.

Politics and other issues of computer-driven security systems

Okay, we've now run through the process of including a semi-automatic security remediation system in your security plan. You're probably wondering if your company will buy off on it and what the fallout might be. Let's go through a couple of issues that can hit you, do you'll be prepared to dodge.

This is an expense. Sure is. The price might be coming down as they grow and gain marketshare, but whatever the price, *somebody* is bound to balk at it. You've gotta be persuasive here. I'd balance an argument for this product against the argument of hiring additional staff to keep up with security threats. Ask for the staff first, and then add (once you get them going on the exorbitance of buying fresh bodies) that you *have* found a device that'scheaper and easier to manage and maintain than a human being—that might clinch the deal.

And, since you're replacing bodies with machinery, there's bound to be a bit of a turf war in larger organizations. Decentralized companies like restaurant chains or store chains won't offer that argument because they don't have the decentralized IT staff to begin with. However, larger organizations with their own IT fiefdoms may put up a security turf barrier or two. That's where "approval" teams come into the picture.

When I was CIO of True North Communications, I dealt with about a dozen "local-division" CIOs across the world. With those based in the U.S., I held a monthly meeting to "decide" where we were going. It was like the President wrangling with Congress—but it worked, and I recommend it as a turf-war-deflection tactic. At monthly meetings, you can turn the terroritorial instinct aside by presenting the most current security scans along with the most current Hercules-suggested remedies. Letting folks "decide" which updates should be run gives them a voice in the system, and enables to you get their backing and, potentially, some bucks if you need more budget—which you *are* going to need for this system.

PROTECTION: SET UP AN INTRUSION DETECTION SYSTEM (IDS)

In tandem with assigning responsibilities and fixing problems, you need to set up a proper intrusion detection system so that if you get hit again mid-fix, you'll know who's hitting you and how they' re hitting you. What you do *not* want to do is to back up systems that have been hacked! This is one time you don't want a new backup. Think of it: If you back up a back-door operation like the Doom back door, you've effectively put a back door into your offsite storage—not good[4].

The best thing to do? Immediately erect an **intrusion detection system** while you're assigning responsibilities and conducting the fixes that the security analysis software found. There are two good, basic choices for this, and one *outstanding* choice in software.

Snort and HenWen 

Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It performs protocol analysis and content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more.

Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture.It has a real-time alerting capability, as well, incorporating alerting mechanisms for syslog, a user-specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's SMB client.

Named after the ancient Welsh sow goddess and Lloyd Alexander's temperamental oracular pig (check out *The Book of Three* for a great all-ages read), HenWen is a network intrusion detection package for Mac OS X that makes it easy to configure and run Snort. HenWen simplifies setting up and maintaining software that scans

4. See *Responding to a hacker attack or other security breach* on page 216for more info.

network traffic for undesirables that a firewall may not block. Everything you need is bundled in, with no compiling or command-line use necessary.

Security Manager 

NetIQ's Security Manager takes over where Security Analyzer leaves off by creating a system for automated response, host-based intrusion detection, event log consolidation, and security configuration management. The automatic response portion of this product is amazing. Security Manager provides more than 30 out-of-the-box responses, ranging from administrative tasks, including enabling or disabling user IDs, to operations tasks that include starting and stopping services and terminating applications and processes. Security Manager is more than an intrusion detection system, it's a defense intelligence system.

You're probably wondering what exactly constitutes a defense intelligence system for a network. Chances are, you've already encountered one in your childhood. What am I talking about? Actually, it's a who—every bad kid's nightmare, Sister Mary Knucklebuster of the School of Perpetual Sorrows, that's who. She's the nun who just sorta Knew All about your every questionable deed. No matter what you did, she knew it—and then *whack!* Sore knuckles for a day. It didn't matter if you screwed up in gym—*whack!* The lunch hall—*whack!* Putting prickles in Janey-Sue's pigtails—*whack!* She had eyes *everywhere*. How? Because each and every person around you was snitching, that's how. She *knew* because she had her tentacles wrapped tightly around every kid-snitch, teacher, and assistant in school, that's how. Very, *very* effective. Now think about Sister Mary's effectiveness if the only person in school who reported on your misbehavior was the principal—you'd be sailing off scot-free.

A network defense intelligence system is Sister Mary Knucklebuster's mechanical doppleganger. The heart of any defense intelligence system is the security auditing console—probably a SQL server running on your network somewhere that gathers all your security logs from the different security monitoring silos, such as your corporate firewall logs, your network intrusion detection system reports, each key user's personal firewall logs, etc. NetIQ's Security Manager is such a device.

The defense intelligence system gathers all the various security reporting tools' logs mentioned above, and puts the pieces of each individual security audit into their proper place in the puzzle. Recently, Wyatt Banks of NetIQ was up at the office, and we were discussing the role of such a system. As he pointed out, "It isn't

that situation A being reported by audit software is bad in and of itself. Nor is it that situation B is very bad in and of itself. However, when situation A happens in tandem with situation B, you might have an attack on your hands." And the only way to build a secure system is to make all of your tools work *in concert* with each other. Let's walk through what it takes to build such a system.

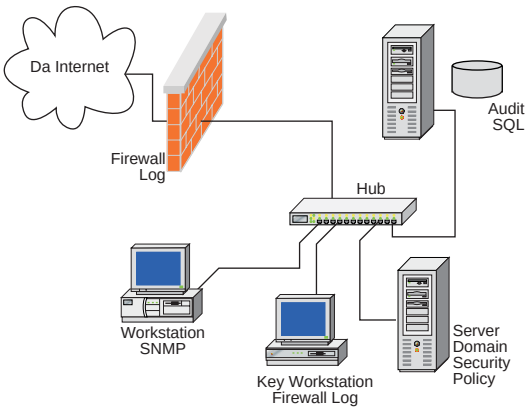


Figure 9-16. Security audit system

Know what is where

As we've stressed in our network corruption chapter, in *The computer's-eye view of your network* on page 153, one of the most critical baselines of networking is the name game: You must name your computers so that you can find them and differentiate them from each other. If you need to know how to do that, go back to the network corruption chapter (see *Network corruption* on page 125) and follow those directions.

Set local security policies

You can set the local security policy on Windows NT, 2000, and XP computers by enabling Windows security auditing, which enables the operating system to send security-specific events to the security event log on your computer, which is picked up by the security audit console. You can set these security audit settings in **User Manager** in NT and **Local Security Policy** in 2000 and XP. Both are in

the **Start > Programs > Administrative tools folder**. Keep at least 20 MB of space for your logs.

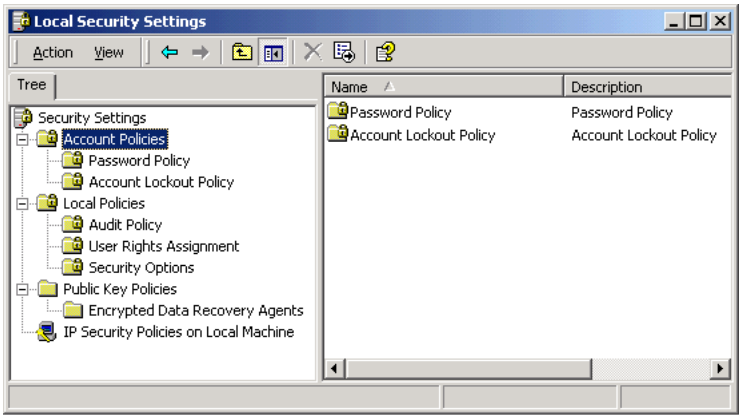


Figure 9-17. Windows 2000 Local Security Settings

While Mac OS X does indeed have a firewall system built into each computer (you can access its setting through the Sharing folder in System Preferences), it's not entirely clear whether or not any attempts or breaches are logged to the syslog file on the computer. As we learn more about this operating system, we'll update our information on our www.backupbook.com website. Even if the firewall isn't logging information that the audit console can read, it won't hurt to turn the firewall on anyway—but you'd do well to get additional computer-based security tools like BrickHouse or FireWalk for OS X.

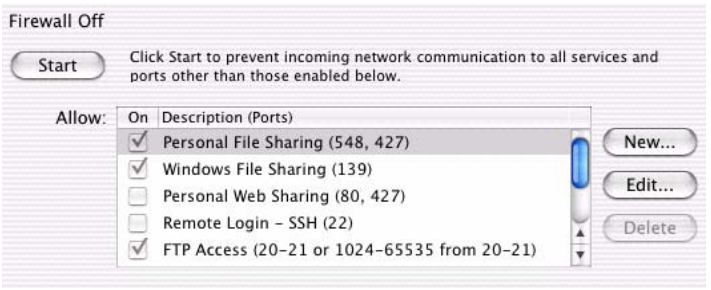


Figure 9-18. OS X built-in firewall

For Linux, you'll be working with ipfwadm. No, I didn't abandon spellcheck: **ipfwadm** is a utility to administer the IP accounting and firewall services offered by the Linux kernel. Firewall rules can be created or deleted via the **setsockopt(2)** system call. Via the same mechanism, a filter's default policy can be changed. Filters can be inspected by reading the following pseudo-files in the proc file system:

/proc/net/ip_input /proc/net/ip_output /proc/net/ip_forward

Each of these files lists the default policy, followed by the details of all rules (if any) belonging to that filter, in a compact format. The **ipfwadm** command provides a command-level interface for managing the Linux firewall facilities, and can be used to change or inspect all aspects of the kernel filters.

Standardize your security policies beforehand, document and then post them somewhere in your intranet for your security team to read and review on a regular basis. They'll be a bit different for each OS, but that's par for the course.

Set domain security policies

You'll need to set your **domain security policies** on the Windows 2000 domain controller (this isn't available in Windows NT 4.0). Domain security policies override local security policies and allow you to use the policy to enable security auditing[5]. In Microsoft Windows NT Server 4.0, *domain security policy* referred to an associated group of items considered critical to the secure configuration of a domain. These include:

- Audit Policy to control what types of events are recorded in the security log. User Password, or Account Policy to control how passwords are used by user accounts.
- User Rights are applied to groups or users and affect the activities permitted

5. This and other security information can be found on Microsoft's support website, and in more detail in their white paper: "Introduction to Windows 2000 Group Policy," http://www.microsoft.com/windows2000/techinfo/howitworks/management/grouppolicyintro.asp.

on an individual workstation, a member server, or on all domain controllers in a domain.

In Windows 2000, Microsoft has reconfigured these components into one consistent hierarchy or tool, the Security Settings snap-in in the Group Policy Editor. This may be useful if you want to know the proper group policy object to change.

To configure security settings intended to span a domain, use the Group Policy Editor snap-in, with its focus set to the "Default Domain Policy" Group Policy object (GPO):

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Active Directory Users and Computers**.

2. Right-click the appropriate domain object, and then click **Properties**.

3. Click the **Group Policy** tab to view currently linked group policy objects.

4. Click the **Default Domain Policy** GPO link, and then click **Edit**.

After you start the Group Policy Editor snap-in, you can gain access to domain security policies from the following node:

Console Root\"Default Domain Policy" Policy\Computer Configuration\Windows Settings\Security Settings

At this point in the hierarchy, you have access to both the account policies and the local policies.

When a computer is joined to a domain with the Active Directory and group policy implemented, a local **Group Policy Object** is processed. The Windows Group Policy is administered through the use of these Group Policy Objects, data structures that are attached in a specific hierarchy to selected Active Directory objects, such as sites, domains, or organizational units. These GPOs, once created, are applied in a standard order: LSDOU, which stands for (1) Local; (2) Site; (3) Domain; and (4) OU; the later policies are superior to the earlier applied policies.

Set Windows group policy

Windows 2000 domain controllers pull some security settings only from Group Policy objects linked to the root of the domain (the domain container). These set-

tings from Group Policy Objects are not applied on the domain controller's organizational unit, because a domain controller can be moved into a different organizational unit. Using the domain container allows these setting to be applied regardless of which organizational unit the domain container resides.

Because domain controllers share the same account database for the domain, certain security settings must be set uniformly on all domain controllers to ensure a consistent experience for all domain members regardless of which domain controller they use to log on. Windows 2000 accomplishes this task by allowing only certain settings in the Group Policy to be applied to domain controllers at the domain level. This Group Policy behavior is different for member server and workstations.

The following settings are applied to domain controllers in Windows 2000 only when the Group Policy is linked to the domain container:

- All settings in Computer Configuration/Windows Settings/Security Settings/Account Policies (This includes all of the Account Lockout, Password, and Kerberos policies)
- The following three settings in Computer Configuration/Windows Settings/Security Settings/Local Policies/Security Options:

  Automatically log off users when logon time expires

  Rename administrator and guest account

If you have a large network with multiple subordinate managers, each of your subordinate managers will more than likely want to set their own Group Policies. Therefore, you should delegate the authority for editing this Group Policy. Because this is an involved process, we've listed the steps for both setting the policy and delegating the authority at the end of this chapter in the chapter appendix beginning on page 253.

Integrate your defense intelligence system with Windows SCM

SCM is Microsoft's Security Configuration Manager, which can be integrated into defense management and intelligence systems such as NetIQ's Security Manager software. Because this is specific to the software you use, we can't really cover it in

detail here. Let's just say that each of the packages will present you with its own integration wizard, and they all seem to work like a champ.

Monitor devices that provide network services

Next, tie all of your devices that support *any* type of network services into your defense intelligence system. This includes, but is not limited to the following:

- Devices that run SNMP (Simple Network Management Protocol), which is used to monitor and administrate network-connected devices. You want to tie SNMP devices into the mix because SNMP uses an unencrypted string for authentication known as a **community string** (read *public* for those who haven't configured their systems otherwise). This lack of encryption poses *major* security risks, even for devices that are supposed to have SNMP running—so make sure that these devices are monitored by your intelligence system.
- If you have a Remote Access Service (RAS) device (it's bundled with Windows NT 4) that provides remote access to the network, monitor that system as well. This is a major-league door to the outside world (if it's hooked up); thus presenting a serious security risk. In Windows 2000, this is called RRAS: Routing and Remote Access Service. And, of course, for every instance of RAS or RRAS that you don't want running, make sure that your monitoring system has shut them down.
- IIS servers are notorious for security holes. You'll *absolutely* want to tie them into the monitoring system.
- Of course, all your firewalls should be added to the monitoring system, as well. Any Cisco PIX, Check Point firewall, Check Point reporting server, and Check Point firewall management servers should be on this list.

As you can see, there's a lot to setting up and managing intrusion detection systems and Windows 2000 systems in a secure network, and you need some solid tools to collect and then monitor your policies and logs within your network—it just can't be done any other way. And, if you're serious about inside-the-firewall security management, you'll put both the policies and the tools to work.

ASSIGN RESPONSIBILITIES & CREATE YOUR TEAM

Once you've got your intrusion detection or defense intelligence system running, assign the hacker or virus attack fixes to the most likely candidate in your organization—probably the individual server's administrator, and, in the case of workstations with holes and vulnerabilities, whoever takes care of network security.

By assigning the various fixes for each box to the person responsible, you have now set a date by which the hole should be fixed, and you can later verify that those fixes have been accomplished. Yes, you must continue to set update dates frequently, making sure that each of your patches is both current and correct.

For a team effort, share the responsibility for security-hole-testing any new platform you add to the network with the security administrator. Before you back up the new device, check off on whether or not it is secure. The security administrator must keep current on patch updates for software for each of the platforms in your system. The security admin should then communicate which patches are available for which applications to both the backup admin and the training coordinator (who should be teaching a security class to folks who run network services on their computers). While the security admin runs monthly scanning probes on the network (and full-time monitoring programs), emergency response to a hack attempt should be shared with the backup administrator.

	Equip. Testing	Updating Patches	Monthly Testing	Response Tm	Comm. & Training
Backup Admin	Yes			Yes	
Security Admin	Yes	Yes	Yes	Yes	Yes
Help Desk Admin				Yes	Yes
Training Coordinator					Yes
VAR	Yes			Yes	

Table 9-2. Responsibilities list

Why should an emergency response to a hack attempt be shared with a backup administrator? Think of it this way—if someone were to deface your website, unless the defacement was obvious, how would you know? The easiest way to find defaced pages on a website is to create a backup report. Your backup software can and will quickly tell you which pages of the website are *newer* than the last ones it

backed up. A quick coordination check with the webmaster will reveal which page changes are from your staff—and which are courtesy of the hacker.

We've also listed your VAR in the group. If your VAR is building and prepping your systems for you before they're sent to your organization, you'll want to keep her in the loop about the newest builds, images, and security patches you're working with. And, if you keep her in the loop at the best of times, your VAR might be able to come to the rescue at the worst of times with a loaner machine or other valuable services—but if it comes down to the tumbril, I wouldn't count on her pulling a Sidney Carton for you.

For everyone in your team, make sure you know whether or not they can be contacted in the evenings and on weekends, and keep multiple methods of contact on file.

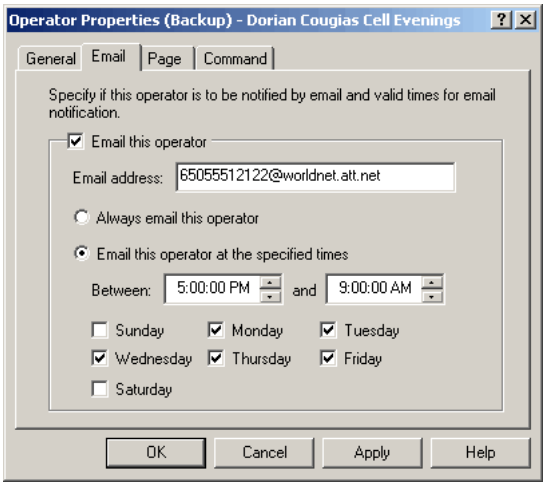


Figure 9-19. Operator Properties

As with the settings for configuring the notification groups of NetIQ, set up the user twice: for business-hours and for after-hours contact. In the window depicted in Figure 9-19. I've set myself up for evenings. I've added my cell phone number, which is set up by AT&T so that it can receive short (255 characters) text-based e-mail messages called SMS messages; this works great for evening calls when I'm not going checking regular e-mail. I then set up another account for myself during the day to send messages directly to my normal e-mail address.

Fixing the holes

Plain and simple: Fixes should be done as quickly and painlessly as possible. Your scanning software should include not only the problems it's discovered, but the recommended solutions to those problems, as well. Several Internet databases archive security vulnerabilities, but Cisco's Secure Encyclopedia (CSEC)[6] actually helps security personnel prioritize potential problems by relating security faults to specific industries. The service is free, but you do have to sign up for a Cisco ID to use it.

A quick tour of their headlines at the site turned up this little tidbit (Figure 9-20.), which not only describes the problem, but details its severity, its consequences—and its fix.

Multiple Vulnerabilitiesfound in Microsoft IIS

Severity:Vulnerability Type:Exploit Type: High SeverityNetwork Access

Description
Microsoft advisory MS02-0018 describes ten vulneralilities related to the IIS HTTP / FTP server component of Windows NT / 2000 / XP.

The vulnerabilities included in the Microsoft advisory are:
- Buffer overrun in Chunked Encoding mechanism
- Microsoft-discovered variant of Chunked Encoding buffer overrun
- ...

Consequences
A remote attacker may be able to execute arbitrary code on the web server with varying degrees of privilege depending upon the version of IIS. Denial of service attacks could render the IIS service unavailable. Cross site scripting issues could lead to the exposure of sensitive information such as HTTP session cookies.

Countermeasures
Apply the cumulative patch listed in the referenced Microsoft advisory for your particular platform(s).

Access Required:Network connectivity to the vulnerable web server.

Access Gained:Ability to execute arbitrary code or denial of service depending on the vulnerability exploited.

Figure 9-20. CESC vulnerability report

As you download your patches and apply them to the boxes on your network, communicate with your team members to coordinate your applied patches so that you don't have multiple people doing the same research.

If you're using Citadel's Hercules system, you don't have to worry about much of the above, as it's done automatically for you (and within minutes versus hours).

6. http://www.cisco.com/pcgi-bin/front.x/csec/csecHome.pl

All you have to do is import the security scan into their system (as mentioned earlier in *Immediate remediation* on page 197), decide which of the updates to apply, and then apply them. No fuss, no muss.

Verify that the holes are fixed

The best way to verify the efficacy of the fixes? Rerun the tests on the computers and check your original scores against the after-patch-and-repair scores.

If the same holes remain, someone has dropped the ball (Security Analyzer lets you run a comparison report to tell the difference between two scans).

If you can get in through a new hole, go back and put new patches in.

Once you've verified that your holes are fixed:

1. Re-image the system, and then
2. Back up the system, so if something gets hit and you have to restore, you can restore to the last known level of security before adding additional patches. If you don't, you could lose some patches you've already applied.

RESPONDING TO A HACKER ATTACK OR OTHER SECURITY BREACH

You're the backup administrator, and you're at lunch. Your feet are propped up on your desk, and you're scarfing a sloppy sub, chuckling at the latest *Mad* magazine. All is well—until you glance at your security management console and see 30 new breaches!

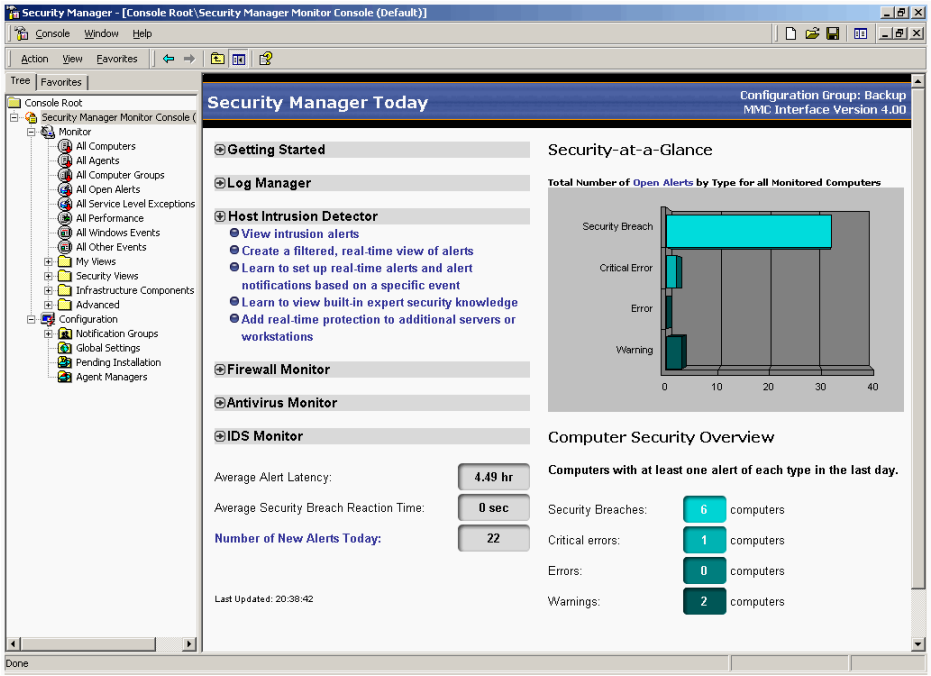


Figure 9-21. Security Manager console showing more than 30 breaches

Once you've swallowed hard and your pulse returns to somewhere near normal, if you're using NetIQ, you'd simply click the **Security Breaches** button below the bar chart to get a list of all six computers with breaches. Accessing the list, you see individual devices and then open up each of the alerts to assign them to someone to fix one-by-one. If you're using an intrusion detection system like HenWen or Snort that doesn't provide a console, you'd walk into the office and scan 30+ e-mails with the same information. Because it presents the most consolidated front

and has the better interface of the three products we've reviewed, we'll use NetIQ's Security Manager to demonstrate how to examine a condition, assign the fix, and then find a resolution. With this product, you can select each breach and open the property for that alert, as shown in the window below (Figure 9-22.).

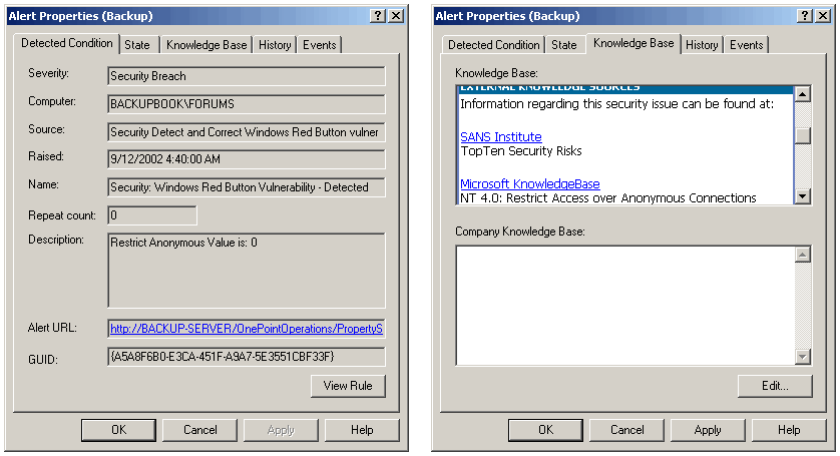


Figure 9-22. NetIQ's alert properties window showing Condition (left) and KnowledgeBase (right)

Before you assign someone in the team to fix the alert (assuming the "team" isn't just you) check the KnowledgeBase to find out more about the alert and how to fix it. As we've noted above, any intrusion detection worth its salt gives you a location to find out more about the alert and ways to fix that specific security breach. In this case, the window (right side, Figure 9-22.) reveals two locations the administrator can examine to sniff out more about this specific problem.

This particular alert sends the administrator off to the SANS Institute as well as the Microsoft website, in search of very specific answers. Once you know what in the world has happened to the device, you can assign fixing the problem to the particular person or group responsible for this aspect of your security management arena.

If you're using NetIQ, move over to the **State** tab of the dialog, assign the problem to an owner to be alerted, and then fix the problem (Figure 9-23. on page 218).

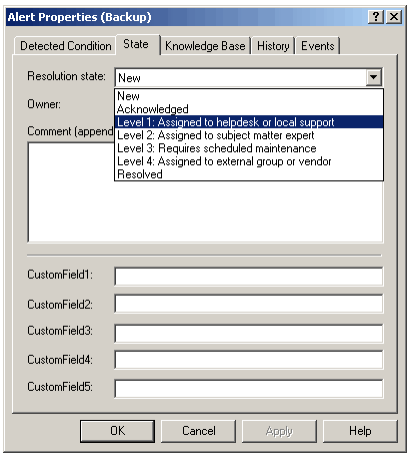


Figure 9-23. Assigning the security breach to an owner

If you're using Snort or HenWen, simply forward the e-mail notification to the appropriate person with your own notes on what to do.

Quarantine the device

The next step? *Immediately* quarantine your device. Any attacked system, especially if attacked by a virus or Trojan horse, must be taken off the normal network to stop the spread of the attacking program. Quarantining your device means *taking it completely off of the network* so that you can make a disk image of the hard drive to send to the authorities or your security personnel for analysis.

Restore the backed-up contents to a temporary replacement server

If the quarantined system is a key element in the work process, you, as backup administrator, must get a secondary computer up and running to take its place by swapping computers and doing an emergency restore of the backed-up data to the new or temporary box taking Device X's place.

This might mean moving a "hot" swap box into place, or doubling up another server to act as two for the time being. Whatever your decision, use your own business continuity plan for disaster recovery in this instance. It's one heck of a way to

test your theories, but they should work, and you should be in business as soon as you can restore the data of Server X to your temporary location on Server Y.

Now, *before* you decide to restore any of the data, check which files were modified by the hacker, virus, or Trojan horse. Also, check which methodology your backup software uses when backing up and restoring files. Knowing which methodology your backup software uses to back up files is important—especially if your system is one like Retrospect,which backs up the first instance of a file it sees (this is called **single-instance storage**), and then *doesn't* back up any other file with the same metadata (name/modify date/creation date/byte size/host file system, etc.), but merely creates a pointer to the file on the tape or disk backup.

Let's say that there are two computers being backed up: a workstation and a server. When it comes to backing up common files, such as dlls that have the same metadata, Retrospect and other backup software like it backs up the first instance of the file (in the case of Figure 9-24. , the workstation version) and then, instead of backing up the same file again, the next computer has a *pointer* to the place on the tape (or disk) where the initial file was backed up.

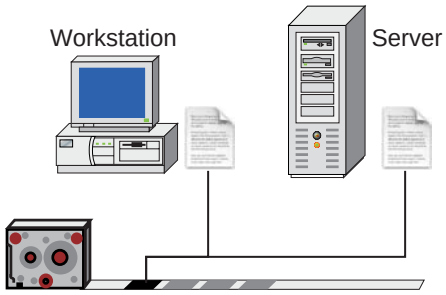


Figure 9-24. Backing up the .dll file once and then using a pointer to it afterward

So, let's say that on day 1 of the backup, everything was hunky-dory and went smoothly. But on day 4 , a hacker hits the server right before it's backed up so that you can't get there and stop the process quick enough. You have now successfully backed up a hacked file. The *good news*? As long as any piece of the metadata has changed (for example, the file has a new modification date and its size is different), the file will be different enough that no other device (unless it, too, was hacked) will have a pointer to this file. The server's snapshot of day 4's backup points to the newly backed-up (hacked) version of the dll (Figure 9-25. on page 220).

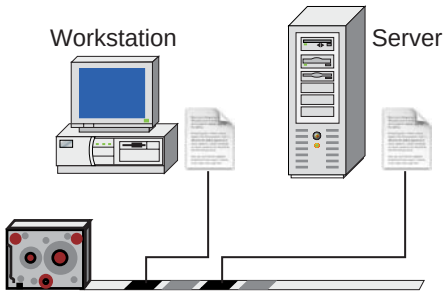


Figure 9-25. Backing up a hacked dll

You can see from the above that restoring using the day 4 snapshot of the server will restore the hacked file. So what you *don't* want to ever do after an attack is to set your backup software's preferences to restore *the most recent* version of the backup. First, you must find out when you got hit and restore the information that was backed up before that date. You won't have to worry about other restores to other computers—they'll use the undamaged file. Using snapshots, you never get a different file from one that originally resided on a given machine at that date and time. In the case of Retrospect, the program doesn't use a newer date to decide that the file is "better"—it's completely date-agnostic, treating older or newer dates as different, but never preferring one to the other.

Allow limited access to this information and plan for a separate backup

Once you've the restored data onto a working substitute server, if that server is one that you've had to double-up on, make sure that you keep your backup of this data separate from the rest of the server's normal data, so if you have to restore that server, you won't accidentally put these documents back on the substitute computer at a later date.

And since this is a temporary home for your data, treat it like a *good* friend's house: Don't invite all of your rowdy pals over to have at the data. Based on this server's new load, you might want to restrict access to those who need immediate access to the data, and have the others in the organization tackle something else on their busy schedules for the day.

Wipe the system

You'd be just plain old goofy to want to put the infected Server X back onto your network. *Wipe the system completely.* Reformat the hard drives and reinstall the system, either from your drive image or from the backup files themselves.

Don't waste your time with cleaning systems, virus removers, etc. Sterilize that baby and start from scratch. You won't have to worry about having missed something. And then don't bring the system back online until you've found how the hacker entered the system in the first place and you've found and applied the patches that close that open door the hacker crept in through.

Because you're probably rebuilding the system from an image, and then updating your image for new security patches, this is a great time to update the drive image master, as well.

Change your address

I know this sounds hokey, but you might want to change the address of the affected system, too. This makes it just a few minutes harder for the hacker to find it again, but at least you'll have those few minutes to spot the incoming attack while Mr. Creepo's rummaging around looking for the new address.

If your backup of the old server was "hard coded" to an IP address, make sure that you have the new address and you've tested your new address in the backup system.

Run another backup

Before you put the new system live on the network again, back it up once more. It's now spanking clean, and having one full backup in this state sure won't hurt your chances for future survival.

A SECURITY CHECKLIST FOR BRINGING A NEW SYSTEM ONLINE

1. Change the defaults for your device as much as you can. Ensure that the default user names are changed to something else that fits your naming strategy, as discussed earlier. Doubly ensure that all default passwords are changed to something other than their "boxed" version. Change the root directories of your web server's files. People attack what they know. If you've moved things, the hacker will be like a blind person navigating a stranger's house.

2. Build computer-based defense systems around the computer you're going to deploy. Does the device need a "personal" firewall? How will anti-virus software work with this computer? Have you installed *all* of the known patches for the system—you've checked them out at Cisco's CSEC site? Should this device be monitored by the organization's defense intelligence system?

3. Run a penetration scan using the tools you've run on the network or individual devices, on this one. If you've run an up-to-date vulnerability scan on the device and everything is clean, you're ready for a full backup and image creation.

4. Image the device—or, if you aren't into imaging your devices, get ready for failure in whatever fashion you so choose for your network. Have the information "locked and loaded," as they say in the military, so that when something hits, you're ready for the bombardment.

5. Plan for failure. If this device gets hit and you have to take it offline, how will you replace it? Will you replace it with a spare computer? Will it stay offline until you've wiped it and then restored it? Will this device be "doubled up" onto another device—if so, have you tested your theory to see if it'll work?

6. Test your recovery theory. If you don't test it in the absence of panic, when there *is* a panic, it's likely that you won't be ready and you'll flub the recovery effort. Always test when you have time to think through issues that come up at the last second. To test the theory, once everything is set up, bring the box down and then move the data over to whatever your plan suggests. Get it working, and time how long it takes. This will be useful if something really does go wrong: You'll be able to say (with confidence) that you'll be up and running again within the time-frame you already tested. Make sure you add

15 minutes to your estimate, so that when you finish early, everyone will realize your true genius.

7. Bring the system fully online and then run one more test hit against it. Did the Intrusion Detection System (IDS) pick up the hit? Did you get notified? If so, you're finally ready. If not, make sure that a hit to the system can get picked up by your IDS and that you get notified in whatever fashion you've set.

8. Then go get some Krispy Kremes and relax. You deserve it. You are ready. And send us an e-mail. We want to know how it all worked out.

SETTING UP AN E-MAIL DEFENSE SYSTEM

An **e-mail defense system** is a bit different than a hacker defense system, as you don't have to scan the entire network to find the point of failure. It's simple—it's your e-mail server and the content that comes in through it. E-mail resources have never been more vulnerable, and the incidence of hostile attacks continues to climb. More than 90 percent of all viruses can be attributed to e-mail, and who knows how many new viruses are unleashed each month? Here are some interesting numbers about SPAM and e-mail viruses:

> 650 – The percent of increase in spam during 2001. Source: SurfControl
>
> $2,000 – The amount that a Washington State small-claims court awarded Bennett Haselton, who sued spammers over four messages they sent him. Each spammer had to pay $500. Source: Spamcon Foundation (www.spamcon.org)
>
> 15 – The age of "suid," the Israeli kid who admitted writing the "Pentagone" virus, which attacked millions of Windows users through Outlook and ICQ. Masquerading as a screensaver, the program deleted anti-virus software and installed a Trojan horse that allowed remote control via Internet Relay Chat (IRC). Source: McAfee

Spam is getting so bad that some ISPs[7] have begun issuing ultimatums to corporate customers: Meet baseline standards or take your business elsewhere. Whether they're successful or not (most service providers have not yet created a formal list of security requirements), many have some kind of policy that attempts to dictate what companies can and can't do as customers, and the kinds of security systems that must be in place before they can purchase services. These service providers want to see IT managers install encryption and authentication products, firewalls that interact with intrusion detection software, dedicated servers, and VPN links to secure data. They also want IT shops to use tools such as anti-virus software, specified intrusion detection systems, and anti-spam content filtering[8]. In this sec-

7. Service providers that have such a policy include Ameritech, AT&T and CTC Communications, and national ISPs, EarthLink, Exodus Communications, and PSINet.

tion, we'll describe the things you need to know about anti-spam content filtering and anti-virus software for your e-mail server.

Simple mail server anti-virus defense

I followed a recent discussion in Google Groups among a bunch of friends who had encountered a new and interesting e-mail-borne virus. I've gotten their permission to give you details, but have taken their names and most of the rest of the conversation out of the loop. This is more or less what was said:

Friend 1 I tried to open an attachment today that was labeled !"#$. It wouldn't open, but later in the day, I got an e-mail saying that I had a virus. It's called sulfnbk.exe. I did a search and found it in the WINDOWS\COMMAND folder. So, if anyone gets that attachment in their mail (mine was from *Friend X*), DON'T OPEN IT!

Friend 2 Note to everybody: You might not get sulfnbk.exe as the attachment. The actual virus/worm is called Magistr and it sends out a variety of infected .exe files. cfgwz32.exe is another of its victims (I got an e-mail with this one from Friend X myself, but I didn't run the attachment). Note that these are *real* Windows files (sulfnbk is for managing long file names, for instance), and if you delete the infected versions, it would be a good idea to replace them from your original Windows installation disks. You don't have to worry about replacing anything if you just delete the e-mails and their attachments without running the attachments.

I got 2 e-mails this morning from *Friend 1's* address with a stars.exe and sulfbnk.exe. I deleted the 2 e-mails and didn't attempt opening the files. I also got an e-mail from *Friend X* yesterday with sulfbnk.exe too. Ugh! I hate viruses.

Friend 3 This was my first experience with a virus, and I've sure learned my lesson! I hardly ever open attachments. Unfortunately, with all the e-mails going back and forth about *Friend Y*, I thought the attachment "from *Friend X*" had something to do with him. So I naively clicked on it.

It turns out that this virus is smart enough (once activated) to enter a user's address book, get the list of names in the address book, grab *one* of the names in the address book, and set it as the outgoing e-mail address (whether it was that user's

8. "Security Strategies: Get serious about security—ISPs are demanding that IT shops get their security policies in order, and companies are complying," by R. Tadjer, *InternetWeek (*2001).

particular computer or not), and then send itself to all other users in the address book. In reality, it wasn't *Friend X* who had sent those two e-mails—it was another friend not even mentioned in this dialog. Ouch. I asked him directly if I could interview him for this book. What I learned was (as usual)…

- The Windows machines are at *great* risk, there's *some* risk to the Macintosh machines, and little risk to the Linux machines.
- Some of the mail servers and users in question had anti-virus software running on them, and some of those were completely up-to-date with the latest virus definitions.
- Each of the users whose computers had become infected indeed lost the Windows files that the virus attacked. There's no fix for this other than restoring the files from original CD, disk image, or backup application.

What are we to conclude? At minimum, you'll need anti-virus software on your mail server, such as Norton's AntiVirus for Microsoft Exchange for Windows, or the built-in McAfee Virex for 4D (Webstar) Mail. *This doesn't obviate the necessity of running anti-virus software on your desktops and other servers, as well.*

Norton AntiVirus for Microsoft Exchange secures your Microsoft Exchange environment against virus attack by monitoring all public folders and mailboxes on your Exchange servers, scanning the body and attachments of e-mail messages, including files in compressed and encoded formats. The Auto-Protect feature of Norton AntiVirus detects viruses in real-time, managing them according to your specifications. You can configure Norton AntiVirus to repair, quarantine, delete, or log detected viruses and send e-mail alerts to selected administrators and users to keep them informed of virus activity. You can use LiveUpdate to keep virus protection current. With LiveUpdate, Norton AntiVirus for Microsoft Exchange connects *automatically* (which is great, because you don't have to worry about keeping it up-to-date) to special Symantec sites and determines if virus definitions need updating. If updates are needed, the required files are downloaded and installed in the proper location.

4D Mail provides support for virus scanning through McAfee Virex. If the administrator enables this feature, 4D Mail passes all incoming mail for local addresses to the virus scanner. The server immediately refuses to accept a message that Virex flags as infected, sending an error to the sending SMTP server explaining the failure. Administrators can choose to keep a copy of rejected messages in the Quarantine folder. Administrators are not notified, since the message is automatically

rejected (and the sender is aware of the problem), but they can review the Quarantine folder for troubleshooting purposes. Just like Norton's, McAfee Virex's virus definitions can be automatically updated to ensure your safety from any new virus attacks.

In essence, both programs work the same way. The application intercepts all incoming e-mail and then checks it and all its attachments for viruses. If a virus is detected, the message is rejected, returning an error to the sender or sending SMTP server and then putting the offending message into a quarantine directory on the mail server for troubleshooting or legal purposes. If, on the other hand, the message has no viruses in it or attached to it, it's forwarded to the mail server's normal operations, and all is fine in the world.

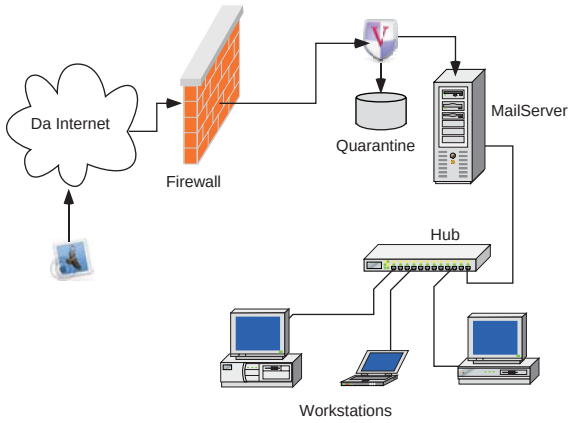


Figure 9-26. Direct Server-based protection

The advantage of adding anti-virus capabilities to your organization's mail server is that the mail server–based anti-virus program doesn't care whether the sender or receiver's computer is a Mac, Windows, or Linux computer. At the mail server level, a known virus is a known virus. This is not always the case on the end user's computer. Some programs check for viruses for only the particular workstation OS that the program is running on—some Mac software checks for Mac-only viruses; Windows for Windows-only, etc., which means that a Mac user could pass on a virus to a Windows user, and vice versa.

Perimeter protection for your e-mail server

To be honest with you, putting e-mail anti-virus protection on the e-mail server itself is very much like having a guard dog that lets thieves into the building but doesn't let them leave. Call me silly, but I'd much rather keep the bad guys out of the building completely than let them into a quarantined room of the building. By moving your protection to a perimeter service, you can set up your anti-spam mechanism much more effectively than if you ran it from your own servers.

In establishing a well-protected e-mail system, you also have to guard against server misuse by hackers and spammers. Spammers are not only bothersome, but the spam that comes through the mail server hogs mail server processes as well as network bandwidth.

One type of spam/hacking, the **Directory Harvest Attack (DHA),** is designed to obtain valid e-mail addresses in the organization to send members of the organization more spam. In a DHA, spammers attempt to deliver messages to multiple addresses, such as johndoe@yourco.com, jdoe@yourco.com, and john@yourco.com. Addresses that aren't rejected by the receiving mail server are determined valid, compiled into lists and repeatedly sold to other spammers. A successful DHA can net a spammer thousands of corporate e-mail addresses in just a few minutes. Users whose addresses are harvested quickly begin receiving an ever-growing amount of junk e-mail, as spammers resell and exchange lists of known valid addresses[9].

To further complicate issues, e-mail servers overloaded by DHA traffic may be unable to accept legitimate incoming traffic, thus appearing unreachable to customers and business partners who are attempting to send real business-related messages. Sure, it's possible to identify DHA hackers by manually examining e-mail server logs. However, attackers aren't stupid—just lazy and unprincipled—they frequently switch IP addresses to stay under the detection radar. And by the time a DHA hacker is identified through log file analysis, the horse already done run outa the barn: Your organization's addresses are in the spammer's database, and they've moved on to the next mark.

9. "Directory Harvest Attacks Pose Significant Security Threat to Corporate E-mail Systems," Postini, http://www.postini.com/press/pr/pr081302.html (2002).

The best thing to do? Identify these attacks in real-time, and on a server other than yours, so that your organization can proactively protect the proprietary personnel list stored on your e-mail systems using a pertinacious approach. (My apologies: I adore alliterative excess!) Oh, by the way, this is a good way to get rid of general spam, as well.

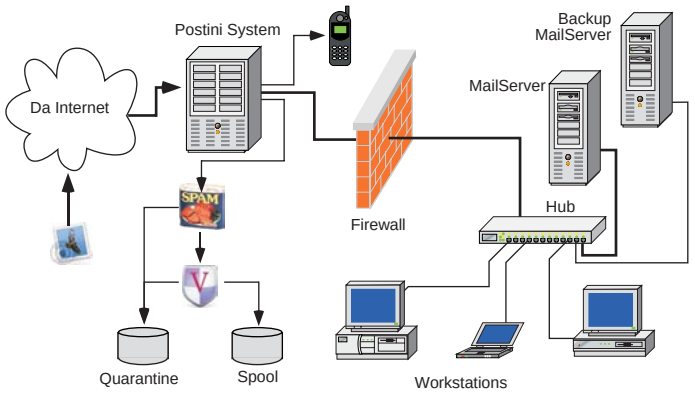


Figure 9-27. E-mail system perimeter protection by Postini

The Postini service is the best we've seen. They set up their server so that it's a "bump in the road" of the flow of e-mail traffic—*before* the e-mail hits your server (they call it an inline SMTP utility service). Figure 9-27. shows the Postini system setup, in which all e-mail (good, bad, and indifferent) hits the Postini service first, filtering out unwanted spam as well as viruses and DHA attacks, before the mail you want is then forwarded through to your company. The small phone on the top of the diagram above indicates that Postini can even contact you via wireless cell phone, blueberry, blackberry, raspberry, whateverberry, to tell you if your e-mail service can't be reached and is therefore "down" for some reason. The thick line in the diagram follows the route that "good" mail takes through the system as it passes through the Content Manager and then Content Filter of the Postini perimeter protection system.

As you can see in Figure 9-27. , I've also placed a *backup* e-mail server. When you move your e-mail gathering and filtering system outside your network, you can also use the Postini service as a mail spool, so that if your mail server dies, you can quickly bring a secondary server online and have Postini redirect e-mail to *it*, ensuring that you keep your e-mail downtime to a minimum. A very nice service to have at the ready, it makes e-mail backup and recovery that much easier.

Setting up the Postini system

As you can see from the form below that shows the organizational information, and Figure 9-29. on page 231 on the following page, the Postini setup is quite detailed. New service, junk-mail notification, virus suspicion, and virus alert e-mails for the organization are set up (so that the user gets a custom message that fits the organization's personality). At the same time, an initial list of "approved" junk-mail senders and blocked senders can be added. Once that's done, it's time to configure the attack-blocking mechanisms (which is the real reason you've chosen this service).

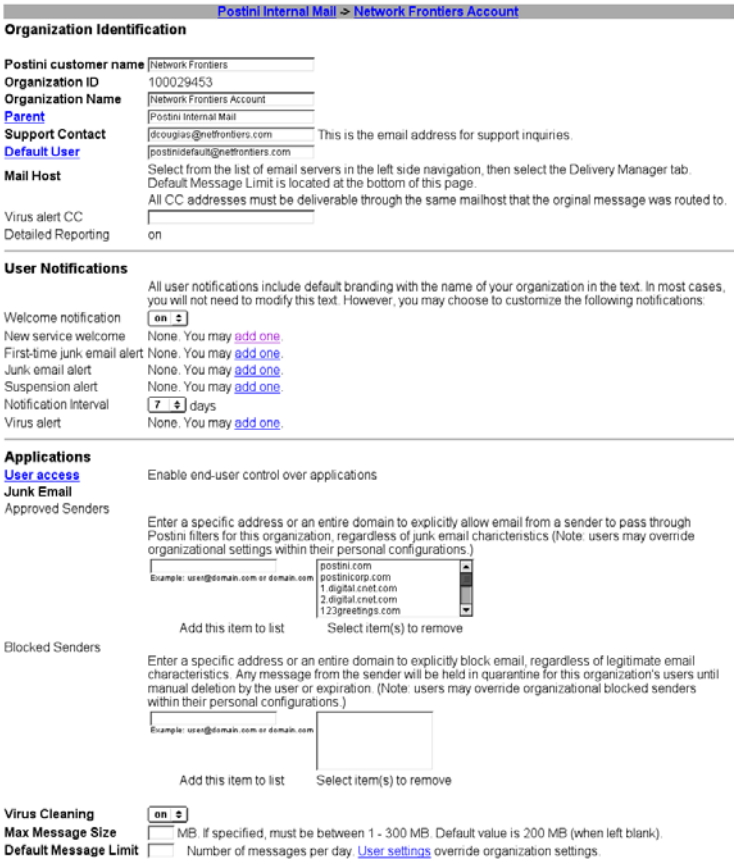


Figure 9-28. Organizational management setup

To identify attacks, Postini's system performs statistical and content analysis. They wouldn't tell us the syntax or threshold values of the rules definitions, because we're writing a book and they want to keep the spammers guessing.

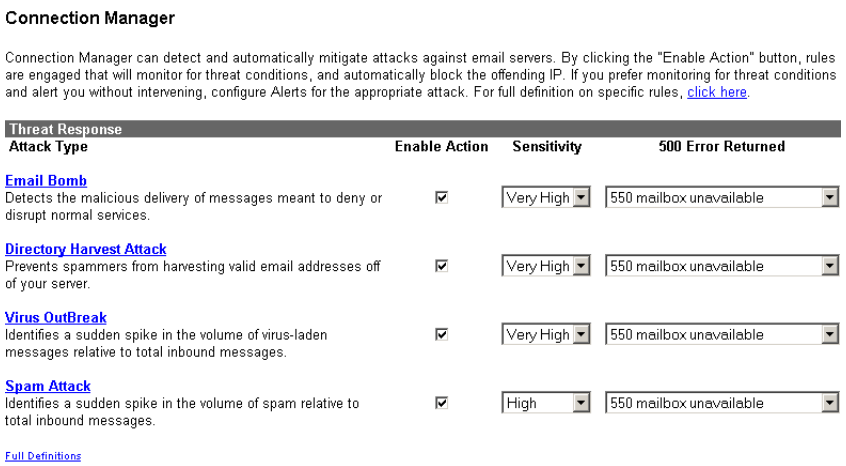


Figure 9-29. Blocking attacks through Postini

A Very High sensitivity triggers a particular rule when the standard deviation from normal traffic characteristics is low. Conversely, a Very Low sensitivity requires a more substantial change in the characteristics of the traffic to trigger a rule. Normal is the recommended setting, recognizing the majority of attacks. The Normal setting is calibrated on the ongoing analysis of threat conditions across the nearly 20 million messages that Postini processes each day. Here are their targets:

E-mail bomb

E-mail bombs are denial of service attacks in which unusually large messages or an unusually high volume of messages are sent repeatedly. Postini's Connection Manager identifies spikes in message volume that violate standard variance in message traffic. Similar messages sent repeatedly, messages of particular size characteristics, and a high ratio of suspect to valid e-mail are classified as e-mail bombs.

Directory harvest attack

A **directory harvest attack**, also known as an **e-mail harvest attack**, is a series of delivery attempts that result in 550 errors. Your e-mail server happily responds to each request, issuing potentially thousands of 550 errors. When the spammer lucks into a valid address, a spam may be delivered, and the address is logged as valid. Sensitivity allows a variance in the ratio of valid to invalid messages per session or per source IP. Very Low sensitivity doesn't block the IP if a single valid

address is in the session. Very High sensitivity ranges up to a ratio of 1:5 valid addresses.

Virus outbreak

To identify this type of attack, Postini looks at a number of criteria. A **virus outbreak** is tracked by monitoring both the ratio of infected messages to valid e-mail and the total volume of infected messages from the network connection during a specific interval. If the ratio changes in a statistically significant manner, the network connection will be blocked for several hours.

Spam attack

To identify this type of attack, Postini examines a number of criteria. A **spam attack** is tracked by monitoring both the ratio of spam to valid e-mail, as well as the total volume of spam from the network connection during a specific interval. If the ratio changes in a statistically significant manner, the network connection will be blocked for several hours.

Allowing individual user preferences for filters

Once you've set up your organizational information, you need to set up your user accounts. These can be automatically gleaned from Postini's surveillance of who is getting e-mail, or manually entered into the system through uploading a spreadsheet. Either way is effective.

One of the great things about the Postini system is that each user can set individual thresholds for five different categories of e-mail: bulk, sexually explicit, get-rich-quick schemes, racially insensitive, and special offers.

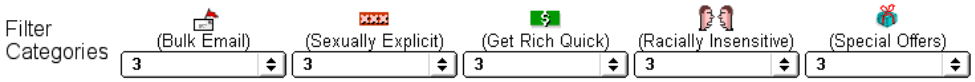


Figure 9-30. Filter categories

Reporting

After it's set up, the system does some nice reporting—one of the parts of the system that I really, *really* like, because it gives me visual evidence of what Postini does for me every day. Postini will report on everything from normal mail processes through amount of spool activity (they can spool your mail for you so that if your

server goes down, nobody in the outside world will know), and the number of attacks blocked on a daily or weekly basis.

The daily address information shows you who got what mail. This is a great place to notice the fact that folks who aren't even at your organization anymore (like our editor, Cass) are still getting mail (four years post-departure!). It's also a pretty decent indicator of where your staff members are spending their time. One of our clients' staff members usually gets a ratio of 85 spam and virus-laden messages to every single good message. A quick check over his shoulder proves that he spends *a lot of time* on the Internet visiting all sorts of various and sundry sites.

Traffic by Domain - Daily for 09-11-2002
For Domain: netfrontiers.com

Recipient	Messages	Bytes	Account Messages	Forwarded	% of Msgs	% of Bytes	Quarantined	% of Msgs	% of Bytes
lheiberger@netfrontiers.com	55	444,582	0	55	--	--	0	--	--
lynn_heiberger@netfrontiers.com	15	59,370	0	15	--	--	0	--	--
heiberger@netfrontiers.com	10	59,739	0	10	--	--	0	--	--
dcougias@netfrontiers.com	10	30,400	10	9	90.0	82.6	1	10.0	17.4
cass_kovel@netfrontiers.com	4	15,715	0	4	--	--	0	--	--
dorian_cougias@netfrontiers.com	4	10,883	0	4	--	--	0	--	--
kkoop@netfrontiers.com	2	987,030	0	2	--	--	0	--	--
dcouglas@netfrontiers.com	1	18,324	0	1	--	--	0	--	--
Grand Total	**101**	**1,626,043**	**10**	**100**	**90.0**	**82.6**	**1**	**10.0**	**17.4**

Figure 9-31. Daily e-mail traffic flow

You can also use this information to see how people are misaddressing mail to your users. Once you know how folks on the outside are misaddressing your mail, you can set up aliases of the misspellings and have the mail forwarded to the right person in the organization.

For those of us who like graph views of the same information, Postini provides graph views, as well, as shown in Figure 9-32. on page 234. We've highlighted two lines in the graph for you. The gray bars represent the total number of messages that have hit the system. The top line shows directory harvest attacks, e-mail bombs, etc. The line slightly below it shows the number of spam blocks that are currently running. Over to the right, we've highlighted the event tracker list that shows when the directory harvest attacks hit. You can use that list to back-track to those domains that are continuously attacking you, and then add them to your black-hole list of mail that never goes through.

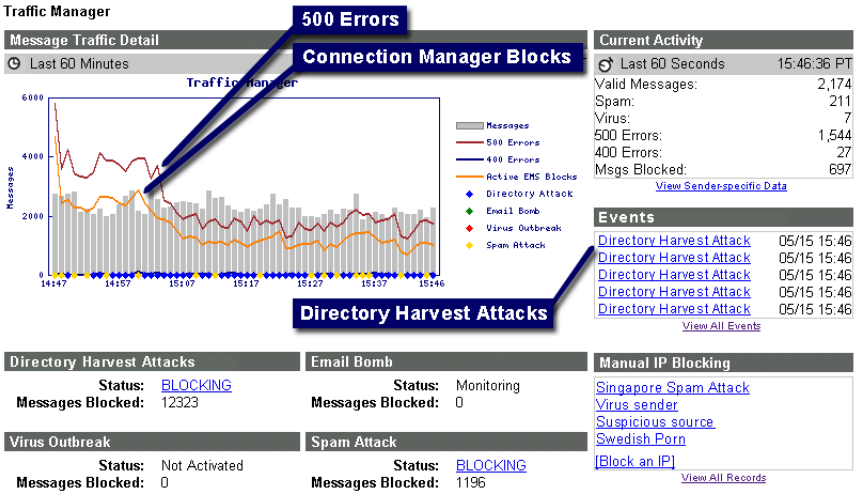


Figure 9-32. Traffic Manager reports

DNS SECURITY

During its most recent DNS health survey[10], Men & Mice found that 13 percent of the Fortune 1000 companies were running with *known* security vulnerabilities that could lead to denial-of-service attacks in their BIND (Berkeley Internet Name Domain) DNS software. Exploiting this vulnerability would cause the BIND server to shut down and leave the Internet blind to the location of the organization's client-facing servers.

An attacker can cause shutdown by sending a specific DNS packet designed to trigger an internal consistency check. However, this vulnerability doesn't allow an attacker to execute arbitrary code or write data to arbitrary locations in memory. The internal consistency check that triggers the shutdown occurs when the `rdataset` parameter to the `dns_message_findtype()` function in `message.c` is not `NULL` as expected. The condition causes the code to assert an error message and call `abort()` to shut down the BIND server[11]. It's also possible to accidentally trigger this vulnerability using common queries found in routine operations, especially queries originating from SMTP servers.

Exploitation of this vulnerability causes the BIND server to abort and shut down. As a result, the BIND server won't be available unless it is restarted.

And that's only the most current problem

Security holes in BIND have been found since hackers have found BIND. One of the interesting things that we've found regarding DNS security is that many DNS managers don't take security as serious as they should. A case in point is the *last* known security bug that was announced back in 2001.

After a major press run about the security bug in 2001 that affected roughly 17 percent of all companies, only about 5 percent of those affected fixed the bug![12]

10. "Domain Health Survey for .COM," http://www.menandmice.com/6000/61_recent_survey.html, Men & Mice (August 2002).

11. http://www.kb.cert.org/vuls/id/739123

The majority of news coverage of the bug list (there were four major bugs) focused on only the most serious of them[13]. Why so little attention to DNS security? It just isn't sexy, and there's not a lot anyone can do about it. It's not a big splash when it gets hit, and it isn't huge news unless it's a huge hacker attack. And the way to fix the attack is simply to upgrade the DNS server to a version that is patched for the attack: no packet analyzer teams swooping in, no intrusion detection system's bells and whistles blaring; simply a CERT warning that needs to be heeded and a patch that must be applied.

What can happen?

Hackers utilize BIND vulnerabilities to gain root access to the host or to turn the host into a launching platform for DOS attacks. An improper or insufficiently robust BIND configuration can also "leak" information about the hosts and addressing within the intranet. Miscreants can also take advantage of an insecure BIND configuration and poison the cache, thus permitting host impersonation and redirecting legitimate traffic to black holes or malicious hosts. Let's look at what can happen.

DNS spoofing and triggered cache poisoning

Triggered cache poisoning happens when a hacker induces a name server, either directly or indirectly, to query another name server under the hacker's control and then cache the bogus records. In July 1997, Eugene Kashpureff used an indirect triggered cache poisoning attack against the InterNIC's website. By poisoning the InterNIC's name servers' cache, he was able to spoof his alternic.net websites onto InterNIC's DNS system.

12. "Domain name system security still lax," http://www.cnn.com/2001/TECH/internet/03/02/lax.on.DNS.idg/index.html, CNN.com (2001).

13. "Fix for DNS software hole released," http://www.infoworld.com/articles/hn/xml/01/01/29/010129hnhole.xml?0129mnpm, *InfoWorld* (2001).

"Sleep tight, don't let the BIND bugs bite," http://www.thestandard.com/article/display/0,1151,21785,00.html, *The Standard* (2001).

"Software flaw may mean more Web outages," http://news.cnet.com/news/0-1003-201-4638816-0.html?tag=mn_hd, CNET (2001).

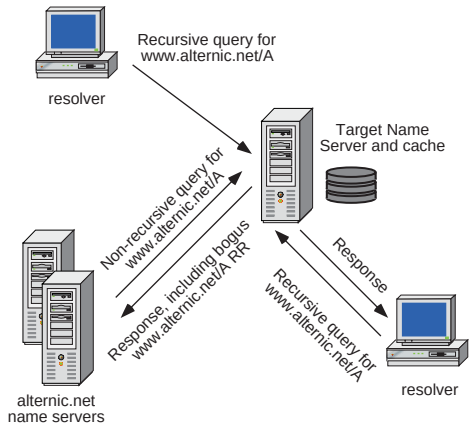


Figure 9-33. Kashpureff's attack

When spoofing, attackers can use the recursive mechanism described in the recursive call commands within legitimate DNS procedures by predicting the request that a DNS server will send out, and then replying with false information before the real reply arrives, as shown in the diagram above (Figure 9-33.). Each DNS packet has an associated 16-bit ID number that DNS servers use to determine the identity of the original query. In the case of BIND, the prevalent DNS server software, this number increases by 1 for each query, making the request easier to predict[14]. By providing false host name and mapping information, the attacker can misdirect name-resolution mapping while exposing network data to the threat of corruption. DNS involves a high trust relationship between client and server, and it is this trust that makes DNS vulnerable to spoofing.

After recursive querying, a second DNS vulnerability lies with DNS caching, which can be exploited through triggered cache poisoning. DNS servers cache all local zone files (hints file, and information for all zones the DNS server authorizes) and the results of all recursive queries they've performed since their last startup, to save time should they receive a similar query again. The length of time that recursive query results are held in the DNS cache (TTL–time to live) is configurable. The default is for a RR (resource record) to inherit the TTL of the zone (name domain) it's in.

14. This has been fixed in the later versions of BIND, in which DNS packets are assigned random numbers.

DNS cache poisoning involves sending a DNS server incorrect mapping information with a high TTL. The next time the DNS server is queried, it replies with the incorrect information. DNS cache poisoning occurs when malicious or misleading data received from a remote name server is inadvertently saved (cached) by another name server. This "bad" data is then made available to programs that request the cached data through the client interface. It's possible to limit exposure to this DNS cache poisoning attack by reducing the time that information is stored in the cache (the TTL), but this makes a negative impact on the server's performance.

Denial-of-Service attacks

Simple **denial-of-service attacks** can come in two forms. The first exploits DNS implementation flaws by responding to name server responses with parroted responses.

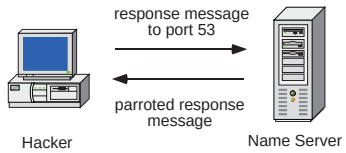


Figure 9-34. DNS denial-of-service parrot response attack

The second method overwhelms the server with zone name transfer requests. A **zone (name domain) transfer** is the transfer of the DNS database to a secondary server. It allows name servers that are authoritative for the same domain to stay in sync with each other. DNS servers should be configured to allow zone transfers between primary and secondary DNS servers only, because the information in a zone transfer, such as the IP addresses of important hosts, is very attractive to a hacker. Zone transfer attempts are often the first indication that a network is being probed.

Takeovers

Another attack involves breaking into the target network's DNS server per se; for example, the buffer overflow vulnerabilities of earlier BIND versions allowed root access to attackers. Once attackers gain control of the underlying DNS platform, they have control of the network environment.

Security measures you need to take

Let's walk through setting up DNS security for an organization laid out like the one in the following diagram that has two offices, with the main office in Ogunquit, Maine (with two subnets), and the branch office in Pismo Beach, California (with a single subnet).

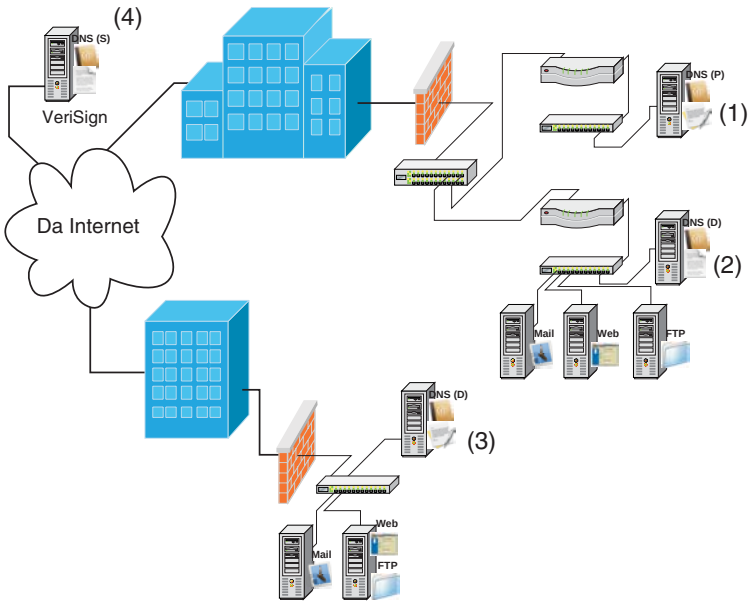


Figure 9-35. Two office organization

In this diagram, we show four name servers:

1. This is the Primary, or WhoIs listed authoritative server for the myco.com network. This server is a Unix box running BIND.

2. This is a Windows 2000 Server running Active Directory and Windows DNS. It has a delegated zone ogun.myco.com.

3. This is a Windows 2000 Server running Active Directory and Windows DNS. It has a delegated zone pismo.myco.com.

4. The organization has chosen to partially outsource DNS slave services to a third party (in this case,VeriSign) to protect the organization in case of disaster at either or both of their locations.

Setting up the firewall

First, restrict the number of services running on the name server itself and then restrict access to the servers at your border router and bastion host. Eliminating unnecessary network services reduces the potential holes in your server's security. Limiting protocol delivery sets up strong anti-spoofing measures.

From	Source Port	To	Dest. Port	Protocol	Purpose
Any	Any	Name Server (1)	53	UDP or TCP	Queries from the Internet
Name Server (1)	53	Any	Any	UDP or TCP	Name Server responses
Name Server (1)	Any	Any	53	UDP or TCP	Queries from your name server
Any	53	Name Server (1)	Any	UDP or TCP	Responses from your name server

Table 9-3. Firewall setup for name servers

Set this up for all firewalls that have name servers behind them so that the "public" queries are delivered only to your primary name server and not to any of the "local" delegated name servers, such as name servers 2 and 3.

Setting up an offsite name server

Because your building(s) can fall down, you need to find an offsite slave name server for one or more of your zones. Far too many organizations have a single point of failure (their own network) in their DNS infrastructure. Get into the habit of moving at least one of the authoritative name servers offsite in case your primary site has problems.

You can ask your local ISP whether or not slave name service is included (or at least available) in their package. Companies such as Nominum[15], SecondaryDNS[16], and VeriSign[17] offer secondary slave name server services. VeriSign's High-Availability service provides companies with robust DNS support for Inter-

15. http://www.nominum.com/.

16. http://www.secondarydns.com/

net systems (like websites and e-mail) through VeriSign's substantial investment in DNS infrastructure. VeriSign name servers are located around the globe and currently support the .com, .net, and .org domains. As such, they respond without interruption to more than 6 billion queries per day. Companies can now be sure their websites, e-mail, and other online systems are supported by the most robust and reliable DNS infrastructure available.

Or you can ask your local VAR who supplies your computer equipment for a slave name server service. Partnering with a local VAR whom you already trust is a really, *really* good idea.

TSIG

With BIND 8.2 and later name servers, you can use **transaction signatures** (**TSIG**) to cryptographically authenticate and verify zone data. TSIG uses shared secret codes and a one-way hash function to authenticate DNS messages such as updates.For this to work properly, you must configure a key on your primary master name server (as well as all slave name servers) and instruct them to use the key to sign communication with each other[18]. In essence, this means that you have to set the primary master's named. conf as well as synchronizing the time between all name servers involved.

Once TSIG is configured correctly, the name server (or client attempting to update its own record) adds a TSIG record to the additional data section of a DNS message, thus "signing" the DNS message to indicate that it isn't forged by some assailant to the system.

Set this up for all of your name servers running BIND. Windows 2000 DNS servers aren't TSIG compliant, so they should be relegated to specific zone services such as dynamic user zone services.

17. http://www.verisign-grs.com/mdns/ha/

18. This is covered thoroughly in pages 145–147 of C. Liu's "Defining a TSIG Key," *DNS and BIND Cookbook* (O'Reilly, 2002).

Limiting queries

Next, make sure you limit the devices from which your name servers accept queries. Accepting recursive queries from the Internet makes your name servers vulnerable to spoofing attacks, wherein your server is forced to query their server, which then sends back bogus data that could end up in your cache. Disabling recursion puts your name servers into a passive mode, telling them never to send queries on behalf of other name servers or resolvers. To deal with this, restrict recursion as well as the addresses the name server responds to when receiving queries. A non-recursive name server is very difficult to spoof.

A caching-only name server should accept queries *only* from the IP addresses of resolvers it serves. You can't disable recursion on a name server if any legitimate resolvers query it, or if other name servers use it as a forwarder.

An authoritative-only name server must accept queries from any IP address, but shouldn't accept any recursive queries.

And then you need to black-hole (deny) any private, experimental, or multicast networks. Rob Thomas maintains an outstanding list of those sites[19].

In our example (Figure 9-35. on page 239), you'd set up your Primary DNS server (1) to accept queries from any IP address, not *not* to accept recursive queries.

Restricting zone transfers

You must restrict zone transfers only to slave name servers and other authorized software. This prevents others from taxing your name server's resources, and prevents hackers from listing the contents of your zones to identify such targets as mail servers and other name servers, or even gaining "host demographic" information such as the number and models of your computers and their device names.

- In our example, you should restrict zone transfers from name server 1 to *only* name servers 2, 3, and 4.

19. "Secure BIND Template," Rob Thomas, http://www.cymru.com/Documents/secure-bind-template.html (2002).

- Name servers 2 and 3 should be able to copy information from name server 1, giving them a local copy of the master zone data.
- Name server 4 *should not* be allowed to transfer any of these zone's data files at all, since it's a slave server of 1 through 4. It's easy to forget that you also need to restrict zone transfers from your slave name servers, but it's just as simple to transfer a zone from a slave as it is from the primary master name server.

Setting the rules for Windows 2000 and BIND to play nicely

Next, you need to restrict dynamic updates, because they have near-complete control of the name server. To do this, you must set up an organizational policy and DNS structure for end-user machines such as Windows 2000 devices.

By default, each Windows 2000 client tries to update its own name-to-address mapping on the DNS server. If your organization has a small, trusted group of people and you want to permit these updates, simply allow dynamic updates to your forward-mapping zone (the zone that contains your hosts' A records) from any IP address that a Windows 2000 client might have in your network address range. A small, trusted group reduces the chance that some malcontent will meddle with your DNS system. But since this kind of meddling is an easy task, we don't suggest this method for most organizations for that simple reason.

The better approach is to configure the DHCP server to handle updating the client's name-to-address and address-to-name mappings on the DNS server. Since all dynamic updates come from the DHCP server's address, you can allow *that one* address (the DHCP server) to create dynamic updates in your DNS system. However, pay attention to this caveat: First, set an organizational policy that defines which zones can be updated dynamically and which cannot.

The diagram immediately to the right depicts the DNS hierarchy and pinpoints where we believe that you should allow dynamic updating of DNS records.

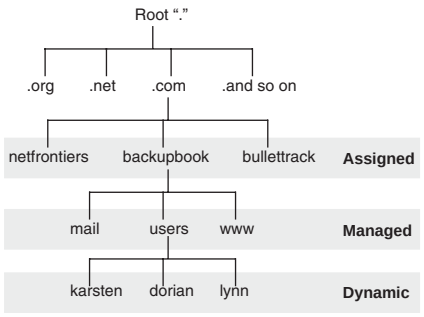


Figure 9-36. Dynamic range of DNS entries

Now that you've set the rules for who gets to be "king of the naming pile," it's time to build the sandbox that the Windows

devices can play in. Unfortunately, the DHCP server simply accepts the information from a client when it specifies its fully qualified domain name. In other words, if JoeBo considers it a riot to name his Windows 2000 client with the domain name www.backupbook.com, the DHCP server will delete any conflicting address record at that domain name and add a record pointing www.backupbook.com to that Windows device's IP address—and that ain't good. To construct this sandbox, you create your own domain zone for the Windows devices themselves. By default, Windows 2000 assumes that the domain name of your device's forward-mapping zone is the same as the name of the Windows 2000 domain they belong to. However, Windows 2000 also allows you to specify the domain name of the zone independently.

Right-click on **My Computer** and choose **Properties**. Then navigate to **Network Identification** > **Properties** > **More…** to bring up the dialog below.

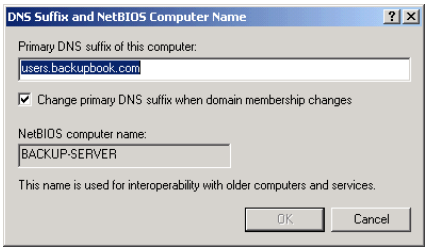


Figure 9-37. More… info for setting up a domain zone

Notice that we added the "users" label to the backupbook.com domain. By adding this new zone, the computer's address will be backup-server.users.backupbook.com. This new subzone, "users", is just for the Windows 2000 devices on the network.

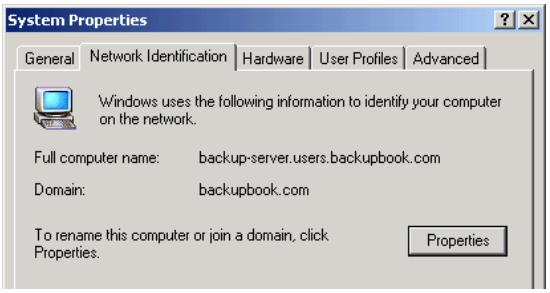


Figure 9-38. Windows client name in its own user zone

Thus, we'd configure the organization's Windows 2000 DNS/Active Directory servers (servers 2 and 3) to accept dynamic updates to that zone from only the DHCP server that's giving out their addresses, effectively building them their own little sandbox to play in. This can either be done directly through the DNS MMC in the Windows 2000 DNS server, by right-clicking on the properties of the zone and setting the **Allow dynamic updates** pop-up to **Yes**; or through Men & Mice's QuickDNS for Active Directory software.

Once you've done this, of course, since they are progeny of Mr. Bill, the Windows 2000 devices will more than likely trash each other's names, address records, etc. But at least they're limited to their own sandbox with their own king of the hill.

Run named as a user instead of root

To ensure that a vulnerability in BIND doesn't give a hacker root access to your hosts, you need to run *named* as a user instead of root on your BIND name servers (in our diagram, that's name server 1)[20]. Ensure that *named* can read your named.conf and zone data files, and if you use dynamic updates for your user zones, can write those zone data files. You'll also need to ensure that the user can write to *named*'s PID file. To make doubly sure that even if you *do* get hacked, your hacker doesn't get far, you'll want to run your BIND name server in a chroot() jail so that the hacker has very limited access to the host's file system[21].

Aggregate your logs to your IDS system & test your name servers

Use log file monitoring tools like NetIQ's Security Manager to alert your staff when hack attempts happen. Then sign up for services like Men & Mice's DomainHealth service[22], which monitors the health of your name servers.

---

20. "Running a Name Server as a user other than Root," *DNS & BIND Cookbook*, by C. Liu, pp. 144–145 (O'Reilly, 2002).

21. "Running a Name Server in a chroot() Jail," *DNS & BIND Cookbook,* by C. Liu, pp. 143–144 (O'Reilly, 2002).

22. http://www.menandmice.com

CHAPTER APPENDIX

How does a hacker hack? What does he look for?

Holes. Like well-trained infantrymen on the front lines, they've learned to scout, scan, and detail their prey (read *you*). What hackers look for is an open door[23]. According to a "white hat" hacker[24] who sports the grandiose alias of Epic, "90 percent of all attacks stem from poor configuration and administrators who do not consistently update the software they use." The same article quoted a hacker by the nickname Hackah Jak, who stated, "I can in minutes code a scanner to scan the Internet for two-year-old known vulnerabilities… I've hit a lot of workstations this way and then worked my way through the network to the server." In addition to making simple configuration mistakes, most administrators don't keep up with the updates and patches released by software vendors. Doors open when administrators don't update services running on the system, or set up permissions and software settings the wrong way on the web server.

Let's walk through this so that you know what might happen to you and how it might happen.

The scouting process

Scouting, **profiling**, and **footprinting** are all the same terminology for one process: Gathering basic information about a potential target. When a hacker scouts a site, he's looking for infrastructure information, knowledge of the network, and the physical environment. Let's take a quick look at what a hacker can find out about you just from common tools on the Internet.

---

23. "Hackers Find Open Doors," by Dan Verton, *Computerworld*, 7/22/2002.

24. The terms "white hat" hackers and "black hat" hackers recall old cowboy movies in which the good guys wore white hats, and the bad guys wore —you guessed it—black. White hats hack into companies on the company's behalf or just for research, but black hats do this maliciously. Unconscious racism—or archetypal imagery? You decide.

Whois.net The Whois database maintained on all domains allows you to get information about the owner of a domain name and its DNS servers. The default Whois server is whois.net. When searching for Network Frontiers information, we discovered our address, one of our staff's *real* e-mail addresses (instead of an alias such as hostmaster), and direct phone lines to our support staff (instead of general numbers).

```
Registrant:
Network Frontiers, Inc. (NETFRONTIERS-DOM)
  155 Bovet Road, #101
  San Mateo, CA 94402

  Domain Name: NETFRONTIERS.COM

  Administrative Contact:
    Gila, Joy  (NUCDRNKGGI)jgila@INTERPUBLIC.COM
    The Interpublic Group of Companies
    676 St. Clair St.
    Chicago, IL  60611
    US
    312.425.6926 312.425.6924
  Technical Contact:
    Interpublic Hostmaster  (NISGZUDGXO)hostmaster@INTERPUBLIC.COM
    The Interpublic Group of Companies, Inc.
    676 St. Clair St.
    Chicago, IL 60611
    US
    312.425.6926

  Record expires on 28-Feb-2003.
  Record created on 27-Feb-1995.
  Database last updated on 3-Aug-2002 09:55:30 EDT.

  Domain servers in listed order:

  NS1.TRUENORTH.COM          199.221.98.5
   NS2.TRUENORTH.COM         204.149.81.10
```

Figure 9-39. Whois registration info

However, around 50 percent of all hackers are *insiders*[25], so much of the work is already done. They know where the data center is, they know where the server room is, they have an up-to-date user list, and they know the address to the firewall. Scouting the target point isn't hard, nor is it time consuming. To start the process, the outside hacker is looking for a single IP address—which can be found easily enough at the www.netcraft.com website.

25. According to the 2001 Computer Security Institute/FBI Computer Crime and Security Survey. Before 2001, this was as high as 80 percent, so it's dropped significantly.

Netcraft Netcraft is supposedly a website "uptime" monitoring service that checks pretty much any site on the Internet that it can (we never asked them to check ours). According to their own website, "We report a site's operating system, web server, and netblock owner together with, if available, a graphical view of the time since last reboot for each of the computers serving the site.[26]" However, while this might be useful to some, to a hacker, it's a candy store. Just a quick trip to Netcraft can determine the make and model of a server for a planned attack. A quick check at Netcraft (by typing in www.netfrontiers.com) gives me *much* more information about the server itself:

OS	Server	IP address
MacOSX	Apache/1.3.23 (Darwin) DAV/1.0.2	64.175.139.131

Table 9-4. NetCraft info

Within three minutes of search, we now know that Network Frontiers is located in Los Altos, is running Apache on a Mac OS X box, and the server's address is 64.175.139.131. The hacker can now fine-tune the method to our specific vulnerabilities. Great. Just wonderful.

The scanning process

Hackers scan the network to create a list of network devices that are active and that have services that are potentially exploitable. Again, this is a three-minute process. It starts with matching names to addresses and then scanning each named entity on the network for service ports.

Name scan Name-scanning software uses a starting and ending TCP/IP address, and then performs a DNS query to find out what you've named your computers. Hackers look for things like "www" for web server, "intranet" for intranet server, "pop, smtp, mail" for mail server, and so on.

IP	Name
64.175.139.131	www.netfrontiers.com
64.175.139.132	Mail.netfrontiers.com
64.175.139.148	Surveys.netfrontiers.com

Table 9-5. Name Scan info

26. http://uptime.netcraft.com/up/graph/

The above list is just a small sampling of the information that can be found. So, now the hacker has a *list* of computers that might be running something fun. It's time to find out what kind of services and running ports are active on the computers in the list above.

Port scan

Once the hacker knows what type of computer he's dealing with, he looks for each of the software services—called **ports** in the IP world—running on each individual computer. Here's a list of network services running on of *one* of the computers that was in the original list:

Port	Service Running
25	smtp - Simple Mail Transfer
53	domain - Domain Name Server
80	www-http - World Wide Web HTTP
110	pop3 - Post Office Protocol - Version 3
143	imap2 - Interim Mail Access Protocol v2
389	ldap - Lightweight Directory Acess Protocol
497	retrospect - Retrospect Backup software
548	AppleShare IP Server

Table 9-6. Port Scan info

Port scanning software like **7thSphere** or **nmap** then begin to fingerprint devices based on how they register their services, and provide the same type of information about the operating system software as Netcraft provides: Identifying a Mac OS X box, Windows 2000 box, Sun Solaris box, etc. This gives hackers the basic information about the network, its devices, and the software available to target on each of those devices.

Now it's time for the hacker to go to work and create a detailed exploitable plan of ways to hit each of these computers and software processes in their vulnerable spots. They focus on four "biggie categories" of network service vulnerabilities: CGI vulnerabilities, unshielded directories, Trojan horses, and service protocols that are left wide open.

The detailing process

The **detailing process**, or, as some call it, the **enumeration** process, involves accessing the different software processes running on each machine that might allow a way into documents, databases, file systems, or the web server per se.

A greater number of applications do this type of thing than you think. The information below has been compiled by Jean-Baptiste Hervet of Lagoon Software, makers of the great Macintosh analysis program called MacAnalysis that we mentioned earlier, the folks at Foundstone who make FoundScan, the gang at Security, and the team at Common Vulnerabilities and Exposures (CVE) who provide a list of standardized names for vulnerabilities and other information security exposures[27]. In reality, there are around 1,000–1,500 different tests that should be run *against each machine* to thoroughly scan.

CGI vulnerabilities One of the first scans you undertake should be the available CGIs on your server, warning you about possible security holes if they're exploited. The CGI Syntax should then try to exploit every security hole found in "services/protocols holes" and "CGI vulnerability". If your software displays "Vulnerability Syntax Found: /scripts/test.idq?/../../ ," someone could exploit the hole in their very own browser by typing "http://www.yourserver.com//scripts/test.idq?/../../".

Unshielded directories Many types of computers ship with certain folders pre-shared so that anybody in the world can gain access to them. It's common to see directories whose content is not viewable by everybody, like private folders, scripts, systems, etc (ex : /script, /private). Know that unprotected directories can compromise your server's security, because they can reveal precious information and allow anybody to see information you would not want exposed.

Installed Trojan horses The most dangerous thing that can compromise your server is a Trojan horse, which allows anybody to remotely connect to your computer and execute network administrator (root) tasks. If, during a scan, your tool finds an installed Trojan, double-check that you haven't mistakenly launched a service on that port because your tool may interpret your daemon as a Trojan horse.

Enabled service protocols Some tools have the ability to match a service protocol to a particular server. The software must first determine if the server is enabled, and then look in its database to find a relationship between the version of your service and a known security hole.

The two best sites for finding information about this type of detailing and exploiting are www.packetstorm.org, and the Common Vulnerabilities site, cve.mitre.org.

27. CVE provides a full list of all known vulnerabilities and exposures at their website, cve.mitre.org.

To determine your company's vulnerabilities, you must run your own tests, up to the point of actual exploitation. I decided to hit my own computer first, to see how things were configured and running. Why not start at home, hmm? So I ran the test, thinking that nothing would happen, and voilà! The software found two shared folders I didn't know I had and a CGI syntax that could hit me. Ouch! Taking my own medicine, I hit a standard Apple OS X file/web/mail server that we set up—again, directly out of the box and following their instructions. Only one hole—but a very high–risk hole. Apple zealot that I am, I decided to test the Windows server configured for running Windows 2000 and a very popular PHP/SQL-based forum package. The system was set up by direction, and we'd applied all of the security patches that we could from the Microsoft website. The last set of tests was run against a Network Attached Storage (NAS) box fresh out of the shipping container and set up according to the manufacturer's own guidelines for security. You aren't going to *believe* the holes in this thing! We've italicized Web:80 below because of the nature of its importance and the number of attacks that can run through it.

	Mine	OS X Server	BBS	NAS Box
Folders				
/demo	•			
/manual	•			
Service Holes				
Samba:139 (VH)		•	•	•
LinePrinter:515 (VH)			•	•
Web:80 (VH)				•
FTP:21 (H)				•
SU_NFS:2049 (M)				•
Proxy:8080 (M)		•		
Echo:7 (L)			•	
Daytime:13 (L)			•	
CGEN:19 (L)			•	
PortMap:111 (L)				•
CGI Syntax				
/./	•	•	•	
/php/php.exec?c:\			•	
/\...\			•	

Table 9-7. Attack test results

Notice that the *NAS box has three very high–risk holes!* Think of it: You buy a NAS box because it's a "simple" way to add quantities of storage to your network. You believe in the manufacturer, so you follow their instructions and set it up according to their manual, only to find out that the thing has three *very* vulnerable security holes from the moment you turn it on. Yeow—that smarts! Let's examine how these holes could be exploited and what can happen to your system.

Service/Protocol holes found in common systems

While having folders open and available isn't great, and having CGI syntaxes available and open is bad, having service holes open can go from negligible to very high risk. Here's a rundown of the very high–risk service holes we found, and what could happen to your system if they were exploited.

Samba:139 is active (Risk: Very High)

Some buffer overflows can be used to gain root access.

The fix: Disable the service or upgrade the version.

LinePrinter:515 is active (Risk: Very high)

Various buffer overflows exist in implementations of the line printer daemon, especially under Linux, BSD, and IRIX, which can lead to a root compromise. Linux lpd has proven to be exploitable to spawn a root shell.

The fix: upgrade to the latest lpd version; disable the service.

WEB:80 is active (Risk: Very High)

There are about two pages of things that can happen to the server when this port is wide open. We'll just go through a few of them, so that you get the idea.

It has been reported that the authentication methods supported by a given IIS server can be revealed to an attacker through the inspection of returned error messages, even when anonymous access is also granted.

Vulnerability has been discovered in Microsoft IIS that may disclose the internal IP address or internal network name to remote attackers. This vulnerability can be exploited if an attacker connects to a host using HTTPS (typically on port 443) and crafts a specially formed GET request. Microsoft IIS will return a 302 Object Moved error message containing the internal IP address or internal network name of the server.

A flaw exists in version 5.0 of Microsoft IIS that makes it subject to a potential denial of service attack. The problem occurs when the server is preparing the MIME headers for the response to an HTTP request for a certain type of file

The server doesn't restrict access to certain types of files when their parent folders have less restrictive permissions.

There you have it—that's how they get in.

Setting Windows Group Policies

Normally, administrators use a Windows 2000 Active Directory to distribute Group Policy settings. In this situation, the administrator creates Group Policy objects that contain policy settings, and then uses Active Directory to target the delivery and application of these settings. When you use Windows 2000 clients in an environment where there is no Active Directory, you can distribute policy settings using Windows NT 4.0-style system policies or Local Group Policy[28].

- If the needed settings are available for editing with the Poledit.exe tool, administrators can create a policy on a Windows 2000–based server with the Poledit tool and save it as a Ntconfig.pol file. On the user's workstation, the administrator must modify the NetworkPath registry value in the following location:

  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Update\NetworkPath

If this value doesn't exist, the value must be added as a REG_SZ data type. For example, if the policy file is named Ntconfig.pol, and it's saved in the shared Directory \Policies on a Windows NT server, the value of NetworkPath should contain the following universal naming convention (UNC) path: \\Servername\Policies\Ntconfig.pol.

28. This information is courtesy Microsoft, and can be found on their website, at http://support.microsoft.com/default.aspx?scid=kb;en-us;Q274478

After adding the above registry entry, modify the registry entry UpdateMode(same location in the registry) to a value of 2. This sets the workstation to Manual Update mode, which is what you're setting by specifying the registry information above. If this entry, UpdateMode, is *not* changed, the workstation stays in Automatic mode, which means that it will look for the NTCONFIG.pol file in the default location instead of in the location you've specified.

When the users log onto the workstation, Windows 2000 can read the policy file specified by NetworkPath, and then apply the appropriate policy to the computer or user.

- If the settings that the administrator wants to enable are available on the user and computer level from the Group Policy Microsoft Management Console (MMC) snap-in, the administrator should use Local Machine policies. Since it may be difficult to visit each client to distribute and configure Local Group Policy, you can use the following two methods to configure Local Group Policy on multiple clients:
- Local Group Policy can be configured for a single system; then it can be cloned. The Microsoft System Preparation (Sysprep) tool can be used in conjunction with other third-party software to clone the computers. The cloned computers can retain the settings.
- Administrators can also configure a Local Group Policy on one client computer, and then copy the associate's pieces that make up the Local Group Policy Object (LGPO) to other clients.

Note that the only settings you can transfer from one client to another are those from Administrative Templates.

To edit the LGPO and to configure Local Group Policy settings on a local computer, and then to distribute to other computers, perform the following steps:

1. At the client requiring the policy settings, log on as an administrator and run the Group Policy snap-in (the Gpedit.msc file). Then focus the Group Policy snap-in on the Local Group Policy of the client.
2. Configure the LGPO on the client.
3. Edit and configure the policy settings you require.

4. Take the entries found in the Local Group Policy Object that are stored in the %Systemroot%\System32\GroupPolicy folder, and copy them to other clients where you also want to apply these Local Group Policy settings.

5. The settings under User Configuration normally take effect the next time the user logs on, and the settings under Computer Configuration normally take effect when you restart your system.

6. It may be necesary to edit the %systemroot%\system32\grouppolicy\gpt.ini and change the version entry so that the policy gets applied.

The preceding settings are stored in the LGPO on that client. If this client later joins a Windows 2000 Active Directory, Active Directory can override the settings in the LGPO using Group Policy distributed from Active Directory.

Security permissions on this folder can be changed to deny access to administrators to ensure that the policy does not apply to the local administrators.

If you use the preceding method, you must exercise much care because anything set on the original system that is specific to that particular computer is unsuccessful on the new target computer. In particular, many of the security settings for the computer should be avoided. If you're interested in only the administrative templates settings, copy the Registry.pol file to the target computer.

Setting the access control lists (ACLs) on the folder to prevent the local administrators from being affected works with any local built-in group, as well. When a change to the policy settings is required, the local administrator must take ownership, change the ACLs, make the change to the LGPO, and then change the ACLs back to *deny* for the local administrator. Combined with a certain set of policies, a failure to do this could render any further changes impossible.

Consider, for example, if the "Disable registry editing tools" or "Take ownership of files or other objects" policy is set and the "Deny access to this computer from the network" policy is set. This can lead to a situation in which administrators are locked out of a system.

In general, to avoid problems, be aware of the following suggestions:

- Each setting variance needs to be methodically tested prior to implementation.

- An administrator's strategy must be based on all clients having remote network access with Windows 2000.

If you wish to delegate the authority

Here are the steps to do so[29]:

1. Create an organizational unit (OU) and create a new GPO directly linked to this OU, by clicking **Properties** on the context menu of the OU, clicking the **Group Policy** tab in the Properties dialog box, and clicking the **New** button. Once the GPO has been created, launch the Delegation Wizard. The Delegation Wizard provides a step-by-step process in which specific functionality may be delegated easily, with a high degree of detail.

2. To start the Delegation Wizard, select the OU and right-click it. Then select Delegate Control. This starts the Delegation of Control Wizard.

3. Directly access the security settings for the GPO itself, by clicking **Properties** on the context menu of the specific GPO, and clicking the **Security** tab. Add your non-administrator user to the list of users for whom security is defined.

4. Provide your user **Full Control - Allow** privilege. Full Control gives the user the ability to write to the GPO, and also to change security permissions on the GPO. If you want to prevent this user from setting security, you may decide to give her only the **Write - Allow** permission. You may also decide that the user should be exempt from the application of this policy, and this may be accomplished by clearing the **Apply Group Policy - Allow** privilege.

To simplify administration for the user, launch the management console (Mmc.exe) and add the **Group Policy** snap-in. Browse for and add the GPO that you're configuring for delegation. Once this MMC session is appropriately configured, save the MMC session and give it to the user. The user can now utilize and administer her GPO with no additional setup.

29. http://support.microsoft.com/default.aspx?scid=kb;en-us;Q221577

SECURITY RESEARCH SITES

cve.mitre.org Common Vulnerabilities and Exposures, the authoritative list of vulnerability definitions

icat.nist.gov The National Institute of Standards and Technology's search engine for the CVE database

project.honeynet.org The Honeynet Project: Learn how to do detailed forensics after a compromise

www.2600.com Home of the venerable *Hacker Quarterly* magazine

www.atstake.com/research/advisories/index.html Formerly LOpht Advisories, this is an excellent list of advisories that often covers applications skipped by other lists

www.cert.org The CERT Coordination Center at Carnegie Mellon provides one of the best resources for security advisories and best-practices information

www.cisco.com/pcgi-bin/front.x/csec/csecHome.pl Cisco Secure Encyclopedia, a central warehouse of security knowledge

www.insecure.org Home of the definitive port scanner nmap, plus a great list of security tools

www.linuxsecurity.com All things Linux security

www.microsoft.com/technet/security Microsoft Product Security Notification Service and Microsoft's security vulnerability mailing list

www.ntbugtraq.com A Windows-specific vulnerability website and mailing list

www.sans.org Includes the SANS Institute's vulnerability list, white papers, and port scan statistics from monitors spread around the Internet

www.securityfocus.com Home of the Bugtraq mailing list archive, plus a good source of security white papers

www.securityportal.com Security news and commentary

www.wiretrip.net/rfp/ The "skinnable" home of Rain Forest Puppy provides detailed information on exploits and has been first to list several prominent vulnerabilities

Figure 9-40. I should have never *turned off that spam filter!*