

How Vulnerable?

We tested five VA

scanners to see

how well they illuminate

holes in your systems.

by JOEL SNYDER

INSIDE ...

- 34 What is Vulnerability Analysis?
- 36 Caveat Emptor
- 43 Tuning IDS
- 45 Comparison Chart

Quick: What's on your network? What's it running? Is it patched? Up to date? Properly configured? Are you *vulnerable*? A vulnerability analyzer (VA) is designed to help you answer these questions.

Many security managers first see the results of a vulnerability analyzer when a consultant drops an annual audit report on their desk. But as January's SQL Slammer worm reminded us, exploits aren't timed to coincide with audits. Like Code Red in 2001, Slammer exploited a well-documented vulnerability; and as with Code Red, a patch was available well before the worm struck. The point is, admins need an up-to-date picture of what's running on their network, where the holes are, and what's patched and what's not. We tested five¹ tools to see which had the best detection engines and reporting tools, and which did the best job managing the data from their findings:

- **Internet Security Systems'** Internet Scanner 6.21 (www.iss.net)
- **eEye Digital Security's** Retina 4.9 (www.eeye.com)
- **Symantec's** NetRecon 3.5 (www.symantec.com)
- **SAINT's** SAINT 4.1 (www.saintcorporation.com)
- **Nessus** 1.2.6 (www.nessus.org) and NessusWX 1.4.2 (www.nessuswx.org)

To determine how well these tools support real-world tasks, we staged our testing in phases to get a full picture of our lab network and its vulnerabilities:

- First, we did a standard audit-style run, looking for all known vulnerabilities on our test network.
- Then, we moved into network mapping mode, specifically to support tuning of our Sourcefire IDS (*see "Not-So-Fine Tuning," p. 43*).
- After getting a good handle on the network, we tested the ability of the vul-

nerability analyzers to react to changes, as systems and services were turned on and off, and as vulnerabilities were introduced and fixed. (This one turned out to be a major discriminator between products.)

- Finally, we moved the VA tools outside our firewall to verify correct operation and configuration of the firewall.

Our test network was large enough to give the analyzers a run for their money. We scanned 14,025 IP addresses, with more than 1,130 active systems. Then, we randomly picked 25 of those systems and investigated them in depth (*see Vulnerability Hunters, below*).

Because scanning is network and processor intensive, all VA products offer different types of scans, usually called "policies." We set the vulnerability analyzers at their highest, or most intensive, setting. We took the advice in each product's documentation to pick the heaviest "safe" scan—safe in the sense that the scan was not supposed to crash any system, *although this definitely wasn't the case* (*see "Caveat Emptor," p. 36*).

We installed all the analyzers on dual-processor 1.7 GHz systems with 512 MB of memory.

Based on our test results, each product received letter grades in four evaluation categories, and a final grade based on overall performance (*see Comparison Chart, p. 45*). The four criteria were:

1. **Mapping.** Many network managers aren't even sure exactly what's running where. Sometimes, surprises spring

VULNERABILITY HUNTERS

As part of our evaluation of the five vulnerability assessment tools, *Information Security* conducted intensive testing on 25 systems, seeing which tools identified 10 known vulnerabilities and which failed. Internet Scanner did the best job discovering the top 10 vulnerabilities selected among 25 test hosts, missing only one. (*For a complete report card on each product, see p. 45*)

VULNERABILITY	PRODUCT				
	SAINT	NetRecon	Internet Scanner	Nessus	Retina
TCP Sequence Prediction			✓		
Anonymous FTP		✓	✓	✓	
SNMP Writeable	✓	✓	✓	✓	✓
SSH Old Version	✓	✓	✓		
Dormant W2K Accounts			✓		
SNMP Readable		✓	✓	✓	
Cisco HTTP Bug			✓	✓	
VPN Server	✓				✓
VNC and RDP Running			✓	✓	✓
UDP Echo	✓	✓	✓	✓	✓

¹Several other vendors in the VA space didn't participate for a variety of reasons. Foundstone, maker of FoundScan, and eSecurityOnline, which offers Advisor, declined to participate.BindView submitted a buggy beta version of its bv-Control product, but was dropped when it failed to submit a working version. Harris said its STAT Scanner is more of a host assessment than a network assessment tool. Since the scope of this article covered only licensed commercial and free vulnerability assessment tools, our testing didn't include vulnerability assessment services, such as Qualys.

What Is Vulnerability Analysis?



As the number of security threats to networks and servers grows, security managers have turned to vulnerability analysis tools to identify a wide variety of potential problems on their networks. While host-oriented patch tools such as UpdateEXPERT from **St. Bernard Software** (www.stbernard.com) and HFNetChkPro from **Shavlik Technologies** (www.shavlik.com) focus on the myriad patches needed to keep Windows servers up to date, network vulnerability analyzers look for more than just missing patches.

These tools can search for misconfigured application servers, such as Web servers; and network components, such as switches and routers, that are vulnerable to known problems. They look for out-of-date applications, especially those with known problems. And they often search for applications that are

enabled by default—but perhaps shouldn't be, such as RPC services on Unix or the UDP ECHO program on Windows NT/2000. Vulnerability analyzers are also security oriented, so they often look for "information leakage" from systems through DNS and other avenues, including SNMP and Windows registry.

Most vulnerability analyzers take a three-phase approach to testing:

1. Given a network range by the security manager, the VA attempts to determine which IP addresses are in use. This phase usually includes tools such as ping.
2. The VA attempts to determine which applications and services are running on these systems, and their configurations. The VA uses a variety of techniques, ranging from simply trying to connect (a port scan) to gathering actual socket information out of SNMP.
3. The tool employs a long series of tests to find out if each system is susceptible to a particular known bug or problem. Smarter products iterate between phases two and three, learning more and using that information to launch additional tests. Others have ways of pruning their decision tree to save time and minimize the risk of overloading the target systems.

There are many variations within these three phases. Some products try to brute-force guess passwords on accounts. Others assume a "friendly" environment and connect to servers with administrative access to look for problems at the system level. Some are more devious, and will try to evade a network IDS. ▶

—JOEL SNYDER

up even when they think they know what's out there. Network mapping is a critical first step in any network security project.

2. Vulnerability analysis. Most of these scanners come with more than 1,000 tests to find software and configuration problems. But do those tests work? How many vulnerabilities will slip in under the radar?

3. Data management. Large networks generate hundreds, if not thousands, of records of network map and vulnerability information. We evaluated how these tools let network managers sort, sift and report on all that data—and how hard they have

to work to do it.

4. Performance. These tools aren't designed to run in real-time, but normally you'd want a scan of even a large network to complete in less than a day.

Mapping the Network

The first task of a vulnerability analyzer is to discover what's on the network, and what it's running. In addition to discovering systems and services, we tested each product's ability to spot services running on nonstandard ports, and what OSes were running on each system.

Discovering systems: In our behind-the-firewall tests, all systems would respond to a simple ping, so finding the systems was easy. However, ping isn't really adequate when testing through a firewall or validating a firewall configuration. All of the vulnerability analyzers had additional techniques for discovering resources, although not all were very well documented.

In testing outside the firewall, Nessus offers the best control, with six port-scanning techniques, from simple ping to SNMP discovery to actually trying a TCP connection. eEye's Retina allows you to scan a host even if it doesn't respond to ping packets.

Discovering services: Pinging systems is one thing, but discovering what TCP/IP network services are on each system, such as Web or mail servers, is another matter entirely.

ISS's Internet Scanner identified the most services correctly, throwing in a few false positives, and only missing a few. Nessus came in with a slightly lower score, but with no false positives.

Retina and SAINT did well in most cases, but both had major functional flaws. On two of our selected systems, Retina simply went berserk, reporting hundreds of nonexistent TCP and UDP services, which casts doubt on the accuracy of everything it did.

SAINT generally did an excellent job, even finding some services that the others missed (for example, an SMTP server hidden on port 2525). However, its internal database was confused by a DNS trap we laid for it, and one system couldn't be scanned or reported on by IP number or DNS name. It also performed poorly on a Windows system with many simultaneous Web servers, missing not only the nonstandard ones, but one running on port 80 as well.

Symantec's NetRecon is useless as a mapping tool. While it might be possible to dig through the alerts and reports and

Caveat Emptor

Before you start scanning your network, be forewarned: These vulnerability analyzers need to be used with extreme care. They are dangerous, and they will crash your systems. Although most of the configuration tools have options to disable “dangerous” or “denial-of-service” scanning, that isn’t always sufficient to keep them up.

Nessus gets our vote for “Most Unsafe Program to Have on Your Network.” We not only crashed servers and clients consistently with Nessus; we even confused our GPS-based NTP server enough that it had to be REFLASHed with new firmware. But Nessus wasn’t alone in taking systems down: Every one of the VAs crashed at least one system or application during the month of testing. Sometimes it was the firewalls that suffered: our normally rock-solid NetScreen firewall locked up once during testing.

This is an important and critical point. You can’t just take VAs and let them loose on your network to keep scanning it over and over. Even if you don’t get consistent crashes, you’ll discover a race condition or load-related problem eventually, which will cause the analyzer to crash something. We also have complaints about the huge number of alarms the scanning process creates. We had to completely shut off our IDS while testing, because the large number of false alarms rapidly overwhelmed it. ▶

—JOEL SNYDER

discover what’s running, NetRecon’s designers definitely didn’t envision this as a use for their product.

Locating services on nonstandard ports: Identifying TCP/IP network services as “open” isn’t sufficient for most network managers. They want to know what services are actually running on what ports. For example, it’s easy to assume that port 80 is running HTTP, but what if it’s really running an SMTP server? Or, what if someone has started an HTTP server on port 25? These kinds of configuration exceptions point to holes in a security infrastructure.

This capability is essential for a VA tool. It’s the kind of mapping data needed to keep the IDS accurate and comprehensive.

Nessus and Retina were tops at finding TCP/IP network services running on nonstandard ports. Still, they had some problems. Retina identified all of the services we had stashed on nonstandard ports, but didn’t follow through. For example, an HTTP server on port 81 was correctly tagged and analyzed, setting off alarms. However, an SMTP server on port 80 was identified, but Retina didn’t call it out as a relay—even though it flagged the same SMTP server running on port 25.

We also ran into GUI defects and bugs in Retina. For example, although it knows that the server running on a particular port isn’t what you’d expect, the GUI gives you no clue. You have to click on something clearly labeled as a Web server to drill down

and discover that Retina knows it’s really a mail server.

Nessus did a good job of identifying services on nonstandard ports, but as with many open-source products, it was inconsistent on quality control. Since each vulnerability test is actually a short script written by a contributing volunteer, each script has to be port-independent. In one case, we got both HTTP and SMTP alerts on the SMTP server running on the HTTP port because of a poorly written script.

NetRecon and SAINT failed to flag any TCP/IP network services on nonstandard ports. Internet Scanner can search for vulnerabilities on HTTP servers on particular ports, but only if you already know they’re there.

OS identification. This is a fairly perilous feature, since it’s easy to get wrong—and most of the products did, most of the time. Internet Scanner and Retina were best, getting the OS right about half of the time.

Many of these products take OS identification into account when scanning, both to expedite the scan process and to reduce false alarms. This led NetRecon to miss a critical vulnerability in our network—incorrectly identifying a Cisco switch as an OpenVMS server. It also failed to alert us to a major Cisco-specific bug.

Conversely, getting the operating system right is no guarantee of reduced false alarms: Retina showed us over a dozen Unix-specific alerts when it was analyzing an OpenVMS server.

Performance

Nessus performed best on a full-scale scan of our 14,025-address network, covering it in less than 19 hours. But don’t read too much into that number—most of the competition was dismal, either failing to complete a full scan or scanning too slowly to be practical. Nessus almost failed, too. Several times during our large scans, Nessus locked up and never completed the scans.

We had a similar problem with Internet Scanner. It would get to what appeared to be the end of the scan (although we could never really tell) and then hang forever. So, we couldn’t calculate an actual finishing time for Internet Scanner, but it seemed to be on par with Nessus.

Retina performed well on the tests that worked, but we couldn’t complete the scans with all features turned on. The Retina developers worked out the bugs we discovered, after which we completed a scan in 17 hours. To do so, however, we had to disable SNMP and SQL testing. When eEye sent fixes, we re-enabled SNMP and SQL tests, but Retina crashed again.

SAINTE had a different problem. Because of the technique SAINTE uses to detect hosts, we had to keep splitting our large scan into smaller parts. Still, in some of our single-host tests, SAINTE ran much more quickly than, for example, Internet Scanner, but at a price—the scans were significantly less thorough.

NetRecon should get a prize for never once failing. Every time we ran our large scan, it *ground* and *hacked* and *coughed* through it without error. It’s the most stable of the products tested, but while “slow and steady wins the race” may be good enough for tortoises, it doesn’t cut it for security applications. Our big scan took NetRecon 71 hours to complete, more than three times longer than Nessus, Retina or Internet Scanner.

NetRecon was slow at just about everything. We couldn’t work with the whole data set generated by NetRecon when we tried to analyze our subset of 25 special hosts. Instead, we had

Vulnerability Analyzers

to rescan just those hosts. The alternative would be to have 15 to 20 seconds of delay every time we clicked the mouse.

Vulnerabilities

The core of VA products is their engine. Each of the engines had problems both with false positives (a vulnerability reported which was not actually there) and false negatives (failing to report a known problem). We were happier to see false positives—which could easily be ignored in future runs—than false negatives.

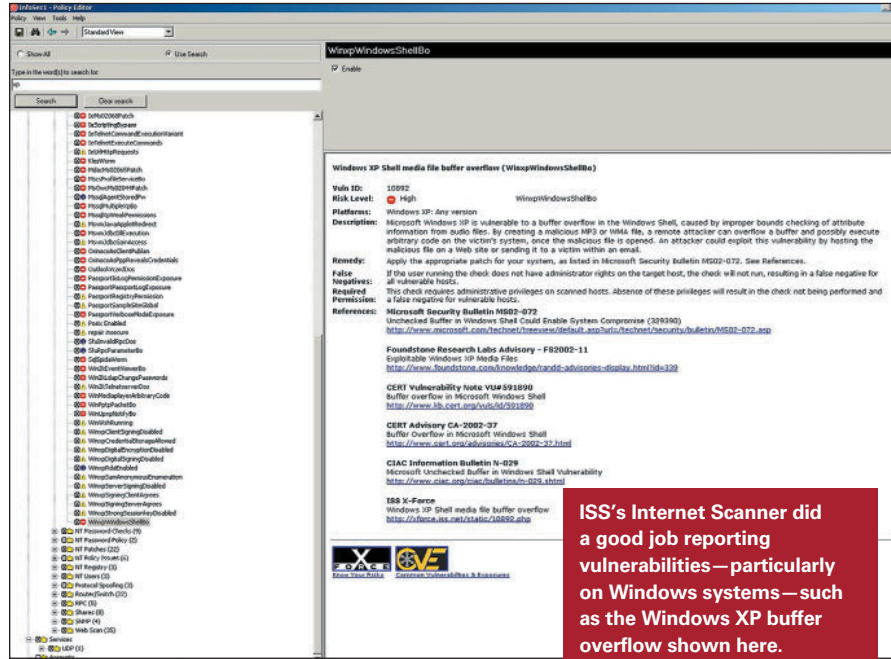
Internet Scanner, with especially good Windows coverage (see screen, right), and Nessus (see screen, below) performed best overall. Internet Scanner gave us the fewest false negatives, but a number of false positives. Nessus was a close second, giving us better information on *nix systems than on the Windows side, and more false positives across all architectures. Retina, SAINT and NetRecon missed a lot of problems on all the servers and systems we tested, but had a fairly low false positive rate.

The “weighting” of vulnerabilities is a real concern. Most of the engines use a three-level score—low, medium and high. (NetRecon was an exception, with a 100-point scale.) Most network managers will work on vulnerabilities identified as the highest risk, and the engines encourage that.

But these ratings were sometimes inaccurate. For example, Internet Scanner said that we were running a VNC server on one Windows system (true) with no password (false). VNC is a popular remote-control application, similar to pcAnywhere or Timbuktu. If this had been an actual vulnerability, it would have been an ultra-critical problem. But Internet Scanner merely rated it in its lowest-risk “information” category.

Let’s look at some examples to illustrate the ups and downs of ferreting out vulnerabilities with these tools:

- There was simply no excuse for some errors. For example, one of our test systems was a Cisco switch with an 18-month-old vulnerability allowing anyone with a Web browser full administrative access. Internet Scanner and Nessus found it; the others missed it. Also, most of the systems had an easily guessed SNMP community string. Internet Scanner, NetRecon and Nessus each found only one insecure string; Retina and SAINT failed to flag such strings in more than 20 percent of the systems that had them.
- Windows 2000 servers require privileged access to be audited remotely, but we didn’t give the analyzers that informa-



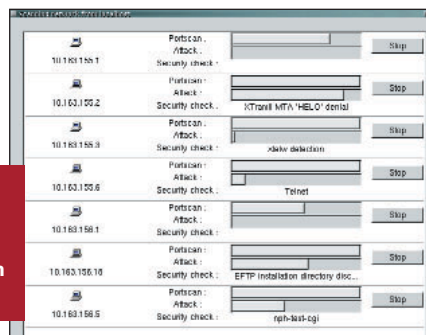
ISS's Internet Scanner did a good job reporting vulnerabilities—particularly on Windows systems—such as the Windows XP buffer overflow shown here.

tion. We tested what they could get via null sessions, remote registry and SNMP. Internet Scanner and Retina gave full details on the Win2K servers, including password issues with accounts and dormant accounts. Nessus only alerted us to a null session misconfiguration, while SAINT sidestepped the whole issue by simply asking, “Is your NetBIOS configured securely?”—with the singularly unhelpful advice to turn off NetBIOS over TCP/IP. NetRecon didn’t appear to use the null session or remote registry information, but did troll heavily for information through the SNMP Management Information Base (MIB). It didn’t, however, appear to pull any Windows-specific vulnerabilities out of SNMP.

- A few results confused us as well. For example, Retina flagged a buffer-overflow problem on one of the six Telnet servers it found, although none of the six were actually vulnerable. It’s difficult to understand how Retina reported a false positive on one of the servers, NetRecon on two and Nessus on one—but not the same server Retina flagged.

With new vulnerabilities being published with alarming frequency, keeping these tools current is essential. The best fully automated updates are in Retina, Internet Scanner and NetRecon. SAINT offers a Perl plug-in to help simplify the process as well. Nessus’ vulnerability database is updated regularly, but you have to go to the Web site.

While these tools may come with a fairly comprehensive set of tests, admins often need to create custom tests quickly and easily to examine specific conditions on their network—preferably without having to be a master programmer. For example, on our network, a common management application needed patching. We tried to write custom tests for each tool that would let us detect the unpatched application by



Nessus, a free open-source tool, was one of the two top performers in our lab test. This screen shows a scan of seven hosts in progress. The entire scan, or a scan of a given host, can be stopped with a mouse click.

Vulnerability Analyzers

the version number in its welcome banner.

Retina makes it very easy to add some types of tests, such as checking for a particular banner on a service, a registry key or some script on a Web server. You use a wizard-like GUI to define what to look for, and you're done in just a few seconds. More complex tests have to be written in a programming language.

Nessus facilitates test creation with Nessus Attack Scripting Language (NASL), written for vulnerability testing. Because Nessus is open source, you get more than 1,000 templates.

Internet Scanner doesn't make the job easy: additional vulnerability tests are written in either C or Perl, and you get just a single example to work from.

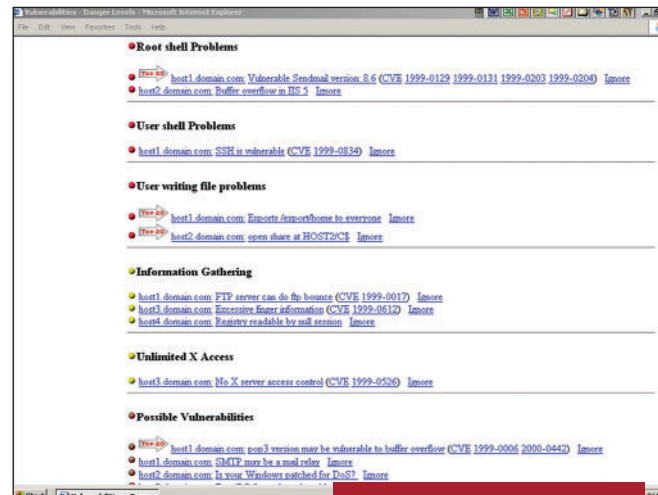
SAINT is better. Although you still have to write code in a traditional programming language, SAINT offers you many more examples.

NetRecon has no provision for creating custom tests.

Data Management

Data management is vital, especially if you plan to scan your network more than once. It's important to be able to see results in many different formats, pivoting across different views, and comparing data over time. We were disappointed in the functionality we found.

None of the five test products stood out for their data manage-



SAINT's use of hyperlinks facilitates navigation, as with this dynamic data analysis of discovered vulnerabilities. It's difficult, however, to export the hyperlinked information.

ment capabilities.

Internet Scanner, marginally the best of the tools in this category, has an easy-to-navigate and robust data management tool. Vulnerability scans are organized by "session," which combines host lists and scanning policies. A session has a set of systems to be scanned and a policy for which vulnerabilities to look for. Multiple scans across the same session are saved separately in an ODBC database. Unfortunately, there are few intrasession manipulation tools, so you're stuck looking at each session one at a time. We were disappointed, too, that Internet Scanner really didn't have a good way to manipulate data over time—for example, to show trend analysis of multiple scans of the same systems.

Viewing session data is easy. Internet Scanner lets you pivot across four views: by host, by vulnerability, by service (such as SMTP or HTTP) or by account. Its vulnerability database is outstanding. Each discovered vulnerability is accompanied by comprehensive documentation, including remedies, false positives and negatives, and multiple URLs for research.

NetRecon's data manipulation tools gave us the sense of having great control over our data. That's important, because NetRecon generates a lot of data. For the 25 systems we looked at in detail, NetRecon generated 4,984 records, but more than 4,000 of these were entries like "this system responded to ping" or "I found this name in the DNS." You can drown in irrelevancies. NetRecon has great slice-and-dice capabilities, but the problem is that you're slicing and dicing an enormous amount of data, mixing important and unimportant, critical and irrelevant.

In addition to host- and vulnerability-oriented sorting, NetRecon sorts by "objectives." For example, pick "find SMTP vulnerabilities," and it will show you all of the vulnerabilities related to SMTP.

The list of vulnerabilities can be sorted by any one of 24 different columns, ascending or descending, with a single click—assuming you're looking at a small number of hosts. When we tried to do this on our entire network, NetRecon's analysis tools were too slow to be usable.

SAINT does a poor job integrating the results of scans, their configuration and some repeatable way to run the same scan. It's the best tool for exploring the characteristics of your network, and for running ad hoc vulnerability tests as you explore the web of trust. But in a more production-oriented environment with regular, repeated, tests, the tools just aren't there.

Vulnerability Analyzers

The reporting capabilities in Symantec's NetRecon are its strong suit, with numerous options and great flexibility. Users can report by host, vulnerability or risk level.

SAINT's Web-based GUI was both its strength and weakness. The developers took the idea of hyperlinking to heart, with the result that you can drill down, through, across and over your data very quickly (see screen, p. 40). Start, perhaps, by looking at a list of vulnerabilities, then click on one to see which hosts are affected, then click on an affected host to see what other problems it might have, or services, and then click...it's a maze of possibilities which lets you jump around very quickly in your data set. Unfortunately, even with all this functionality, it's hard to export the hyperlinked data stream.

Nessus itself doesn't have a data management tool. We decided to try NessusWX, an open-source Windows-based client that has more sophisticated data management functions than the Nessus native tool. A Unix tool is also available.

Like Internet Scanner, NessusWX organizes vulnerability scans by session. At any time, you can bring up an old session and have Nessus rescan using the same set of saved parameters and vulnerability tests. It's also easy to compare the results of two scans.

Unfortunately, NessusWX isn't a very sophisticated information browsing tool. You can only look at your results by host, sorted alphabetically (not numerically). Plodding through vulnerabilities one at a time was pretty tedious. On the plus side, you can mark a vulnerability as a false positive, which means that it won't show up in reports. The vulnerability description text associated with each alarm is thin compared to Internet Scanner and Retina.

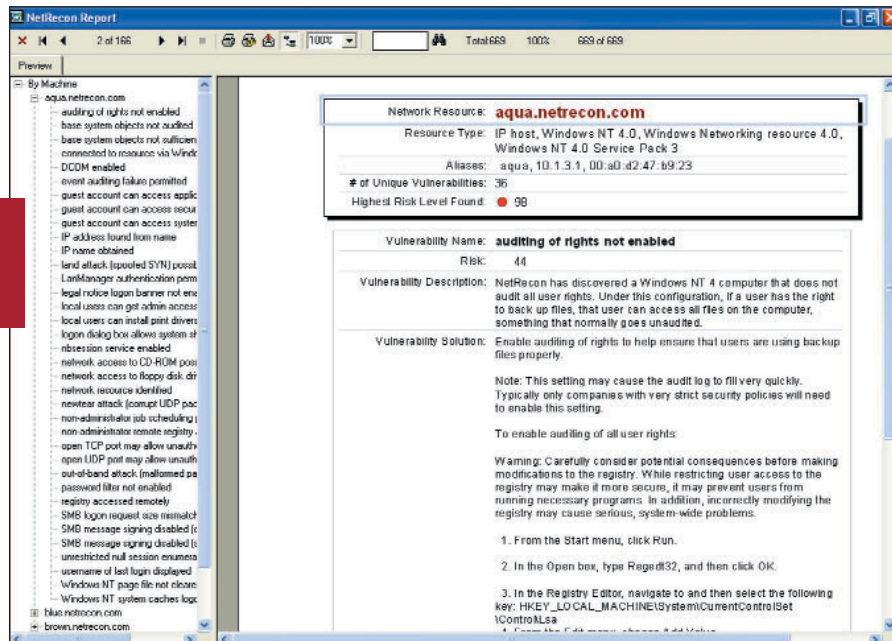
We got off to a bad start with Retina. eEye has a suite of Web-based management tools that wrap around the scanner. Unfortunately, they didn't work very well for us, and we spent so much time working out bugs in the scanner itself that we didn't get back to these. (However, in an off-site demo after our testing was completed, eEye was able to demonstrate its enterprise-level management system, which would have improved our perception of the capabilities of the product.)

We fell back to just looking at Retina—with a sigh of relief! The Windows-based GUI on Retina doesn't have a lot of data management features, but it's one of the nicest we've seen at giving you an overview on a system-by-system basis of the problems found in vulnerability analysis. Unfortunately, Retina only allows you to look at your results on a system-by-system basis.

Reporting

The ability to generate reports in different formats and with different data sets is important in summarizing and presenting information.

Internet Scanner and NetRecon earned the highest marks as



reporting tools. The others were just fair.

Internet Scanner's reporting is sophisticated and flexible, with varying levels of detail. You can sort by host name or address, OS, service or vulnerability severity. You can pick which scan to report on, and which vulnerabilities, hosts and services to include. Reports are available in HTML, PDF and RTF formats.

NetRecon's powerful built-in reporting tool, based on Crystal Reports, lets you generate one of three types of reports—executive summary, detail sorted by host or detail sorted by vulnerability. Reports also can be filtered and trimmed, so you can select particular hosts, vulnerabilities or risk levels (see screen, above). You can print the report directly, or save it in Excel, Word or HTML.

SAINT's hyperlink format is especially hard to represent in a printed report. Reports are only useful when pushed out in HTML format—the ASCII report we created was formatted so poorly that it was useless. Although you do have a number of different types of reports, you can't modify which data go into them. Its reporting component, SAINTwriter, generates a lot more graphs than the standard reports, but these are limited. For example, one required us to distinguish between 20 different colors across a 15-screen report, remembering the difference between light-light-green and medium-light-green.

We found that it was easier to run reports with NessusWX than to stay with the Nessus GUI. The report writer is slim, but has the features we needed for our testing. Reports can be sorted by host name or vulnerability, filtered by the severity of the vulnerability. The report writer will generate text, HTML or PDF files.

Retina reports are also limited. Retina offers three types of reports, and while there are many options to customize the style, there are no options for customizing content (see screen, p. 44). The Retina report writer only generates HTML.

Making a Choice

None of the products we looked at excelled in all areas. Almost any might be good for a once-a-year scan of your network. But as day-to-day tools in the real world of corporate security, each had significant weaknesses.

That being said, ISS's Internet Scanner, a commercial prod-

NOT-SO-FINE TUNING

Vulnerability scanners prove mediocre tools for helping IDSes protect the network perimeter. by JOEL SNYDER

To test the value of our five vulnerability analyzers as practical tools for systems managers, we turned them to the very real-world task of tuning our Sourcefire network IDS.

We looked at three ways in which we wanted to use VA results to make our IDS more useful. We got mixed results, reflecting the weaknesses the VA tools demonstrated throughout our testing.

1. Nonstandard ports. Sourcefire can look for Web server attacks on any TCP port, but is initially configured with the most common ones—80, 8000 and 8080. Knowing where we have Web servers running would give us a chance to act on alerts before a vulnerable server caused serious problems.

It wasn't very easy collecting a list of ports on which our Web servers were running, since our VA tools sort reports either by vulnerability or by host. Only Retina and Nessus found servers on non-standard ports, so we ran a simple Perl script against their reports written to summarize the Web server results. If we were in a production environment, we would have dumped the events into a database.

In the real world, we could apply that solution to other kinds of servers, such as e-mail and database, to extend the coverage of our IDS.

2. Identifying Web servers. Because we had a little of virtually everything on our network, knowing who was running IIS and who had Apache was extremely valuable. Since the majority of attacks are crafted against those two, we broke our network down into three areas—IIS, Apache and everything else. Once again, the reporting structures didn't give us exactly what we wanted, so we wrote our own vulnerability test to simply tag a server as IIS, Apache or "other." SAINT, Nessus and Retina made this easy, with their robust tools for creating custom tests. You can't create custom tests with NetRecon, and it was too difficult with Internet Scanner to be worth the effort.

Actually, ISS's answer to our problem might be to buy their own RealSecure IDS, which can be linked with another ISS product, SiteProtector, to integrate vulnerability analysis and intrusion detection by feeding information from Internet Scanner to the network IDS.

3. Matching alerts to vulnerabilities. The ultimate in fine-tuning an IDS is the ability to match specific alerts with specific vulnerabilities. For example, if version 2.1 of a particular CGI script was vulnerable to a problem, we wanted to use VA results to enable or disable the IDS alert based on version information.

Our results were less than satisfying. Although we found that we could manually track at least some of our IDS alerts to software packages to version information to vulnerabilities, the job was difficult and time-consuming. For example, we picked an IDS alert, then had to see what package caused the alert—not too hard. Version number information wasn't so easy, but most IDS alerts are linked to the MITRE CVE database, which assigns a unique identifier to each problem and often has pointers to additional information. The only problem is that vulnerability analyzers don't list their vulnerabilities by CVE number.

So we went to Plan B, using the results from the VAs as a filter *after* the IDS alert had occurred. Before bothering a system manager with an IDS alert, we would use the VA tool to discover what it could about the system or software involved. In this case, we weren't so much "tuning" the IDS as filtering or

post-processing its output.

This was a little more successful, but still difficult. You would never want to undertake that kind of correlation without dumping your VA results into a database, both to speed the lookup and because you would want to reorganize the data slightly.

The problem with vulnerability scanners that put their own results in a database is that the schemas weren't particularly helpful for this task. They have no specific field, for example, that stores the name of a software package separately from the version number or the CVE vulnerability number.

The bottom line is that matching vulnerabilities and IDS alerts is possible, and can help enormously in the tuning process. However, unless you have the time and budget to write your own software tools, you'd probably want to turn to third-party products, such as Lightning Console from **Tenable Security** (www.tenablesecurity.com) rather than rolling your own. The alternative is buying an IDS with built-in links to the vendor's own vulnerability analyzer. ▸



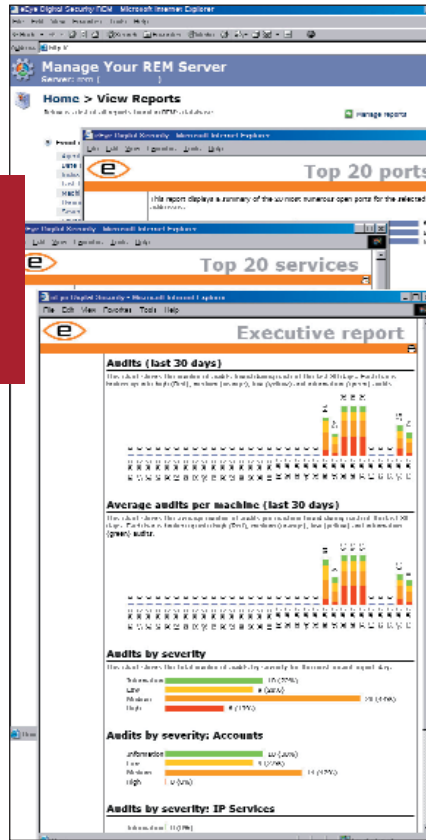
Vulnerability Analyzers

eEye's Retina gives the user great flexibility in customizing the style of its three types of reports, but there's no way to customize report content, and output is limited to HTML. Cascading windows here show an executive report of audits over the last 30 days, with other reports behind.

uct, and the open-source Nessus/Nessus-WX team were the best of the five tools we tested. Both did well at finding, reporting and managing vulnerability information on our network.

Internet Scanner stood out as being good in almost all areas, with an especially strong ability to track down problems in Windows environments.

While Nessus did well tracking problems all over the network, the lack of an automated update service and weak data management tools keep it from being a certain win. However, if you haven't



done much with vulnerability analysis before, Nessus is an outstanding start—and you may find that you never go commercial.

eEye's Retina is one of the most promising products we looked at, but isn't mature. Our experience wasn't outstanding because of multiple bugs, and the data management and reporting weren't very impressive. But Retina looks like the one to watch, and does have features (such as the ability to detect problems on nonstandard ports) that Internet Scanner does not.

We're less enthusiastic about SAINT and NetRecon. Neither stood out in any of the areas we cared about most: data management, documentation, reporting, or accuracy and completeness. ▀

JOEL SNYDER (joelsnyder@opus1.com) is a senior partner at Opus One, an IT consulting firm in Tucson.

Comparison Chart

Vulnerability Analysis Scanners

					
Product Name	SAINT v4.1 Saint www.saintcorporation.com	NetRecon v3.5 Symantec www.symantec.com	Internet Scanner v6.21 Internet Security Systems www.iss.net	Nessus v1.2.6 and NessusWX v1.4.2 The Nessus Project www.nessus.org	Retina v4.9 eEye Digital Security www.eeye.com
Pricing	From \$845 (10 hosts)	From \$3,995 (254 hosts)	From \$1,198 (10 hosts)	Open-source freeware	From \$6,520 (256 hosts)
Platform	Linux/x86/SPARC Solaris/HP-UX/ FreeBSD/OpenBSD	Windows	Windows	Nessus: Unix NessusWX: Windows	Windows
Network Mapping	C Name-to-address problems kept it from working well.	D Not useful as a mapping tool.	B Only a few false positives.	B+ Could be more accurate, but did detect nonstandard port usage.	C Generally excellent when it worked, but went berserk on two systems, calling other findings into question.
Vulnerability Testing	C- Missed some critical problems.	C Average results.	B Good balance between errors/accuracy; excellent Windows coverage.	B Good balance; excellent Unix coverage.	B- Didn't do well on nonstandard ports, despite good mapping.
Data Management	C+ Few tools to help the network admin, but good hyperlink displays.	B- Good tools, but slow performance.	B Good data manipulation, but nothing to help with repeated scans.	B- Nice data management, but difficult to sort through information.	B- Only one way to look at the data, but that one gives excellent overview of problems found.
Reporting	C- Weak reporting overall.	A Excellent reporting tools.	A Excellent reporting tools.	C Good selection of reports, but no ability to trim and filter.	C Good report selection; no filtering tools.
Performance	B- Had to break up scans into pieces. Fast when it worked.	C Slow	B- Fast, but didn't always run to completion.	B Scanned in appropriate length of time.	B- Fast, but didn't always run to completion.
Verdict	C Works, but has lots of room to grow.	C Overkill in some areas, underkill in others.	B Solid, balanced product, but can still learn from the competition.	B Solid and comprehensive. Excellent customization. Free.	C+ Good foundation, but needs bug fixes and enhancements.