Chapter 1

Vulnerability Assessment

Solutions in this Chapter:

- What Is a Vulnerability Assessment?
- Automated Assessments
- Two Approaches
- Realistic Expectations

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

In the war zone that is the modern Internet, manually reviewing each networked system for security flaws is no longer feasible. Operating systems, applications, and network protocols have grown so complex over the last decade that it takes a dedicated security administrator to keep even a relatively small network shielded from attack.

Each technical advance brings wave after wave of security holes. A new protocol might result in dozens of actual implementations, each of which could contain exploitable programming errors. Logic errors, vendor-installed backdoors, and default configurations plague everything from modern operating systems to the simplest print server. Yesterday's viruses seem positively tame compared to the highly optimized Internet worms that continuously assault every system attached to the global Internet.

To combat these attacks, a network administrator needs the appropriate tools and knowledge to identify vulnerable systems and resolve their security problems before they can be exploited. One of the most powerful tools available today is the vulnerability assessment, and this chapter describes what it is, what it can provide you, and why you should be performing them as often as possible. Following this is an analysis of the different types of solutions available, the advantages of each, and the actual steps used by most tools during the assessment process. The next section describes two distinct approaches used by the current generation of assessment tools and how choosing the right tool can make a significant impact on the security of your network. Finally, the chapter closes with the issues and limitations that you can expect when using any of the available assessment tools.

What Is a Vulnerability Assessment?

To explain vulnerability assessments, we first need to define what a vulnerability is. For the purposes of this book, *vulnerability* refers to any programming error or misconfiguration that could allow an intruder to gain unauthorized access. This includes anything from a weak password on a router to an unpatched programming flaw in an exposed network service. Vulnerabilities are no longer just the realm of system crackers and security consultants; they have become the enabling factor behind most network worms, spyware applications, and e-mail viruses.

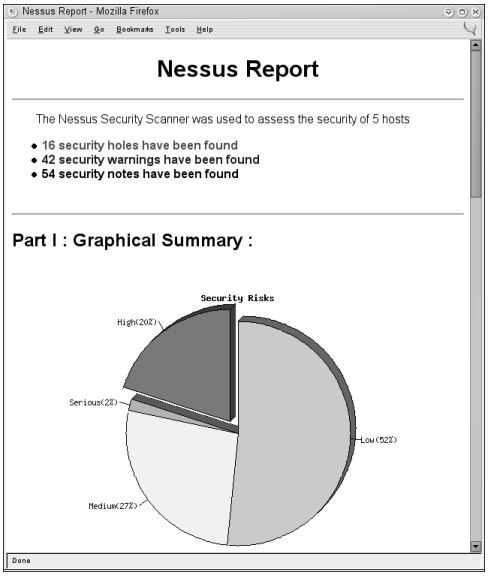
Spammers are increasingly relying on software vulnerabilities to hide their tracks; the open mail relays of the 1990s have been replaced by compromised "zombie" proxies of today, created through the mass exploitation of common

vulnerabilities. A question often asked is, "Why would someone target my system?" The answer is that most exploited systems were not targeted; they were simply one more address in a network range being scanned by an attacker. They were targets of opportunity, not choice. Spammers do not care whether a system belongs to an international bank or your grandmother Edna; as long as they can install their relay software, it makes no difference to them.

Vulnerability assessments are simply the process of locating and reporting vulnerabilities. They provide you with a way to detect and resolve security problems before someone or something can exploit them. One of the most common uses for vulnerability assessments is their capability to validate security measures. If you recently installed a new intrusion detection system (IDS), a vulnerability assessment allows you to determine how well that solution works. If the assessment completes and your IDS didn't fire off a single alert, it might be time to have a chat with the vendor.

The actual process for vulnerability identification varies widely between solutions; however, they all focus on a single output—the report. This report provides a snapshot of all the identified vulnerabilities on the network at a given time. Components of this report usually include a list detailing each identified vulnerability, where it was found, what the potential risk is, and how it can be resolved. Figure 1.1 shows a sample Nessus Security Scanner report for a network of only five systems; the number of vulnerabilities is already over 100!

Figure 1.1 Sample Nessus Report



Why a Vulnerability Assessment?

Vulnerability assessments have become a critical component of many organizations' security infrastructures; the ability to perform a networkwide security snapshot supports a number of security vulnerability and administrative processes. When a new vulnerability is discovered, the network administrator can perform an assessment, discover which systems are vulnerable, and start the patch installation process. After the fixes are in place, another assessment can be run to verify that the vulnerabilities were actually resolved. This cycle of assess, patch, and re-assess has become the standard method for many organizations to manage their security issues.

Many organizations have integrated vulnerability assessments into their system rollout process. Before a new server is installed, it first must go through a vulnerability assessment and pass with flying colors. This process is especially important for organizations that use a standard build image for each system; all too often, a new server can be imaged, configured, and installed without the administrator remembering to install the latest system patches. Additionally, many vulnerabilities can only be resolved through manual configuration changes; even an automated patch installation might not be enough to secure a newly imaged system. It's much easier to find these problems at build time when configuration changes are simple and risk-free than when that system is deployed in the field. We strongly recommend performing a vulnerability assessment against any new system before deploying it.

While many security solutions complicate system administration, vulnerability assessments can actually assist an administrator. Although the primary purpose of an assessment is to detect vulnerabilities, the assessment report can also be used as an inventory of the systems on the network and the services they expose. Since enumerating hosts and services is the first part of any vulnerability assessment, regular assessments can give you a current and very useful understanding of the services offered on your network. Assessments assist in crises: when a new worm is released, assessment reports are often used to generate task lists for the system administration staff, allowing them to prevent a worm outbreak before it reaches critical mass.

Asset classification is one of the most common nonsecurity uses for vulnerability assessment tools. Knowing how many and what types of printers are in use will help resource planning. Determining how many Windows 95 systems still need to be upgraded can be as easy as looking at your latest report. The ability to glance quickly at a document and determine what network resources might be overtaxed or underutilized can be invaluable to topology planning.

Assessment tools are also capable of detecting corporate policy violations; many tools will report peer-to-peer services, shared directories full of illegally-shared copyrighted materials, and unauthorized remote access tools. If a long-time system administrator leaves the company, an assessment tool can be used to detect that a backdoor was left in the firewall. If bandwidth use suddenly spikes, a vulnerability assessment can be used to locate workstations that have installed file-sharing software.

One of the most important uses for vulnerability assessment data is event correlation; if an intrusion does occur, a recent assessment report allows the security administrator to determine how it occurred, and what other assets might have been compromised. If the intruder gained access to a network consisting of unpatched Web servers, it is safe to assume that he gained access to those systems as well.

Notes from the Underground…

Intrusion Detection Systems

The difference between vulnerability assessments and an IDS is not always immediately clear. To understand the differences between these complimentary security systems, you will also need to understand how an IDS works. When people speak of IDSs, they are often referring to what is more specifically called a network intrusion detection system (NIDS). A NIDS' role is to monitor all network traffic, pick out malicious attacks from the normal data, and send out alerts when an attack is detected. This type of defense is known as a *reactive security measure* as it can only provide you with information after an attack has occurred. In contrast, a vulnerability assessment can provide you with the data about a vulnerability before it is used to compromise a system, allowing you to fix the problem and prevent the intrusion. For this reason, vulnerability assessments are considered a *proactive security measure*.

Assessment Types

The term *vulnerability assessment* is used to refer to many different types and levels of service. A host assessment normally refers to a security analysis against a single

system, from that system, often using specialized tools and an administrative user account. In contrast, a network assessment is used to test an entire network of systems at once.

Host Assessments

Host assessment tools were one of the first proactive security measures available to system administrators and are still in use today. These tools require that the assessment software be installed on each system you want to assess. This software can either be run stand-alone or be linked to a central system on the network. A host assessment looks for system-level vulnerabilities such as insecure file permissions, missing software patches, noncompliant security policies, and outright backdoors and Trojan horse installations.

The depth of the testing performed by host assessment tools makes it the preferred method of monitoring the security of critical systems. The downside of host assessments is that they require a set of specialized tools for the operating system and software packages being used, in addition to administrative access to each system that should be tested. Combined with the substantial time investment required to perform the testing and the limited scalability, host assessments are often reserved for a few critical systems.

The number of available and up-to-date host assessment solutions has been decreasing over the last few years. Tools like COPS and Tiger that were used religiously by system administrators just a few years ago have now fallen so far behind as to be nearly useless. Many of the stand-alone tools have been replaced by agent-based systems that use a centralized reporting and management system. This transition has been fueled by a demand for scalable systems that can be deployed across larger server farms with a minimum of administrative effort. At the time of this publication the only stand-alone host assessment tools used with any frequency are those targeting nontechnical home users and part-time administrators for small business systems.

Although stand-alone tools have started to decline, the number of "enterprise security management" systems that include a host assessment component is still increasing dramatically. The dual requirements of scalability and ease of deployment have resulted in host assessments becoming a component of larger management systems. A number of established software companies offer commercial products in this space, including, but not limited to, Internet Security System's System Scanner, Computer Associates eTrust Access Control product line, and BindView's bvControl software.

Network Assessments

Network assessments have been around almost as long as host assessments, starting with the Security Administrator Tool for Analyzing Networks (SATAN), released by Dan Farmer and Wietse Venema in 1995. SATAN provided a new perspective to administrators who were used to host assessment and hardening tools. Instead of analyzing the local system for problems, it allowed you to look for common problems on any system connected to the network. This opened the gates for a still-expanding market of both open-source and commercial network-based assessment systems.

A network vulnerability assessment locates all live systems on a network, determines what network services are in use, and then analyzes those services for potential vulnerabilities. Unlike the host assessment solutions, this process does not require any configuration changes on the systems being assessed. Network assessments can be both scalable and efficient in terms of administrative requirements and are the only feasible method of gauging the security of large, complex networks of heterogeneous systems.

Although network assessments are very effective for identifying vulnerabilities, they do suffer from certain limitations. These include: not being able to detect certain types of backdoors, complications with firewalls, and the inability to test for certain vulnerabilities due to the testing process itself being dangerous. Network assessments can disrupt normal operations, interfere with many devices (especially printers), use large amounts of bandwidth, and create fill-up disks with log files on the systems being assessed. Additionally, many vulnerabilities are exploitable by an authorized but unprivileged user account and cannot be identified through a network assessment.

Automated Assessments

The first experience that many people have with vulnerability assessments is using a security consulting firm to provide a network audit. This type of audit is normally comprised of both manual and automated components; the auditors will use automated tools for much of the initial legwork and follow it up with manual system inspection. While this process can provide thorough results, it is often much more expensive than simply using an automated assessment tool to perform the process in-house.

The need for automated assessment tools has resulted in a number of advanced solutions being developed. These solutions range from simple graphical user inter-

face (GUI) software products to stand-alone appliances that are capable of being linked into massive distributed assessment architectures. Due to the overwhelming number of vulnerability tests needed to build even a simple tool, the commercial market is easily divided between a few well-funded independent products and literally hundreds of solutions built on the open-source Nessus Security Scanner. These automated assessment tools can be further broken into two types of products: those that are actually obtained, through either purchase or download, and those that are provided through a subscription service.

Stand-Alone vs. Subscription

The stand-alone category of products includes most open-source projects and about half of the serious commercial contenders. Some examples include the Nessus Security Scanner, eEye's Retina, Tenable Security's Lightning Proxy, and Microsoft's Security Baseline Scanner. These products are either provided as a software package that is installed on a workstation, or a hardware appliance that you simply plug in and access over the network.

The subscription service solutions take a slightly different approach; instead of requiring the user to perform the actual installation and deployment, the vendor handles the basic configuration and simply provides a Web interface to the client. This is primarily used to offer assessments for Internet-facing assets (external assessments), but can also be combined with an appliance to provided assessments for an organization's internal network. Examples of products that are provided as a subscription service include Qualys' QualysGuard, BeyondSecurity's Automated Scan, and Digital Defense's Frontline product.

The advantages of using a stand-alone product are obvious: all of your data stays in-house, and you decide exactly when, where, and how the product is used. One disadvantage, however, is that these products require the user to perform an update before every use to avoid an out-of-date vulnerability check set, potentially missing recent vulnerabilities. The advantages of a subscription service model are twofold: the updates are handled for you, and since the external assessment originates from the vendor's network, you are provided with a real-world view of how your network looks from the Internet.

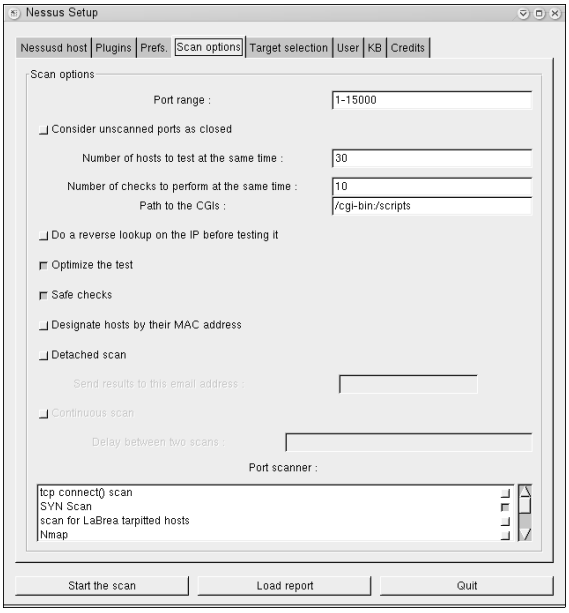
The disadvantages to a subscription solution are the lack of control you have over the configuration of the device, and the potential storage of vulnerability data on the vendor's systems. Some hybrid subscription service solutions have emerged that resolve both of these issues through leased appliances in conjunction with user-provided storage media for the assessment data. One product that

implements this approach is nCircles' IP360 system, which uses multiple dedicated appliances that store all sensitive data on a removable flash storage device.

The Assessment Process

Regardless of what automated assessment solution is used, it will more than likely follow the same general process. Each assessment begins with the user specifying what address or address ranges should be tested. This is often implemented as either a drop-down list of predefined ranges or a simple text widget where the network address and mask can be entered. Once the addresses are specified, the interface will often present the user with a set of configuration options for the assessment; this could include the port ranges to scan, the bandwidth settings to use, or any product-specific features. After all of this information is entered, the actual assessment phase starts. Figure 1.2 shows the assessment configuration screen for the Nessus Security Scanner.

Figure 1.2 Nessus Scan Options



Detecting Live Systems

The first stage of a network vulnerability assessment determines which Internet Protocol (IP) addresses specified in the target range actually map to online and accessible systems. For each address specified by the user, one or more probes are

sent to elicit a response. If a response is received, the system will place that address in a list of valid hosts. In the case of heavily firewalled networks, most products have an option to force scan all addresses, regardless of whether a response is received during this stage.

These types of probes sent during this stage differ wildly between assessment tools; although almost all of them use Internet Control Message Protocol (ICMP) "ping" requests, the techniques beyond this are rarely similar between two products. The Nessus Security Scanner has the capability to use a series of TCP connection requests to a set of common ports to identify systems that might be blocking ICMP messages. This allows the scanner to identify systems behind firewalls or those specifically configured to ignore ICMP traffic. After a connection request is sent, any response received from that system will cause it to be added to the list of tested hosts. Many commercial tools include the capability to probe specific User Datagram Protocol (UDP) services in addition to the standard ICMP and TCP tests. This technique is useful for detecting systems that only allow specific UDP application requests through, as is commonly the case with external DNS and RADIUS servers.

Identifying Live Systems

After the initial host detection phase is complete, many products will use a variety of fingerprinting techniques to determine what type of system was found at each address in the live system list. These fingerprinting techniques range from Simple Network Management Protocol (SNMP) queries to complex TCP/IP stack-based operating system identification.

This stage can be crucial in preventing the assessment from interfering with the normal operation of the network; quite a few print servers, older UNIX systems, and network-enabled applications will crash when a vulnerability assessment is performed on them. Indeed, the biggest problem that most administrators encounter with automated assessment tools is that they can disrupt network operations. Often, the administrator will have to spend time rebooting devices, retrieving garbage printouts from network-attached print servers, and debugging user problems with network applications. This identification stage can often be used to detect and avoid problematic systems before the following stages can cause problems.

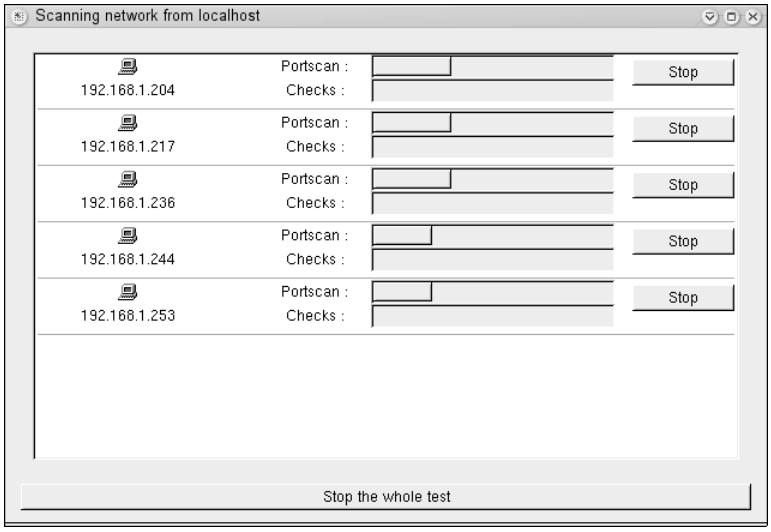
Enumerating Services

Once the host detection and identification steps are complete, the next stage is normally a port scan. A port scan is the process of determining what TCP and

UDP services are open on a given system. TCP port scans are conducted by sending connection requests to a configured list of port numbers on each system. If the system responds with a message indicating that the port is open, the port number is logged and stored for later use. UDP port scanning can often provide inconsistent results, since the nature of the protocol makes obtaining consistent results difficult on most networks.

There are 65,536 available TCP ports; however, most assessment tools will only perform a port scan against a limited set of these. Limiting the scan to a subset of the available ports reduces the amount of time it takes to perform the assessment and substantially decreases the bandwidth required by the assessment (in terms of packets per second, not the total number of bytes). The downside of not scanning all available ports is that services that are bound to nonstandard, high port numbers are often completely ignored by the assessment. The Nessus Security Scanner provides an option that allows the user to define how these ports are treated. The default is to consider all nonscanned TCP ports open, which can take quite a bit of time during the assessment, especially in cases where heavy packet filters or firewalls are in place. Figure 1.3 shows the Nessus Security Scanner performing the service enumeration phase of the assessment.

Figure 1.3 Nessus Enumerating Services



Identifying Services

After the port scan phase, many assessment tools will try to perform service identification on each open port. This process starts with sending some common application requests and analyzing the responses against a set of signatures. When a signature matches a known application, this information is stored for the later use and the next service is tested. Although not all assessment tools perform this stage, the ones that do can provide much more accurate results, simply by knowing which vulnerabilities to check for on what ports.

The Nessus Security Scanner includes a robust service identification engine, capable of detecting more than 90 different application protocols. This engine uses a set of application probes to elicit responses from each service. After each probe is sent, the result is matched against a list of known application signatures. When a matching signature is found, the port number and protocol are stored for future use and the engine continues with the next service. If the Secure Sockets Layer (SSL) transport protocol is detected, the engine will automatically negotiate SSL on the service before sending the application probes. This combination of transport-level and service-level identification allows the system to accurately detect vulnerabilities even when the affected service is on a nonstandard port.

The HyperText Transfer Protocol (HTTP) is a great example of a service that is often found on a port other than the default. Although almost all standard Web servers will use TCP port 80, literally thousands of applications install an HTTP service on a port other than 80. Web configuration interfaces for many Web application servers, hardware devices, and security tools will use nonstandard ports. E-mail protocols such as Simple Mail Transfer Protocol (SMTP), Post Office Protocol 3 (POP3), and Internet Message Access Protocol (IMAP) are often configured with the SSL transport protocol and installed on nonstandard ports as well. A common misconfiguration is to block spam relaying on the primary SMTP service, but trust all messages accepted through the SSL-wrapped SMTP service on a different port. Additionally, this phase prevents an application running on a port normally reserved for another protocol from being ignored completely by the scan or resulting in false positives.

Identifying Applications

Once the service detection phase is complete, the next step is to determine the actual application in use for each detected service. The goal of this stage is to identify the vendor, type, and version of every service detected in the previous stage. This information is critical, as the vulnerability tests for one application can

actually cause another application to crash. An example of this is if a Web server is vulnerable to a long pathname overflow. If any other vulnerability tests send a request longer than what is expected by this system, the application will crash. To accurately detect this vulnerability on the Web server instead of crashing it, the system must first identify that specific application and then prevent any of the problematic vulnerability tests from running against it.

One of the most common problems with most assessment tools is that of the *false positive* where the tool reports a vulnerability that does not actually exist on the tested systems. False positives can produce a huge amount of verification work for the assessment engineer. When application identification information is either missing or incomplete, test results will often include false positives. When the developers of these assessment tools write the vulnerability tests, they often assume that the system they are interacting with is always going to be the product in which the vulnerability was discovered. Different applications that offer the same service will often respond to a probe in such a way that the vulnerability test logic registers a vulnerability. For this reason, application identification has become one of the most critical components of modern assessment tools.

Identifying Vulnerabilities

After every online host has been identified, each open port has been mapped to a known service, and the known services have been mapped to specific applications, the system is finally ready to begin testing for vulnerabilities. This process often starts with basic information-gathering techniques, followed by active configuration probes, and finally a set of custom attacks that can identify whether a particular vulnerability exists on the tested system.

The vulnerability identification process can vary from simple banner matching and version tests, to complete exploitation of the tested flaw. When version detection and banner matching are used to identify a vulnerability, false positives often result due to application vendors providing updated software that still displays the banner of the vulnerable version. For this reason, version numbers are often consulted only when there is no other way to safely verify whether the vulnerability exists.

Many common vulnerabilities can only be identified by attempting to exploit the flaw. This often means using the vulnerability to execute a command, display a system file, or otherwise verify that the system is indeed vulnerable to an attack by a remote intruder. Many buffer overflow and input manipulation vulnerabilities can

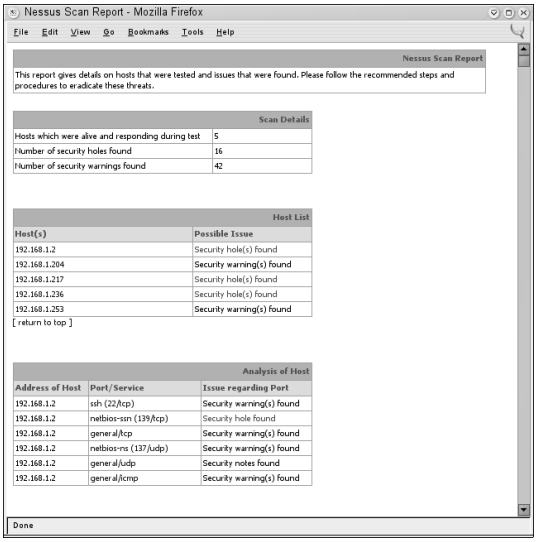
be detected by triggering just enough of the flaw to indicate that the system has not been patched, but not enough to actually take down the service. The assessment tool has to walk a fine line between reliable vulnerability identification and destructive side effects.

Vulnerability tests that use banner checks will encounter problems when the tested service has been patched, either by the vendor or system administrator, but the version number displayed to the network has not been updated, or at least when it has not been updated in the way the vulnerability test expects. This is a relatively common practice with open-source UNIX-based platforms and certain Linux distributions.

Reporting Vulnerabilities

After the analysis is finished, the final stage of the assessment process is reporting. Each product has a unique perspective on how reports should be generated, what they should include, and in what formats to provide them. Regardless of the product, the assessment report will list the systems discovered during the assessment and any vulnerabilities that were identified on them. Many products offer different levels of reporting depending on the audience; it is useful to provide a high-level summary to management giving a system administrator a report that tells him or her what systems need to be fixed and how to do so. One of the popular features in many assessment tools is the capability to show trend reports of how a given network fared over time. Figure 1.4 shows the Nessus Security Scanner's HTML report summary.

Figure 1.4 Nessus Report Summary



Two Approaches

When performing an automated vulnerability assessment, the actual perspective of the test can have a huge impact on the depth and quality of the results. Essentially, there are two different approaches to vulnerability testing: administrative and outsider. Each has distinct advantages and disadvantages, such that many of the better assessment tools have migrated to a hybrid model that combines the best features of both approaches. Understanding these different approaches can provide insight into why two different assessment tools can provide such completely different results when used to test the same network.

Administrative Approach

The administrative approach performs the assessment from the perspective of a normal, authenticated system administrator. The assessment tool might require that it be launched by an authenticated administrative user or provided with a user account and password. These credentials can be used to detect missing patches, insecure configuration settings, and potentially vulnerable client-side software (such as e-mail clients and Web browsers).

This is a powerful approach for networks that consist of mostly Windows-based systems that all authenticate against the same domain. It combines much of the deep analysis of a host assessment with the network assessment's scalability advantages. Since almost all of the vulnerability tests are performed using either remote registry or remote file system access, there is little chance that an assessment tool using this method can adversely affect the tested systems. This allows assessments to be conducted during the day, while the systems are actively being used, without fear of disrupting a business activity.

The administrative approach is especially useful when trying to detect and resolve client-side vulnerabilities on a network of workstations. Many worms, Trojans, and viruses propagate by exploiting vulnerabilities in e-mail clients and Web browser software. An assessment tool using this approach can access the registry of each system and determine whether the latest patches have been installed, whether the proper security settings have been applied, and often whether the system has already been successfully attacked. Client-side security is one of the most overlooked entry points on most corporate networks; there have been numerous cases of a network with a well-secured perimeter being overtaken by a network simply because a user visited the wrong Web site with an outdated Web browser.

Unfortunately, these products often have some severe limitations as well. Since the testing process uses the standard Windows administrative channels—namely, the NetBIOS services and an administrative user account—anything preventing this channel from being accessed will result in inaccurate scan results. Any system on the network that is configured with a different authentication source (running in stand-alone mode, on a different domain, or authenticating to a Novell server) will not be correctly assessed. Additionally, these products may have issues similar to the issues of host-based assessment tools, network devices, UNIX-based servers, and IP-enabled phone systems may also be completely missed or return incomplete results.

Network and host-based firewalls can also interfere with the assessment. This interference is a common occurrence when performing assessments against a system hosted on a different network segment, such as a demilitarized zone (DMZ) or external segment behind a dedicated firewall. Additionally, network devices, UNIX-based servers, and IP-enabled phone systems might also be either completely missed or have only minimal results returned. An example of this is a certain Windows-based commercial assessment tool that will report missing Internet Information Server (IIS) patches even when the Web server has not been enabled or configured.

This type of testing is very helpful to verify a networkwide patch deployment, but should not be relied upon as the only method of security testing. Microsoft's Security Baseline Scanner is the best example of an assessment tool that uses this approach alone. Many of the commercial assessment tool offerings were originally based on this approach and have only recently started to integrate different techniques into their vulnerability tests. The differences between administrative and hybrid solutions is discussed at length in the section *The Hybrid Approach*.

The Outsider Approach

The outsider approach takes the perspective of the unauthenticated malicious intruder who is trying to break into the network. The assessment process is able to make decisions about the security of a system only through a combination of application fingerprinting, version identification, and actual exploitation attempts. Assessment tools built on this approach are often capable of detecting vulnerabilities across a much wider range of operating systems and devices than their administrative approach counterparts can.

When conducting a large-scale assessment against a network consisting of many different operating systems and network devices, the outsider approach is

the only technique that has a chance of returning accurate, consistent results about each discovered system. If a system is behind a firewall, only the exposed services will be tested, providing you with the same information that an intruder would see in a real-life attack. The reports provided by tools that use this hybrid approach are geared to prevent common attacks; this is in contrast to those tools using the administrative approach that often focus on missing patches and insecure configuration settings. In essence, the outsider approach presents a much more targeted list of problems for remediation, allowing the administrator to focus on the issues that would be the first choices for a potential intruder.

Although this approach is the only plausible method of conducting a vulnerability assessment on a heterogeneous network, it also suffers from a significant set of drawbacks. Many vulnerabilities simply cannot be tested without crashing the application, device, or operating system. The result is that any assessment tools that test for these types of vulnerabilities either provide an option for "intrusive" testing, or always trigger a warning when a potentially vulnerable service is discovered. Since the outsider approach can only detect what is visible from the point in the network where the assessment was launched, it might not report a vulnerable service bound to a different interface on the same system. This is an issue with reporting more than anything else, as someone reviewing the assessment report might not consider the network perspective when creating a list of remediation tasks for that system.

The Hybrid Approach

Over the last few years, more and more tools have switched to a hybrid approach for network assessments. They use administrative credentials when possible, but fall back to remote fingerprinting techniques if an account is either not available or not accepted on the tested system. The quality of these hybrid solutions varies greatly; the products were originally designed with only the administrative approach in mind have a difficult time when administrative credentials are not available, whereas the products based on the outsider approach often contain glitches when using an administrative account for tests. It seems that the latter has better chances at overcoming its hurdles without requiring a re-write. Overall, though, these products provide results that are often superior to those using a single approach. The Nessus Security Scanner and eEye's Retina product are examples of tools that use this approach.

One of the greatest advantages of tools using the outsider approach is that they are often able to determine whether a given vulnerability exists, regardless of

whether a patch was applied. As many Windows network administrators know, installing an operating system patch does not actually guarantee that the vulnerability has been removed. A recent vulnerability in the Microsoft Windows Network Messenger service allowed a remote attacker to execute arbitrary code on a vulnerable system. Public exploits for the vulnerability started circulating, and companies were frantically trying to install the patch on all their internal workstations. Something that was overlooked was that for the patch to take effect, the system had to be rebooted after it was applied. Many sites used automated patch installation tools to update all their vulnerable systems, but completely forgot about the reboot requirement.

The result was that when an assessment was run using a tool that took the administrative approach, it reported the systems as patched. However, when an assessment was run using the Nessus Security Scanner, it reported these systems as vulnerable. The tool using the administrative approach simply checked the registry of each system to determine whether the patch had been applied, whereas the Nessus scan actually probed the vulnerability to determine if it was still vulnerable. Without this second assessment, the organization would have left hundreds of workstations exposed, even though the patches had been applied. The registry analysis used by many tools that take the administrative approach can miss vulnerabilities for a number of other reasons as well. The most common occurrence is when a hotfix has been applied to resolve a vulnerability, and then an older service pack is reapplied over the entire system. The changes installed by the hotfix were overwritten, but the registry entry stating that the patch was applied still exists. This problem primarily affects Windows operating systems; however, a number of commercial UNIX vendors have had similar issues with tracking installed patches and determining which ones still need to be applied.

Recently, many of the administrative and hybrid tools have developed new techniques for verifying that an installed patch actually exists. Shavlik Technology's HFNetChk Pro will actually check the last reboot time and compare it to the hotfix install date. The Nessus Security Scanner actually accesses the affected executables across the network and verifies the embedded version numbers.

The drawbacks to the hybrid approach are normally not apparent until the results of a few large scans are observed; because the administrative approach is used opportunistically, vulnerabilities that are reported on a system that accepts the provided user account might not be reported on a similar system that uses a different authentication realm. If the administrator does not realize that the other system might be vulnerable as well, it could lead to a false sense of security. These missed vulnerabilities can be difficult to track down and can fall under the radar

of the administrator. Because there is a higher chance of these systems not being patched, the hybrid approach can actually result in more damage during an intrusion or worm outbreak. Although the administrative approach suffers from the same issue, tools using the administrative approach take it for granted that systems outside of the authentication realm will not be tested.

Realistic Expectations

When the first commercial vulnerability assessment tools started becoming popular, they were advertised as being able to magically identify every security hole on your network. A few years ago, this might have been close to the truth. The number of publicly documented vulnerabilities was still quite small, and tracking vulnerability information was an obscure hobby. These days, the scenario is much different, whereas there were a few hundred well-documented vulnerabilities before, there are literally thousands of them now, and they don't even begin to scratch the surface when it comes to the number of flaws that can be used to penetrate a corporate network.

In addition to the avalanche of vulnerabilities, the number and type of devices found on an average corporate network has exploded. Some of these devices will crash, misbehave, or slow to a crawl during a network vulnerability assessment. A vulnerability test designed for one system might cause another application or device to stop functioning altogether, annoying the users of those systems and potentially interrupting the work flow. Assessment tools have a tough job; they have to identify as many vulnerabilities as possible on systems that must be analyzed and categorized on the fly, without reporting false positives, and at the same time avoid crashing devices and applications that simply weren't designed with security in mind. Some tools fare better than others; however, all current assessment tools exhibit this problem in one form or another.

When someone first starts to use a vulnerability assessment system, he or she often notices that the results between subsequent scans can differ significantly. This issue is encountered more frequently on larger networks that are connected through slower links. There are quite a few different reasons for this, but the core issue is that unlike most software processes, remote vulnerability testing is more of an art form than a science. Many assessment tools define a hard timeout for establishing connections to a service or receiving the result of a query. If an extra second or two of latency occurs on the network, the test could miss a valid response. These types of timing issues are common among assessment tools; however, many other factors can play into the consistency of scan results.

Many network devices provide a Telnet console that allows an administrator to reconfigure the system remotely. These devices will often set a hard limit on the number of concurrent network connections allowed to this service. When a vulnerability assessment is launched, it might perform multiple tests on a given port at the same time; this can cause one check to receive a valid response, while another gets an error message indicating that all available connections are being used. If that second check was responsible for testing for a default password on this particular device, it might completely miss the vulnerability. If the same scan was run later, but the default password test ran before one of the others, it would accurately detect the vulnerability at the expense of the other tests. This type of timing problem is much more common on network devices and older UNIX systems than on most modern workstations and servers, but can ultimately lead to inconsistent assessment results.

Tools & Traps…

Assessing Print Servers

Almost all vulnerability assessment tools have one thing in common; they are capable of eating a print server alive. The problem stems from the fact that many print servers offer a variety of network services that can be used to spool documents directly to the attached printer. The most problematic of these services is the Direct Print Protocol, which is a TCP service. This can cause problems with automated assessment tools, as the service identification phase can often cause reams of paper to printed out, covered in what appears to be garbage. Another common issue relates to the custom FTP service that many print servers run. This service will allow authentications using any username and password combination and simply prints out any files that are uploaded. If the assessment tool is looking for insecure FTP configurations, it might end up printing out a test file when running against a print server. To compound matters, quite a few print servers have such shoddy TCP/IP implementations that a simple port scan can take them offline, and a full power cycle is required to return them to service.

Dynamic systems are the bane of the vulnerability assessment tools. If an assessment is in full swing and a user decides to reboot his workstation, the assessment tool will start receiving connection timeouts for the vulnerability tests. Once the

system comes back online, any subsequent tests will run normally; however, all tests launched during the period of downtime will result in missing vulnerability results for that system. This type of problem is incredibly difficult to detect when wading through a massive assessment report, and at this time only a handful of commercial systems offer the capability to detect and rescan systems that restart during the assessment process.

Despite the extraordinary amount of refinement and testing that most assessment tools have undergone, false positives continue to annoy network administrators and security consultants alike. As we discussed earlier in the chapter, a false positive is simply a vulnerability that is reported, but does not actually exist on the tested system. These annoyances can build to quite a bit of verification work—before you throw out Nessus or any vulnerability assessment application for the false positive load, take the time to tune it as we show you later in this book. Nonstandard Web servers, backported software packages, and permissive match strings inside vulnerability test scripts are the top causes for false positives.

The Web server software that provides a configuration console for many network devices is notorious for causing false positives; instead of returning a standard "404" error response for nonexistent files, these systems will often return a success message for any file that is requested from the system. In response, almost all of the popular assessment tools have developed some form of Web server fingerprinting that allows their system to work around these strange Web servers. These solutions range from incredibly robust, such as the one found in the recent versions of the Nessus Security Scanner, to almost not worth the bother, as in certain commercial products.

The Limitations of Automation

Vulnerability assessment tools are still no replacement for a manual security audit by a team of trained security experts. Although many assessment tools will do their best to find common vulnerabilities in all exposed services, relatively simple vulnerabilities are often missed. Custom web applications, written under tight deadlines and for small user bases, often perform inadequate security checks on user input, but automated assessment systems may not find these flaws. Although the chances of an automated assessment tool being able to find a vulnerability in this software are slim, a security analyst experienced with Web application testing could easily pinpoint a number of security issues in a short period of time. Just because an automated assessment does not find any vulnerabilities does not mean that none exist.

Summary

As the number of discovered vulnerabilities increases every day, networks are becoming increasingly difficult to keep secure. Vulnerability assessments have become the preferred method of managing security flaws for many organizations. The ability to quickly identify misconfigured and unpatched systems, combined with the ease of use and accuracy of many assessment tools, has changed the way many administrators manage their systems. Network vulnerability assessments provide the wide view of security weaknesses on a given network, supplemented by host assessment solutions that provide granular hardening steps for critical systems.

The traditional process of system hardening and patch application has been left in the dust; as the sheer quantity of vulnerabilities is more than most administrator teams can keep track of, especially for diverse networks. Automated assessment solutions have come to the rescue, with both stand-alone and subscription-based options. The average administrator no longer needs to become a security savant simply to keep his or her systems secure. The same repeatable process allows administrators to track, resolve, and verify vulnerabilities.

Although almost all assessment tools advertise their capability to detect and report all critical vulnerabilities, the way these systems are designed and the techniques they use for vulnerability tests vary widely. Not all assessment solutions are created equal; tools using the administrative approach are almost useful when it comes to identifying vulnerabilities in network devices and across large networks. At the same time, tools using the outsider approach are restricted by the technical limitations of the vulnerabilities themselves, often ignoring vulnerabilities that they simply are unable to test. Fortunately, many of the more popular solutions have solidified around a hybrid approach for vulnerability testing, allowing for unprecedented levels of accuracy and depth.

Vulnerability assessments are not a security panacea; although they excel at detecting vulnerabilities in widely deployed products, even relatively simply flaws can be missed. The current market of assessment tools can often cause problems with network devices, slow internetwork links, and custom applications. No matter what tool you use, false positives will always be a significant problem; although many solutions have made huge steps in reducing these, backported patches and vague version identifiers will guarantee that these never entirely disappear. The depth and flexibility of a manual security assessment will always be better than any automated solution; there is no replacement for a skilled analyst manually reviewing your systems, network architecture, and in-house applications.

Solutions Fast Track

What Is a Vulnerability Assessment?

- ☑ A vulnerability is any flaw that an attacker can use to gain access to a system or network.
- ☑ Vulnerability assessments provide a snapshot of the security posture of your network.
- ☑ Host assessments provide detailed information about the weaknesses on a system.
- ☑ Network assessments pinpoint flaws that a remote attacker can use to gain access.

Automated Assessments

- ☑ Manual assessments are no longer feasible for entire networks due to the sheer number of vulnerabilities that exist.
- ☑ Stand-alone and subscription assessment models each have distinct advantages.
- ☑ Automated assessments tend to follow the same process regardless of the tool.
- ☑ The assessment process is essentially staged information gathering.

Two Approaches

- ☑ Two assessment tools can provide very different results depending on their approach.
- ☑ The administrative approach is often safest, but might not be reliable.
- ☑ The outsider approach provides the same information an attacker would have.
- ☑ Robust assessment tools use a hybrid approach for maximum vulnerability coverage.

Realistic Expectations

- ☑ Assessments can cause a myriad of side effects on an average corporate network.
- ☑ Consecutive between assessments is often less than ideal.
- ☑ False positives will always be an issue, but recent tools are making progress.
- ☑ Manual security audits still provide better results than any assessment tool can.
- ☑ Penetration testing can provide a deeper, if not wider, view of your network, from the perspective of an attacker.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: I am planning to use a vulnerability assessment tool at my organization. Is there any reason to assess the internal networks as well as the external?

A: While systems exposed to the Internet should always be incorporated into a vulnerability assessment plan, internal assessments can actually reduce the risk to the organization even more. When a new worm appears that exploits one or more known vulnerabilities, the first step an organization should take is to secure all external and internal systems. An internal assessment can be used to verify that internal assets are not at risk to an automated attack. Internal networks are vulnerable to infection through users who are compromised through their e-mail clients and Web browsers; a worm infection on an internal network segment can result in the inability for the business to function. Additionally, unethical consultants, disgruntled employees, and visitors using the network can leverage insecure systems to gain access to sensitive information.

Q: What is the difference between a vulnerability assessment and a penetration test?

A: One of the biggest problems with the security industry is consistent naming of services. A strong contributing fact is that many near dishonest security firms are selling "penetration tests" that are nothing more than a vulnerability assessment using automated tools. A vulnerability assessment is the process of identifying vulnerabilities on a network, whereas a penetration test is focused on actually gaining unauthorized access to the tested systems and using that access to the network or data, as directed by the client. A penetration test is a great way to determine how well your security measures respond to a real-life attack and what an attacker could accomplish or compromise, but may not result in a detailed analysis of every system on your network.

Q: Can a vulnerability assessment find users with weak passwords?

A: Although manual vulnerability assessments can include password auditing, automated vulnerability assessment tools are rarely able to detect common or weak passwords. The reason behind this is not that the tool is not technically able to perform the check, but that the process of testing each user could result in an account lockout. This is primarily the case with Windows domains; however, it can also apply to many commercial UNIX systems. While some automated assessment tools will test for accounts with a default or blank password, they would still not be able to detect an account with a simple one-character password. Finally, automatic tools might slow the application or network being tested. This is a part of the security assessment process that needs to be very carefully coordinated with administrators to achieve maximum success while causing a minimum of negative effects for users.

Q: My organization uses an intrusion prevention system (IPS). What complications will this cause with a vulnerability assessment?

A: The goal of an IPS is to block hostile traffic before it reaches a potentially vulnerable system. Many automated assessment solutions depend on being able to send a specially crafted attack probe and to determine whether the system is vulnerable by analyzing the response. If the IPS blocks the initial probe, the vulnerability assessment will not be able to accurately detect that vulnerability. The solution to this is either to configure the IPS to specifically ignore traffic originating from the vulnerability assessment tool, or only run

the tool from the protected side of the IPS. Most assessment tools are not designed to bypass these systems; however, an advanced intruder could easily detect the IPS and find a way to exploit a vulnerability while avoiding the IPS's block. Evading intrusion detection and prevention could easily be a book of its own; however, sufficient it to say that what the IPS is looking for might not be what the intruder uses to successfully exploit the vulnerability.