

BROADENING THE SCOPE OF PENETRATION-TESTING TECHNIQUES

“The Top 14 Things Your Ethical Hackers-for-Hire Didn’t Test.”

Ron Gula
Vice President
Intrusion Detection Products
Enterasys Networks
rgula@enterasys.com

Abstract

It has become commonplace for organizations to test the security of their networks with automated vulnerability-analysis tools or by engaging a “tiger team” to simulate hacker network attacks. This exercise is known as a “penetration test.” Too often, such methods are not thorough enough and leave many aspects of a network untested. This paper reviews several network vulnerabilities that are often left untested by penetration testing and discusses possible reasons for their omission.

Paper originally published in 1999 prior to the acquisition of Network Security Wizards by Enterasys Networks.

Table of Contents

Introduction.....	3
Modern Penetration Testing	3
Why Test?	3
Common Tools	3
White, Grey and Black Hat Testing	3
Problems with Penetration Testing	4
How Much Should a Test Cost?	4
Doing It Yourself	4
Us vs. Them	4
The Slippery Slope.....	4
Zero-Day Exploits	5
Fratricide	5
Common Omissions	6
1. DNS Spoofing	6
2. Third-Party Trust	7
3. Custom Trojan Horses	8
4. Database	9
5. Routing Infrastructure	10
6. Testing the IDS.....	11
7. WWW Server Side Includes	11
8. TCP Hijacking	12
9. Testing the Firewall	12
10. ISDN Phone Lines	13
11. Network Brute-Force Testing.....	13
12. Testing non-IP Networks.....	14
13. Ethernet Switch Spoofing	14
14. Exploiting Chat Tools.....	14
Final Thoughts	15
Glossary	16
References.....	17
Contributors	17



Introduction

Penetration testing is one of the most exciting information security fields that an analyst may become involved in. A typical test employs an automated tool to identify and organize vulnerabilities found in a target network. Many tests also attempt to exploit subsets of the vulnerabilities found in order to demonstrate how unauthorized access could be achieved. The problem is that a penetration test is really testing the network as it is on a particular day and the same test performed a week later can have dramatically different results. What follows is a short description of modern penetration techniques, problems with penetration testing in general, and the fourteen most common vulnerabilities that are overlooked.

Modern Penetration Testing

Why Test?

Organizations choose to conduct penetration testing for a number of reasons. The most common reason is to assess the amount of vulnerable systems that they own. Such tests are reduced to a simple percentage of systems that can be compromised. This percentage is tracked over time to track trends in a network's security posture. Another reason for testing may be driven by marketing. Many Internet Service Providers (ISPs) hire third-party security consultants to conduct penetration tests so that the results can be shown to current and prospective customers. Using a third party provides a sense of objectivity to the assessment. Other organizations may conduct penetration testing as the first step in a network security overhaul. The raw data from a penetration test is used to drive network changes and upgrades, which enhances security.

Common Tools

There is a wide variety of network tools that may be used in any penetration test. These tools may include mundane programs such as PING, NSLOOKUP and even web browsers. However, most serious penetration tests make use of at least one automated vulnerability analysis tool. Typically, these tools map out a target network and interrogate identified systems for available services. Information from the available services (such as banners) is compared against a database of vulnerabilities. Some tools even attempt to exploit identified vulnerabilities in an attempt to eliminate false positives. A false positive occurs when a tool identifies a vulnerability in a tested system that does not have the vulnerability.

The two most common commercial tools used for this type of testing are ISS Scanner and Cybercop. Cybercop is produced by Network Associates and was formerly the product known as Ballista produced by Secure Networks. ISS Scanner is obviously produced by Internet Security Systems. ISS Scanner is produced for the Windows NT platform while Cybercop is available for NT and Linux. Both tools include a large number of vulnerability tests. Neither tool allows the user to write custom attacks. However, Cybercop does include the CASL tool, which can be used to generate custom attack packets. These examples have considered IP networks, but there are many other types of automated testing tools that test large numbers of phone lines, IPX networks and mainframe security.

White, Grey and Black Hat Testing

Penetration tests can be loosely grouped into two categories based on a target network's knowledge of the test. A black-hat penetration test is only known by a small number of network operators. The test is designed to find vulnerabilities, but to ultimately engage the humans in the loop to see how they react. White-hat testing uses the full cooperation of a target network. For example the test may include employee interviews, insider network access, physical facility inspections and security policy review. Grey-hat testing combines various features of white- and black-hat testing into a custom test plan. Some security consultants have a slightly different model for white-hat and black-hat testing. In these cases, white-hat testing makes use of automated commercial tools and black-hat testing makes use of manual hacker tools.

Problems with Penetration Testing

How Much Should a Test Cost?

It may be hard to believe, but selecting a third party to test your networks is not an easy task. If you find someone and they are very good, then they will probably be very busy. The better someone's reputation is, the more you can expect to pay for their services. When dealing with consultants, ask for four things. One, ask for references. Two, ask to talk directly with the person(s) doing the testing. Three, ask the testers what methods they use to limit unintentional damage to your network and what they do with the data when they are finished testing. And four, make sure to get a signed NDA between you and the testing company. After you have received a quote for the security testing, obtain a second or third quote from different types of consultants. You may want to obtain quotes from larger security firms and also from smaller firms that may even be locally based in your area. The bottom line with any penetration test is deciding what discovering a set of vulnerabilities is worth to you.

Doing It Yourself

Authorizing an internal penetration test presents many different problems. First of all, the people selected for the testing must be trustworthy and not easily swayed by emotion. There have been too many horror stories of penetration testers making a beeline straight for the CEO's e-mail account. Most CEO's don't take kindly to the intrusion, regardless of any vulnerabilities discovered. Any penetration team must be mature enough to weigh the impact of local politics when conducting the test. During the test, sensitive information about people, projects and many other facets of an organization may be revealed. This information must be handled discreetly and maturely such that vulnerabilities are discovered without invading the privacy of individuals. It is also recommended that the senior manager of any network operations or MIS group be informed of the testing. This can avoid resource-wasting investigations into hardware failures and other network interruptions that may have been caused by the penetration testing. It can also limit overreaction by employees who may discover evidence of the penetration attempt.

Us vs. Them

If network operators have any knowledge of the penetration test before it happens, there is a high chance that they will take extra steps to secure the network. This may represent a higher state of security than what is normally available. There have been some instances of network administrators trying to secure the network even as the testing was progressing.

The human factor may also become involved in penetration testing. It is quite conceivable for an administrator to make claims about the security of a network at staff meetings and to their management. When a penetration test shows vulnerabilities, emotions may become involved. These emotions may over-emphasize the seriousness of a vulnerability or the exact opposite. It is just as likely for a penetration test team member to search for any vulnerability, no matter how small, as it is for an administrator to downplay test results.

The Slippery Slope

Experienced security testers are very capable people, but when they do not find easy ways to compromise a network, one or two courses of actions ensue. First, the testers claim that with more time, they might have been able to break in. In this case, the tester should be able to demonstrate the data to support those claims. If the data is promising, you may wish to authorize a second test.

Second, if a compromise has not been achieved, the tester will point out possible denial of service attacks, sensitive information that should not be on public web sites and many other things that don't directly affect the security of the network. The seriousness of these suggestions should be considered, but they are usually indications that the testers were not able to break into the test network.

Zero-Day Exploits

Experienced network penetration testers should have access to the latest vulnerability and exploit information. Since they test networks for a living, they have the resources to attempt new attacks, monitor information security sources and even develop their own tools. Many of these brand new attacks are christened “zero-day” exploits. Many network administrators do not have the resources to keep up with these vulnerabilities. If a penetration tester breaks into a network with a zero-day exploit, it only proves that the tester has access to vulnerability information that the network administrator does not have. In this case, the results of the test should be considered in terms of realistic expectations. For example, breaking in with a zero-day exploit may demonstrate that a networked organization needs to have their resources augmented with extra staff and possibly extra equipment.

Understanding the nature of zero-day exploits is another way to understand the nature of penetration testing. If the testers break into a test network by guessing passwords or using common (older) system exploits, then this may indicate that the network has not been secured. Testers may complain that there were too many ways to break in, and that they didn't need to break out the new exploits. On the other hand, if the testers had to work three days straight to find a way to make the latest vulnerability work against a test system, this indicates a network that is much more robust. In both examples, each network was vulnerable, but the degree of difficulty to accomplish the break-in was much different.

Fratricide

It happens more often than one would think, but there have been many cases of penetration tests launching attacks against networks that were not authorized for testing. Sometimes this occurs when the test is starting and a tester incorrectly enters a network address into an automated scanner. Other times it occurs during the heat of testing when trust relationships are being exploited. Fratricide affects the results of penetration testing primarily by delaying the execution of the test.

Common Omissions

The following is a list of the top fourteen things that normally go untested as considered by the author. Of course, some of these areas are tested by various penetration testers, but most of them are not. This list is meant to illustrate the broad range of network attacks that are available to a hacker. It does not consider physical attacks or social engineering. The list is also not in any particular order, as many of these attacks are equally as likely to succeed.

1. DNS Spoofing

The DNS system is used to convert IP addresses to domain names and vice versa. The protocol has no authentication and hosts “believe” anything they receive that looks like a correct answer. Here are some techniques that can be used to modify how the DNS system works:

A. Break into the target network DNS server

Many versions of BIND have buffer overflows that can result in direct compromises. Exploits for Linux DNS systems are easily executed and widely available. Exploits for more obscure platforms such as HP-UX and x86-based Solaris have also been demonstrated. It may also be possible to break into the machine via a different vulnerable service on the same system. Once a DNS system is “owned,” it is trivial to affect the information given out.

B. Spoof DNS responses

If an attacker is in the position of observing DNS queries and responses, then they can trivially spoof bogus answers. These bogus answers will be believed by other DNS servers or the client computer making the query. Common “positions” used to observe DNS queries are other systems on the target network, systems outside the network firewall, and systems on the same networks as queried DNS servers.

C. DNS Cache Poisoning [1]

Many DNS servers can cache the information they process for a finite amount of time. This speeds up DNS resolution with the theory that if someone asked a question once, they will probably ask it again soon. There are many techniques available to poison a DNS cache. The Cybercop scanner tests for some of these.

So what can spoofed DNS information do for a penetration tester you ask? Traditionally, penetration testers have used spoofed DNS information to exploit UNIX systems via the rlogin and NFS services. The rlogin service has a variety of trust mechanisms, some of which depend on DNS. Many NFS servers would also export file systems to machines that had a matching DNS name. These techniques are still valid, but have fallen out of practice with many penetration testers because of the decline in the use of rlogin and NFS. Here are some DNS spoofing penetration testing techniques that can be used in modern network environments:

A. Redirect web browsers to you

There are many vulnerabilities in the Netscape and Internet Explorer browsers. Using DNS spoofing to force a browser to come to your WWW server instead of the real site can demonstrate the ability to attack web clients. Spoofing DNS names for common start-up pages such as Netscape’s and Microsoft’s home pages may net browsers as they are initially started. Also, spoofing names of likely browsed web sites such as Internet search engines is another technique. One could imagine a sophisticated web server used for this sort of testing that would recognize the browser in use and then automatically serve it the appropriate Java or ActiveX vulnerability.

B. Make logins come to you

There are many clear-text protocols still in use today. Telnet is still alive and well. FTP and POP e-mail both use clear-text protocols. Creating a server to emulate any of these services is trivial. Combining that and some DNS spoofing can cause “normal” traffic to come to your fake servers where the usernames and passwords can be obtained. For example, consider a web server that is a clone of Microsoft’s Hotmail server (www.hotmail.com) in that its splash page looks identical. By spoofing DNS, a penetration tester may be able to force a Hotmail user to unwittingly divulge his or her password. The facade may become even more believable, if the server automatically forwarded the data to the real Hotmail server.

C. Make network traffic come to you

When spoofing DNS, it is also possible to redirect a variety of other network traffic. This traffic may be of interest to a penetration tester. Some examples include SMTP messages and SNMP queries. An SMTP message may contain sensitive corporate data and can be easily forwarded onto the proper destination. SNMP queries are also useful because they contain the SNMP community string.

There are many reasons that DNS spoofing is not attacked during penetration tests. It is very common for penetration testers to be unwilling to negatively impact a target network's normal operations in any way. Manipulating DNS can cause outages and extra work for network administrators. For example, some DNS overflows are "one shot" in that they either work or crash the server. Another reason that DNS is seldom tested is because the cure is more painful than the vulnerability. For example, not allowing POP e-mail or web browsing by employees from their desks is usually frowned upon, even though it is a huge security risk.

2. Third-Party Trust

There are many forms of third-party trust. Third-party trust includes any network relationship where one party is granted some sort of privileged access based solely on who they are. Here are some examples:

A. Allowing certain IP addresses/ISPs through a firewall

Many networked organizations allow remote access to their networks, and this is usually accomplished via a firewall rule. The authentication is based totally on the source IP address of the network traffic. Some companies have been observed to allow entire ranges of IP addresses from known ISPs because of the lack of fixed IP addresses.

B. Trusting network information and services from an ISP

We've already talked about DNS, but there are many other types of network information that an ISP can provide to a customer. Routing information is crucial for proper network connectivity. If the network is complex enough, BGP may even be used. Some ISPs provide monitoring of customer equipment. This monitoring may be in the form of SNMP queries, Syslog monitoring and even Telnet/Shell access. And finally, an ISP may actually be hosting a customer server. In these cases, the ISP owns the hardware and operating system, while the customer owns the data.

C. Virtual private networks

A virtual private network (VPN) is a secure communications protocol that encrypts traffic between two endpoints. At each endpoint, the traffic is decrypted. Traffic can consist of one-to-one, one-to-many and many-to-many conversations. A typical application for a company would be to place a VPN at a satellite location. All traffic between the location, other satellite locations and corporate headquarters is automatically encrypted.

Each of the above examples is now examined for possible attacks and why those attacks usually aren't tested during a penetration test. We also offer some real-world recommendations to compensate for this.

A. Allowing certain IP addresses/ISPs through a firewall

The attack is obvious and an old one. Simply break into one of the trusted set of IP addresses and then use that to access the target network. IP spoofing is a valid attack, but let's assume that the penetration team is not in a position to exploit that technique.

As simple as it sounds, in practice, this is almost never tested. Why? Many times the target network does not own any assets of the trusted network. For example, if a target network allowed access through the firewall from a range of AOL IP addresses, the testers may not know that. If they did, they could easily obtain access to AOL at a small cost, but then AOL may observe the security testing and become suspicious. In other cases, accounts to the trusted network cannot be easily obtained. Consider the numerous military and government organizations that trust two or three other organizations. In the commercial world, many companies have these third-party relationships with other companies.

To accurately test this vulnerability, two things must be accomplished. First, the trust relationship must be verified as a threat. To complete this, invoke what the NSA calls the "Infosec Advantage." This simply means to work with the security testers and give them as much information as possible. Together, all of the trust

relationships can be accurately determined. Second, the testing team needs to simulate the trusted access. Options include obtaining permission for the testers to break into the trusted site, giving an account on the trusted site to the testers and also opening up the firewall to allow the testers into the target network as if they were coming from the trusted network.

B. Trusting network information and services from an ISP

We've said before that an ISP provides a great deal of information to its client networks. Many times modification of this information can be used to obtain access to a target network. Modifying DNS, SMTP and routing traffic can be a great asset to a penetration testing team. For example, if all e-mail could be redirected for scanning of keywords, this could contain a lot of sensitive information. If routing could be alerted, traffic could be monitored. Traffic could also be monitored if the ISP were compromised.

Obviously, most ISPs would not want to be targeted by a penetration test of one of their customers. The network security status varies from ISP to ISP. Some have very strong security, while others can have their NOCs compromised relatively easily. ISPs do not want anything to interrupt their operations, because they have "service level agreements" (SLAs) that they must meet with their customers. Any interruption could cost the ISP money and resources. Similarly, most target networks also wish to avoid network outages.

To simulate this vulnerability, one option that a penetration team can exercise is to deploy a computer system on the target network's perimeter. Ideally, this would be placed just outside the firewall or border router. The team should first concentrate on obtaining as much passive network data as possible. On modern networks, it is still common to find sensitive information in clear text. Some of the more common clear text information obtained includes web username and passwords. If passive analysis of the network data does not yield fruit, then an active approach may be necessary to demonstrate an ISP's impact on security. From the monitoring host's vantage point, DNS spoofing, TCP hijacking and IP spoofing should each be readily accomplished. More complex attacks such as altering network routes should be accomplished with the cooperation of the network administrators of the target and possibly the ISP.

C. Virtual private networks

There are two basic attacks that can be performed against a VPN. The first attack is to defeat the encryption used to protect data. If the encryption can be trivially broken, then all sorts of passive network monitoring attacks can be executed. The VPN encryption may also be able to be broken using an active technique. For some VPNs, such as Microsoft's PPTP, passive analysis and active packet spoofing can result in access to the VPN. See [2] for more information. The second technique is to directly attack a network, which is behind a VPN. Theoretically, access to such a network will result in access to the entire network protected by the VPN. Several VPN devices also allow for firewall rules to be placed on the VPN link, but these rules are usually less secure than if they were protecting access from the Internet.

Many times, a tiger team does not test the VPN because they do not know about it. Also, if it is known, then the client software to test connectivity with the VPN may also not be readily available. Sometimes VPNs are used with trusted networks and many of the topics discussed in that section also apply. Some advanced attacks against VPNs that exchange keys in the clear may also be accomplished if the tiger team can execute man-in-the-middle attacks. However, unless the VPN was set up incorrectly, the most likely path of attack is to find a network node of the VPN, which can be compromised.

Simulating this attack is difficult. Many times, an independent review of the VPN's topology and implementation by a third party may be in order, rather than relying on the tiger team. A tiger team should still use a network analyzer to capture and analyze network traffic that is supposed to be encrypted.

3. Custom Trojan Horses

There are several types of Trojan Horse programs. These range from binary programs for a variety of platforms to documents that contain macros. They all contain code that most end users are not aware of. Traditionally, such code has been destructive in nature. Viruses and logic bombs have cost countless hours of productivity for many network administrators. However, when this custom code contains backdoors that bypass network security mechanisms, the security of an entire organization can be compromised.

Recently, there has been a surge of Microsoft Trojan Horse programs such as Back Orifice, Back Orifice 2000, NetBus, NetBus Pro and many others. These “programs” can be surreptitiously placed on a target computer. Once there, full control of the target computer is usually obtained through a remote client. The backdoor can be quickly placed with physical access or even hidden in a “trusted” application. These “trusted” applications can include executable e-mail attachments, programs downloaded from the Internet and even demo programs delivered on physical media.

Microsoft platforms have also recently had to endure a barrage of macro viruses that effect products such as Microsoft Word and Excel. Many of these malicious macros have been disruptive in nature, but the macro languages are sophisticated enough to download programs and execute them. Some macros are sophisticated enough to actually become the Trojan server. There are many other data-processing applications that are vulnerable to Trojan Horse macros.

For penetration testing purposes, these programs are hardly used because of time and reliability. Time is a factor because it may be very difficult to identify a vulnerable target. Imagine the highly paid ethical hacker using her network probes to identify Windows 95 machines. She will have a tough time targeting those specific machines with a Trojan program or macro. There may be some e-mail information that could be used to blindly send to an unsuspecting target employee, but that is not precise. Social engineering could be used to identify specific people or maybe even send them some fake software updates on physical media. But once more, this takes time. The second issue is reliability. There is no guarantee that sending an e-mail to someone actually targets the customer’s computers. For example, many Internet users forward multiple e-mail addresses to a central location. Imagine a user who is working from home receiving an e-mail containing a Trojan horse from an ethical hacker. Sometimes Trojan Horse programs are not widely tested and have unanticipated effects on target networks.

There have been some cases of ethical hackers testing a target network with this technique. A common ploy is to find as many company e-mail accounts as possible and simultaneously send them forged, “Spam” e-mail containing an executable attachment. The theory is that someone will run the attachment. If the attachment is a client-server application, then the tester may still have to bypass a firewall. If the attachment is more sophisticated, it may try to automatically connect to the ethical hacker’s computer to receive instructions.

In many cases, the solution to this type of vulnerability is employee education, deployment of malicious code detection software, a PKI infrastructure to verify e-mail transactions and possibly even changes in basic Internet access. Since these changes are encompassing, it is recommended that the CEO, CIO, or the appropriate decision maker be given a full demonstration of the vulnerability. This demonstration should not be canned. It should contain several demonstrations on the target network that may involve spoofed e-mails, DNS spoofing, custom Trojan Horses, common Trojan Horses and in some cases, extraction of company data via a Trojan Horse. The important thing for the ethical hacker to consider is to convince the CEO or CIO of the vulnerability (if there is one) without negatively impacting the target network’s operation.

4. Database

Many networks have had one or more databases on them for some time now. Network users access the databases with custom applications. Today, it is very common for web-based “front ends” to be written that automatically query databases for information. By and large, there are very few exploits or “zero-day” hacks that allow unauthorized users to access database resources. Regardless, the database is often not directly targeted during an ethical hacking test.

There is one type of database test that is gaining popularity. During web-server testing, it may be possible for an ethical hacker to obtain the source code to a “server” program. Typically, these are CGI-BIN and Active Server Pages programs. Very often, if these programs need to exchange information with a database, then they may have authentication information embedded in their source code. For example, a Windows NT IIS web server may make network queries to a Microsoft SQL server from an ASP application. If the source code of the ASP contains usernames and passwords to the SQL server, then the SQL server may be directly accessed with a generic SQL client.

Most ethical hackers do not attempt to test database servers because they are not familiar with them. In general, network security professionals do not ascend from the ranks of database administrators. There are some, but most

security analysts haven't even set up an SQL database. Of all the types of security experts that are out there, those that concentrate on NT networks seem to have a better grasp of databases such as Microsoft SQL.

ISS has recently released a security product that scans database implementations. This is an excellent trend and a one-of-a-kind product. Databases have their own complexity, and many implementers make common configuration mistakes that effect security. For starters, someone out there must be teaching Oracle DBAs to add an 'oracle' user account with a password of "oracle."

It is recommended that an ethical hacker become familiar with at least one type of database technology. For NT systems, Microsoft's SQL server is commonly deployed. For Unix networks, there are a variety of different SQL technologies. MySQL is a free version of SQL that many universities and small ISPs are using to track data. Larger organizations tend to use Oracle databases, which is available for both NT and Unix platforms. A knowledge of database operation is required to find vulnerabilities and to recommend security fixes.

It's also interesting to note that the database is where a majority of the most sensitive information is stored.

5. Routing Infrastructure

There are two ways that a routing infrastructure can generally be attacked. The first is to alter the routing logic through spoofing of the native routing protocol. The second is to compromise a network node that is making routing decisions and then directly change routing information. Both of these methods are hardly used in ethical hacking tests.

Attacking routers has a high chance of negatively affecting a target network. If a mistake occurs, there is a high chance that the target network will experience a network outage. For commercial consulting scenarios, this is unacceptable. For most hackers, it is also unacceptable because the cause of the outage will be investigated.

The skill of the ethical hackers is also a reason why routing infrastructure may not be detected. A majority of the hacking exploits that exist today are for Unix and NT platforms. It is very unlikely for these platforms to be routing packets although it does occur. More likely, Cisco routers are used to move packets across a network. Except for trying some default usernames and passwords, there are few direct attacks against a particular router. SNMP control and web-based administrator access have been some recent paths of attack, but many times that access cannot be used to dramatically alter how traffic is flowing.

The topology used in a test can also influence an ethical hacker's attack path. For example, consider a small office that owns a class C network block and only has one router. There really isn't a complex topology to target. If the network is protected by a firewall and it is configured correctly, there are very few attack points for a typical "scan and cash" ethical hacker probe. The topology that a hacker would exploit in this case is the network of routers that feeds the target network. Most likely, these routers are owned by an ISP and not by the target network. They are off limits to most ethical hacker tests. A hacker would want to control those first few routes in order to sniff traffic, spoof DNS, hijack network connections and many other techniques to leverage access.

Protocol spoofing can be used to alter almost every major routing protocol. RIP and OSPF have many attacks that can be effected simply by injecting fictitious route information in the form of spoofed packets. The general hacker community has provided some publicly available RIP tools, while OSPF tools remain the property of sophisticated ethical hackers and high-end hackers. Spoofing ARP packets in order to overcome Layer 2 security partitioning is another technique to overcome topology segmentation.

In many routing products, the protocols involved include a mode that encrypts all routing information. This makes it very difficult for anyone else to spoof routing traffic. However, many network engineers do not enable this feature because it has a negative performance impact. Demonstrating illegal control of routing is necessary to convince network administrators who believe it is not possible.

Hacking routers is a not a common art. Typically, most security analysts are comfortable configuring a small Cisco router, but when making changes to a high-end router with multiple OC-3 links, most ethical hackers have not had the chance to do this. One way to combat this is for an ethical-hacker team to hire a network engineer for their expertise. Another way to overcome this is to use a sympathetic network engineer from the target network.

Typically, ethical hackers have gone after three types of routing access. First, if the Cisco configuration file is obtainable, some password information may be obtained through decryption. I've heard of some organizations that

publish their Cisco configurations on the Internet. This is almost as bad as publishing non-shadowed password files. Second, hackers will target SNMP in the hopes of obtaining write access. Many networks still use the SNMP v1 network management protocol. That version has no encryption and uses a clear text string called the “community string” to control access. If the community string can be guessed or obtained, then in some cases, control of the router can be accomplished. The last thing that hackers traditionally try is to sniff password information and in some cases, attempt to hijack router telnet sessions. There are other attacks, such as searching for out-of-band modems used to administer the routers.

Lately, there has been an increase in the amount of attacks used to gain access to Cisco routers. It seems that source code to the Cisco IOS operating system is widely available in the hacker community and it is being searched for vulnerabilities. It is difficult for many ethical hackers to obtain IOS source code legally. However, any well-paid ethical hacker should possess the skills necessary to reverse engineer a CERT or vendor security advisory to result in a usable exploit.

6. Testing the IDS

Many networks run some sort of network- or host-based intrusion detection system. The ethical hackers or tiger team should specifically attempt to identify the IDS and attempt to launch attacks that bypass it. There are several types of IDS testing that can occur. These range from simply launching attacks and seeing if anyone is watching the IDS, to crafting specific attacks that go undetected.

Identifying the IDS may be easier than most people think. One would be surprised how many people name things “ids.company.com” or even “nfr.school.edu” which makes identification of the IDS platform trivial through DNS. Of course, the DNS information could be a deception, but in many cases it isn't. Many organizations also make claims about their security on their web pages. ISPs, universities and government organizations are notorious for this. If someone says they are using ISS RealSecure to protect their network, they probably are. Other more sophisticated identification techniques include watching for SNMP traps from the IDS sensors, scanning for particular open TCP and UDP ports, and even using the same IDS client in an attempt to connect to an IDS central console.

When attacks are launched, it is very important to record exactly which attack was launched from where and when. This log should be compared to what the target network is aware of. For instance, a slow ping sweep may not be detected at first while a brute force HTTP password attack generates all sorts of alerts and logs. By choosing low intensity attacks first and then gradually increasing their verbosity, a penetration test can find out how sensitive a network is to probes and attacks.

If the IDS is known, then technical attacks that bypass the IDS should be attempted. The many techniques used to bypass an IDS are a subject of much debate. However, there is nothing like a real-world test. If an ethical-hacker team wants to modify their attacks to avoid detection, then they should be allowed to do so. Several tools such as Fragrouter [3] can be used to convert a variety of attacks into packet streams that will be incorrectly reassembled by most packet-based IDS systems. Fancy attacks aside, the latest zero-day exploits may also yield low chances of detection.

All in all, if the IDS is to be tested, its logs should be compared with the actual attack list and sequence in an after-test meeting. There may be lots of finger pointing at this meeting which results more from politics than technology. In any case, the following questions should be answered. One, at what point was the target network aware that they were being probed and/or under attack? Two, how much advanced notice did the target network have of the impending security test? Three, what steps would the target network have taken to contain the ethical hackers? There are probably more questions, but if these are outlined up front, there is less room for hurt pride and embarrassment for all involved parties. If possible, a mediator should run the meeting.

7. WWW Server Side Includes

Many complex web servers use some sort of server side include (SSI) to “keep state.” This allows a web server to recognize a previous visitor and maintain the illusion of a session. For example, with the use of an SSI, a web server may be able to recognize a web query from a recent user. This may allow the web user to custom generate HTML code for the particular user. Unfortunately, sometimes the SSI feature is used for security purposes. By spoofing the SSI, a web user may be able to access other “sessions,” which may contain sensitive information.

During a penetration test, an ethical hacker will almost never use this exploit. There is a good chance that an exploit of this type may divulge a customer's or user's sensitive data. It may even disrupt an e-commerce transaction.

Typically, if the server is tested, it is tested off line with a white box inspection. Close in examination of CGI-BIN, ASP or even JAVA code can discover blatant security problems which may be more difficult to find remotely on a live server.

8. TCP Hijacking

As common of an attack as TCP hijacking is, it is almost never used in a penetration testing scenario. Typically, the type of sessions that can be hijacked contain enough sensitive information in them such as clear text passwords, that session hijacking isn't needed. However, passwords can only get people so far. Session hijacking is very useful for a number of things that often go untested by an ethical hacker.

One of the most common things that ISPs do to give web page administration access to their customers is to issue SecurID tokens. These tokens use a one-time password to authenticate the user to the FTP server. This makes password sniffing totally useless. However, the FTP protocol may be hijacked. In other words, an attack could wait for an authenticated FTP session to start and then hijack it. Typically, Telnet is not used with SecurID, but is replaced with SSH. For many reasons though, FTP is still in use by most ISPs.

Hijacking is also useful when an IP filter is present. Routers with ACLs, firewall rules and even TCP Wrappers can prevent sessions such as Telnet and FTP from occurring. With TCP hijacking, it is possible to avoid those security mechanisms. It is very common for a router engineer to secure their Telnet access with an ACL that limits login from a few IP addresses. If an attacker can see the Telnet session, she can take it over with a hijacking tool.

Depending on the topology and the protocols involved, hijacking may be a vulnerability that many ethical hackers talk about in their final penetration testing report, but that few actually do during an assessment.

9. Testing the Firewall

Almost every penetration tester uses some sort of port scan to find attack paths into a target network. But it is very rare for an ethical hacker to attempt to identify the firewall, or identify how it is configured. There are many ways to identify a firewall. Once a firewall is known, it may be attacked. Identifying how a firewall is configured is also useful for determining the type of attacks one could try to bypass it.

Typically, ethical hackers who rely on tools such as Nmap, ISS Scanner or CyberCop don't get a good overall picture of how the network topology effects security. These tools simply scan a list of IP addresses for available services and then interrogate them for vulnerabilities. Without a sense of topology, ethical hackers will miss out on alternate attack paths. For example, many network organizations configure a DMZ to place SMTP and HTTP servers on the Internet without exposing the internal network. If a DMZ server could be compromised, it's possible that its firewall rules may not be as restrictive as those directly from the Internet.

There are many ways to identify firewalls. Products like Checkpoint and Interlink have many default ports that can be used for identification. In some cases, a firewall will use the host's IP stack and TCP/IP fingerprinting can be used to find out the operating system. If all traceroute paths seem to go through an NT or Linux machine, then it may be the firewall. And just like the IDS, don't be surprised if you see a "firewall.company.com" address when evaluating the DNS. Discovering Cisco routers or routers in general that are filtering packets is also a possibility.

Discovering how a firewall is configured is also useful to an ethical hacker. Tools such as Firewalk [4] are extremely noisy, but can produce an accurate model of how a router or firewall is configured to drop certain types of traffic. For instance, if high ports (above 1024) are open, then scanning for X-Window servers and high RPC ports may be worth while. If there is enough time, scanning the entire UDP and TCP space may be accomplished, but this is a very noisy and very slow technique. Knowledge of a firewall's configuration can help focus large-port scans.

Along the lines of figuring out how a firewall is configured, attacks to bypass the firewall may also be chosen. Attacks such as fragmentation (good luck on that working), sending in RFC 1918 traffic, encapsulating protocols such as IP and IPX and even source porting can yield a variety of results.

Of course, the best way to test a firewall is to sniff traffic inside and to send traffic from the outside. With this technique, a large amount of traffic can be generated to brute-force test the firewall. Any traffic that leaks through the firewall should be evaluated for its usefulness in an attack.

10. ISDN Phone Lines

Many security consultants offer ethical hacking of a customer's phone lines. There are all sorts of systems that are connected to networks and to a phone line. Out-of-band router and server access is pretty common on many large ISPs. Testing those lines for common passwords and password-less access is becoming very common.

Increasingly, many people are placing systems on ISDN phone lines. There is greater bandwidth and in some cases, greater security. When a non-ISDN war dialer attempts to dial an ISDN phone line with a digital data service over the B-channels, the analog modem war-dialer will not detect the data service. The ISDN signaling (Q.931/Q.2931) will not allow for the successful routing and connection between an analog line and a known ISDN data service.

All that an ethical hacker would need to continue their testing would be an ISDN terminal adapter capable of supporting the ISDN protocol stack and data services. She would then need to write the software to interface with the ISDN equipment, which would scan known ISDN data numbers and attempt to connect. The ISDN technology will automatically negotiate a typical PSTN dial tone. If the connecting phone number is an ISDN line, then the connection won't be disconnected. This sort of testing is necessary in Europe where ISDN is prevalent.

11. Network Brute-Force Testing

Brute-force tests invoke the miracle of computer automation. It is trivial to program a computer to attempt almost any sort of network login and record successful logins. If weak passwords are prevalent in a network organization, then it is more likely that a brute-force attack of this type will be successful. The key to a successful brute-force attack is to select a target that has a high degree of success and a small chance of being logged.

What type of targets have a high degree of success? There are two types. The first is to exploit the public. Anytime that a large number of users is involved, there is a high chance of discovering a few users that have chosen bad passwords. If a user list is obtainable, this attack is very easy to execute. Simply using a small common dictionary of passwords against each username will result in a few successful logins. If a user list is not available, the attack takes exponentially longer because the users must be brute forced also. The second type of exploit is to search for common username and password pairs that are typical to large network organizations. Some hackers pride themselves on the password and username pairs they've run across through the years. For example, using a username of "oracle" and a password of "oracle" may grant a successful login in many more places than one would expect.

Another notion to consider is the amount of time between login attempts. If the amount of time is more than a second, then it may be impossible to make large numbers of attempts in a small timeframe. The typical three tries and then you're out rule can make brute forcing difficult, as a new session is required. If this session is over a modem, then the dialing, answering and modem negotiation can take an excessively long time.

Historically, the POP and REXEC services have been targets of brute force attempts simply because they did not have their login failures logged. Today, various SSH servers can be used to attempt brute force password guessing without failures being logged. REXEC (thanks to Tivoli and HP Openview installations) is also becoming very common at large ISPs.

To give them credit, ethical hackers tend to avoid brute-force attacks in favor of less noisy methods. However, when most attacks during a penetration test fail, the brute-force attack method is seldom used. Sometimes the attack does not even need to be a direct compromise. Some ethical hackers search for the finger service and then use a dictionary of common usernames to build user lists. A similar technique can be used on many MTAs with the "VRFY" command.

One thing that ethical hackers are trying today is to brute force SNMP community strings. There are two problems with this technique. First, for each failure, most SNMP devices will send an error message to a central SNMP server. Large numbers of these can easily be detected. Second, the attacks themselves are inaccurate because of the UDP protocol. It is trivial to send in thousands of SNMP queries a second in as many UDP packets, but it is difficult to tell exactly how many are being received by the target. For large networks, it is better to attempt the brute-force attack over the entire network rather than bombarding one SNMP node.

12. Testing non-IP Networks

There are many other network protocols than the Internet Protocol (IP). IPX, SNA and NetBEUI are still quite popular and usually operate on the same network segments as the IP networks. Most ethical hackers tend to concentrate in the TCP/IP and Unix world. Mainframes, Novel networks and some Microsoft network technologies have been around for quite some time. There are many legacy networks that have been retrofitted to run IP over them.

This results in a situation where an ethical hacker may be able to compromise a target via an IP pathway and then load up a set of tools that will probe and attack a non-IP network. One of the best examples of this type of tool is Simple Nomad's IPX exploits for the Linux operating system [5]. With tools like this, non-IP networks may be attacked by ethical hackers.

Another avenue of possible exploitation is from encapsulating non-IP packets in IP packets. Many devices understand the IPInIP or GRE protocols. Both protocols allow for an IP packet to carry other IP packets from point to point. The same technique can be used to carry IPX, SNA and many other protocols. Scanning tools such as CyberCop actually have checks for these types of packets.

Typically, ethical hackers do not test these other topologies because they do not have knowledge that they exist in the first place. Using a samba client over IP is not the same as scanning for Windows 95 and Windows NT servers that do not have IP stacks. In order to simulate and find these vulnerabilities, ethical hackers should sniff the network for non-IP traffic once access is obtained. If access is not obtained, sending in IPX or NetBEUI packets encapsulated in IP packets to many network devices is a valid scanning technique. Analysis of any returned traffic may be able to shed light on the protocols and possible vulnerabilities associated with the target network.

13. Ethernet Switch Spoofing

Many networked organizations claim enhanced security by using Layer two switching. This technology, also called VLAN, only sends traffic to a computer that is destined to that computer. The motivator for this is performance. Older technologies would broadcast each Ethernet packet to all connected computers. Any computer could have been running in "promiscuous" network mode where all packets were recorded. Such attacks were able to divulge passwords and other sensitive data. The Ethernet switch added a certain amount of security by only sending traffic to its proper destination. All connected computers did not receive all traffic. They only received broadcast traffic and traffic that was sent to them. VLAN technology actually allows complex switches to establish "virtual" LANs. These VLANs can simulate small broadcast networks or small switched networks.

The Achilles Heal of this technology is in how these switches handle broadcast traffic. A typical attack is to use a program that sends fake ARP requests and replies. Typically, these switches keep tables of IP addresses and Ethernet addresses. By sending in Ethernet packets with a broadcast source address, the switch may think that some or all IP addresses actually broadcast Ethernet addresses. This causes some switches to broadcast all IP traffic to all listening devices. Some TCP hijacking tools such as "hunt" [6] actually use these techniques to defeat some Layer two switches.

Ethical hackers tend not to test these switches because of their unpredictability. Unless an ethical hacker has had experience with a Cisco 5500 or comparable switch, she may not be as willing to try some newer techniques on a customer's network. The best way to test these vulnerabilities is with the cooperation of the local network engineer. It will fall on them to fix the problem if there is one, so getting them involved from the start is a good thing. In many cases, there are configuration changes that can be made to the switch to avoid the Layer 2 spoofing.

14. Exploiting Chat Tools

Many large organizations have users that use a variety of chat tools. Traditionally, Internet Relay Chat (IRC) has been the most common chat tool, but new products such as ICQ and AOL's Internet Message tool are competing heavily in their ease of use and popularity.

Many clients who connect to chat rooms face security vulnerabilities. Ethical hackers may wish to target individuals from a network organization by searching for users from that domain in various chat rooms. Most of the chat rooms have user search features that allow on-line users to be located quickly. Also, sometimes the information may reveal IP addresses and username information.

Ethical hackers usually do not attempt to attack target networks this way. The notion of directly attacking a target network through a third party is questionable. On the legal side, there is also a big problem with possibly attacking the wrong target. Protocols such as DCC may allow two users to talk directly with each other and they may also act as new attack paths. Tools such as Cybercop actually have features that check for online IRC users and can be configured to launch attacks against their chat clients.

Another technique that is often overlooked by ethical hackers is to use social engineering. Finding someone online from a target network can provide a variety of “free” data. Getting someone’s e-mail address, a few co-worker names or even their operating system may be very useful during a penetration test. This information gathering is often very slow and does not have any guarantees of success.

Final Thoughts

There is much more to penetration testing than running a few tools and producing a report. For as many vulnerabilities that are checked by those testing tools, there are as many additional techniques that are available to an ethical hacker for finding vulnerabilities. While ethical hackers are usually bound by time, legal permission and experience, they have an obligation to provide as realistic of an assessment as possible.

To get the most bang for the buck, don’t forget about your “Infosec Advantage.” As soon as the sneaky testing is over, walk on over to those target servers and get a manual inspection of them. Usually, you will be surprised by what you find.

Glossary

AOL	America Online
ARP	Address Resolution Protocol
BALLISTA	See Cybercop
BIND	A DNS server program
CEO	Chief Executive Officer
CIO	Chief Information Officer
CYBERCOP	Network Associate's automated vulnerability testing tool (www.nai.com)
DCC	Data Communication Channel
DBA	Database Administrator
DMZ	Demilitarized Zone
DNS	Domain Name System
GRE	General Routing Encapsulation
HP-UX	Hewlett Packard's version of UNIX
ICQ	A more recent chat tool. See http://www.icq.net
IIS	Internet Information Server (Windows NT Web/FTP server)
IPX	A protocol commonly used on Novell office networks
IRC	Internet Relay Chat
ISP	Internet Service Providers (MCI, PSI, AOL, etc.)
LINUX	A "free" version of Unix that is very common
MIS	Management Information Systems (folks who run office networks)
MTA	Mail Transfer Agents
NDA	Non-Disclosure Agreement
NFS	Network File System (share files and directories on UNIX networks)
NOC	Network Operations Center
NSLOOKUP	A tool to convert between domain names and network addresses
OC-3	A type of fiber-optic data channel
PING	A diagnostic tool that is used to test if hosts are alive
PKI	Public Key Infrastructure
POP	Post Office Protocol
PSTN	Public Service Telephone Network
REXEC	Remote execution
SQL	Structured Query Language
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSI	Server Side Include
VLAN	Virtual Local Area Network
VPN	Virtual Private Network

References

- [1] The ADM Crew, "DNS ID Hacking"
<http://packetstorm.harvard.edu/ADM/ADM-DNS-SPOOF/ADMID.txt>
- [2] Bruce Schneier and Dr. Mudge, "Cryptanalysis of Microsoft's PPTP Authentication Extensions,"
Counterpane Systems and L0pht Heavy Industries
- [3] Fragrouter, a program available with Anzen's (www.anzen.com) network intrusion detection benchmarking software (nidsbench)
- [4] David Goldsmith and Michael Schiffman, "Firewalking: A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists," Cambridge Technology Partners
- [5] Simple Nomad's Pandora IPX/Novell security testing software, available at
<http://www.nmrc.org/pandora/index.html>
- [6] Pavel Krauz's Hunt TCP/IP hijacking tool, available at <http://www.cri.cz/kra/index.html#HUNT>

Contributors

Jonathan Hackmann, International Network Services, hackma_j@ins.com

Chris Scott, CScott8989@aol.com

J. John Skordas, Jr., GTE Internetworking, jskordas@bbn.com

North America

35 Industrial Way
Rochester, NH 03867
U.S.A.
(603) 337-1600

50 Minuteman Road
Andover, MA 01810
U.S.A.
(978) 684-1000

Europe/Middle East/Africa

Network House
Newbury Business Park
London Road, Newbury
Berkshire, England RG13 2PZ
44-1635-580000

Asia Pacific

85 Science Park Drive
#03-01/04
The Cavendish
Singapore 118259
65-775-5355

Unit 10, 14A Rodborough Road
Beacon Business Park
Frenchs Forest NSW 2086
Australia
61-29950-5900

Latin America

Periferico Sur No. 3642
Piso 6
Colonia Jardines del Pedregal
Mexico City DF
Deleg. Alvaro Obregon
C.P. 01900
Mexico
525-490-3400

Av Nações Unidas, 12.551,
18° Floor
Brooklin-São Paulo
04578-903-Brazil
55-11-5508-4600

Copyright © 2001 Enterasys Networks, Inc. All rights reserved. NOTE: Enterasys Networks, Inc. reserves the right to change specifications without notice. Please contact your representative to confirm current specifications.