Global Information Assurance Certification Paper

A Management Guide to Penetration Testing

David A. Shinberg

SANS Hacker Techniques, Exploits, and Incident Handling (GCIH)

Practical Assignment

Version 2.1a

Abstract

Penetration tests are an excellent method for determining the strengths and weaknesses of a network consisting of computers and network devices. However, the process of performing a penetration test is complex, and without care can have disastrous effects on the systems being tested. This paper provides guidance, primarily focused around planning and management, on how to conduct a penetration test comprised of five phases – Preparation, Public Information, Planning, Execution and Analysis and Reporting. However, due to the technical and sometimes sensitive nature of penetration testing only a cursory overview how to compromise a system is provided.[1]

1. Introduction

Penetration tests are used to evaluate the security of computer systems. A methodical approach is required to maintain both the integrity of the results and the stability of the systems being tested. The following describes what the reader will find in the remaining sections of this paper.

- Preparation: The first part of the paper will be devoted to preparing for the penetration test. This includes obtaining appropriate approval to perform the test. A short discussion of an appropriate tool suite for performing the penetration test will also be discussed. The paper will not discuss how to install the tools, as people performing penetration test should have the skills required to install the necessary tools.

- Public Information: The second part of the paper will discuss what can be done before sending probe packets to the network. This includes accessing public whois information, and normal web access to obtain company information. No scanning for vulnerabilities or active network probes (e.g., traceroutes) will be performed. Depending on the goals of the penetration test, this is the appropriate time to involve the network administrators and security staff who can provide some useful details about the network. However, this information must be verified during the testing and not consider to be true a priori.

- Planning: Based on the publicly available information, the next step is to produce a detailed plan for performing the penetration test. At this point, specific tools will be chosen based on what was learned from the public information. This section will also discuss issues related to the network and host impact of the penetration test. One point to consider is should an IDS detect the penetration test.

---

[1] Prior approval for this topic was obtained from certify@sans.org because the topic is not one of the specified topics.

- Execution: This part of the paper will discuss how to run the tools to perform the penetration test. Part of this section will include the necessity to modify the plan, as more information becomes known. Good record keeping as opposed to analysis will be the focus of this section.

- Analysis and Reporting: The final section of the paper will present an analysis of the results obtained from the various tools. It will include suggested items that should be present in the report given to the owner of the network being tested. In addition to the list of vulnerabilities detected, corrective actions are an important part of the final report.

1.1 Scope

To help the reader develop a better understanding of the process of performing a penetration test, several sections will demonstrate how to use some common tools and provide sample results. The demonstration will be performed on a fictitious organization called shinberg.org that is assigned a Class B address space. All testing will be performed using an external IP address. That is, testing of wireless access points, modem lines and scanning from inside the target network are specifically excluded from this paper.

2. Preparation

There are two distinct activities involved in preparing for a penetration test. The first part is developing the required technical acumen and creating a suitable system, or group of systems to perform the testing. The other, process focused part is to define the purpose and associated limits of the penetration test.

2.1 Technical Preparation

A good penetration tester must be technically competent and methodical. In many situations, a test team is more appropriate than an individual tester.[2] Care must be taken in selecting, installing and configuring the platforms used to perform the testing. Although there are several commercial tools that can be used to perform penetration tests such as Internet Scanner® from Internet Security Systems[3], free tools will be used throughout this testing. Kurtz and Prosise make an excellent point when they claim; "Running a commercial vulnerability scanner is penetration testing" is a myth.[4] There are several problems with simply running a vulnerability scanner and assuming that a complete penetration test has been performed. The first is that the vulnerability scanners are only as good as the person running them. As will be discussed latter in this paper, there is more to performing a penetration test than just finding

---

[2] Naturally, the testing performed in support of this paper will be performed by the author only.

[3] http://www.iss.net/products_services/enterprise_protection/vulnerability_assessment/scanner_internet.php

[4] http://www.infosecuritymag.com/articles/september00/features4.shtml

vulnerabilities. Careful planning is required before packets are sent on the network and reports need to be written once the testing is complete.

A UNIX based system is best for most types of penetration testing; however, it is important to have access to a machine running a current version of Microsoft Windows.[5] A windows system provides easy tests file and print sharing using the net use command. Most tests can be performed from a Linux box that is configured with the proper tools. The machines used to perform the testing described in this paper are:

- IBM ThinkPad T23 capable of dual booting into Windows 2000 Professional and Red Hat 8.0. This machine also runs vmware[6] to allow switching between Linux and Windows without rebooting.

- HP Omnibook XE2 also capable of dual booting into Windows 2000 Professional and Red Hat 8.0. A D-Link 100BT Ethernet card was used to provide network access.

Both machines used for testing are kept current with patches released from Microsoft and Red Hat. Additionally, virus protection software is run under Windows 2002. It is important to ensure that the machines used in penetration testing are secure and administered with security in mind. Remember, that during the course of penetration testing, a tester may obtain access to the target network. The last thing that a tester wants to have happen is for his machine to introduce malicious code into the customer's network.

The network for launching the tests is connected to the Internet via a cable modem. The internal network consists of a combination firewall router and a hub. Both test machines are on the hub so that they can be used to observe all packets used in testing. A diagram of the network used to perform the testing is shown in Figure 2-1: Network used for penetration testing.

2.1.1 Penetration Testing Tools

A myriad of tools are available for performing penetration tests. This section will discuss only a few and focus on the tools that are applicable to this paper. The following tools are installed on the test machines; however, not all of the tool will necessarily be used during the testing. This section provides an overview of the tools; later sections will provide details on how the tools were used to perform the penetration testing.

---

[5] As of the writing of this paper, a current Microsoft Windows operating system is Windows 2000 and later.

[6] vmware is a tool that allows virtual machines to be run under a host operating system. Using vmware Linux is run as a process under Windows, which allows a tester to easily switch between the two operating systems. More information on vmware can be found at: http://www.vmware.com/.

nmap Nmap is the defacto network scanning tool. It is available for almost all modern operating systems, including Windows. Nmap works by creating specially crafted network packets to elicit responses. The packets used by nmap are designed to trick the TCP stack on the target machine into responding and revealing more information than a normal ping would reveal. Nmap is a command line tool with approximately 50 options. Clearly, this can make running nmap a little daunting. For the less experienced, nmapfe provides a graphical front end to nmap. Nmap is provided as part of Red Hat 8.0; however, versions that are more recent are often available. More information on nmap, including the most recent distribution can be found at: http://www.insecure.org/nmap/index.html.

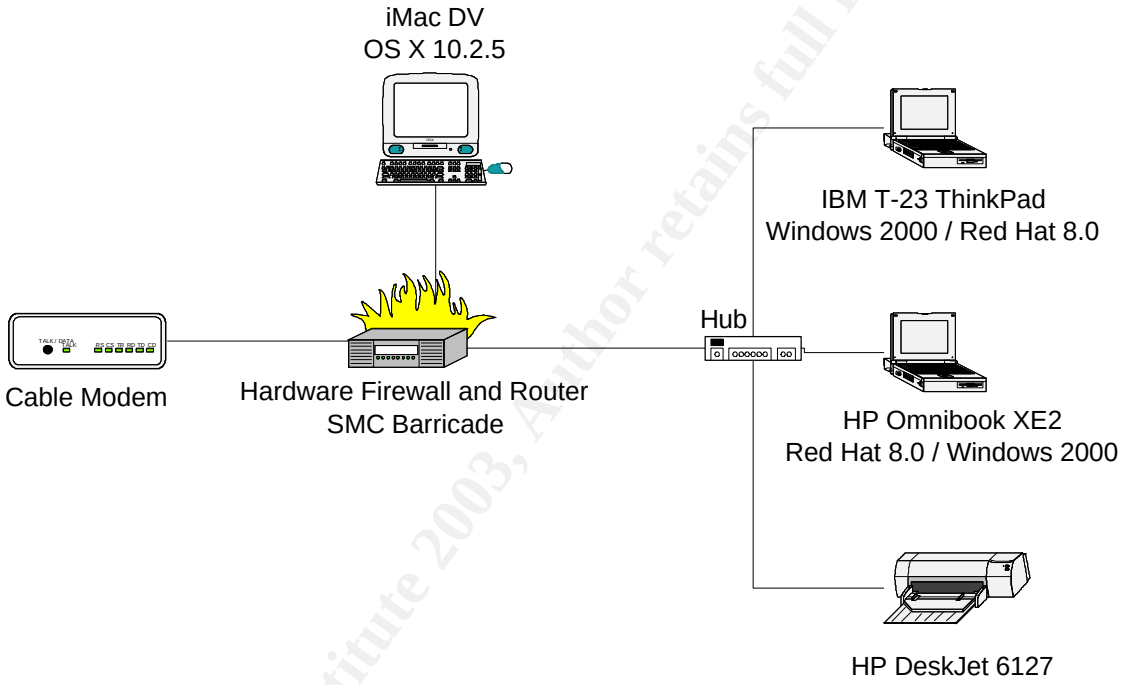

Figure 2-1: Network used for penetration testing

cheops-ng Cheops-ng is described as the "the network Swiss army knife"[7] in that it can scan hosts and map networks. Cheops-ng uses a client server architecture to isolate the user interface from the scanner. This also allows the scanner to be run inside a network and the user interface to reside outside of the network. More information about cheops-ng can be found at http://cheops-ng.sourceforge.net.

Nessus The Nessus Security Scanner is a required tool for penetration testing. Nessus functions as a network scanner to identify hosts, and

[7] http://cheops-ng.sourceforge.net

then continues to scan each host for known vulnerabilities. Like cheops-ng, Nessus uses a client server architecture. The benefit of Nessus in performing penetration tests is Nessus will report on vulnerabilities present on each host. Nessus contains plugin modules to scan for vulnerabilities and the user can write custom plugins if necessary. The maintainers of Nessus do a good job of providing plugins for new vulnerabilities as they are announced. Another feature of Nessus is the ability to test several hosts simultaneously. This is important when scanning larger networks with many accessible hosts. Nessus relies on nmap for performing some scans including ones to identify the operating system on the host. More information about Nessus can be found at: http://www.Nessus.org.[8]

netcat Netcat is another tool that claims to be a "network Swiss army knife"[9], and in a sense, it is depending on you definition of a Swiss army knife. Netcat is a simple and small tool for sending data over a network using TCP and UDP packets, and is available on Unix and Windows. Netcat acts as both a client and a server depending on how it is invoked. For this reason, netcat is a favorite tool of hackers. Netcat can be used to install simple backdoors on hosts by listening on a given port. Further, netcat can be used to establish a connection to compromised system by sending a shell prompt out of a firewall protected network via a connection from the host to port 80 (http) on a machine owned by the hacker.[10] Additionally, because netcat can read and write to standard input and output, it can be incorporated into scripts.[11] This allows one to write a script to test an application without needing to deal with the intricacies of establishing a network connection. The network connections established by netcat are unidirectional, unlike telnet. That is, when one is using netcat the characters sent are not echoed back to the user. This is a little disconcerting and a good reason to drive netcat using scripts. More information about netcat can be found at: http://www.atstake.com/research/tools/network_utilities/.

---

[8] http://www.Nessus.org/features.html

[9] http://www.atstake.com/research/tools/network_utilities/

[10] The term "owned" is used lightly. In this case "owned" means a machine that the hacker can control.

[11] On a UNIX system, standard input (stdin) is usually associated with the keyboard and standard output (stdout) is usually associated with the terminal or screen. It is possible to use special characters to redirect both standard input and standard output. For example: `ls > ls.txt` will place the output of the ls command into the file ls.txt. The less than sign (<) is used to redirect standard input and the greater than sign (>) is used to redirect standard output.

snort Technically speaking snort is not a tool for performing a penetration test. However, it is very valuable in monitoring the penetration test. Although snort started as a network sniffer similar to tcpdump, it has grown into a world class Intrusion Detection System (IDS). Snort contains many rules that allow it to detect malicious activity. By running snort on the network, or host perform the penetration test, it is possible to demonstrate not only what packets were sent, but what a good IDS should have reported. More information on snort can be found at: http://www.snort.org/.

firewalk Firewalk is an active reconnaissance network security tool that attempts to determine what layer four protocols a firewall will pass. This information is useful because it allows one to determine what ports are open on a firewall. Firewalk works by sending out TCP or UDP packets with a time to live one greater than the targeted gateway. If the gateway allows the traffic, it will forward the packets to the next hop where they will expire and elicit an icmp message. If the gateway host does not allow the traffic, it will likely drop the packets and no response will be seen.[12] More information on Firewalk can be found at: http://www.packetfactory.net/projects/firewalk/.

2.1.2 Team formation

Penetration testing is best performed by a team, which provides several benefits from both a technical and a managerial standpoint. One technical benefit is different team members bring different sets of skills. Some people can perform network mapping including penetrating firewalls, but are not adept a breaking into machines. Many hackers do not like taking notes of their work; however, a team member can assist with the note taking.

Two managerial benefits are efficiency and accountability. A team will be more efficient than a single person will because multiple tasks can be performed at the same time. A team, especially one with a strong leader, shares the responsibility for damaging a network or compromising information.

2.2 Process Preparation

Most people assume that a penetration tester is technically competent, or a least proficient in the tools that will be used. With some tools, such as Nessus, performing a simple penetration test and generating reports requires nothing more than a few clicks of the mouse.

Technical competence alone does not make a good penetration tester. Penetration testing should be undertaken as a scientific process.[13] The scientific

---

[12] http://www.packetfactory.net/projects/firewalk/

[13] Although, it is very tempting, a discussion about the difference between scientists and engineers is outside of the scope of this paper. Suffice it to say

method is applicable to performing penetration tests. A good guideline for performing penetration tests that parallels the scientific process is:

1) Define a problem statement. In the case of penetration testing this is the objective of the penetration test.

2) Make hypotheses and plan experiments. A possible hypothesis could be that the network is secure with the exception of the hosts in the de-militarized zone (DMZ).

3) Perform experiments recording all observations even ones that seem meaningless. In the case of penetration testing most experiments, involve scanning networks and hosts. However, a search of public information may also lead to some useful insights. The reason that snort is mentioned in the tools section is that it can be used to record observations.

4) Produce a report. A clear and concise report that documents the procedures used and results obtained is critical in documenting the findings from penetration testing.

The need to document both the process as it is performed and the results cannot be overstated. Consider the hypothetical case that a production server crashes during the time when a penetration test is being performed. Clear documentation and network logs will allow you to determine if the penetration caused the crash.

There are several other aspects to preparation, which are discussed in the following sections.

2.2.1 Goals and limits

The goals of a penetration test must be defined and agreed upon by both the testers and the owner of the network being tested. Associated with the goals are some limits that the testers must not cross.

In preparing the goals of a penetration test, there are several items to consider. The first is how much of an impact is acceptable on the network of interest. Some questions to answer are:

When can the tests be run? Penetration tests can deplete the resources of networks, and potentially cause an unintended denial of service condition. Some customers will prefer to run penetration tests during off hours, such as at night or on the weekends. A disadvantage to running the tests during off hours is that some desktop machines may be powered down.

---

in general, that engineers apply known methods to solve problems, and scientists strive to understand how things work at a fundamental level.

What type of tests can be performed? Some customers might be happy just knowing that their machines are vulnerable based on network and host scans. Others will want proof that the testers were able to break into their machines. If the customer instructs a tester to break into a machine whenever possible a clear understanding of what the tester is allowed to do once the machines is compromised is required. Often more than one vulnerability must be used to gain full access to a computer.[14] In addition, some scans might actually crash the service of machine.

Are any machines off limits or excluded from some tests? Certain machines may be excluded from the penetration tests or tested in a limited manner. One reason for this is to protect sensitive information. For example if an external network scan can reach the main financial machine, this is a significant concern in and of itself.

Where will the penetration testing be performed? Typically, penetration tests, as their name implies are performed from outside the network of interest. However, given the current news about insider threats performing an internal scan, which is from inside the corporate firewall, may be a prudent activity.[15]

Should wireless network scanning be performed? Wireless networks based primarily on 802.11 are prevalent in use and a prime target for hackers. Recent legislation in New Hampshire is saying that an unprotected wireless network is fair game. "A bill that's breezing through New Hampshire's legislature says operators of wireless networks must secure them -- or lose some of their ability to prosecute anyone who gains access to the networks."[16]

Are there any situations when the customer needs to be notified immediately? This is a rhetorical questions, as there are some obvious cases were immediate remediation is required. One example is accounts without passwords. A customer may want to be notified of other findings such as well-known vulnerabilities before the final report is delivered.

Another important related topic is how to manage any proprietary information discovered during a penetration test. The organization performing the penetration test should provide a written policy on how it will protect any information discovered during the penetration test. Some companies consider their network topology to be proprietary. Naturally, all vulnerabilities found should be proprietary. Not treating them as such is like telling robbers where the keys to your house can be found.

---

[14] W. Fithen, *et al*., "Formal Modeling of Vulnerability"
[15] Husted, Bill, "Hacker may sit in next cubicle"
[16] McWilliams, Brian, "Licensed to War Drive in N.H."

Memorandum for File

Subject: Vunerability Assessment and Penetration Testing Authorization

Date: 1 April 2003

To properly secure this organization's information technology assets, the information security team is required to assess our security stance periodically by conducting vulnerability assessments and penetration testing. These activities involve scanning our desktops, laptops, servers, network elements, and other computer systems owned by this organization on a regular, periodic basis to discover vulnerabilities present on these systems. Only with knowledge of these vulnerabilities can our organization apply security fixes or other compensating controls to improve the security of our environment.

The purpose of this memo is to grant authorization to specific members of our information security team to conduct vulnerability assessments and penetration tests against this organization's assets. To that end, the undersigned attests to the following:

1. David Shinberg has permission to scan the organization's computer equipment to find vulnerabilities. This permission is granted for from 10 April 2003 until 16 May 2003.
2. Charles I. Officer has the authority to grant this permission for testing the organization's Information Technology assets.

This authorization is solely applicable to the address and domains assigned to the Networking.

Signature: _____ Signature: _____

Charles I. Officer
Director Networking

David Shinberg
Penetration Test Leader

Date: _____ Date: _____

Figure 2-2 Authorization Letter

2.2.2 Customer Agreements

The above goals and limits once agreed upon must be documented. In addition to the goals document, which the tester provides to the customer, the customer must also sign an authorization letter providing documentation that the penetration test has been authorized. Ed Skoudis frequently called this a “get out

of jail free card"[17] The purpose of the letter is to protect the tester from both legal and organizational actions as a result performing the penetration test. A sample letter provided by Ed Skoudis[18] was used as a basis for the agreement signed before the penetration tests were performed by the author. The agreement is shown in Figure 2-2 Authorization Letter. It is important to have a lawyer review the letter before it is used. Several items need to be included including who is authorized to perform the tests, and what networks will be tested. The person signing the letter for the customer must have the authority to authorize the penetration testing. Because the tester will not always be sure that the signer has the authority, the letter states that the signer has the authority to authorize the tests.

Now that the preparation phase is complete, the penetration testing can begin.

3. Public Information

To begin a penetration test, do not just start scanning the network. There is a wealth of information that can help a tester learn about a network without even sending a single probe. Most of this information comes from the network registries. An excellent place to start is with determining the IP address of a host on the network of interest. In most cases, the customer will provide this information; however, it is always a good idea to verify it. The dig command available on many UNIX systems is an excellent choice to map a machine name to an IP address. An example of using dig is shown below.

```
[das@localhost das]$ dig www.shinberg.org

; <<>> DiG 9.2.1 <<>> www.shinberg.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61668
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1,
ADDITIONAL: 0

;; QUESTION SECTION:
;www.shinberg.org.          IN    A

;; ANSWER SECTION:
www.shinberg.org.         36117  IN  A     A.B.255.50[19]

;; AUTHORITY SECTION:
```

[17] Skoudis, Ed, SANS San Francisco, Track 4, Hacker Techniques, Exploits, and Incident Handling, December 15-20, 2002.

[18] http://www.counterhack.net/permission_memo.html.

[19] The A.B is used to denote the first and second octets of the IP address to protect the identity of the actual network that was scanned.

```
shinberg.org.    36117    IN     NS  ns1.shinberg.org.

;; Query time: 47 msec
;; SERVER: 192.168.2.1#53(192.168.2.1)
;; WHEN: Sun May  4 12:12:47 2003
;; MSG SIZE  rcvd: 65
```

The output from dig provides lots of useful information. First, it shows that the IP address of www.shinberg.org is A.B.255.50. This will be useful when we attempt to penetrate hosts on the network, as web servers are a prime candidate to be exploited. In addition, we learn that ns1.shinberg.org is the Authoritative DNS server for shinberg.org. This means if we can perform a zone transfer from A.B.255.50, then we can find all of the hosts listed in shinberg.org. An interesting observation is that the DNS server is running on the same machine as the web server. This is bad! Both DNS and web servers are commonly attacked. Now if a hacker can compromise the web server or DNS server process, the entire machine will likely be compromised.

Now that the IP address is know, the next step is to see who owns the address space allocated to shinberg.org. This can be done using a whois server with web interfaces such as www.arin.net. ARIN is the American Registry for Internet Numbers and manages the Internet numbering resources for North America, a portion of the Caribbean, and sub-equatorial Africa.[20] Going to ARIN's webpage and performing a whois query on A.B.255.50 reveals the following information.

```
OrgName:    GEEKS 'R US Inc
OrgID:      GEEKS 'R US
Address:    GEEKS 'R US Lane
Address:    Room 42
City:       Gotham
StateProv:  ST
PostalCode: 12345
Country:    US

NetRange:   A.B.0.0 - A.B.255.255
CIDR:       A.B.0.0/16
NetName:    GEEKS 'R US-A-B
NetHandle:  NET-A-B-0-0-1
Parent:     NET-A-0-0-0-0
NetType:    Direct Assignment
NameServer: DNS.GEEKSRUS.COM
NameServer: DNS-1.GEEKSRUS.COM
NameServer: DNS-2.GEEKSRUS.COM
```

[20] http://www.arin.net/about_us/index.html

```
Comment:
RegDate:    1997-05-14
Updated:    2001-07-11

TechHandle: CONTACT1-ARIN
TechName:   GEEKS 'R USInc.
TechPhone:  +1-xxx-xxx-xxxx
TechEmail:  contact1@GeeksRUs.com

OrgTechHandle: CONTACT2-ARIN
OrgTechName:   Last, First
OrgTechPhone:  +1-xxx-xxx-xxxx
OrgTechEmail:  contact2@lGeeksRUs.com

# ARIN WHOIS database, last updated 2003-05-16 20:10
# Enter ? for additional hints on searching ARIN's WHOIS
database.
```

Now we learn that shinberg.org is really part of Geeks 'R Us Inc. This is a surprise and might mean that the person who authorized the scans did not have the proper authority. We also learn that the network used by shinberg.org is A.B.0.0/16, which is a Class B network, again a little surprising.

More information can be found by searching other whois servers. In this case, a search of whois.networksolutions.com showed:

```
[das@localhost das]$whois -h whois.networksolutions.com
shinberg.org
[whois.networksolutions.com]
Registrant:
Geeks 'R Us Inc. (SHINBERG3-DOM)
   Main Rd
   Any City
   State,Zip
   US

   Domain Name: SHINBERG.ORG

   Administrative Contact, Technical Contact:
      Ralph, Malph  (RR14414)   rm@GEEKSRUS.COM
      Geeks 'R Us Inc.
      Main Rd
      Any City
      State,Zip
      xxx-xxx-xxxx (FAX) xxx-xxx-xxxx
```

```
   Record expires on 15-Jun-2003.
   Record created on 15-Jun-1999.
   Database last updated on 4-May-2003 12:18:57 EDT.

   Domain servers in listed order:

   NS1.SHINBERG.ORG                A.B.255.50
   NS2.SHINBERG.ORG                A.B.1.21
```

The important item to notice is that again we get the names and IP addresses of the name servers returned. Notice that the names are different from the original query to whois.arin.net. The record is about to expire, and it is not uncommon for small companies to forget to renew their registration. This fact should be written down and mentioned in the final report.

Based on the large address space, and different data from the various whois servers, more investigations are required. An ICMP echo request sent to the first name server (A.B.255.50) responded; however, there was not response from the second name server (A.B.1.21).

A possible explanation for having two DNS servers with such a large difference in addresses is that shinberg.org is supported by two different Autonomous System Numbers (ASN). There might be two different ways for packets to reach shinberg.org. "Autonomous System numbers (ASNs) are globally unique identifiers for Autonomous Systems. An Autonomous System (AS) is a group of IP networks having a single clearly defined routing policy, run by one or more network operators."[21]

Several route servers on the Internet provide information about the mapping between IP addresses and ASNs. An excerpt of a query to route-views.oregon-ix.net is shown below.

```
[das@localhost das]$ telnet route-views.oregon-ix.net
Trying 198.32.162.100...
Connected to route-views.oregon-ix.net.
Escape character is '^]'.

route-views.oregon-ix.net>show ip bgp A.B.0.0
BGP routing table entry for A.B.0.0/16, version 3375939
Paths: (57 available, best #26)
…
  1668 6347 WXYZ
    66.185.128.48 from 66.185.128.48 (66.185.128.48)
      Origin IGP, metric 15, localpref 100, valid, external
```

[21] http://www.apnic.net/services/asn_guide.html

```
…
  7500 2497 701 WXYZ
    202.249.2.86 from 202.249.2.86 (210.173.176.242)
      Origin IGP, localpref 100, valid, external
…
```

The information shows that the ASN associated with A.B.0.0/16 is WXYZ, and that there are indeed to other autonomous systems that provide a path to WXYZ. Now that the ASN is known, a query to whois.arin.net can be used to learn more about the ASNs. Only select portions of the response are shown below.

```
[das@localhost das]$ whois -h whois.arin.net ASWXYZ
[whois.arin.net]

ASNumber:   WXYZ
ASName:     GEEKS 'R US-WHATS
ASHandle:   ASWXYZ
Comment:
RegDate:    1997-05-05
Updated:    1999-01-11

 [das@localhost das]$ whois -h whois.arin.net AS701
[whois.arin.net]

OrgName:    UUNET Technologies, Inc.
OrgID:      UU
Address:    22001 Loudoun County Parkway
City:       Ashburn
StateProv:  VA
PostalCode: 20147
Country:    US
ASNumber:   701 - 705
ASName:     ALTERNET-AS
ASHandle:   AS701
Comment:
RegDate:    1990-08-03
Updated:    2002-11-27

[das@localhost das]$ whois -h whois.arin.net AS6347
[whois.arin.net]

OrgName:    SAVVIS Communications Corporation
OrgID:      SAVV
Address:    1 SAVVIS Parkway
City:       Town and Country
StateProv:  MO
PostalCode: 63017
```

```
Country:    US

ASNumber:   6347
ASName:     DIAMOND
ASHandle:   AS6347
Comment:
RegDate:    1996-03-15
Updated:    2002-07-17
```

An interesting point, not shown above is that yet another different contact person is associated with shinberg.org.

Before proceeding to the technical planning phase a short summary is in order. The company Geeks 'R Us contains an organization called shinberg.org that maintains its own network. This network is a Class B with an address of A.B.0.0/16. Two different autonomous systems provide a routing path the shinberg.org, which has an ASN of WXYZ. Only one active DNS server could be found even though two are listed with the various registrars.

The above information was determined by sending only a few icmp echo request packets to both name servers. The pings to the name servers would not be considered unusual traffic and should not have been reported by the Intrusion Detection System (IDS).

4. Technical Planning

Now that a tremendous amount has been learned about the network to be tested, the actual penetration test can be planned. The purpose of planning is so that the testers do not need to think about what to do next. This allows them to work more methodically, and avoid some careless errors. In addition, a quick review of the results should be performed when the testing is being done.

Since a penetration test is an authorized activity, one does not need to be concerned with being detected. However, it is critical that proper people in the organization be aware that the penetration test is being performed. The reason for this is to prevent any unwarranted responses. There is some debate about whether or not to inform the staff that is responsible for protecting the network and machines. In the case of shinberg.org, the decision was made to inform only the managers responsible for the network and not the technical staff. This way, the managers would be able to determine whether the staff could detect an attack on the network. The managers were also told the IP addresses used for the scanning so that they could quickly distinguish between the penetration test and a real attack.

Because shinberg.org has a large address space, and the customer wanted the testing to occur over a two-week period primarily at night and on the weekends, a high-speed connection was used to perform the testing. Only external testing

was requested by the customer. Additionally, the customer performed a physical audit and instructed the test team that war driving, or in this case, war walking was not to be performed.

The first tests to be performed are various nmap scans of the entire address space. This will identify hosts, and services running on the hosts. It turns out that many other tools make use of nmap either directly (e.g., Nessus) or by sending nearly identical types of packets. In addition to nmap, cheops-ng will be used to provide a graphical view of the network.

Nessus will later be used on the identified hosts to probe for vulnerabilities. Although Nessus can scan a network, its scans will not find anything that nmap doesn't find. Snort will be used to monitor the testing and provide another mechanism for recording the results. Additionally, tcpdump will be used to monitor some tests.

Prior to starting the tests, all machines were updated to the latest release of the tools. This included updating to Nessus 2.05, as well as applying several patched from RedHat.

5. Execution

The execution of the penetration testing was broken down into three parts based on the tools used. Please note that one purpose of the penetration test was for the author to become more familiar with the tools. Therefore, some tests were performed multiple times. In a real penetration test, the team should ensure that the tools will work the first time by performing a dry run on its own network.

Snort was installed on the system used for testing both to monitor the traffic and to demonstrate to the customer what should have been noticed by their staff. To eliminate the collection of extraneous information about the testing system, snort was configured to only monitor only connections to the target address range. The following line is used to configure snort.

```
var HOME_NET A.B.0.0/16
```

The above line makes snort process packets as if it existed on the A.B.0.0/16 network. The three parts of the penetration test are described in the following sections.

5.1 Testing with nmap

Nmap is an excellent tool for finding hosts on a network; however choosing the correct set of options can be a bit tricky. To provide consistency in the tests as well as reduce errors scripts were used to automate the running of nmap. A sample script is shown below:

```
LOGFILE=scan-1-ping
date > ${LOGFILE}.log
```

```
nmap -sP -v –n --randomize_hosts -oN ${LOGFILE}.txt \
     -oG ${LOGFILE}.grp A.B.0.0/16 >> ${LOGFILE}.log
date >> ${LOGFILE}.log
```

The above script, which must be run as root, performs a ping scan of the network while randomizing the host addresses.[22] This scan simply identifies hosts on the network that respond to the icmp echo request packet. The first line sets the name of the log file that should be indicative of the test being performed. The second line creates a new log file and sets the first line to be the time and date that the script started running. The next line actually runs nmap and requires a little more explanation.

nmap The name of the nmap executable, which exists in the UNIX Path.

`-sP` Perform a ping scan using an icmp echo request packet

`-v` Verbose mode, which provides more information about what nmap is doing.

`-n` Prevents nmap from performing a reverse DNS lookup to determine the name associated with an IP address. This option should usually not be used; however, the DNS server for shinberg.org does not provide reverse lookups.[23]

`--randomize_hosts` Instructs nmap to randomize the hosts. Otherwise, nmap will scan the hosts in order of their IP address. Many Intrusion Detection Systems recognize network scans by detecting this sequential pattern.

`-oN ${LOGFILE}.txt` Instructs nmap to log the results of the scan in human readable format in the file ${LOGFILE}.txt

`-oG ${LOGFILE}.grp` Instructs nmap to log the results of the scan in a file that can easily be parsed by grep or other similar programs.

---

[22] Nmap requires root privileges to send and receive packets over the network. The script must be run by root, not setuid to root. It is well known that setuid shell scripts are a huge security hole. A hacker could create his own nmap executable and place it into the path before the real nmap executable. This would provide an easy mechanism to become root on the testing machine. It is possible to create safe scripts, but it was not necessary in this situation.

[23]
```
[das@localhost das]$ host -a A.B.255.50
Trying "50.255.30.135.in-addr.arpa"
Host 50.255.30.135.in-addr.arpa not found: 2(SERVFAIL)
```

`A.B.0.0/16` The address network that nmap should scan. This could also be a single IP address (e.g., A.B.255.50), or a range of addresses (e.g., A.B.0.0 – A.B.255.255).

The results of the above scan showed that only 36 hosts responded to the ping packets. This is a hit ratio of 0.05%! That means that either there are very few hosts on the network, or those that exists are protected behind a filtering router or a firewall.

Luckily, nmap can perform more advanced scans that are often more capable of identifying hosts behind firewalls and filtering routers. These scans are commonly called stealth scans and take advantage of unanticipated condition within the TCP/IP protocol specification.

In an attempt to find more hosts, a FIN scan was performed using the following nmap command line:

```
nmap -sF -v -n -oN ${LOGFILE}.txt -oG ${LOGFILE}.grp
     A.B.0.0/16 >> ${LOGFILE}.log
```

This scan differed from the pervious scan because it instructed nmap to send a FIN packet using the –sF option instead of the ping packet. Also, the randomize host option was not used for this scan. The results of the FIN scan on host are shown below.

```
Interesting ports on  (A.B.255.66):
(The 1596 ports scanned but not shown below are in state:
closed)
Port       State      Service
136/tcp    filtered   profile
137/tcp    filtered   netbios-ns
138/tcp    filtered   netbios-dgm
139/tcp    open       netbios-ssn
445/tcp    open       microsoft-ds
```

The results likely indicated that this is a windows host because ports 139 and 445 are open. The filtered state means that some device is returning an icmp-filtered message. A better configuration is to drop the packet so that there is no response. In scanning a network, every response provides information to the person doing the scanning. Nmap and other tools can be used to attempt to identify the remote operating system.

The –sO option instructs nmap to attempt to identify the host's operating system. Nmap uses packets that do not conform to the TCP/IP standard to elicit responses from the host being probed. Different hosts responded differently to these unusual packets. An example of nmap identifying the operating system of a Cisco router is shown below.

```
Host  (A.B.255.63) appears to be up ... good.
Initiating SYN Stealth Scan against  (A.B.255.63)
Adding open port 2001/tcp
Adding open port 6001/tcp
The SYN Stealth Scan took 29 seconds to scan 1601 ports.
For OSScan assuming that port 2001 is open and port 1 is
closed and neither are
firewalled
Interesting ports on  (A.B.255.63):
(The 1594 ports scanned but not shown below are in state:
closed)
Port       State       Service
136/tcp    filtered    profile
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn
445/tcp    filtered    microsoft-ds
2001/tcp   open        dc
6001/tcp   open        X11:1
Remote OS guesses: Cisco 3600 running IOS 12.2(6c), Cisco
router running IOS 12.
1.5-12.2(6a), Cisco IOS 12.1(5)-12.2(7a)
TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
IPID Sequence Generation: All zeros
```

Operating system identification allows a penetration tester to focus the tests to a particular operating system. This allows for more efficient testing.

5.2 Testing with cheops-ng

Cheops-ng provides a nice graphical user interface to map networks. The primary advantage of cheops-ng is the user interface and display; however, cheops-ng crashed frequently. Nonetheless, some useful results were obtained. The layout algorithm used to arrange the nodes on the display is very poor, and while positioning the nodes by hand makes them more readable, it is still not very useful. Cheops-ng used nmap to identify the operating systems of the network devices and then displayed an appropriate icon. In the case of shinberg.org, cheops-ng indicated that CISCO routers are used. A sample display from cheops-ng is shown in Figure 5-1.

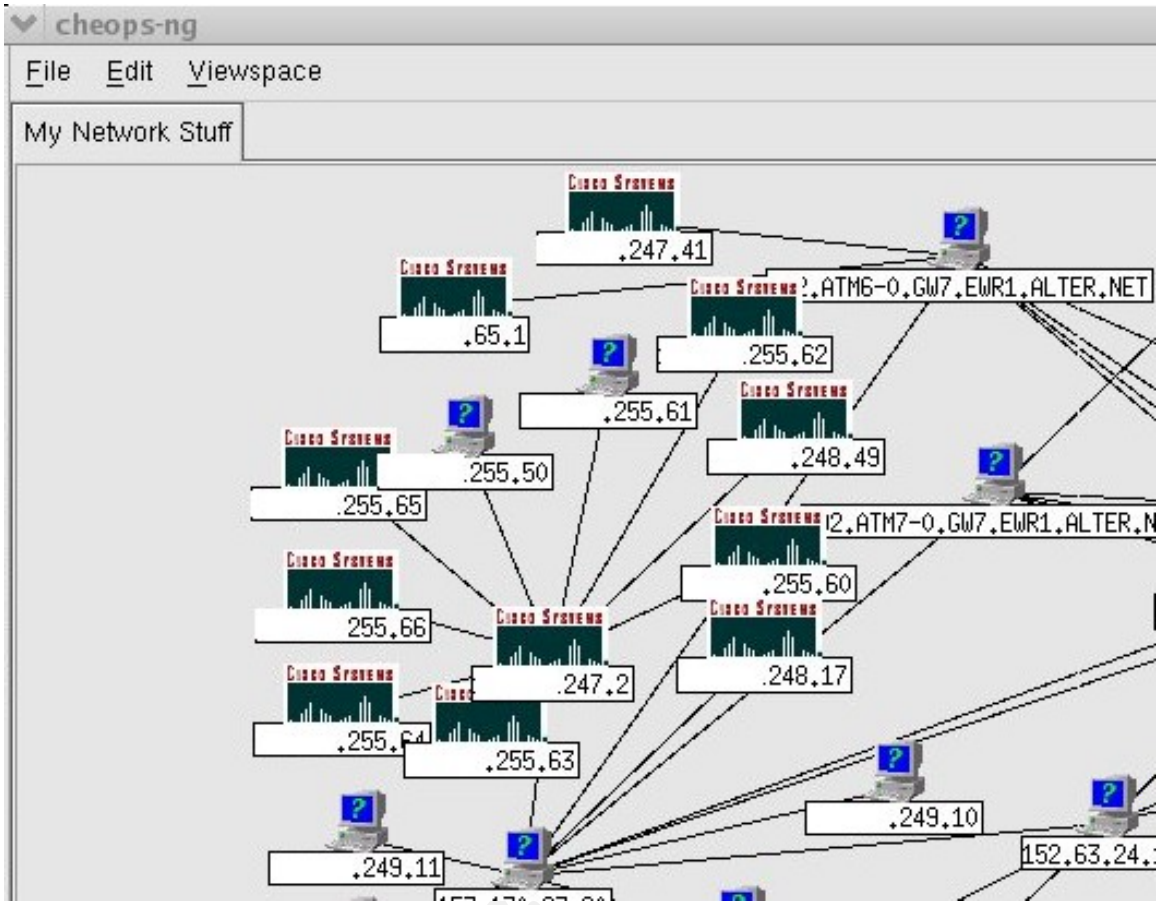

Figure 5-1 Sample cheops-ng output

5.3 Testing with Nessus

The one tool a penetration tester should not be without is Nessus. Nessus provides almost all of the functions required in one package. From a simple graphical user interface a scan of an entire network can be performed.[24] Beyond just identifying hosts on the network, Nessus provides the ability to scan for vulnerabilities.

This is the real power behind Nessus. This section will highlight some of the configuration options available within Nessus. Additionally, the graphical user interface provided to analyze the results of a scan will be discussed.

---

[24] Version 2.05 of Nessus was used while the author was researching this paper and performing the penetration tests. Unfortunately, this version of Nessus consistently crashed before completing the scan of the Class B network. Therefore, to expedite the testing, and therefore the creation of this paper, Nessus was configured to scan only the hosts identified during the nmap scanning of the network.

The extensive vulnerability scanning capabilities in Nessus are built based on the Nessus Attack Scripting Language (NASL). NASL was designed to allow developers easily to write their own security checks. However, Nessus provides up to date scripts for most published vulnerabilities. Because NASL is an open language, a tester can write his own scripts to check any proprietary systems being used.

Figure 5-2 Tabs of the Nessus graphical user interface

The Nessus interface is divided into eight tabs as shown in Figure 5-2. The more significant and commonly tabs are described below.

Nessusd host This tab is used to login into the host that is running the Nessus daemon (nessud). Because Nessus uses a client server architecture, the daemon must be started before a user can login into the daemon.[25]

Plugins This tab allows the user to select the various plugins used to perform the vulnerability scans. A button allows the user to enable all plugins with a single mouse click. Another button enables all but the dangerous plugins.

Prefs. This tab allows the user to configure various details about how Nessus scans the network. Because Nessus uses nmap to perform the network mapping, many of these options are analogous to options used with nmap. Other options on this tab allow the specify default passwords for logins. One particularly useful option is the ability to specify the SNMP community string. The SNMP community string is used to gain access to SNMP devices. Some companies ship their devices with a well-known default string, and administrators, may not change it to something more secure.[26]

Scan options Defines the port range to be scanned and specifies if a reverse DNS lookup should be preformed.

[25] This sounds obvious; however, I have solved the problem that people had logging in by simply starting the daemon.

[26] This is only slightly better than using the default community strings of “public” and “private.” Both read and obviously write SNMP access should be protected as it provides a way to gain information about a network, or worse actually change the configuration of a device.

Target Selection Nessus can accept almost any target specification ranging from host names to CIDR notation. It can read a list of hosts from a file, or even perform a DNS zone transfer to find hosts to scan.

Although Nessus provides a graphical user interface, there are still some options worth mentioning. The most significant option is “Enable all but dangerous plugins.” In most situations, this is the best way to select the plugins to use. Nessus has some plugins that may cause a denial of service on the device being scanned. The “enable all” button on the Plugins tab will instruct Nessus to run these dangerous plugins. More information including examples on how to use Nessus can be found at: www.Nessus.org

After a scan is complete, Nessus provides a graphical user interface for browsing the results. The interface is divided into several frames as described below.

Subnet The subnet frame allows the user to select from the subnets that were scanned.

Host The host frame allows the user to select which host to examine for the chosen subnet

Port The port frames displays a list of open ports detected on the selected host.

Severity Indicates the severity of the problems found.

A final unlabeled frame provides details on any vulnerabilities found. The displayed information tells what an attacker could do to the system. In addition, background information and steps that should be taken to mitigate or eliminate the vulnerability are displayed. A sample report from Nessus that illustrates the above frames is shown in Figure 5-3.

Nessus can also produce reports that are independent of Nessus itself. Two available formats are html and latex. Latex is a complex text layout language that is not commonly used under Microsoft windows. Luckily, there is a tool called pdflatex that converts the latex file to PDF. The html reports can be produced with or without graphs. The graphs provide a nice addition to the final reports.

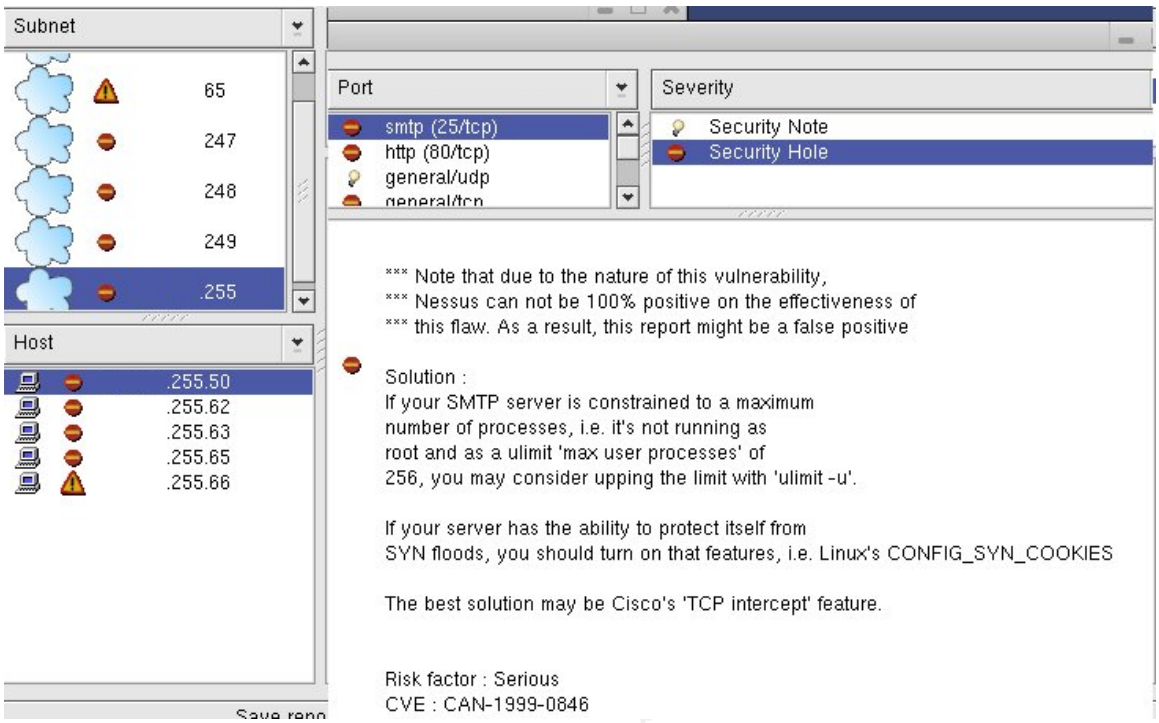

Figure 5-3 Nessus report

6. Analysis and Reporting

The data indicate that this network is not secure and an attacker could gain access to one or more devices on the network. While this is a true statement, a customer will not be happy hearing or reading it. More significantly, the staff responsible for maintaining the network will be upset. Therefore, a report must focus on the technical details and not place blame on anyone. Included in the report should be the steps that can be taken to make the network more secure.

The reports generated by Nessus provide the details needed to summarize the vulnerabilities and provide remedial actions. The latex report produced by Nessus states “Nessus has tested 17 hosts and found 26 severe security holes, as well as 33 security warnings and 68 notes. These problems can easily be used to break into your network.”

An important part of the analysis is how many hosts were found on the network from outside the firewall. In this case, 117 hosts were reported by the various nmap scans. This is fewer than 2% of the addresses available to a Class B network. Therefore, either the firewalls are effective, or there are very few hosts on the network.

The network scans which were at times run using nmap’s aggressive mode, were not detected by the target network. A more correct statement is that the scans were not reported to the mangers responsible for the network. The testers were

informed that snort is running on the target network. The snort instance running on the computers used to perform the testing reported all of the scans. Since the scans were all performed from the same IP address, this indicates a concerted attempt to scan a network for vulnerabilities. It is not uncommon for a scan to precede an attack.

Based on the information discovered during the vulnerability scanning phase, there was no need to break into any machines. A call to the customer informed him of the severity and number of vulnerabilities that were found. Additionally the team verified with the customer that the scans were not reported to management. Therefore, the customer requested that the team provide a report of the vulnerabilities detected and steps required to fix the problems instead of actually breaking into a machine.

6.1 Snort Data Analysis[27]

As stated earlier, a local snort was used to monitor the packets being used to probe the shinberg.org network. The local version of snort consistently detected and logged the scans. Below are some extracts from the alert log generated by snort while the nmap's host identification scan was being run.

```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/18-17:20:35.635351 192.168.2.83:54275 -> A.B.255.63:8080
TCP TTL:55 TOS:0x0 ID:21446 IpLen:20 DgmLen:40
******S* Seq: 0x10A4F4B  Ack: 0x0  Win: 0x1000  TcpLen: 20
```

The above entry indicates that a SYN scan was being performed to port 8080 on A.B.255.63. The SYN scan determines whether there is a server listening on a specific port. Port 8080 is commonly used by web servers that are not started as root because it is an unprivileged port.

```
[**] [1:615:3] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/18-17:20:37.820825 192.168.2.83:54275 -> A.B.255.63:1080
TCP TTL:55 TOS:0x0 ID:37166 IpLen:20 DgmLen:40
******S* Seq: 0x10A4F4B  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://help.undernet.org/proxyscan/]
```

The above entry indicates that a SYN scan was being performed to port 1080 on A.B.255.63. The SYN scan determines whether there is a server listening on a specific port. Port 1080 is commonly used for web proxies.

---

[27] This section could have been titled "What Was Missed by the People Responsible for Network Security"; however, snort data analysis does not assign blame.

```
[**] [1:485:2] ICMP Destination Unreachable (Communication
Administratively Prohibited) [**]
[Classification: Misc activity] [Priority: 3]
05/18-17:20:46.985252 192.168.2.1 -> 192.168.2.83
ICMP TTL:253 TOS:0x80 ID:19214 IpLen:20 DgmLen:56
Type:3  Code:13  DESTINATION UNREACHABLE: ADMINISTRATIVELY
PROHIBITED, PACKET FILTERED
** ORIGINAL DATAGRAM DUMP:
192.168.2.83:54275 -> A.B.255.63:445
TCP TTL:54 TOS:0x0 ID:19214 IpLen:20 DgmLen:40
Seq: 0x10A4F4B  Ack: 0x5C912E24
** END OF DUMP
```

The above entry indicates that a packet was sent to port 445 on A.B.255.63. The Microsoft LanMan server on port 445 in Windows 2000 is vulnerable to a denial of service attack.[28] The first part of the above entry shows that an ICMP Destination Unreachable message was received because this port is filtered. The ICMP filtered message indicates that a router is blocking access to this port on this host.

```
[**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan)
detection [**]
05/18-17:20:51.980743 192.168.2.83:54283 -> A.B.255.63:2001
TCP TTL:55 TOS:0x0 ID:4876 IpLen:20 DgmLen:60
******** Seq: 0xBE665BBF  Ack: 0x0  Win: 0x1000  TcpLen: 40
TCP Options (4) => WS: 10 NOP MSS: 265 TS: 1061109567 0
```

The above entry indicates that a packet was sent to port 2001 on A.B.255.63. The interesting thing about this packet is that there were no TCP flags set. A TCP packet with all flags set to zero (i.e., null) should not happen. This is a null scan being performed by nmap.

```
[**] [1:628:1] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/18-17:20:51.981009 192.168.2.83:54285 -> A.B.255.63:2001
TCP TTL:55 TOS:0x0 ID:47 IpLen:20 DgmLen:60
***A**** Seq: 0xBE665BBF  Ack: 0x0  Win: 0x1000  TcpLen: 40
TCP Options (4) => WS: 10 NOP MSS: 265 TS: 1061109567 0
[Xref => http://www.whitehats.com/info/IDS28]
```

The above entry indicates that a packet was sent to port 2001 on A.B.255.63. The interesting thing about this packet is that the ACK flag was set, but the Ack value is zero. This is an anomalous packet generated by nmap in an attempt to

[28] http://lists.insecure.org/lists/bugtraq/2003/Jan/0138.html

identify the host operating system. Snort correctly identified that nmap was being used for scanning the network.

The excerpts from the snort logs show that snort was effective in identifying the scanning activity that was part of the penetration testing.

7. Conclusion

This documented provided some insight into how to plan and perform a penetration test. A penetration test is best approached as a scientific endeavor and can be based in part on the scientific method.

There are several free tools that can be used to perform penetration tests. The two tools that proved most useful were nmap and Nessus. One should note that Nessus relies on nmap for scanning the network. A significant benefit of Nessus and the reason it should be a tool of choice is that Nessus can scan a host for thousands of known vulnerabilities. Additionally, testers can write their own scripts using Nessus Attack Scripting Language to test to unpublished vulnerabilities.

The penetration testing process described was based on the premise that there was not cooperation between the testers and the organization being tested. Therefore, the testers needed to discover as much as possible about the network from external sources. In addition, all of the testing was performed from outside of the target network.

The use of public information allowed the testers to learn that the shinberg.org network has two separate connections to the Internet as indicated by two different serving Autonomous Systems. Unfortunately, the team did not have the resource to direct traffic through both of the autonomous systems, whereby more devices might have been found.

A more complete view of the security of the network can be developed when personal interviews are performed. The interviews allow the testers to determine whether there is an organizational policy on monitoring security logs and installing system patches. In addition, a physical audit of the premises should be performed to ensure that the network devices are protected from tampering.

The number and severity of the vulnerabilities detected were surprising. The team expected to find one or two out of date versions of software, but nothing so significant.

References

Garfinkel, Simson, Spafford, Gene and Schwartz, Alan, Practical Unix and Internet Security Third Edition, O'Reilly & Associated, Sebastopol, CA, February 2003, ISBN: 0-596-00323-4

Husted, Bill, "Hacker may sit in next cubicle," The Atlanta Journal-Constitution, 5/14/03 http://www.ajc.com/business/content/business/coke/0503/14breakin.html (last accessed 5/17/03)

JANET-CERT "Prevention: Organisations: Penetration Testing" http://www.ja.net/CERT/JANET-CERT/prevention/pentest.html (last accessed 5 May 2003)

Johnson, Greg "Using NMAP and NESSUS to Audit Large Networks" Presentation at MORENet Security Symposium, December 18, 2001 http://bengal.missouri.edu/~johnsong/audit/ (last accessed 5/18/03)

Kurtz, George and Chris Prosise, "PENETRATION TESTING EXPOSED", Information Security Magazine, http://www.infosecuritymag.com/articles/september00/features3.shtml (last accessed 5 May 2003)

Kurtz, George and Chris Prosise, "Penetration Testing: Myth vs. Reality", Information Security Magazine, http://www.infosecuritymag.com/articles/september00/features4.shtml (last accessed 6 May 2003)

McWilliams, Brian, "Licensed to War Drive in N.H.," Wired News, Apr. 29, 2003, http://www.wired.com/news/wireless/0,1382,58651,00.html (last accessed 5/17/03)

nmap manpage supplied with RedHat 8.0

Penetration Tests: "The Baseline for Effective Information Protection," an ISS Whitepaper, http://documents.iss.net/whitepapers/pentestwp.pdf (last accessed 5 May 2003)

Swift, Sean, "Effective penetration testing requires security review, on-target test team," ServerWorld Magazine, September 2000 Issue. http://www.serverworldmagazine.com/sunserver/2000/09/penetration.shtml (last accessed 5 May 2003)

Thompson, Kerry, "Requesting a Penetration Test " March 22, 2002, http://www.crypt.gen.nz/papers/requesting_pen_test.html (last accessed 5 May 2003)

W. Fithen, S. Hernan, P. O'Rourke, and D. Shinberg, "Formal Modeling of Vulnerability" Bell Labs Technical Journal, Network Security Issue, Vol. 8, No. 4. (Forthcoming)