Scanning

INFORMATION IN THIS CHAPTER

- This chapter will introduce the concepts and basic tools used in the scanning phase

CHAPTER OVERVIEW AND KEY LEARNING POINTS

This chapter will:

- explain the importance of the scanning phase of the penetration testing lifecycle
- introduce the networking protocols TCP, UDP, and ICMP
- introduce and explain the basic usage of Nmap
- introduce and explain the basic usage of Hping3
- introduce and explain the basic usage of Nessus

INTRODUCTION TO SCANNING

After the penetration tester has completed the reconnaissance phase of an organization, they will move into the scanning phase. In this phase, the penetration tester can take the information learned about the employees, contractors, and information systems to begin expanding the view of physical and logical information system structures within the organization. Like any of the other phases in the penetration testing lifecycle, the penetration tester can return to earlier phases as needed to gain more information to enhance information gathered in the scanning phase.

The main focus of the scanning phase is to determine specific information about the computers and other devices that are connected to the targeted

Hacking with Kali. DOI: http://dx.doi.org/10.1016/B978-0-12-407749-2.00008-2

network of the organization. Throughout this phase, the focus is on finding live hosts, determining node type (desktop, laptop, server, network device, or mobile computing platform), operating system, public services offered (web applications, SMTP, FTP, etc.), and even possible vulnerabilities. Vulnerabilities at this level are often referred to as, "low hanging fruit." Scanning is done with a number of different tools; however, this chapter will focus on some of the best known and most effective tools including Nmap, Hping, and Nessus. The goal of this phase is to have a listing of possible targets for the next phase of the penetration testing lifecycle: exploitation.

UNDERSTANDING NETWORK TRAFFIC

Network traffic can be confusing to some people; however, a basic understanding of this topic is required to obtain the maximum benefit from the scanning phase. Network traffic is the electronic communication that occurs between computer systems that are connected by a number of different methods. Today the most common methods of networking are Wired and Wireless Ethernet. Understanding of the fundamental principles of Ethernet communication is necessary. This chapter will introduce ports and firewalls, IP protocols including Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Management Protocol (ICMP).

Understanding Ports and Firewalls

One of the most basic methods of defending a network is by implementing a firewall between the internal, often corporate, network and the rest of the world, most likely the Internet. A firewall is simply a computing device with two or more network cards serving as a gatekeeper for the network. Access control lists strictly monitor outbound traffic (*egress*) and inbound traffic (*ingress*). Only traffic that meets the criteria of the access controls is allowed to pass, while the rest are dropped by the firewall. It does this by opening or closing ports to allow or deny traffic.

Ports are the different communication channels used for computer to computer communication. There are 65,535 TCP ports and another 65,535 UDP ports that can be used for communication. A small percentage of these ports are designated for a specific purpose, but are not restricted to this use. For example, the TCP port 80 is most often used for normal Internet web traffic utilizing the Hypertext Transfer Protocol (HTTP), but other traffic can travel over port 80 and Internet traffic can be transmitted over other ports.

One way to think of ports is a large office building with doors leading to different rooms. Each of these rooms has an office staff that does a specific job and manages different functions. The office behind suite number 80 handles the web page request that come in. It is possible for the web department to move to a different office, say the office in suite 8080, continuing to do the same functions, handling web requests, there. A different group could move into suite 80 that has nothing to do with the web requests or the suite could be simply closed, locked, and unused. Visitors trying to find the web team would need to know the web team is now in suite 8080 and no longer in suite 80. A visitor trying to get web information from suite 80 after the web team has moved will be disappointed and not get the needed information as the wrong people will be there or the office will be locked, while a visitor that has the correct address will get the web page requested from the new office in suite 8080.

Understanding IP Protocols

Protocols are rules, whether in real life or on computer networks. Diplomats, politicians, and high-level officials often have special staff members that handle protocol issues. The people in protocol offices make sure every visitor or official message is done in a manner that ensures the message or visitor is received correctly, in the correct format and with the right titles and honors. In the computer world, these protocols ensure communication between systems occurs according to predefined rules. While there is an extremely high number of protocols available for all computer systems, this chapter will address three of the protocols most commonly used by popular scanning applications on Kali Linux use to leverage scanning, vulnerability discovery, and penetration testing: TCP, UDP, and ICMP.

TCP

One of the main protocols used for network communications is the TCP. TCP is a connection-based communication protocol, meaning that the computers on each side of the communications channel acknowledge that the session is open and the messages are being received on each side of the connection. In the past, many people have related this to a phone call.

Phone rings

Charlie:	"Hello"
Denis:	"Hi, is Charlie there?"
Charlie:	"This is Charlie."

While this analogy is a bit backward, it does illustrate a three-way handshake similar to the connection used to initiate a TCP communication stream. In TCP communication, the three-packet exchange, wherein communication is initiated by the computer attempting to connect to another computer. This is done by sending a synchronization flagged packet or request commonly referred to as a SYN. The computer on the receiving end of the communication, and if available, will reply to the sender with a packet using the acknowledgment and synchronization flags set, this TCP packet is known as the SYN/ACK packet. Finally, the computer that initiated the communication sends a packet with the acknowledgment (ACK) flag set to complete synchronization and establish the connection. This communication looks like the illustration in Figure 8.1.

The three-way handshake is the method that all correctly formed TCP communications start and ensures the computers on both ends of the communication channel are synchronized with each other. Later in this chapter, this handshake protocol will be exploited to help identify computers on the network in a way that attempts to avoid detection.

This acknowledgment process continues throughout the communication session between the computers. This helps ensure the messages sent by one computer are all received by the other computer and any packets that do not make the voyage are resent by the first computer. This works similarly to feedback in verbal communication.

Denis:	"I would like you to meet me at the restaurant at 3:00 PM."
Charlie:	"What time did you want to meet me at the restaurant?"
Denis:	"3:00 PM"
Charlie:	"Okay 3:00 PM it is."

This creates a lot of overhead on the network, normally consuming a lot of bandwidth and taking a bit longer for the communications to take place. For this reason, it is often used for communications sessions that need a level of reliability and will not be impacted by the latency of a packet arriving at the distant end out of order (programs using this protocol will reassemble the packets in the correct order even if they arrive out of sequence). Common

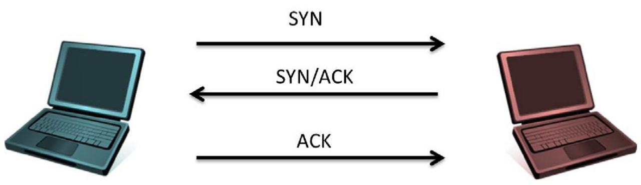


FIGURE 8.1
TCP three-way handshake.

processes that use the TCP communication process include file transfer (FTP), web traffic (HTTP), and email SMTP, POP, and IMAP).

UDP

The UDP is a protocol that has less overhead than the TCP connections. If the TCP communication process is analogous to a phone call where both parties ensure the communication is being received as sent at both sides of the communications channel, UDP is more like a radio broadcast where the communication is sent out and neither the sender or receiver verify, by default, that a communication packet has been received.

> Radio Station: "This is XHAK radio; join us all at the restaurant at 3:00 PM today"

This broadcast is sent over the air and if it is received on the recipients end, great. If part of message is not received at the destination; by default, the receiver will not ask for retransmission of the pack. There are a few exceptions to this rule; unfortunately, this is an advanced topic outside the scope of this chapter. When working with communications utilizing UDP, the receiving end does not confirm the status of the communication link or if packets were dropped during transmission.

This lower overhead communication method is ideally suited for tasks that do not require validation of each package or services that would be adversely impacted if a packet arrived out of order. Applications that use UDP communications value lower overhead and higher speed over the increased reliability, such as streaming video and music.

ICMP

The ICMP is, by design, a health and maintenance protocol for the network. This protocol is used to determine if a device on the network is working as it should be and can communicate correctly. In most cases, end users will never directly use applications that rely on ICMP; however, as with any rule there are always exceptions. In this case, traceroute and Ping are good examples of exceptions. Another difference is that, unlike TCP and UDP communications, the communication method is not designed to carry user data. Instead ICMP transports system messages to and from network devices, computers, and application services.

ICMP messages have a specific type and code, or number set, contained in their header. These sets are used to ask questions or provide information about the various nodes on the network. These type and code sets can assist the penetration tester in determining what the systems are on the target system (Figure 8.2).

Type	Code	Description
0 (Echo Reply)	0	Echo Reply
3 (Destination Unreachable)	0	Destination Network Unreachable
	1	Destination Host Unreachable
	2	Destination Protocol Unreachable
	3	Destination Port Unreachable
	6	Destination Network Unknown
	7	Destination Host Unknown
	9	Network Administratively Prohibited
	10	Host Administratively Prohibited
	13	Communication Administratively Prohibited
8 (Echo Request)	0	Echo Request

FIGURE 8.2
ICMP table.

PING

Ping is likely the most directly used ICMP-based command by an end user or administrator. The Ping command sends an ICMP packet with a type of 8 and a code of 0 indicating this packet is an echo request. Machines receiving this packet, and (*usually by default*) if configured to respond, will reply with another ICMP packet with a type of 0 and code of 0 indicating an echo reply. A successful Ping and response would indicate that the system queried is operating on the network or considered to be a "live host." A Ping request from a Windows platform will by default send the Ping request four times, while Ping requests from a Linux hosts will continue trying to Ping until the request is canceled by the user. To cancel a Linux Ping press the control and "c" keys on the keyboard. A successful and unsuccessful Ping would look like this:

Live Host

```
Ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes = 32 time = 2ms TTL = 64
Reply from 192.168.1.1: bytes = 32 time = 1ms TTL = 64
Reply from 192.168.1.1: bytes = 32 time = 1ms TTL = 64
Reply from 192.168.1.1: bytes = 32 time < 1ms TTL = 64
```

Host Unreachable

```
Ping 192.168.1.200
Pinging 192.168.1.200 with 32 bytes of data:
Reply from 192.168.1.129: Destination host unreachable.
Reply from 192.168.1.129: Destination host unreachable.
Reply from 192.168.1.129: Destination host unreachable.
Reply from 192.168.1.129: Destination host unreachable.
Ping statistics for 192.168.1.200:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
```

Traceroute

Traceroute uses ICMP's Ping command to find out how many different devices are between the computer initiating the traceroute and the target. This command works by manipulating the packets time to live value or TTL. The TTL is the number of times the packet can be rebroadcast by the next host encountered on the network or hops. The command will start with a TTL value of 1 indicating the packet can only go as far as the next device between the initiator and the target. The receiving device will send back an ICMP type 11, code 0 packet (*time exceeded*), and the packet is logged. The sender increases the TTL by 1 and sends the next series of packets. The packets will reach their expected time to live at the next hop along the network; which in turn, causes the receiving router to send another time exceeded reply. This continues until the target is reached, and all hops along the way have been recorded, creating a listing of all devices between the initiating computer and the target. This can be helpful for a penetration tester when determining what devices are on a network. Windows platforms have a default TTL of 128, Linux platforms start with a TTL of 64, and Cisco networking devices have a whopPing TTL of 255.

The traceroute command in Windows is `tracert`. On a Linux system, like Kali, the command is `traceroute`. A typical tracert on a Windows machine would look like the following.

```
tracert www.google.com
Tracing route to www.google.com [74.125.227.179]
over a maximum of 30 hops:
    1 1 ms <1 ms 1 ms 192.168.1.1
    2 7 ms 6 ms 6 ms 10.10.1.2
    3 7 ms 8 ms 7 ms 10.10.1.45
    4 9 ms 8 ms 8 ms 10.10.25.45
    5 9 ms 10 ms 9 ms 10.10.85.99
    6 11 ms 51 ms 10 ms 10.10.64.2
    7 11 ms 10 ms 10 ms 10.10.5.88
    8 11 ms 10 ms 11 ms 216.239.46.248
    9 12 ms 12 ms 12 ms 72.14.236.98
    10 18 ms 18 ms 18 ms 66.249.95.231
    11 25 ms 24 ms 24 ms 216.239.48.4
    12 48 ms 46 ms 46 ms 72.14.237.213
    13 50 ms 50 ms 50 ms 72.14.237.214
    14 48 ms 48 ms 48 ms 64.233.174.137
    15 47 ms 47 ms 46 ms dfw06s32-in-f19.1e100.net [74.125.227.179]
Trace complete.
```

Many of the scanning tools on Kali make use of protocols like TCP, UDP, and ICMP to map out target networks. The result of successful scanning phase is a listing of hosts, IP addresses, operating systems, and services. Some scanning tools can also uncover vulnerabilities and user details. These details will greatly enhance the exploitation phase as attacks in this phase can be better targeted at specific hosts, technologies, or vulnerabilities.

NMAP THE KING OF SCANNERS

Nmap has the ability to determine not only the computers that are active on the target network, but in many cases, it can also determine operating system, listening ports, services, and possibly user credentials. By using a combination of commands, switches, and options against targets, Nmap can be a great asset in the scanning phase of the penetration testing engagement.

The Nmap Command Structure

Nmap's command switches have a very distinct structure allowing command options and targets to be assembled in a manner that supports maximum flexibility. A typical, but quite basic, command is illustrated in Figure 8.3, detailing the several basic parts that tell the scanning engine what to do.

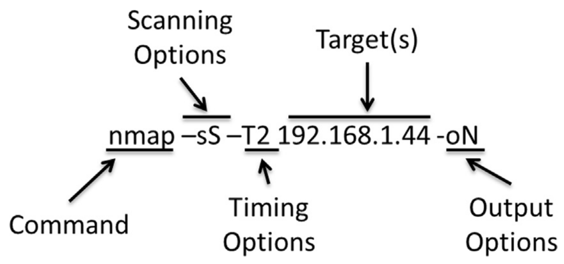


FIGURE 8.3
Nmap command structure.

With the exception of the Nmap command itself, each of these options will be covered in more detail in the sections that follow. The command switches and options tell the operating system what program to run, in this case Nmap, and what is specifically required to properly execute the task. Following the command is the scanning options, in this case the stealth scan is signified by the "-sS" switch. Next is the timing options that tell the engine how much traffic to generate and how fast to generate it, ultimately determining how fast or slow the scan will run. In this example, the target option follows the timing options and is the only other required portion of the command needed to conduct an Nmap scan. The final option in this example is the output option telling the application where to send the results of the scan. Nmap scanning commands can be far more complex or much more basic than the command and options string in Figure 8.3. For example the following is all that is needed to conduct a complete Nmap command statement resulting in a scan of the target. In this case, the target is the Metasploitable2 virtual machine from the lab that was described in an earlier chapter of this book.

```
nmap 10.0.2.100
```

By default Nmap will conduct a stealth scan of the target at 10.0.2.100 using the normal time template (T3) speed if no options are set as in the example above. Additionally, the scan results are output to the monitor (if defined as standard output). This basic scan illustrates one end of the Nmap spectrum, with the other end being complex and lengthy scans that define detailed actions that Nmap will complete. Advanced usage includes executing detailed scripts written for Nmap using the Nmap Scripting Engine (NSE).

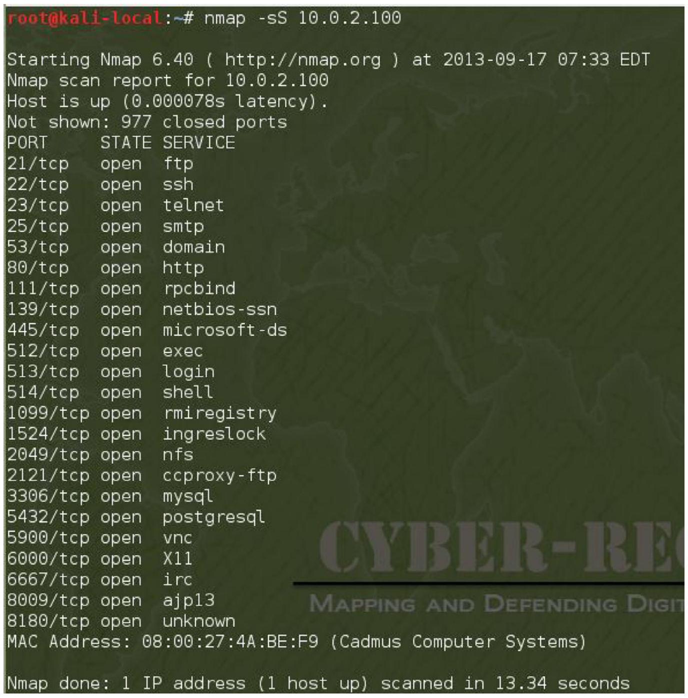
To better understand the details of basic Nmap scans, the next few sections will detail the options that will enhance the use of Nmap as a scanning tool that helps define targets in the penetration testing engagement. These sections not only scratch the surface of Nmap but will give the reader a solid understanding of what the tool can do. The sections will cover the scanning, the timing, targeting, and output options. Following these sections, the basic use of preconfigured Nmap scripts will be covered.

Scanning Options

The use of the "-s" (lowercase s) scanning prefix alerts the Nmap scanning engine that the user is specifying a specific type of scan should be conducted on the target(s) defined in the scan command. The lowercase "s" is followed by an upper case letter that will identify the scan type. The selection of scan type can assist the penetration tester in evading detection by some host and network-based protection systems and may even circumvent network protections like firewalls.

—sS Stealth Scan

The stealth scan is the default scan option used by Nmap when no scan option is defined. The stealth scan can also be intentionally initiated when the —sS option is set in the command string. This scan initiates a TCP connection with the target but never completes the three-way handshake. The Nessus engine initiates the handshake by sending the target machine a SYN packet. The target machine will hopefully reply with a SYN/ACK packet that is not acknowledged by the Nessus engine. This leaves a connection open, as the communication channel is never completely built. Most systems close this connection automatically after a certain time period. In older and poorly configured systems, this type of connection can go undetected, so this type of scan is often associated with a more clandestine and considered less noisy scan of the target. Today many network systems and even hosts can detect the stealth scan; however, this should not deter the penetration tester from scanning with this technique as it will often be harder to detect than other scans and if the system being targeted is poorly configured, the scan may go totally unnoticed even today. This scan technique is illustrated in Figure 8.4.



```
root@kali-local:~# nmap -sS 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 07:33 EDT
Nmap scan report for 10.0.2.100
Host is up (0.000078s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 13.34 seconds
```

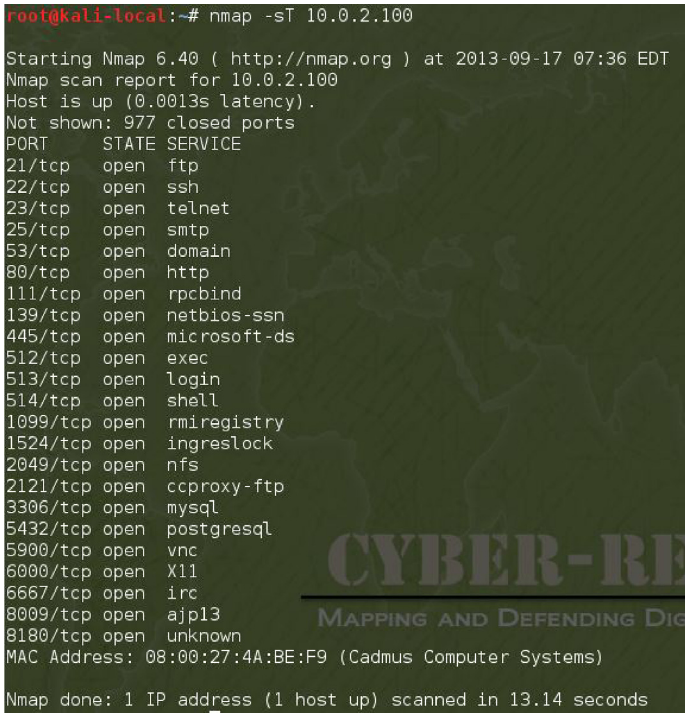
FIGURE 8.4
Stealth scan.

–sT TCP Connect Scan

The TCP connect scan can often be used to gather more information about the target than the stealth scan as a full TCP connection is made with the targeted host. In this case, the Nessus engine initiates a SYN packet that is hopefully acknowledged by the target with a SYN/ACK reply. Unlike the stealth scan, this time the Nessus engine completed the communication path by sending a final ACK packet. This scan is logged on most systems but can normally provide more information than the stealth scan (Figure 8.5).

–sU UDP Scan

The UDP scan assesses the UDP ports on the target system. Unlike scanning TCP ports, UDP scans expect to receive replies back from systems that have the tested ports closed. Packets sent to open UDP ports are not responded; however, if the packet sent elicits a response from the target, then the port being probed is open. If no response is received, then the port could be open or could be filtered by a device like a firewall. Closed UDP ports can be identified by an ICMP response with a type 3 and code 3 response (port unreachable). Finally, ports that are confirmed to be filtered will have an ICMP



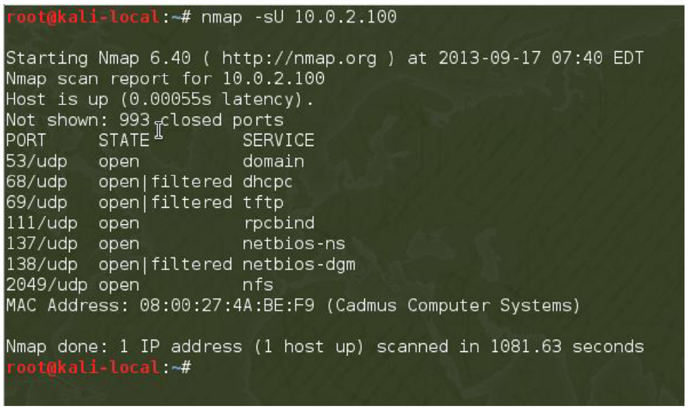
```
root@kali-local:~# nmap -sT 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 07:36 EDT
Nmap scan report for 10.0.2.100
Host is up (0.0013s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 13.14 seconds
```

FIGURE 8.5
Connect scan.

response of type 3 with codes of 1, 2, 9, 10, or 13, indicating various unreachable errors (Figure 8.6).

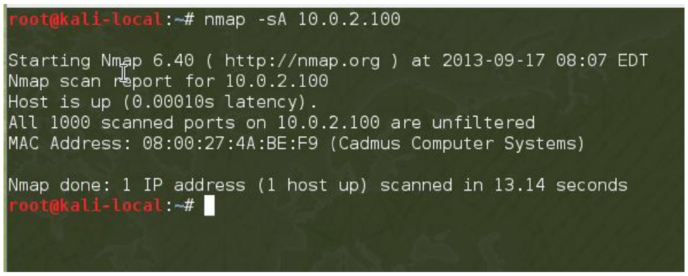
–sA

The ACK scan, –sA, is used to try to determine if a TCP port is filtered or unfiltered. This scan initiates communications with the target with the acknowledgment (ACK) flag set. This type of scan sometimes can bypass certain firewalls by posing as a response (ACK) to an internally sent request. For example, a SYN packet is sent from the target computer, even though this internal computer never sent a request. A reset (RST) response to this scan indicates that the queried port is unfiltered. If no response is received or if a type 3 ICMP response with a code of 1, 2 , 3, 9, 10, or 13 (unreachable error) indicates that the port is filtered (Figure 8.7).



```
root@kali-local:~# nmap -sU 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 07:40 EDT
Nmap scan report for 10.0.2.100
Host is up (0.00055s latency).
Not shown: 993 closed ports
PORT     STATE         SERVICE
53/udp   open          domain
68/udp   open|filtered dhcpc
69/udp   open|filtered tftp
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
2049/udp open          nfs
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 1081.63 seconds
root@kali-local:~#
```

FIGURE 8.6
UDP scan.



```
root@kali-local:~# nmap -sA 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 08:07 EDT
Nmap scan report for 10.0.2.100
Host is up (0.00010s latency).
All 1000 scanned ports on 10.0.2.100 are unfiltered
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 13.14 seconds
root@kali-local:~#
```

FIGURE 8.7
Nmap ACK scan.

Timing Templates

As stated above, the default timing template used by Nmap if no timing switch is set is –T3 or normal. Nmap has the built-in ability to let the user override this functionality to scan the target set faster or slower than the normal default speed. There are a number of different settings that are adjusted based on the timing template that is selected, but the most illustrative are the delays between scanning probes and parallel processing status. For this reason, the scan_delay, max_scan_delay, and max_parallelism options will be used to explain each of the different timing templates. These options provide a good method to measure each of the timing templates to ensure the correct template is set for the engagement and target network. The scan_delay setting sets the minimum pause between probes sent to the target machine while the max_scan_delay indicates the maximum time the scanner will allow the scanning delay to grow based on target and network settings. This can be important as some systems will only reply to probes at a specific rate. Nmap will automatically adjust the probe timing to match the requirements of the system or network up to the max_scan_delay setting. Max_parallelism instructs the system to either send one probe at a time for serial scans or multiple probes at the same time for parallel scans.

The following examples will all use the same target, the Metasploitable2 virtual machine with the –sU (UDP scan) switch set. While it has not been introduced the example will use the port switch (-p) to indicate the first 500 ports should be scanned with the –p 1–500 switch combination. The Nmap command for this will look like the following; however, the hash tag (#) will be replaced with the number of the template to be used for that specific example. This way the timing of the scans can be compared to each other. While the –T# switch is being used in this example, the English text could also be used to achieve the same results, therefore –T5 and –timing insane result in the same scan being run.

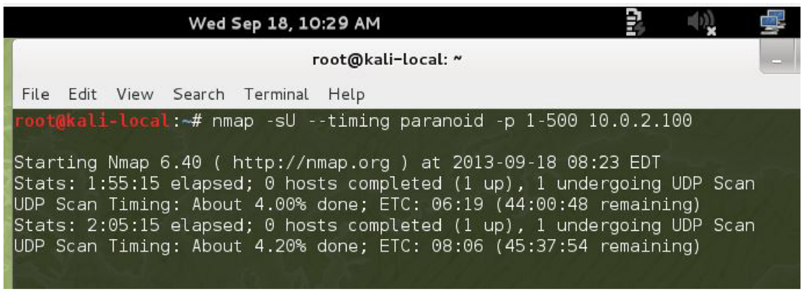
```
nmap –sU –T# p 1-500 10.0.2.100
```

Or

```
nmap –sU --timing paranoid –p 1-500 10.0.2.100
```

–T0 Paranoid

The –T0 or Paranoid scan is used for slow network links or in situations where detection risks must be minimized. This is a serial scan that will pause for a base minimum of 5 minutes; however, the max_delay setting of second is ignored as the base scan_delay is set to a value higher than this default value. It is easy to see the amount of time needed to complete the paranoid scan on only 500 UDP ports on a single computer in Figure 8.8. In



```
root@kali-local:~# nmap -sU --timing paranoid -p 1-500 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-18 08:23 EDT
Stats: 1:55:15 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 4.00% done; ETC: 06:19 (44:00:48 remaining)
Stats: 2:05:15 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 4.20% done; ETC: 08:06 (45:37:54 remaining)
```

FIGURE 8.8
Paranoid scan.

Figure 8.8, the system time is displayed at the top of the figure as 10:29 AM and the scan start time was 8:23 AM indication the scan has been running for over 2 hours. The last line indicates that the scan will complete in another 45 hours and 37 minutes. This scan can be effective but should be used when stealth is required and a lot of time is available.

–T1 Sneaky

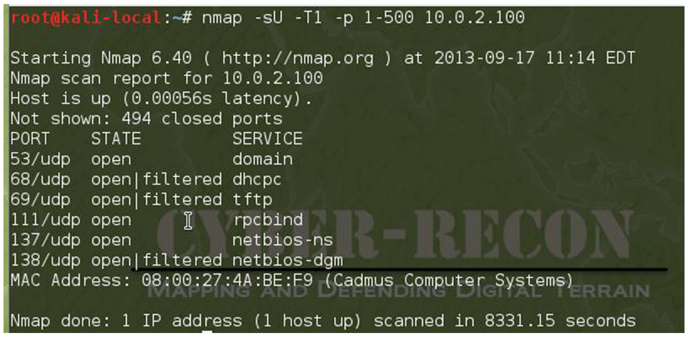
The –T1 or --timing sneaky scan is slightly faster than the paranoid scan, reducing the scan time needed while maintaining some of the stealth inherent in a slower scan. This scan also uses a serial process for querying the target, but reduces the scan_delay quite dramatically to 15 seconds. While the scan_delay is reduced, it is still a larger value than the max_scan_delay so this second value is ignored. The difference in speed between this scan and the –T0 scan is illustrated in Figure 8.9, reducing the scan time to 8331 seconds or 138 minutes.

–T2 Polite

The –T2 or --timing polite scan is an increase in speed again over the –T0 and –T1 scan and is the last scanning template to use the serial scanning technique. The scan_delay for this scan is set to 400 milliseconds, making this the first template to make use of the max_scan delay, a value that is still set to the default value of 1 second. With this template selected Nmap will begin scanning targets using the scan_delay of 400 milliseconds but has the ability to dynamically adjust the delay up to a maximum of 1 second. By examining the time required to complete the polite scan of the same 500 ports, overall scanning time has been reduced to just 544 seconds or just 9 minutes (Figure 8.10).

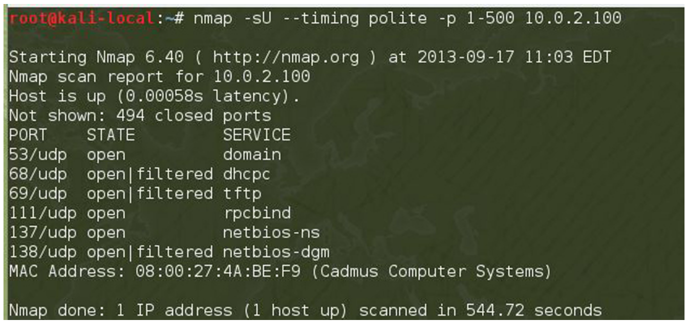
–T3 Normal

The –T3 or --timing normal scan is the default scan for Nmap, meaning that if no timing template or manual timing options are set, the settings in this template will be used for the scan. This template is the first to use the



```
root@kali-local:~# nmap -sU -T1 -p 1-500 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 11:14 EDT
Nmap scan report for 10.0.2.100
Host is up (0.00056s latency).
Not shown: 494 closed ports
PORT    STATE         SERVICE
53/udp  open          domain
68/udp  open|filtered dhcpc
69/udp  open|filtered tftp
111/udp open          rpcbind
137/udp open          netbios-ns
138/udp open|filtered netbios-dgm
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 8331.15 seconds
```

FIGURE 8.9
Sneaky scan.



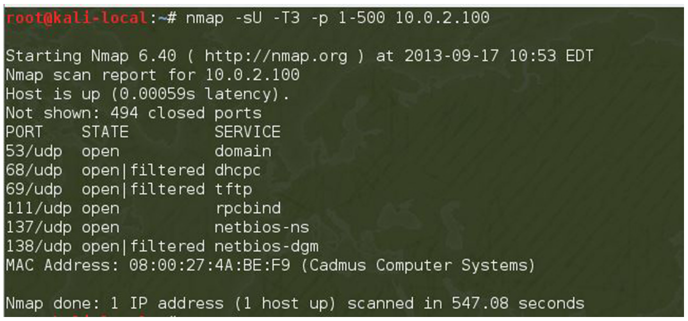
```
root@kali-local:~# nmap -sU --timing polite -p 1-500 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 11:03 EDT
Nmap scan report for 10.0.2.100
Host is up (0.00058s latency).
Not shown: 494 closed ports
PORT    STATE         SERVICE
53/udp  open          domain
68/udp  open|filtered dhcpc
69/udp  open|filtered tftp
111/udp open          rpcbind
137/udp open          netbios-ns
138/udp open|filtered netbios-dgm
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 544.72 seconds
```

FIGURE 8.10
Polite scan.

parallel processing technique, sending multiple probes out simultaneously, increasing the overall speed. This scan has a scan_delay of 0 seconds that can grow to a max_scan_delay that can grow to 1 second, meaning the scan will occur as quickly as possible but after 1 second the current port scan will be abandoned and the next port will be scanned. The normal scan will complete the scan of selected ports on the target computer in 547 seconds, actually slower than the polite scan in this case, however this is not normally the case. This is one of the strange quirks of scanning, at times things will align and a scan that should be slower really is not that much slower. This is why the successful penetration tester should be familiar with all of the tools in his or her arsenal to know how to best employ them (Figure 8.11).

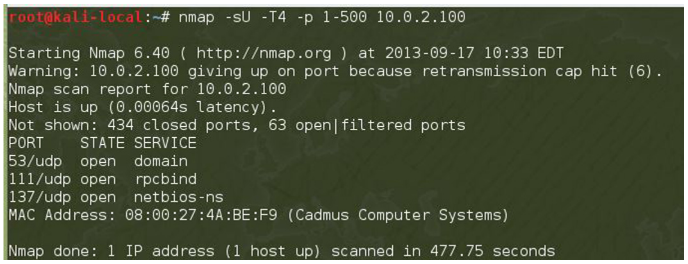
–T4 Aggressive

The –T4 or --timing aggressive template also runs its scanning in parallel increasing speed. The scan_delay for this template is set to 0 seconds and can grow to a max_scan_delay of 10 milliseconds. Scans with a max_scan_delay



```
root@kali-local:~# nmap -sU -T3 -p 1-500 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 10:53 EDT
Nmap scan report for 10.0.2.100
Host is up (0.00059s latency).
Not shown: 494 closed ports
PORT     STATE         SERVICE
53/udp   open          domain
68/udp   open|filtered dhcpc
69/udp   open|filtered tftp
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 547.08 seconds
```

FIGURE 8.11
Normal scan.



```
root@kali-local:~# nmap -sU -T4 -p 1-500 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 10:33 EDT
Warning: 10.0.2.100 giving up on port because retransmission cap hit (6).
Nmap scan report for 10.0.2.100
Host is up (0.00064s latency).
Not shown: 434 closed ports, 63 open|filtered ports
PORT     STATE SERVICE
53/udp   open  domain
111/udp  open  rpcbind
137/udp  open  netbios-ns
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 477.75 seconds
```

FIGURE 8.12
Aggressive scan.

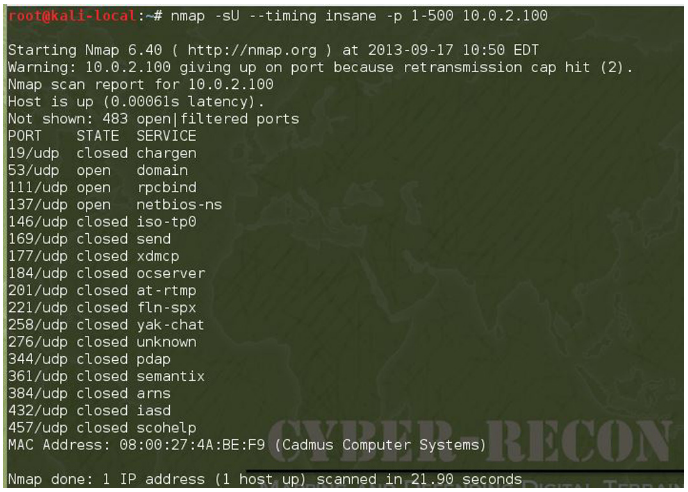
of less than 1 second are prone to errors as some target operating systems have settings that require a minimum delay between probe responses of 1 second. This scan completed the port scan of the metasploit virtual machine in just 477 seconds or just under 8 minutes (Figure 8.12).

–*T5 Insane*

The –T5 or --timing insane timing template is the fastest of the built-in timing templates. This template uses the parallel scanning technique with a scan_delay of 0 seconds and a max_scan_delay of 5 milliseconds. As stated with the –Aggressive scan, this scan can cause errors based on target machine operating systems and settings. This scan, the fastest, completed in just under 22 seconds; however, the results are quite a bit different than all of the scans to this point (Figure 8.13).

Targeting

Identifying the target or target set for an Nmap scan is one of the most important parts of the Nmap command string. Defining the wrong targets can result in scanning empty IP space or worse yet computers that are not covered by the



```
root@kali-local:~# nmap -sU --timing insane -p 1-500 10.0.2.100

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-17 10:50 EDT
Warning: 10.0.2.100 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.0.2.100
Host is up (0.00061s latency).
Not shown: 483 open|filtered ports
PORT    STATE  SERVICE
19/udp  closed chargen
53/udp  open   domain
111/udp open   rpcbind
137/udp open   netbios-ns
146/udp closed iso-tp0
169/udp closed send
177/udp closed xdmcp
184/udp closed ocserver
201/udp closed at-rtmp
221/udp closed fln-spx
258/udp closed yak-chat
276/udp closed unknown
344/udp closed pdap
361/udp closed semantix
384/udp closed arns
432/udp closed iasd
457/udp closed scohelp
MAC Address: 08:00:27:4A:BE:F9 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 21.90 seconds
```

FIGURE 8.13
Insane scan.

Rules of Engagement (ROE). There are a number of ways that a target set can be defined in the scan statement string. Of these methods, the two described in this book are the IP address range and using a scan list.

IP Address Ranges

Defining a set of targets using an IP address range is quite straightforward. For this example the address range will be the 10.0.2.x class c address range. This will mean that the maximum number of hosts that can be included in the scan is 254. To scan all of the hosts, use the following command.

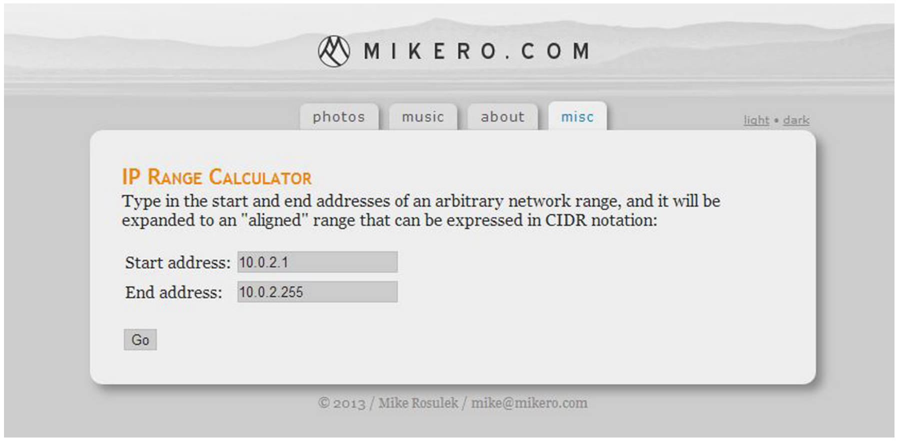
```
nmap 10.0.2.1-255
```

This same scan can be completed using the CIDR method of addressing by using the /24 postfix as follows. CIDR addressing is a quick way to select a range of addresses but CIDR addressing is beyond the scope of this book. A quick way to define a CIDR range without completing all of the calculations is by using one of the online calculators like the one at http://www.mikero.com/misc/ipcalc/. To use this enter the starting and ending addresses in the IP Range boxes, and click the Convert button (Figure 8.14). There are a number of good references that can be used to learn more about CIDR addressing.

```
nmap 10.0.2.1/24
```

A smaller set of IP addresses can be identified in the scan by defining the smaller IP range. In this example, the first 100 addresses will be scanned.

```
nmap 10.0.2.1-100
```



IP RANGE CALCULATOR

Type in the start and end addresses of an arbitrary network range, and it will be expanded to an "aligned" range that can be expressed in CIDR notation:

Start address: 10.0.2.1

End address: 10.0.2.255

Go

© 2013 / Mike Rosulek / mike@mikero.com

FIGURE 8.14
CIDR conversion.

or using the CIDR

```
nmap 10.0.2.0/25
```

Scan List

Nmap can also use a text file as input for the target list. Assume that the following addresses are stored in a file called targets.txt.

```
10.0.2.1
10.0.2.15
10.0.2.55
10.0.2.100
```

The command to use this file would look like the following.

```
nmap –iL targets.txt
```

SELECTING PORTS

Selecting ports can be done by using the –p switch in the scan command. The ports can be continuous by using a dash in the command. Selected ports can also be identified by using commas in the command.

```
nmap –sS –p 1-100
nmap –sU –p 53,137,138,161,162
```

(or use both) `nmap -sS -p 1-100,445,8000-9000`

Output Options

There are many times that the penetration tester does not want the Nmap scan to be output to the screen but rather saved to a file. This can be done by redirecting with the pipe command (|), but for this chapter the Nmap scan output options will be described. These include normal, XML, and GREPable. For all of these examples, the metasploitable target at 10.0.2.100 will be used and the appropriate extension will be used with the file name "metascan".

–oN Normal Output

The normal output option will create a text file that can be used to evaluate the scan results or use as input for other programs.

```
nmap –oN metascan.txt 10.0.2.100
```

–oX Extensible Markup Language (XML) Output

XML output can be used for input into a number of different applications for further processing or evaluation.

```
nmap –oX metascan.xml 10.0.2.100
```

–oG GREPable Output

GREPable output is often used by penetration testers to allow further investigation using tools like GREP, but can also be searched using tools like AWK, SED, and DIFF.

```
nmap –oG metascan.txt 10.0.2.100
```

–oS ScRipT Kidd|# oUTpuT

While not used for serious penetration testing, the script kiddie output can be fun to use from time to time. This output method should not be used for serious scans as it uses the "leet" speak used by many that most penetration testers would call "script Kiddies."

```
nmap –oS metascan.txt 10.0.2.100
```

Nmap Scripting Engine

Building custom scripts for Nmap is beyond the scope of this book; however, the ability to use preconfigured scripts can be quite helpful for conducting penetration tests. The full set of preconfigured scripts can be found at http://nmap.org/nsedoc/. For this example the script to get the targets NetBIOS and MAC address information. To tell the Nmap scanning engine that a script will be used the --script flag is used as in the example.

```
nmap --script nbstat.nse 10.0.2.100
```

Nmap is constantly involved in the development of new scripts for community use. A security tester will want to make sure that the scripting database within Nmap is as up-to-date as possible. It is recommended that the database be updated before heading out on mission. To update the Nmap database:

```
nmap --script-updatedb
```

HPING3

Hping is an application that can be used to manually craft packets to be placed on the network. This is a manual process to create packets in a way that is similar to the way the Nmap engine automatically creates packets. For example, Hping3 can create a series of synchronization packets by using the –S flag.

```
hping3 –S 10.0.2.100
```

Full information about Hping3 can be found in the help file by using the –h switch.

```
Hping3 -h
```

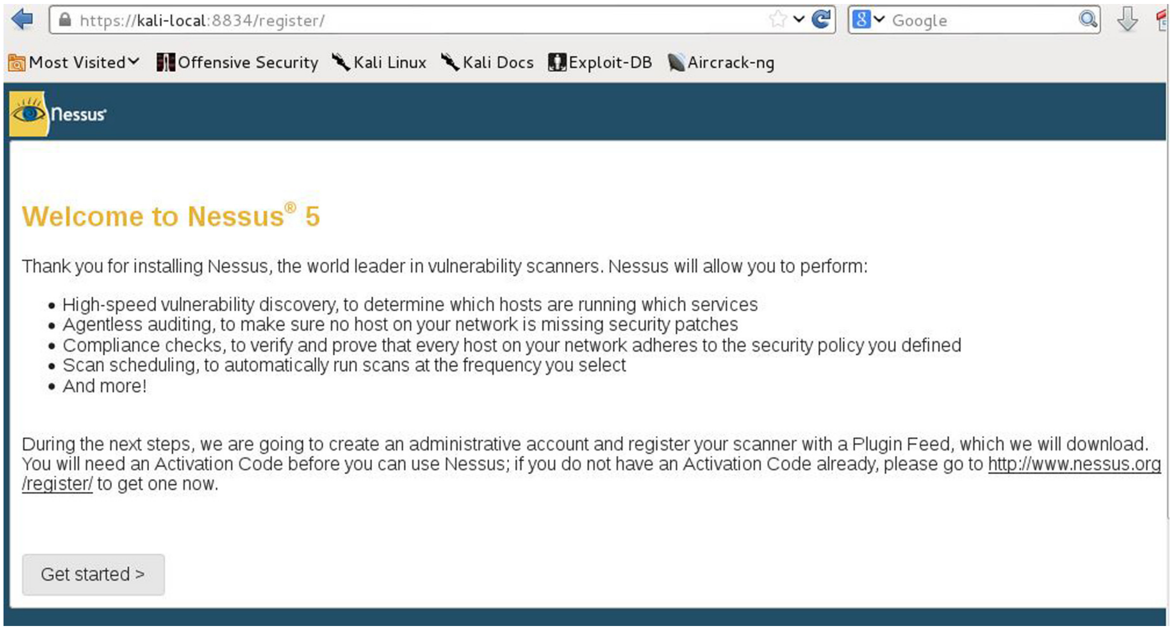
NESSUS

After the installation of Nessus is complete, as described in Chapter 3, start the Nessus scanner with the following command.

```
/etc/init.d/nessusd start
```

Once the scanner is started, open the IceWeasel web browser and navigate to https://localhost:8834/. The number after the colon, in this case 8834, tells the browser to connect to the local machine on port 8834 instead of the default. It is important to check with the Nessus documentation to ensure that the correct port is being used to connect to the console, some versions may use a different port. For web browsers, the default port is 80 and a user trying to access kali-local at the default port of 80 will find it closed or offering services that are not compatible with the IceWeasel web browser.

Connecting on this port with the IceWeasel browser will open the Nessus Console, a GUI that allows the user to set up, configure, and scan using the Nessus engine. The first page that will be displayed will be the registration window as seen in Figure 8.15 that will allow the user to register the application. Registration is required to get Nessus Home Feed updates, files, and



Welcome to Nessus® 5

Thank you for installing Nessus, the world leader in vulnerability scanners. Nessus will allow you to perform:

- High-speed vulnerability discovery, to determine which hosts are running which services
- Agentless auditing, to make sure no host on your network is missing security patches
- Compliance checks, to verify and prove that every host on your network adheres to the security policy you defined
- Scan scheduling, to automatically run scans at the frequency you select
- And more!

During the next steps, we are going to create an administrative account and register your scanner with a Plugin Feed, which we will download. You will need an Activation Code before you can use Nessus; if you do not have an Activation Code already, please go to http://www.nessus.org/register/ to get one now.

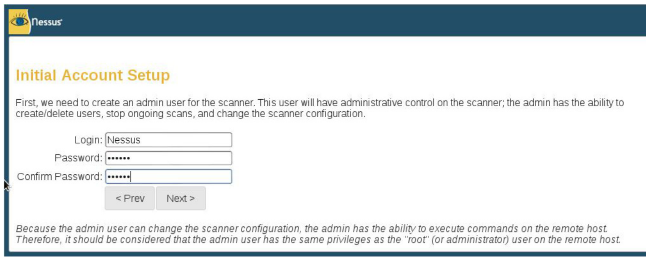
FIGURE 8.15
Nessus registration.

other information. If a valid Nessus activation code is not available click the Get Started button to begin the registration process.

The next screen is used to set up the initial administrator account. Create this account by filling in the login and password fields on this page ensuring the password. For this example the user name will be set as Nessus and password of Nessus will be used, a combination that should only be used in test environments. Select a user name and password combination that will meet system requirements, and click the Next button.

The next screen is used to activate the Nessus Feed Plugin. The "I already have an Activation Code button" is used for users that have previously registered with Nessus. Simply click this button and enter the Activation Code. For this example the "I will use Nessus to scan my Home Network" is selected. Enter first and last name as well as email address. If a proxy is on the network, click the Proxy Settings button and enter the appropriate information. No proxy is used for this example so the next button will be clicked.

Successful registration will result in the display of a screen displaying a successful registration was achieved. A button also displayed on this screen that


Nessus

Initial Account Setup

First, we need to create an admin user for the scanner. This user will have administrative control on the scanner; the admin has the ability to create/delete users, stop ongoing scans, and change the scanner configuration.

Login: Nessus
Password: ••••••
Confirm Password: ••••••

Because the admin user can change the scanner configuration, the admin has the ability to execute commands on the remote host. Therefore, it should be considered that the admin user has the same privileges as the "root" (or administrator) user on the remote host.

FIGURE 8.16
Initial Nessus setup.

will allow the most current plugins to be downloaded. Click the Next: Download plugins button.

Once the plugins have downloaded, the login dialog box will be displayed. Enter the administrator user name and password created earlier. Then click the "Sign in To Continue" button to log in. This will complete the initial Nessus setup and a dashboard similar to Figure 8.16.

Scanning with Nessus

After Nessus has been installed, it is important to understand how to set up and scan a network or system using the application. This example will be completed using the lab created earlier in the book. The metasploitable2 virtual machine has been configured with an IP address of 10.0.2.100 and the Kali virtual machine has an IP address of 10.0.2.15. The virtual machines have had the network adapter setting in the VirtualBox console set to Internal to ensure no unauthorized scanning occurs on the outside network and to ensure the metasploitable2 virtual machine is not accessible by outside users. Once both machines are up and running, Nessus can be configured and the scan can be conducted.

Adding a Nessus User

It is advisable to create a different account for each user that will be using the Nessus Console. The accounts should be linked to individual users, if possible and not be shared. To create a user, select the Users tab and then select the " + New User" button. This will open a dialog box where the user credentials can be entered (Figure 8.17). Use this dialog box to enter the user name and the password (twice). If the user is an administrator, check the "Administrator" check box. Once all of the fields on this form are complete, click the "Create User" button.

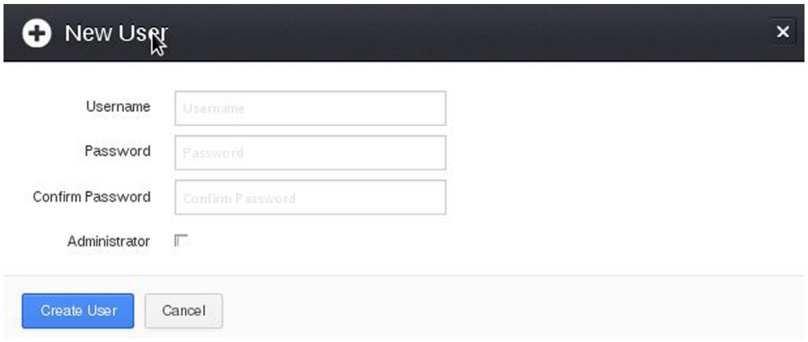


FIGURE 8.17
New user.

Configuration

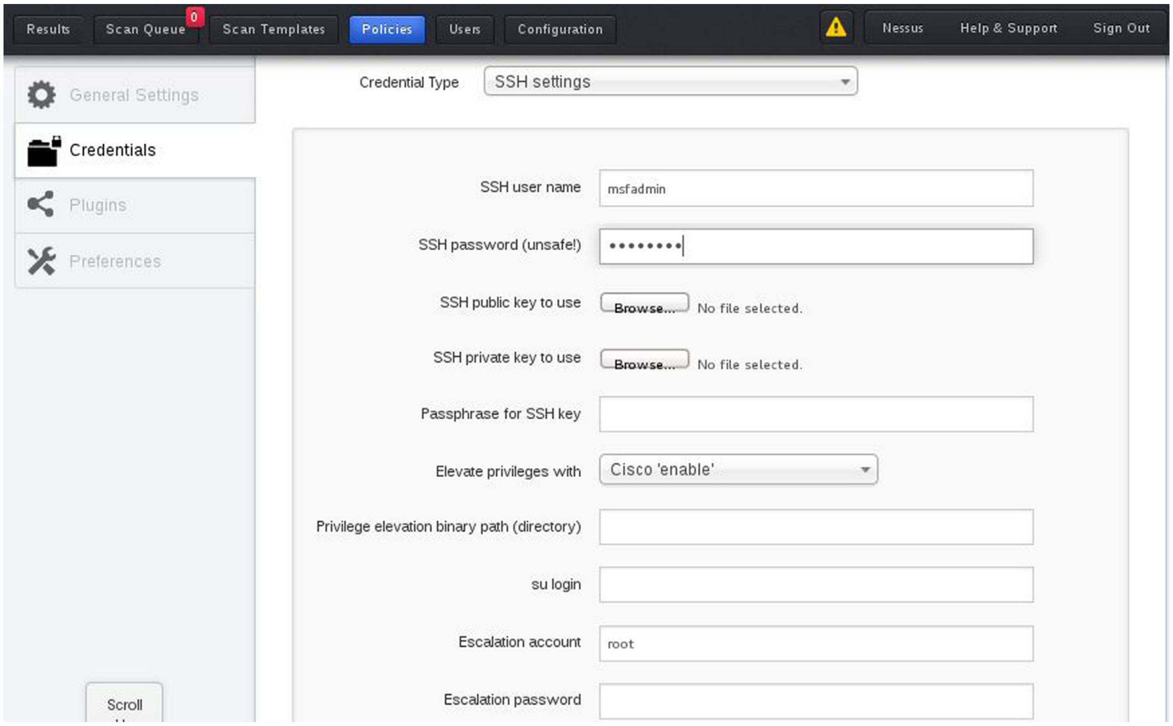
The configuration tab allows the user to fine tune the Nessus Scanner to function as efficiently and effectively as possible. Use this tab to configure proxy ports, SMTP settings, Mobile Settings, Results Settings, a number of Advanced settings, and it also allows the user to configure the Nessus feed and Activation Code. If the Activation Code has not yet been entered, use the "Feed Settings" tab on the System Configuration page to do that at this time. Then update the feed settings by clicking on the "Update Activation Code". Once activated, update the Nessus Plugins by clicking the "Update Plugins" button.

Configuring a Scan

Policies control how the Nessus scan will run including what options and credentials will be used. Developing full policies is beyond the scope of this book, so the focus of this example will be modifying an existing policy. Select the policy tab and then open the "Internal Network Scan" by clicking on the title. This will open the options dialog box, with several tabs.

All of the tabs are useful and should be explored in the lab environment before using the tool in production. For example since the user name and password are known on the metasploitable machine, these credentials could be entered in the credentials tab giving the scanning engine more access to the remote target. Uncovering these credentials often happens in the Reconnaissance phase. Figure 8.18 illustrates entering the user name and password for the metasploitable virtual machine in the credentials tab.

The plugins tab instructs the scanner to scan for specific settings, services, and options. For example, one of the option groups enabled by default is



Credential Type: SSH settings

SSH user name: msfadmin

SSH password (unsafe!): ••••••••

SSH public key to use: No file selected.

SSH private key to use: No file selected.

Passphrase for SSH key:

Elevate privileges with: Cisco 'enable'

Privilege elevation binary path (directory):

su login:

Escalation account: root

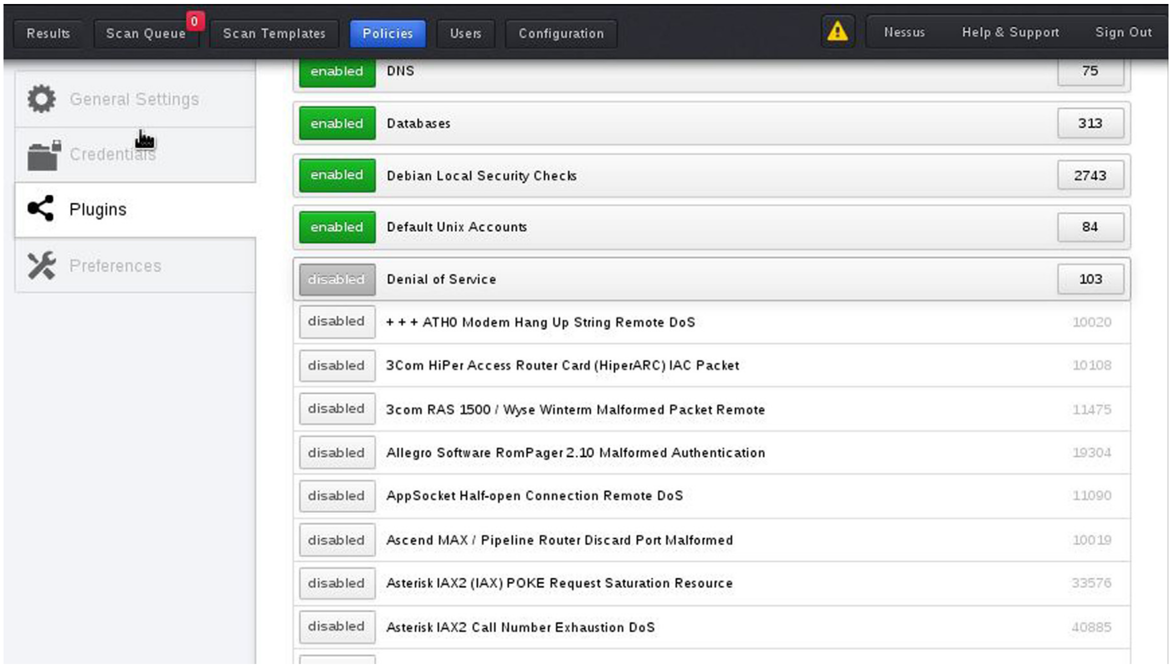
Escalation password:

FIGURE 8.18
Nessus credentials.

DoS. Assuming DoS is not allowed by the current ROE, this options group should be disabled. Do this by clicking the green enabled button. The color should turn to gray and the text should now read “disabled”. To see what checks would be ran by this option group, click the text next to the button and the items that are considered DoS checks will be displayed as in Figure 8.19 The number in the rectangle on the right, in this case 103, indicates the number of checks in the group.

After making these changes return to the “General Settings” tab and enter a new name in the Name Field, in this case the name “No DoS” was entered in the name field (Figure 8.20) and the “Update” button was selected. Once the application updates this new Policy is available in the Policies listings (Figure 8.21).

The last step in setting up the scan is to build the scan template. Create a new template by selecting the “ + New Scan” button. In the “General Scan Settings” Name the new template, for this example, “No DoS Test Scan” was



enabled	DNS	75
enabled	Databases	313
enabled	Debian Local Security Checks	2743
enabled	Default Unix Accounts	84
disabled	Denial of Service	103
disabled	+ + + ATH0 Modem Hang Up String Remote DoS	10020
disabled	3Com HiPer Access Router Card (HiperARC) IAC Packet	10108
disabled	3com RAS 1500 / Wyse Winterm Malformed Packet Remote	11475
disabled	Allegro Software RomPager 2.10 Malformed Authentication	19304
disabled	AppSocket Half-open Connection Remote DoS	11090
disabled	Ascend MAX / Pipeline Router Discard Port Malformed	10019
disabled	Asterisk IAX2 (IAX) POKE Request Saturation Resource	33576
disabled	Asterisk IAX2 Call Number Exhaustion DoS	40885

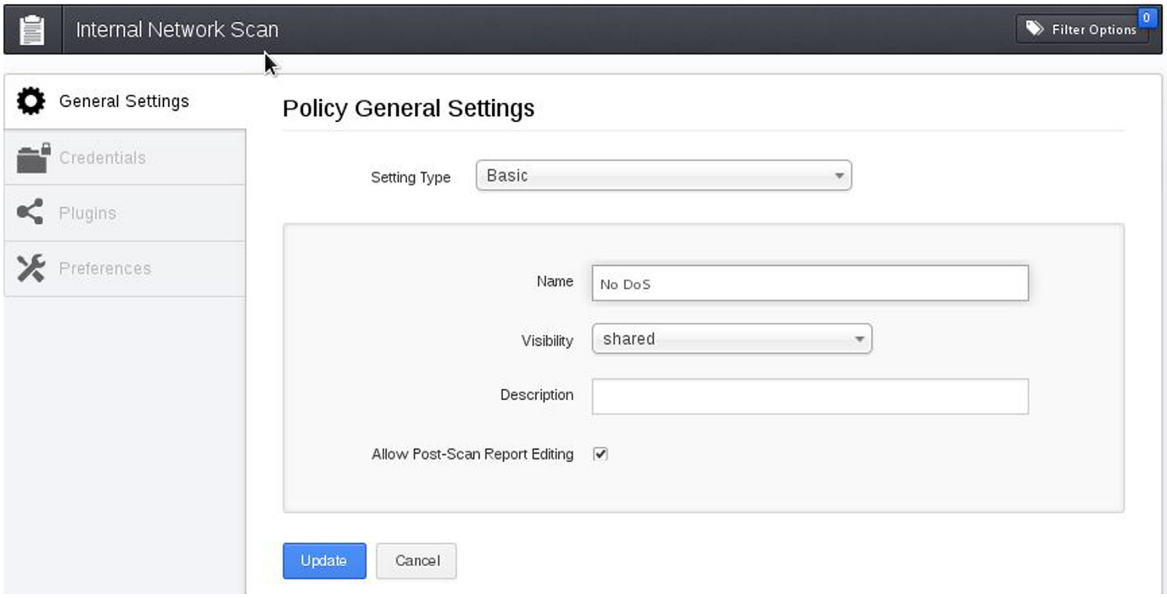
FIGURE 8.19
Removing DoS.

entered as the name, the type was not changed from "Run Now", the policy was set to "No DoS" and the scan target was set to only the metasploitable2 virtual machine at the IP address identified earlier, 10.0.2.100. A text file containing the target list could also be uploaded using the "Upload Targets" "Browse" button.

The email tab can be used to enter email addresses of users that should get information from the template scan. For this to function, the Simple Mail Transfer Protocol (SMTP) service must be configured. This will not be done for this example.

Once the settings have all been double checked, the scan can begin. Do this by clicking the blue "Run Scan" button. This will start the scan of the selected target(s) using the profile selected. The Scan Queue will display the status of the current scan(s) (Figure 8.22).

As the scan executes, the discovered vulnerabilities can be seen on the "Results" tab. Figure 8.23 illustrates the results of the scan of the metasploitable2 virtual machine after the scan has run for only a few minutes and had not yet passed the 0% completion mark. This illustrates how vulnerable



Policy General Settings

Setting Type: Basic

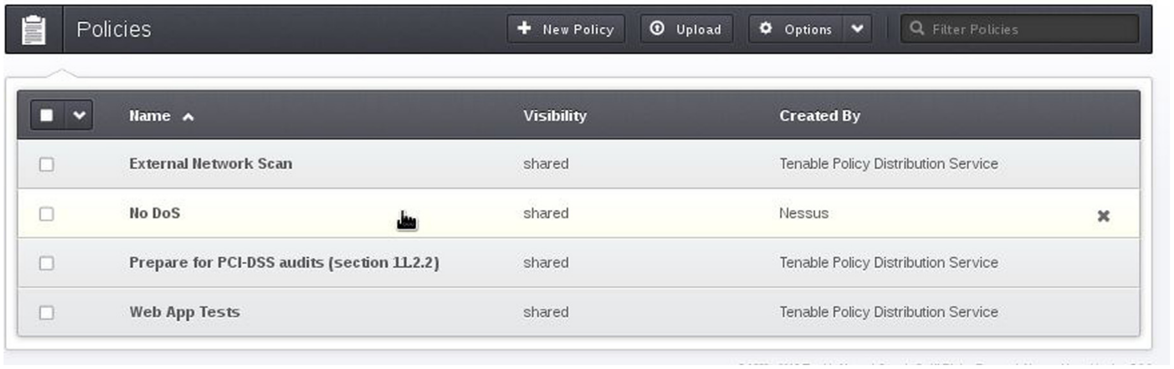
Name: No DoS

Visibility: shared

Description:

Allow Post-Scan Report Editing: ✓

FIGURE 8.20
No DoS rename.

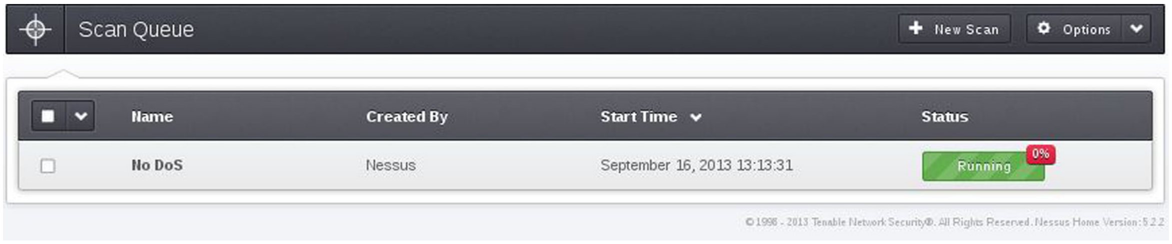


Name	Visibility	Created By
External Network Scan	shared	Tenable Policy Distribution Service
No DoS	shared	Nessus
Prepare for PCI-DSS audits (section 11.2.2)	shared	Tenable Policy Distribution Service
Web App Tests	shared	Tenable Policy Distribution Service

FIGURE 8.21
No DoS listing.

this virtual machine is and why it should never be connected to the Internet directly.

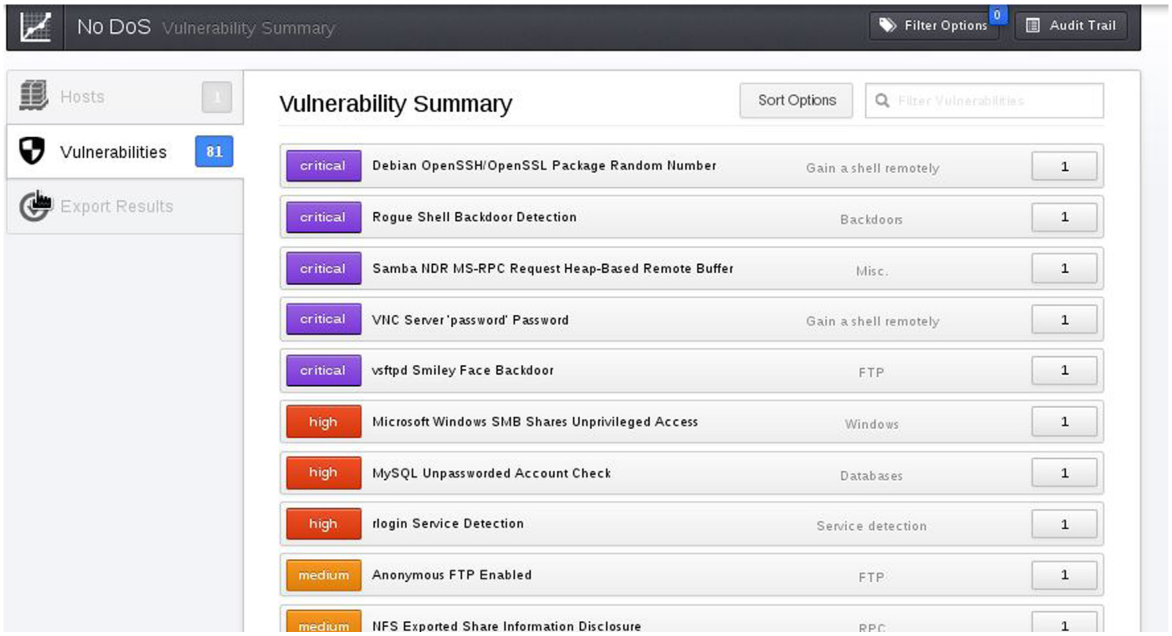
Once the scan has completed, the "Results" tab can be used to export the data in a number of formats including Comma Separated Variable (CSV),



	Name	Created By	Start Time	Status
	No DoS	Nessus	September 16, 2013 13:13:31	Running 0%

© 1998 - 2013 Tenable Network Security®. All Rights Reserved. Nessus Home Version: 5.2.2

FIGURE 8.22
Scan Queue.



critical	Debian OpenSSH/OpenSSL Package Random Number	Gain a shell remotely	1
critical	Rogue Shell Backdoor Detection	Backdoors	1
critical	Samba NDR MS-RPC Request Heap-Based Remote Buffer	Misc.	1
critical	VNC Server 'password' Password	Gain a shell remotely	1
critical	vsftpd Smiley Face Backdoor	FTP	1
high	Microsoft Windows SMB Shares Unprivileged Access	Windows	1
high	MySQL Unpassworded Account Check	Databases	1
high	rlogin Service Detection	Service detection	1
medium	Anonymous FTP Enabled	FTP	1
medium	NFS Exported Share Information Disclosure	RPC	1

FIGURE 8.23
Scan results.

PDF, and HTM, for this exercise the results will be exported as a PDF file. In this example, all of the chapters were included by selecting the "Host Summary (Executive)", "Vulnerabilities By Host", and "Vulnerabilities by Plugin" buttons. Each button turns blue indicating it is selected for export. Select the blue "Export" button to initiate the export (Figure 8.24).

Nessus is a powerful scanning engine that has excellent features. Several books and videos delve deeper into configuring and using the Nessus scanning engine. It is recommended that the application be fully tested in the lab environment before it is used against production systems.

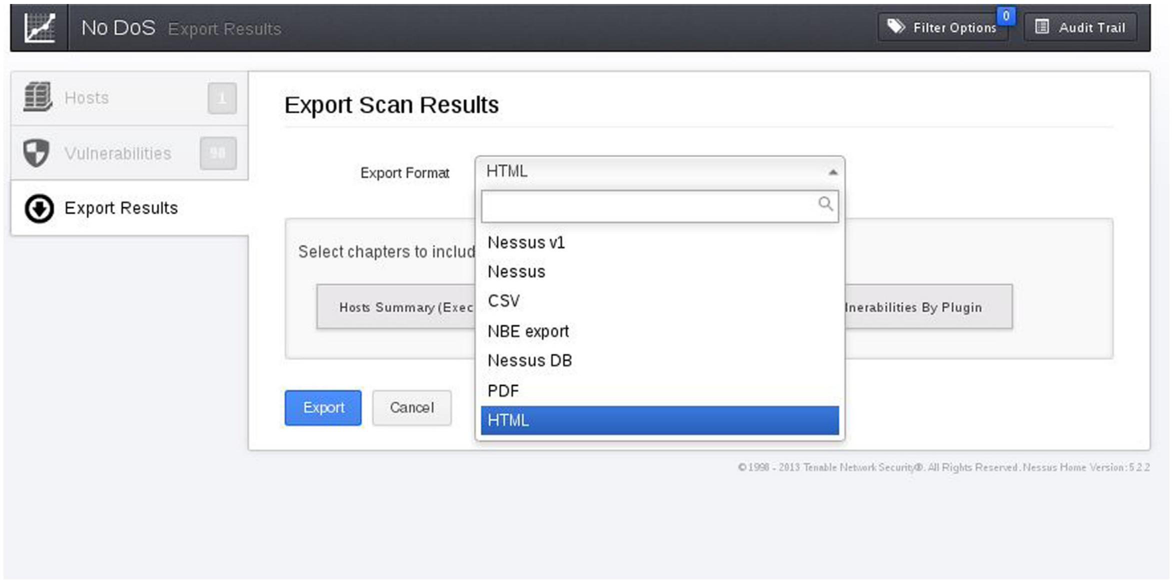


FIGURE 8.24
Scan report.

SUMMARY

There are a number of helpful tools that are packaged on the Kali Linux distribution that can assist with the scanning process. This chapter only touched on three of the most popular tools that can be used in the scanning phase of the penetration testing lifecycle. More about these tools and applications can be found in the man (manual) pages or help files for each tool. Additionally, there are a number of other tools on the Kali Linux distribution that can be used to complete the scanning phase. Results from this phase will be instrumental in assisting the penetration tester in the subsequent phases of the penetration testing engagement.