port scan with namp and netcat

Port Scanning

When you start a pentest, the potential scope is practically limitless. The client could be running any number of programs with security issues: They could have misconfiguration issues in their infrastructure that could lead to compromise; weak or default passwords could give up the keys to the kingdom on otherwise secure systems; and so on. Pentests often narrow your

scope to a particular IP range and nothing more, and you won't help your client by developing a working exploit for the latest and greatest server-side vulnerability if they don't use the vulnerable software. We need to find out which systems are active and which software we can talk to.

Manual Port Scanning

For example, in the previous chapter we saw that exploiting the MS08-067 vulnerability can be an easy win for attackers and pentesters alike. To use this exploit, we need to find a Windows 2000, XP, or 2003 box with an SMB server that is missing the MS08-067 Microsoft patch available on the network. We can get a good idea about the network-based attack surface by mapping the network range and querying systems for listening ports.

We can do this manually by connecting to ports with a tool such as telnet or Netcat and recording the results. Let's use Netcat to connect to the Windows XP machine on port 25, the default port for the Simple Mail Transfer Protocol (SMTP).

```
root@kali:~# nc -vv 192.168.20.10 25
nc: 192.168.20.10 (192.168.20.10) 25 [smtp]❶ open
nc: using stream socket
nc: using buffer size 8192
nc: read 66 bytes from remote
220 bookxp SMTP Server SLmail 5.5.0.4433 Ready
ESMTP spoken here
nc: wrote 66 bytes to local
```

-vv (Very verbose) → Provides detailed connection status and error messages.
192.168.20.10 → Target IP address.
25 → Target port (commonly used for SMTP email servers).

As it turns out, the Windows XP box is running an SMTP server on port 25 ❶. After we connected, the SMTP server announced itself as `SLMail version 5.5.0.4433`.

Now, keep in mind that admins can change banners like this to say anything, even sending attackers and pentesters on a wild goose chase, studying vulnerabilities for a product that is not deployed. In most cases, however, versions in software banners will be fairly accurate, and just connecting to the port and viewing the banner provides a starting point for our pentesting research. Searching the Web for information about `SLMail version 5.5.0.4433` may yield some interesting results.

On the other hand, connecting to every possible TCP and UDP port on just one machine and noting the results can be time consuming. Luckily, computers are excellent at repetitive tasks like this, and we can use port-scanning tools such as Nmap to find listening ports for us.

NOTE *Everything we have done so far in this chapter is completely legal. But once we start actively querying systems, we are moving into murky legal territory. Attempting to break into computers without permission is, of course, illegal in many countries. Though stealthy scan traffic may go unnoticed, you should practice the skills we study in the rest of this chapter (and the rest of this book) only on your target virtual machines or other systems you own or have written permission to test (known in the trade as a get-out-of-jail-free card).*

Port Scanning with Nmap

Nmap is an industry standard for port scanning. Entire books have been written just about using Nmap, and the manual page may seem a bit daunting. We will cover the basics of port scanning here and come back to the tool in later chapters.

Firewalls with intrusion-detection and prevention systems have made great strides in detecting and blocking scan traffic, so you might run an Nmap scan and receive no results at all. Though you could be hired to perform an external pentest against a network range with no live hosts, it's more likely that you're being blocked by a firewall. On the other hand, your Nmap results might instead say that every host is alive, and will be listening on every port if your scan is detected.

A SYN Scan

Let's start by running a SYN scan against our target machines. A *SYN scan* is a TCP scan that does not finish the TCP handshake. A TCP connection starts with a three-way handshake: SYN ▸ SYN-ACK ▸ ACK, as shown in Figure 5-7.

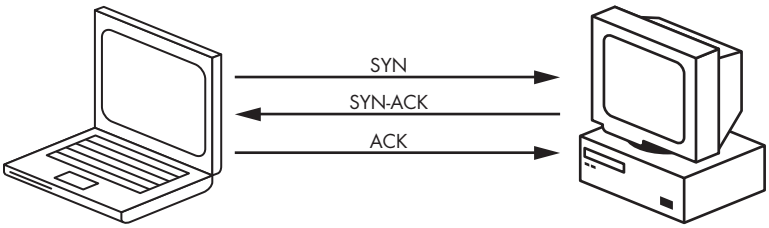

Figure 5-7: TCP three-way handshake

In a SYN scan, Nmap sends the SYN and waits for the SYN-ACK if the port is open but never sends the ACK to complete the connection. If the SYN packet receives no SYN-ACK response, the port is not available; either it's closed or the connection is being filtered. This way, Nmap finds out if a port is open without ever fully connecting to the target machine. The syntax for a SYN scan is the `-sS` flag.

Next, as you can see in Listing 5-6, we specify the IP address(s) or range to scan. Finally, we use the `-o` option to output our Nmap results to a file. The `-oA` option tells Nmap to log our results in all formats: *.nmap*, *.gnmap* (greppable Nmap), and XML. Nmap format, like the output that Nmap prints to the screen in Listing 5-6, is nicely formatted and easy to read. Greppable Nmap (as the name implies) is formatted to be used with the `grep` utility to search for specific information. XML format is a standard used to import Nmap results into other tools. Listing 5-6 shows the results of the SYN scan.

NOTE *It is always a good idea to take good notes of everything we do on our pentest. Tools such as Dradis are designed specifically to track pentest data, but as long as you have notes of everything you did when you get to the reporting phase, you will be okay. I personally am more of a pen-and-paper user, or at best, a*

creating-a-long-Word-document-with-all-of-my-results type. The methods used for tracking results vary from pentester to pentester. Outputting your Nmap results to files is a good way to make sure you have a record of your scan. Also, you can use the Linux command script to record everything printed to your terminal—another good way to keep track of everything you have done.

```
root@kali:~# nmap -sS 192.168.20.10-12 -oA booknmap
Starting Nmap 6.40 ( http://nmap.org ) at 2015-12-18 07:28 EST
Nmap scan report for 192.168.20.10
Host is up (0.00056s latency).
Not shown: 991 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp ❷
25/tcp   open  smtp ❺
80/tcp   open  http ❸
106/tcp  open  pop3pw ❺
110/tcp  open  pop3 ❺
135/tcp  open  msrpc
139/tcp  open  netbios-ssn ❹
443/tcp  open  https ❸
445/tcp  open  microsoft-ds ❹
1025/tcp open  NFS-or-IIS
3306/tcp open  mysql ❻
5000/tcp open  upnp
MAC Address: 00:0C:29:A5:C1:24 (VMware)

Nmap scan report for 192.168.20.11
Host is up (0.00031s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp ❷
22/tcp   open  ssh
80/tcp   open  http ❸
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn ❹
445/tcp  open  microsoft-ds ❹
2049/tcp open  nfs
MAC Address: 00:0C:29:FD:0E:40 (VMware)

Nmap scan report for 192.168.20.12
Host is up (0.0014s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE
80/tcp   open  http ❶
135/tcp  open  msrpc
MAC Address: 00:0C:29:62:D5:C8 (VMware)

Nmap done: 3 IP addresses (3 hosts up) scanned in 1070.40 seconds
```

-sS

1)-sS Sends a SYN packet to the target port.
2)If the port is open, the target responds with a SYN-ACK.
3)Instead of completing the handshake, Nmap immediately sends a RST (Reset) packet to avoid establishing a full connection.
4)If the port is closed, the target responds with a RST (Reset) packet.
5)If no response is received, the port is filtered (possibly blocked by a firewall).

without -sS(default)

default option scan Full TCP handshake

Listing 5-6: Running an Nmap SYN scan

As you can see, Nmap returns a handful of ports on the Windows XP and Linux boxes. We will see as we move through the next few chapters that nearly all of these ports contain vulnerabilities. Hopefully, that won't be the case on your pentests, but in an attempt to introduce you to many types of vulnerabilities you will encounter in the field, our pentesting lab has been condensed into these three machines.

That said, just because a port is open does not mean that vulnerabilities are present. Rather it leaves us with the possibility that vulnerable software might be running on these ports. Our Windows 7 machine is listening only on port 80 ❶, the traditional port for HTTP web servers, and port 139 for remote procedure call. There may be exploitable software listening on ports that are not allowed through the Windows firewall, and there may be vulnerable software running locally on the machine, but at the moment we can't attempt to exploit anything directly over the network except the web server.

This basic Nmap scan has already helped us focus our pentesting efforts. Both the Windows XP and Linux targets are running FTP servers ❷, web servers ❸, and SMB servers ❹. The Windows XP machine is also running a mail server that has opened several ports ❺ and a MySQL server ❻.

A Version Scan

Our SYN scan was stealthy, but it didn't tell us much about the software that is actually running on the listening ports. Compared to the detailed version information we got by connecting to port 25 with Netcat, the SYN scan's results are a bit lackluster. We can use a full TCP scan (`nmap -sT`) or go a step further and use Nmap's version scan (`nmap -sV`) to get more data. With the version scan shown in Listing 5-7, Nmap completes the connection and then attempts to determine what software is running and, if possible, the version, using techniques such as banner grabbing.

```
root@kali:~# nmap -sV 192.168.20.10-12 -oA bookversionnmap

Starting Nmap 6.40 ( http://nmap.org ) at 2015-12-18 08:29 EST
Nmap scan report for 192.168.20.10
Host is up (0.00046s latency).
Not shown: 991 closed ports
PORT     STATE SERVICE      VERSION
21/tcp   open  ftp          FileZilla ftpd 0.9.32 beta
25/tcp   open  smtp         SLmail smtpd 5.5.0.4433
79/tcp   open  finger       SLMail fingerd
80/tcp   open  http         Apache httpd 2.2.12 ((Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k
                              mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0)
106/tcp  open  pop3pw       SLMail pop3pw
110/tcp  open  pop3         BVRP Software SLMAIL pop3d
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn
443/tcp  open  ssl/http     Apache httpd 2.2.12 ((Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k
                              mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0)
445/tcp  open  microsoft-ds Microsoft Windows XP microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
```

```
3306/tcp open  mysql        MySQL (unauthorized)
5000/tcp open  upnp         Microsoft Windows UPnP
MAC Address: 00:0C:29:A5:C1:24 (Vmware)
Service Info: Host: georgia.com; OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.20.11
Host is up (0.00065s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE              VERSION
21/tcp   open  ftp                  vsftpd 2.3.4 ❶
22/tcp   open  ssh                  OpenSSH 5.1p1 Debian 3ubuntu1 (protocol 2.0)
80/tcp   open  http                 Apache httpd 2.2.9 ((Ubuntu) PHP/5.2.6-2ubuntu4.6 with
                                      Suhosin-Patch)
111/tcp  open  rpcbind (rpcbind V2) 2 (rpc #100000)
139/tcp  open  netbios-ssn          Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn          Samba smbd 3.X (workgroup: WORKGROUP)
2049/tcp open  nfs (nfs V2-4)       2-4 (rpc #100003)
MAC Address: 00:0C:29:FD:0E:40 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for 192.168.20.12
Host is up (0.0010s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE        VERSION
80/tcp   open  http           Microsoft IIS httpd 7.5
135/tcp  open  msrpc          Microsoft Windows RPC
MAC Address: 00:0C:29:62:D5:C8 (VMware)

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 3 IP addresses (3 hosts up) scanned in 20.56 seconds
```

Listing 5-7: Running an Nmap version scan

This time we gained much more information about our Windows XP and Linux targets. For example, we knew there was an FTP server on the Linux box, but now we have reasonable assurance that the FTP server is Very Secure FTP version 2.3.4 ❶. We'll use this output to search for potential vulnerabilities in the next chapter. As for our Windows 7 system, we found out only that it's running Microsoft IIS 7.5, a fairly up-to-date version. It's possible to install IIS 8 on Windows 7, but it's not officially supported. The version itself would not raise any red flags to me. We will find that the application installed on this IIS server is the real issue in Chapter 14.

NOTE *Keep in mind that Nmap may report the wrong version in some cases (for instance, if the software has been updated, but the welcome banner is not edited as part of the patch), but at the very least, its version scan gave us a good place to begin further research.*

UDP Scans

Both Nmap's SYN and version scans are TCP scans that do not query UDP ports. Because UDP is connectionless, the scanning logic is a bit different.

In a UDP scan (`-sU`), Nmap sends a UDP packet to a port. Depending on the port, the packet sent is protocol specific. If it receives a response, the port is considered open. If the port is closed, Nmap will receive an ICMP Port Unreachable message. If Nmap receives no response whatsoever, then either the port is open and the program listening does not respond to Nmap's query, or the traffic is being filtered. Thus, Nmap is not always able to distinguish between an open UDP port and one that is filtered by a firewall. See Listing 5-8 for a UDP scan example.

```
root@kali:~# nmap -sU 192.168.20.10-12 -oA bookudp

Starting Nmap 6.40 ( http://nmap.org ) at 2015-12-18 08:39 EST
Stats: 0:11:43 elapsed; 0 hosts completed (3 up), 3 undergoing UDP Scan
UDP Scan Timing: About 89.42% done; ETC: 08:52 (0:01:23 remaining)
Nmap scan report for 192.168.20.10
Host is up (0.00027s latency).
Not shown: 990 closed ports
PORT     STATE         SERVICE
69/udp   open|filtered tftp ❶
123/udp  open          ntp
135/udp  open          msrpc
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
445/udp  open|filtered microsoft-ds
500/udp  open|filtered isakmp
1026/udp open          win-rpc
1065/udp open|filtered syscomlan
1900/udp open|filtered upnp
MAC Address: 00:0C:29:A5:C1:24 (VMware)

Nmap scan report for 192.168.20.11
Host is up (0.00031s latency).
Not shown: 994 closed ports
PORT     STATE         SERVICE
68/udp   open|filtered dhcpc
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
2049/udp open          nfs ❷
5353/udp open          zeroconf
MAC Address: 00:0C:29:FD:0E:40 (VMware)

Nmap scan report for 192.168.20.12
Host is up (0.072s latency).
Not shown: 999 open|filtered ports
PORT     STATE         SERVICE
137/udp  open          netbios-ns
MAC Address: 00:0C:29:62:D5:C8 (VMware)

Nmap done: 3 IP addresses (3 hosts up) scanned in 1073.86 seconds
```

Listing 5-8: Running a UDP scan

For example, on the Windows XP system, the TFTP port (UDP 69) may be open or filtered ❶. On the Linux target, Nmap was able to glean that the Network File System port is listening ❷. Because only two TCP ports responded on the Windows 7 box, it's fair to assume that a firewall is in place, in this case the built-in Windows firewall. Likewise, the Windows firewall is filtering all traffic except to one UDP port. (If the Windows firewall were not in place, our UDP scan might give us more information.)

Scanning a Specific Port

By default, Nmap scans only the 1,000 ports it considers the most "interesting," not the 65,535 possible TCP or UDP ports. The default Nmap scan will catch common running services, but in some cases it will miss a listening port or two. To scan specific ports, use the `-p` flag with Nmap. For example, to scan port 3232 on the Windows XP target, see Listing 5-9.

```
root@Kali:~# nmap -sS -p 3232 192.168.20.10

Starting Nmap 6.40 ( http://nmap.org ) at 2015-12-18 09:03 EST
Nmap scan report for 192.168.20.10
Host is up (0.00031s latency).
PORT     STATE SERVICE
3232/tcp open  unknown
MAC Address: 00:0C:29:A5:C1:24 (VMware)
```

Listing 5-9: Running an Nmap scan on a specific port

Sure enough, when we tell Nmap to scan 3232, it returns `open`, which shows that this port is worth checking out, in addition to the default Nmap scanned ports. However, if we try to probe the port a bit more aggressively with a version scan (see Listing 5-10), the service listening on the port crashes, as shown in Figure 5-8.

NOTE *A good rule of thumb is to specify ports 1 through 65535 on your pentests, just to make sure there's nothing listening on those other "uninteresting" ports.*

```
root@kali:~# nmap -p 3232 -sV 192.168.20.10
Starting Nmap 6.40 ( http://nmap.org ) at 2015-04-28 10:19 EDT
Nmap scan report for 192.168.20.10
Host is up (0.00031s latency).
PORT     STATE SERVICE VERSION
3232/tcp open  unknown
1 service unrecognized despite returning data❶. If you know the service/
version, please submit the following fingerprint at http://www.insecure.org/
cgi-bin/servicefp-submit.cgi : ❷
SF-Port3232-TCP:V=6.25%I=7%D=4/28%Time=517D2FFC%P=i686-pc-linux-gnu%r(GetR
SF:equest,B8,"HTTP/1\.1\x20200\x20OK\r\nServer:\x20Zervit\x200\.4\r\n❸X-Pow
```

```
SF:ered-By:\x20Carbono\r\nConnection:\x20close\r\nAccept-Ranges:\x20bytes\
SF:r\nContent-Type:\x20text/html\r\nContent-Length:\x2036\r\n\r\n<html>\r\
SF:n<body>\r\nhi\r\n</body>\r\n</html>");
MAC Address: 00:0C:29:13:FA:E3 (VMware)
```

Listing 5-10: Running a version scan against a specific port

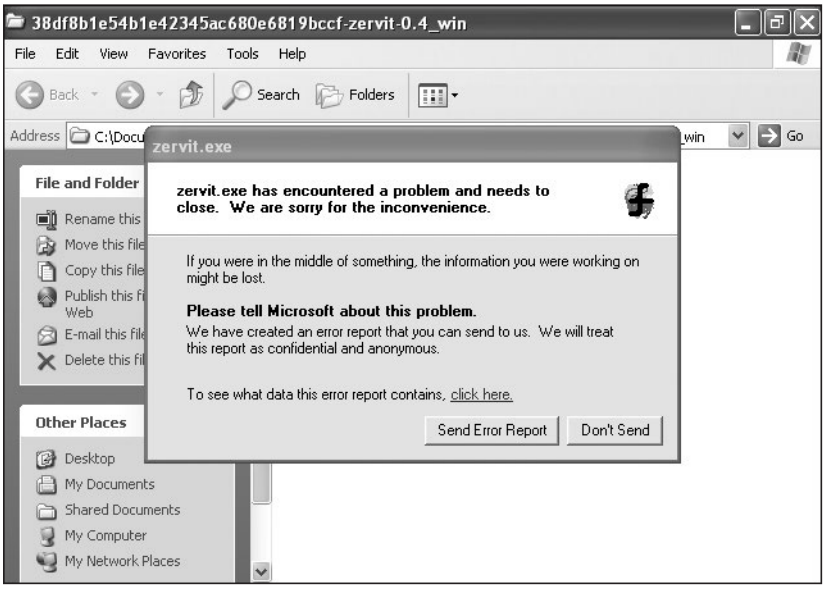

Figure 5-8: The Zervit server crashes when scanned by Nmap.

In the process of crashing the listening service, Nmap can't figure out what software is running as noted at ❶, but it does manage to get a fingerprint of the service. Based on the HTML tags in the fingerprint at ❷, this service appears to be a web server. According to the `Server:` field, it is something called Zervit 0.4 ❸.

At this point, we have crashed the service, and we may never see it again on our pentest, so any potential vulnerabilities may be a moot point. Of course, in our lab we can just switch over to our Windows XP target and restart the Zervit server.

NOTE *Though hopefully you won't make any services crash on your pentests, there is always a possibility that you will run into a particularly sensitive service that was not coded to accept anything other than expected input, such that even seemingly benign traffic like an Nmap scan causes it to crash. SCADA systems are particularly notorious for this sort of behavior. You always want to explain this to your client. When working with computers, there are no guarantees.*

We'll return to the Nmap tool in the next chapter when we use the Nmap Scripting Engine (NSE) to learn detailed vulnerability information about our target systems before beginning exploitation.