Pentest + 6

Pentest + 6

Network attacks come in many forms. Some focus on protocol vulnerabilities or take advantage of specific configurations. Others seek to obtain access to the network or to persuade target systems that they are legitimate servers or the correct network path to send traffic through to allow man-in-the-middle attacks. ***In this chapter, we will explore many of these vulnerabilities and the tools and techniques that can be used to exploit them. Along the way, we will dive into Microsoft Windows network vulnerabilities; attacks against common network services like SMTP, FTP, and DNS; and both wired and wireless network attacks.***

Our scenario continues in this chapter with an onsite penetration test that focuses on acquiring network access and then leveraging that access to penetrate systems that were not accessible from outside the network's security boundary. You will learn how to set up a fake wireless access point and how to gather information about wireless and wired clients and traffic in order to help you gain access to your target. Once you have access to the network, you will work to gain further access, including access to credentials and data exposed by service exploits.

Scenario Part 1: Onsite Assessment

After your successful remote penetration test of MCDS, LLC, the firm has asked you to perform an onsite assessment of its network security. MCDS operates a facility with over 500 employees in your area, with four office buildings spread across a small corporate campus. You must determine how to gain access to its network and then pivot to gain credentials that are useful in its infrastructure. From your previous data gathering, you know that MCDS runs an infrastructure that uses both a Windows 2012 Active Directory domain and quite a few Linux servers that provide web and other services both internally and externally.

As you read this chapter, consider how you would answer the following questions:

1.How would you gain access to the MCDS wired network if it uses a NAC scheme based on a MAC address?

2.What would you do differently if the NAC system used a client-based approach?226 Chapter 7 ■ Exploiting Network Vulnerabilities

3.MCDS uses an 802.11n network, with an open guest network called MCDS_GUEST and a WPA-2 Enterprise network that authenticates via RADIUS to Active Directory for its own internal users. How would you gather information about these networks and the systems that use them?

4.What attacks could you use against the wired network once you gain access?

Conducting Network Exploits

Once you have gained access to one or more systems at a target location, or if you have obtained physical or wireless network access, you should consider how you can exploit the network itself. This can involve attacking network protocols and behaviors, conducting man-in-the-middle attacks to capture traffic that you wouldn't normally be able to see, using denial of service (DoS) attacks to disable services or systems, or conducting attacks against security controls like NAC or encryption.

VLAN Hopping

Virtual local area networks (VLANs) separate broadcast domains into separate sections for security or performance reasons. Many organizations use VLANs to create internal security boundaries between different systems or organizational units. This makes the ability to access a VLAN other than the one you are currently on an attractive opportunity for penetration testers.

There are two common means of conducting VLAN hopping attacks: double tagging and switch spoofing.

Double tagging is used on 802.1Q trunked interfaces. Figure 7.1 shows the internal layout of an 802.1ad Ethernet frame that allows the second VLAN tag to be inserted into the packet. This allows the outer tag or service provider tag found immediately after the source MAC address to be read first and then the inner, or customer, tag to be read second.

Penetration testers can use double tagging to hop VLANs by inserting the native VLAN's tag as the first tag and the target VLAN's tag as the second tag. This causes the packet to be

passed by switches on its native VLAN, with the next switch on its trip reading the second tag. As a result, the packet is sent to the target VLAN, since it looks like it originated on the correct source VLAN.

Double tagging does have a couple of critical flaws that limit its use for penetration testers. **First, since the VLAN tags won't be replicated on responses, no responses will be received by the originating system. Second, double tagging can only be used when switches are configured to allow native VLANs, and many organizations use mitigation techniques to prevent this type of abuse.**

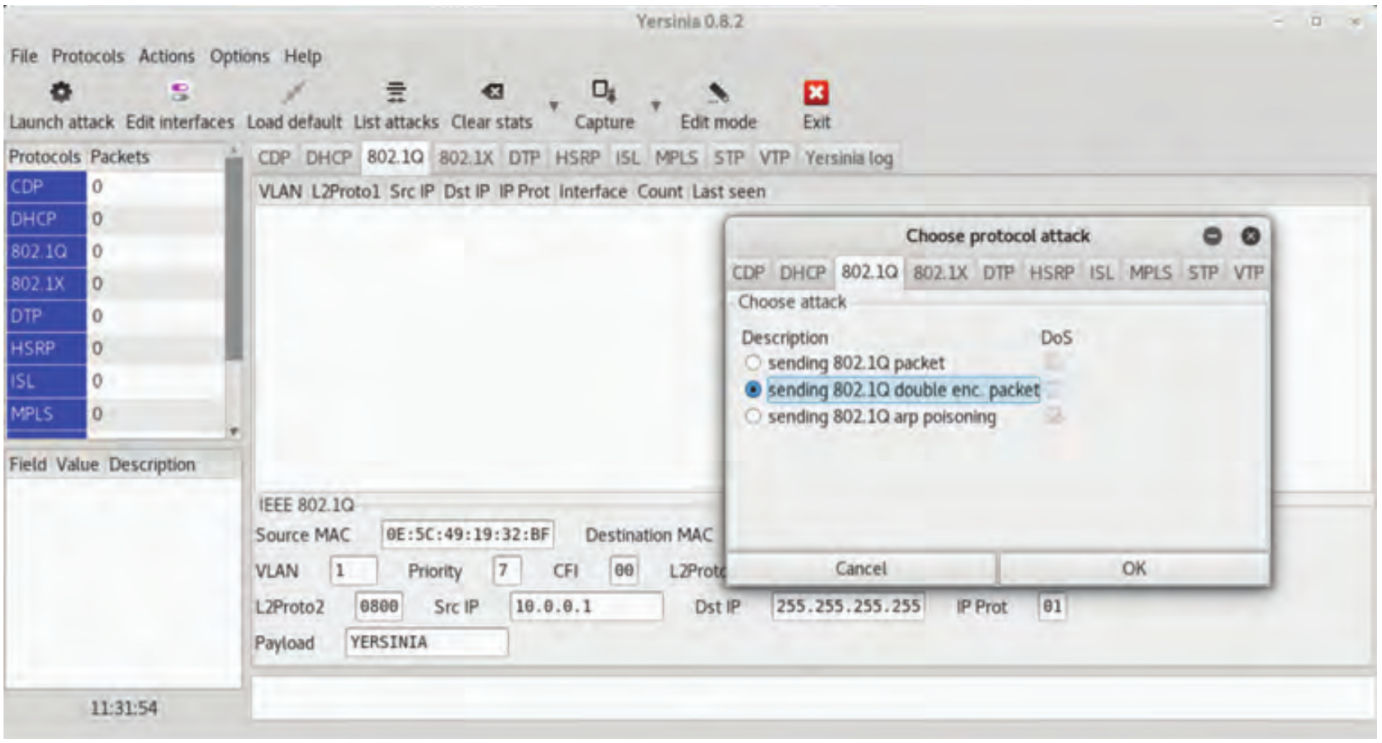
802.1Q trunking (or Dot1q) allows VLANs to work by adding tags to Ethernet frames. Switches and other devices then interpret those tags, allowing the traffic to be handled as part of the virtual LAN. Double tagging is an important capability for Internet service providers who want to properly handle VLAN tagging done by their clients while using their own VLAN tagging, so the ability to do double tagging isn't uncommon.

Switch spoofi ng relies on making the attacking host act like a trunking switch. Because the host then appears to be a switch that allows trunks, it can view traffic sent to other VLANs. Like double tagging, this technique requires that local network devices are configured to allow the attacking host to negotiate trunks (with an interface set to dynamic desirable, dynamic auto, or trunk mode), which should not be the case in a well-configured and -maintained network. I==**f you gain control of network devices or discover a misconfigured or==poorly maintained and managed network, switch spoofi ng can provide additional visibility into VLANs that might otherwise remain hidden.**

Attacks like these can be performed using the Yersinia tool found in Kali Linux. Yersinia provides a wide range of layer 2 attack capabilities, including Spanning Tree Protocol (STP) attacks, Dynamic Host Configuration Protocol (DHCP) attacks, 802.1Q trunking attacks, and quite a few others. Figure 7.2 shows Yersinia's attack module selection and interface.

see bookmarks.. installing gtk.sudo cd

++++++++++=



The PenTest+ exam objectives don't cover Yersinia, so you shouldn't have to practice with it, but if you need these capabilities, you'll want to know that it exists!

Network Proxies

In some cases, you may not be able to load penetration testing tools on a remote host that you have gained access to, but you may have access to common tools like SSH. In other scenarios you may need to have testing traffic originate from specific IP addresses or ranges, or you may want to have access to a specific host through network protections like fi rewalls that you cannot establish directly.

In each of these cases, a network proxy can help. A SOCKS proxy (Socket Secure Proxy via SSH) can securely tunnel traffic through one (or more!) hosts, thus allowing traffic through while making the proxy host appear to be the system originating the traffic.

Setting up an ssh proxy is quite simple. From a Linux command prompt, simply enter the following command using an arbitrary high port, a valid username on the proxy server, and the proxy server's hostname or IP address:

ssh [username]@[proxyserver] -D [port]

Once this is set up, you can simply set your SOCKS proxy for service by configuring the web browser's proxy setting to localhost with the port you set above. This type of proxy can allow you to pivot more easily inside a network. Using the command just shown would allow you to port-scan through the system or perform other activities directly through the proxy! This type of proxy is relatively easily spotted by defenders because the SSH proxy will appear in a list of running processes.

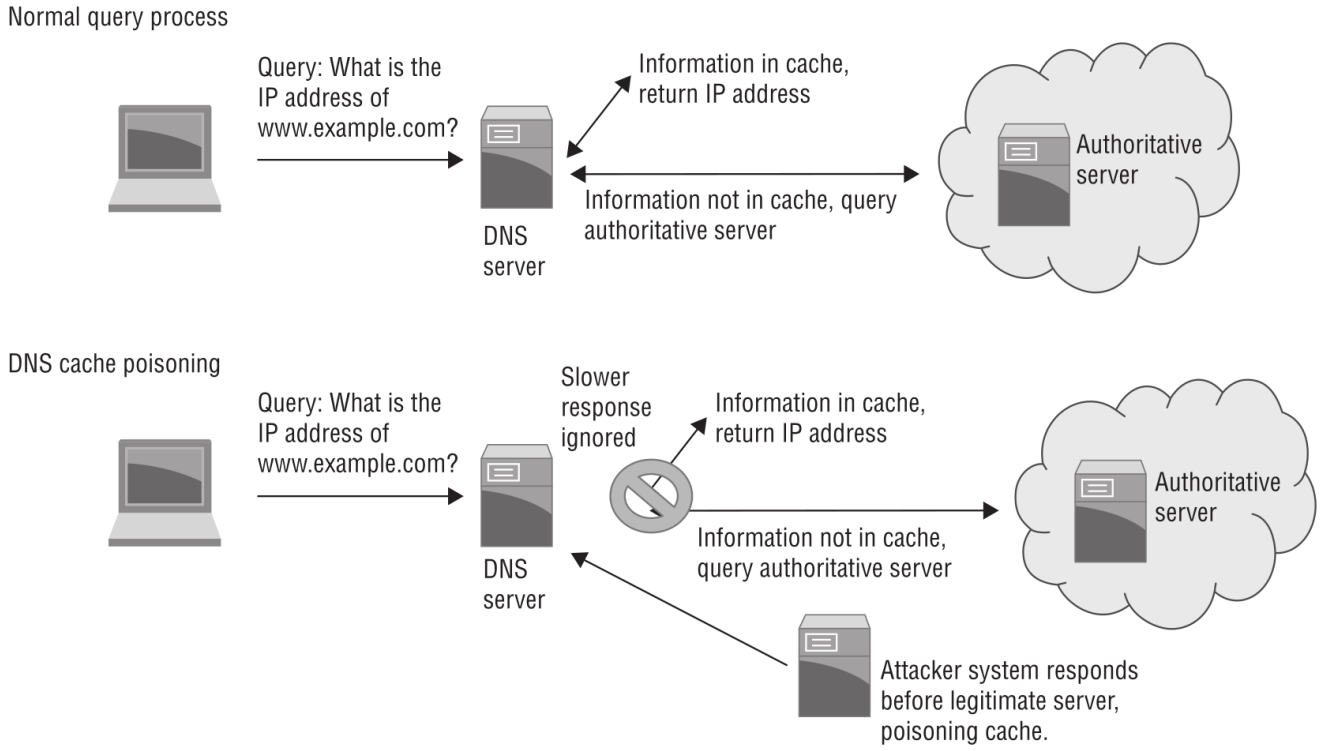
DNS Cache Poisoning

DNS spoofing, also known as DNS cache poisoning, can allow you to redirect traffic to a different host that you control. As shown in Figure 7.3, a poisoned DNS entry will point traffic to the wrong IP address, allowing attackers to redirect traffic to a system of their choice. Most DNS cache poisoning relies on vulnerabilities in DNS software, but improperly secured or configured DNS servers can allow attackers to present DNS information

without proper validation.

The most famous DNS cache poisoning vulnerability was announced in 2008, and it is rare to fi nd a vulnerable DNS server now. Thus, DNS poisoning attacks that rely on very narrow, difficult-to-exploit timing attack windows are the main option for attackers. Unless a new, widespread DNS vulnerability is discovered, DNS cache poisoning attacks are unlikely to be usefully exploitable for most penetration testers.Conducting Network Exploits

FIGURE 7.3 DNS cache poisoning attack



If you want to read up on Dan Kaminsky's 2008 DNS vulnerability, ==Steve Friedl provides a great illustrated guide at http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html, and you can read the CERT ==vulnerability note at https://www.kb.cert.org/vuls/id/800113.==

Penetration testers can take advantage of related techniques, including modifying the local hosts fi le on compromised systems to resolve hostnames to specified IP addresses. While this will not impact an entire network, the effect at a single system level is the same as it would be for a poisoned DNS cache.

A fi nal option for penetration testers is to modify the actual DNS server for a network. If you can gain control of an organization's DNS servers, or cause systems to point to a different DNS server, you can arbitrarily choose where DNS entries send your victims.

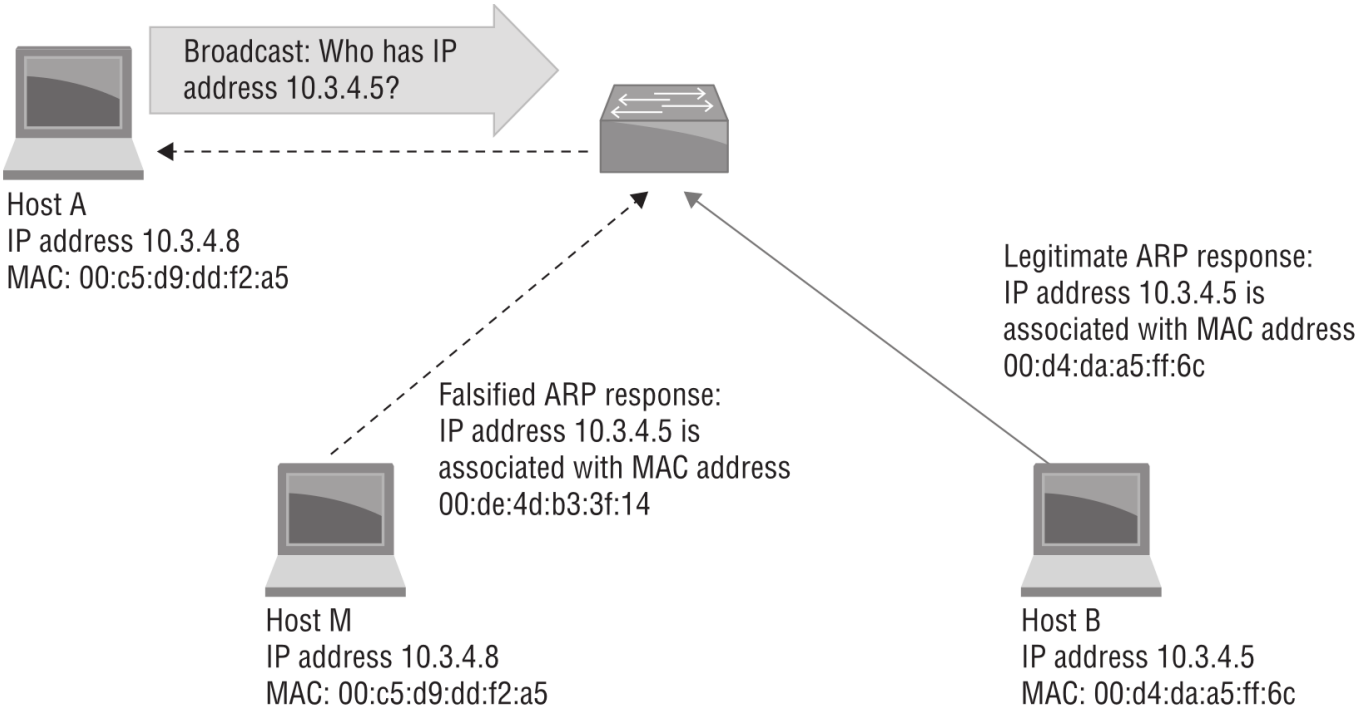
Man-in-the-Middle

Penetration testers often want to capture traffic that is sent to or from a target system, but without control of the network devices along the path, they cannot access that traffic in most cases on a modern switched network. That means they need to fi nd a way to insert

themselves into the middle of the traffic flow, either by persuading the systems involved to send traffic via another path or by compromising network equipment that is in the path of the target traffic, thus acting as a man in the middle.230

ARP Spoofing

The Address Resolution Protocol (ARP) is used to map IP addresses to physical machine addresses (MAC, or Media Access Control, addresses). Because that is how most local networks are tracked for systems, falsifying responses to ARP queries about which address traffic should be sent to can allow attackers to conduct various attacks that rely on victims sending their traffic to the wrong system, including man-in-the-middle attacks.



ARP spoofing occurs when an attacker sends falsified ARP messages on a local network, thus providing an incorrect MAC address to IP address pairing for the deceived system or systems. This information is written to the target machine's ARP cache, and the attacker can then either intercept or capture and forward traffic. **If man-in-the-middle packet capture isn't your goal, the same technique can be used to hijack sessions or cause additional traffic to hit a target system, potentially causing a DoS condition.**

In Figure 7.4, an attacker has conducted an ARP spoofing attack, causing machine A to believe that machine M should receive traffic meant for machine B. Machine M now acts as a proxy and inspects all of the traffic that machine B receives, often without either A or B becoming aware that traffic is not flowing as it should.

A**RP spoofing only works on local networks, which means that you will need to be inside the broadcast domain for a target system to successfully spoof a response. Conducting this attack in Kali Linux is relatively simple using the arpspoof command, where eth0 is our local interface, the target is set with -t, and the router or other upstream**

device is set using the -r flag for the host:

arpspoof -i eth0 -t 10.0.2.7 -r 10.0.2.1

The reverse spoof can also be set up to hallow responses to be captured, and tools like Wireshark can be used to monitor traffic between the two hosts. As you might expect, Metasploit includes ARP poisoning tools in its auxiliary modules (auxiliary/spoof/arp/arp_poisoning).

Defenders may have implemented ARP spoofing detection tools, either using automated detection capabilities or simply via Wireshark. Using an active technique that may be caught by defenders may be dangerous, so the value of an attack like this should always be weighed against the likelihood of detection.

Replay Attacks

A replay attack is a form of man-in-the-middle attack that focuses on capturing and then resending data. Common uses for replay attacks include masquerading to allow an attacker to present credentials to a service or system after capturing them during an authentication process.

One of the most common replay attacks used by penetration testers is an NTLM pass-the-hash attack. **Once a pen-tester has acquired NTLM hashes, they can then identify systems that do not require SMB signing (which prevents the attack). With a list of targets in hand, Responder or other tools with similar features can be used to intercept authentication attempts, and then an NTLM relay tool can be leveraged to drop Empire or another similar tool onto the target machine.**

If you'd like to read a good overview of how to conduct this attack, including leaving the target with Empire running, you can find an excellent writeup here:

https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-under-5-minutes.html

Replay attacks are increasingly harder to conduct now that many services use encrypted protocols for data interchange. As a penetration tester, you may have to take additional steps to successfully conduct a replay attack.

Relay Attacks

Relay attacks can appear very similar to other man-in-the-middle attacks; however, in relay attacks, the man-in-the-middle system is used only to relay attacks without modifying them rather than modifying any traffic. *It is worth bearing in mind that relay attacks are not limited to traditional IP-based network traffic.* As a penetration tester, you may fi nd it useful to query an *RFID card* or other device required to provide authentication or authorization and to relay the response to a device or system that the card is not actually near!232

The same tools used to execute other man-in-the-middle attacks can be used for relay attacks, since the goal is merely to capture or present traffic rather than modify it.

SSL Stripping Attacks

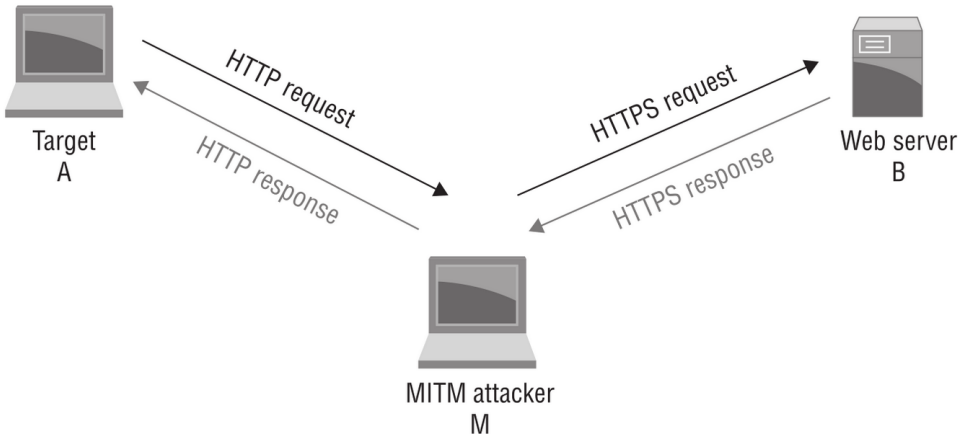
Because an ever-increasing proportion of organizational network traffic for applications and services is carried via HTTPS, downgrading HTTPS connections to HTTP is a pow-

erful tool in the hands of a penetration tester. ***The ability to downgrade the connection and then access the formerly encrypted traffic can provide a massive trove of information, including credentials, passwords, and organizational data.***

SSL stripping attacks are also often called HTTP downgrading attacks. Local policies like certificate pinning, plug-ins like HTTPS Everywhere, and many modern browsers that require HTTPS connections and validate certificate signatures can help prevent or alert users about HTTP downgrade and other related attacks. This means that knowing what security measures are in place in your target environment is important to prevent victims from detecting an SSL stripping attack.

Figure 7.5 shows an example of an SSL stripping attack, which occurs when attacker M intercepts traffic meant for site B, sent by user machine A. When A requests an HTTPS page from B, M intercepts the traffic, forwards it, and creates a secure session from itself to B and forwards responses back to A. M can now monitor session traffic between A and B.

FIGURE 7.5 SSL stripping attack



While SSL stripping is useful, alert users may notice that their connection to a normally secure site is `no longer secure. An alternative method is to provide a secure connection that appears to be legitimate while performing the same interception attack. This works better with applications than web browsers, since most web browsers will flag certificates that aren't signed by a trusted certificate authority (CA). Of course, a fake certificate signed by a legitimate CA is even more useful, but it's typically far harder to acquire.Conducting Network Exploits

Downgrade Attacks

SSL downgrade attacks work by intercepting TLS handshakes and dropping packets, thus modifying them to request weaker encryption methods. Since TLS (like SSL) allows clients to request the ciphers that they can use, this may allow an attacker to more easily read client traffic. Figure 7.6 shows an MITM attacker blocking and ending initial negotiations until the target sends a TLS request that uses weaker encryption.

If you're wondering why an attack on TLS is called an SSL downgrade attack instead of a TLS downgrade attack, it is because the term has been in

use since before TLS replaced SSL. Many practitioners still call TLS SSL out of habit, which can lead to confusion if you're not familiar with the practice!

NAC Bypass

While many network attacks rely on man-in-the-middle techniques to access traffic, gaining access to a network itself may also be required. Many organizational networks now require authentication and authorization to be on the network, and NAC (Network Access Control) is often utilized to provide that security layer.

NAC systems work by detecting when new devices connect to a network and then requiring them to be authorized to access the network. Their detection process typically involves one of the following methods:

■✓A software client that talks to a NAC server when connected

■✓A DHCP proxy that listens for traffic like DHCP requests

A broadcast listener that looks for broadcast traffic like ARP queries or a more general-purpose sniffer that looks at other IP packets

An SNMP-trap-based approach that queries switches to determine when a new MAC address shows up on one of their connected ports234

A penetration tester who wants to bypass NAC needs to determine what detection method the NAC system in place on a target network is using and can then use that information to figure out how they can best attempt to bypass NAC.

Systems that do not require client software and instead rely on information like the MAC address of a device can sometimes be bypassed by presenting a cloned MAC address on the same port that an existing system was connected on. Similarly, DHCP proxies can be bypassed by using a static IP address that the network already trusts.

Kali Linux provides macchanger, an easy way to change the MAC address of a Kali system, including the ability to match known vendor MAC prefi xes as well as to set either arbitrary or randomized MAC addresses. This makes it very easy to use a Kali system to try to defeat systems that rely on MAC addresses for part of their security controls.

More complex systems will require additional work to access the network.

If you want to read more about this topic, Ofir Arkin's 2006 paper on bypassing NAC provides a good overview despite its age:

==***https://www.blackhat.com/presentations/bh-dc-07/Arkin/Paper/***==

bh-dc-07-Arkin-WP.pdf

***DoS Attacks and Stress Testing

For many penetration tests, the rules of engagement specifically prohibit intentional denial of service (DoS) attacks, particularly against production environments. That isn't always true, and some engagements will allow or even require DoS attacks, particularly if the client organization wants to fully understand their ability to weather them. There are three major types of denial of service attacks:

Application layer denial of service attacks, which seek to crash a service or the entire

server.
Protocol-based denial of service attacks, which take advantage of a flaw in a protocol.
A SYN flood is a classic example of a protocol-based denial of service attack.
Traffic volume–based denial of service attacks simply seek to overwhelm a target by sending more traffic than it can handle.
Application layer denial of service attacks are most likely to occur accidentally during a typical penetration test, particularly when attempting to exploit vulnerabilities in services or applications. These unintentional DoS conditions should be addressed in the rules of engagement and communications plans for a penetration test, and typically require immediate communication with a contact at the client organization if the test is conducted against a production environment.
If a DoS attack is allowed in the test scope, penetration testers have a number of tools at their disposal. In addition to commercial load testing and stress test services (sometimes called “stressers”), security testing tools like Hping and Metasploit can be used to create DoS conditions.

Like most of the techniques we discuss in this book, Metasploit includes built-in modules that allow DoS attacks. They include dozens of modules ranging from OS- and service-specific tools to a general-purpose SYN flood module. Figure 7.7 shows the /auxiliary/dos/tcp/synflood tool in use with rhost and rport set to a Metasploitable vulnerable machine’s IP address and a HTTP service port. You can check the impact of this by running Wireshark (or tcpdump) to watch the SYN flood in process.

```
msf auxiliary(dos/tcp/synflood) > set rhost 10.0.2.5
rhost => 10.0.2.5
msf auxiliary(dos/tcp/synflood) > set rport 80
rport => 80
msf auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOST      10.0.2.5         yes       The target address
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new data

msf auxiliary(dos/tcp/synflood) > exploit

[*] SYN flooding 10.0.2.5:80...
```

Metasploit SYN flood

Hping: a Packet-Generation Swiss Army Knife

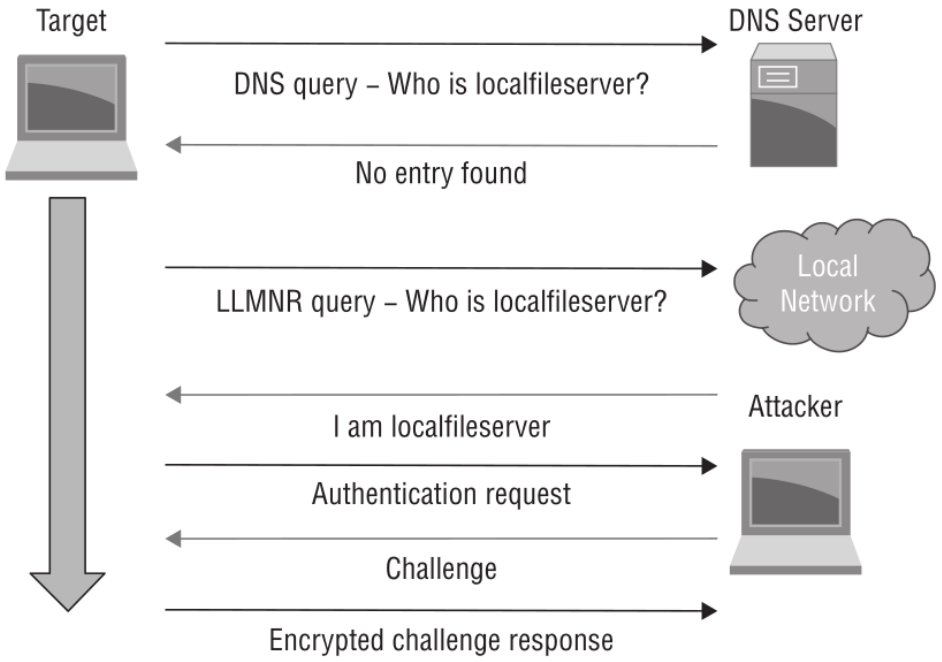
The ability to generate arbitrary packets that meet the specific formatting or content needs of an exploit or attack is a crucial one for penetration testers. In many cases, that's where Hping comes in. Hping is a packet generation (or packet crafting) tool that supports raw IP packets, ICMP, UDP, TCP, and a wide range of packet manipulation tricks including setting flags, splitting packets, and many others. \

Hping3

Hping's full list of capabilities are in the Hping wiki at http://wiki.hping.org/, and the command-line flags can all be found by typing hping -h on a system with Hping installed. Fortunately for penetration testers, Hping3 is part of Kali Linux.
In addition to the modules built into Metasploit, common DoS tools include HTTP Unbearable Load King (HULK), Low Orbit Ion Cannon (LOIC) and High Orbit Ion Cannon (HOIC), SlowLoris, and a variety of other tools. It is very important to verify that you have the correct target and permission before using tools like these against a client organization!

One of the most commonly targeted services in a Windows network is NetBIOS. NetBIOS is commonly used for file sharing, but many other services rely on the protocol as well.
NETBIOS Name Services
When Windows systems need to resolve the IP address for a hostname, they use three lookup methods in the following order:
1.The Local host file found at C:\Windows\System32\drivers\etc\hosts
2.DNS, first via local cache and then via the DNS server
3.The NetBIOS name service (NBNS), first via Link Local Multicast Name Resolution *(LLMNR) queries and then via NetBIOS Name Service (NetBIOS-NS) queries*
At first, it seems like very few queries would make it past the first two options, but that isn't the case. Many, if not most, local networks do not have entries in DNS for local systems, particularly other workstations and network devices. While domain controllers or other important elements of infrastructure may resolve via DNS, many Windows services will end up falling through to the NetBIOS name service. This means that targeting the NetBIOS name service can be a surprisingly effective attack, as shown in Figure 7.8.

FIGURE 7.8 NetBIOS name resolution attack



NOTE

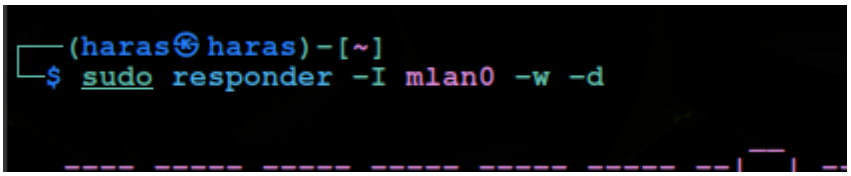
You should memorize the ports used by NetBIOS and remember what service each port is used for, as listed in the table below.

Port/Protocol	Service
135/TCP	MS-RPC endpoint matter (epmap)
137/UDP	NetBIOS name service
138/UDP	NetBIOS datagram service
139/TCP	NetBIOS session service
445/TCP	SMB

Windows sends broadcast queries to the local subnet's broadcast address via LLMNR and NetBIOS, which provides an opportunity for you to respond with a spoofed response, redirecting traffic to a host of your choice. As a stand-alone exploit, this may not be particularly effective, but SMB spoofing using tools like Responder or Metasploit modules like /auxiliary/spoof/nbns/nbns_response and then pairing them with capture tools like Metasploit's /auxiliary/server/capture_smb for authentication hashes can be a powerful option in networks that support less secure hashing methods.Exploiting Windows Services

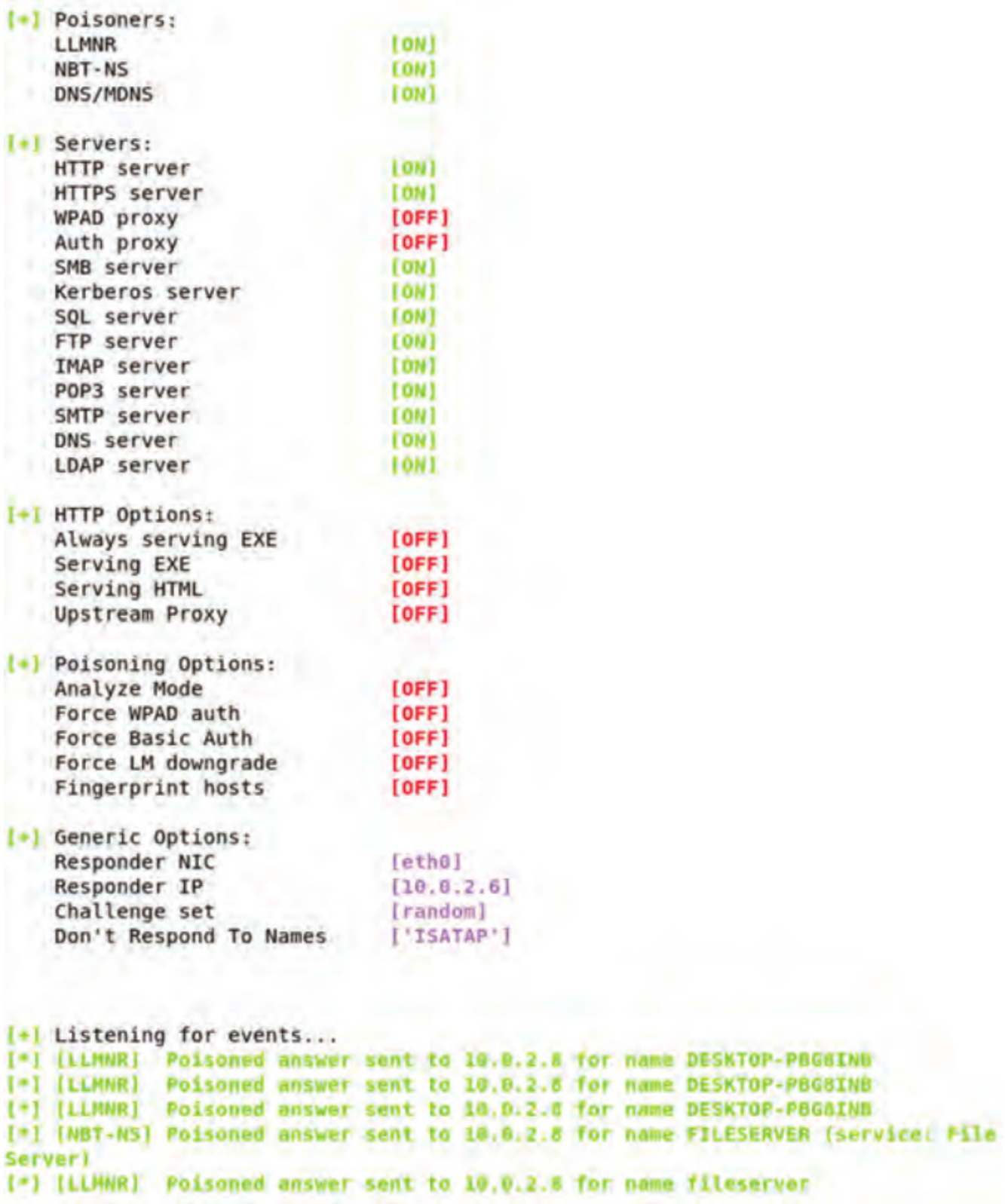
Once you have captured hashes, you can then reuse the hashes for pass-the-hash–style attacks. Doing so requires a bit more work, however, since hashes sent via SMB are salted using a challenge to prevent reuse. Metasploit and other tools that are designed to capture **SMB hashes defeat this protection by sending a static challenge and allowing the use of** rainbow tables to crack the password.

Using Responder



Responder is a powerful tool when exploiting NetBIOS and LLMNR responses. It can target individual systems or entire local networks, allowing you to analyze or respond to NetBIOS name services, LLMNR, and multicast DNS queries pretending to be the system238

that the query is intended for. Figure 7.9 shows Responder in its default mode running poisoners for each of those protocols, as well as multiple servers. Note the ability to provide executable downloads that include shells by serving EXE and HTML fi les.



`nano /etc/responder/Responder.conf`

`sudo responder -I eth0 -Pdv`

Responder sending poisoned answers

Link Local Multicast Name Resolution (LLMNR) is the first service that a Windows system tries if it cannot resolve a host via DNS. LLMNR queries are sent via port 5535 as UDP traffic and use a multicast address of

224.0.0.252 for IPv4 traffic.

Once Responder sees an authentication attempt, it will capture the hash as shown in **Figure 7.10. This is done automatically, allowing Responder to continue running in the background as you attempt other exploits or conduct further penetration testing work.Exploiting Windows Services**

F i g u r E 7.1 0

239

Responder capturing hashes

If you'd like to learn more about how to use Responder, Not So Secure's "Pwning with Responder—A Pentester's Guide" provides a very approachable overview at https://www.notsosecure.com/pwning-with-responder-a-pentesters-guide/.

*Once you have captured credentials as shown in Figure 7.10, you can also use Res*ponder to *relay NTLM authentication to a target; then, if your attack is successful, you can exe-*cute *code. Once you have gained access to the remote system, Mimikatz functionality built into the Responder tool can be used to gather more credentials and hashes, allowing you to pivot to other systems and services.*

Windows Net commands

Exploring Windows domains can be a lot easier if you are familiar with the Windows net commands. Here are a few of the most useful commands:

net view /domain

Lists the hosts in the current domain. You can also use /domain:[domain name] to search *a domain that the system has access to other than the current domain.*

net user /domain

Lists the users in a domain.

net accounts /domain

Shows the domain password policy.

net group /domain

Lists groups on the domain.

net group "Domain Admins" /domain240

Adding a group name like Domain Admins to the net group command lists users in the group, allowing discovery of domain admins.

net share

Shows current SMB shares.

net session

Used to review SMB sessions. Using the find command with this can allow searches for active sessions.

Net share [name of share] c:\directory\of\your\choice /GRANT:Everyone,FULL

Grants access to a folder on the system for any user with full rights. As you would expect, this is easy to change by identifying specific users or permissions levels.

Since the net commands are built into every Windows system you will encounter, know-

ing how to use them can be a powerful default tool when testing Windows targets. As you might expect, PowerShell provides even more powerful capabilities, but access is often more restricted, especially if you don't have administrative credentials.

SMB Exploits

The Server Message Block (SMB) implementation in Windows is another popular target for *penetration testers. Its vulnerabilities mean that unpatched systems can be exploited with relative ease; these include critical remote code execution vulnerabilities in the Windows SMB server discovered in 2017 (MS17-010, also known as EternalBlue). Like most major exploits, Metasploit includes an SMB exploit module that targets the EternalBlue vulnerability.*

Exploiting Common Services

While there are many services commonly found on networks, the PenTest+ exam specifically asks test-takers to be familiar with SMB, SNMP, SMTP, FTP, and DNS exploits. You should make sure you know how to identify these services by typical service port and protocol and that you understand the most common ways of attacking each service.

Exploiting Common Services

Exploiting an SMTP Server

One of the servers you discovered on the MCDS network is a Linux shell host. MCDS's external documentation notes that this host is available for remote logins for many of its engineering staff as well as other employees. You don't have passwords or usernames *for employees, and you want to gain access to the server. Unfortunately, your vulnerability scans don't indicate any vulnerable services. You did discover an SMTP server running on the same system.*

1.How can you gather user IDs from the SMTP server?

2.What tool could you use to attempt a brute-force SSH attack against the SSH server?

Once you have working credentials, what would your next step be to gain further access to the system?

See the following for a demonstration:

https://www.youtube.com/watch?v=YKnHq8qh3-M

SNMP Exploits

The Simple Network Management Protocol (SNMP) is commonly used to gather information about network devices, including configuration and status details. While SNMP is most commonly associated with network devices like switches and routers, it is also used to monitor printers, servers, and a multitude of other networked systems. SNMP operates on UDP port 161, making it easy to recognize SNMP traffic on a network.

SNMP organizes data into hierarchical structures called MIBs, or management information bases. Each variable in an MIB is called an OIT, or object identifier. In a***ddition, SNMP v1 and v2 rely on community strings to determine whether a connected user can read, read and write, or just send events known as "traps."***

Since SNMP can provide a wealth of information about a network and specific devices on it, it can be an important target for a penetration tester. One of the first steps for SNMP

exploitation is to map a network for devices with SNMP enabled. While a port scan can help provide information about which systems are running SNMP services, more information can be gathered with dedicated tools. Kali Linux includes both snmpenum and snmpwalk for this purpose.

```
root@kali:~# snmpwalk -c public -v1 192.168.1.1
iso.3.6.1.2.1.1.1.0 = STRING: "Linux RT-N66R 2.6.22.19 #2 Thu Aug 4 22:19:37 EDT 2016 mips"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (7763) 0:01:17.63
iso.3.6.1.2.1.1.4.0 = STRING: "root@localhost"
iso.3.6.1.2.1.1.5.0 = STRING: "RT66"
iso.3.6.1.2.1.1.6.0 = STRING: "Unknown"
iso.3.6.1.2.1.1.8.0 = Timeticks: (2) 0:00:00.02
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.10.131
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "RFC 2667 TUNNEL-MIB implementation for Linux 2.2.x kernels."
```

shows the output of snmpwalk against a commodity home router; in fact, the output extends for pages, divulging much of the current configuration and status for the system. If the system was not using the community string of public, or was properly configured with SNMP v3 settings, this would not have worked as easily!

Output from snmpwalk

Once you know which devices are running an SNMP daemon, you can query them. The goal for this round of SNMP queries is to determine *the community strings that are configured, often starting with public. If the read community string can be determined, you can gather device information easily. In poorly configured environments, or when administrators have made a mistake, it may even be possible to obtain read/write capabilities via SNMP, allowing you to change device settings via SNMP. In most cases, however, SNMP attacks are primarily for information gathering rather than intended to compromise.*

SNMP

There are three major versions of SNMP that may be encountered on a network:

■■

■■

■■

SNMP v1 has poor security and should be largely deprecated.

SNMP v2 provides added administrative functionality and added security, but the security features require configuration, are quite weak compared to modern designs, and are often not used.

SNMP v3 is functionally equivalent to SNMP v2 but adds additional security capabilities to provide confidentiality, integrity, and authentication.

SMTP Exploits

The Simple Mail Transfer Protocol (SMTP) is the protocol by which email is sent. SMTP operates on TCP port 25 and can typically be easily identified by telnetting to the service port. Much like FTP, SMTP is a very old protocol without much built-in security. That means it has been targeted for years, and most organizations that run SMTP servers have learned to harden them against misuse so that they do not get blacklisted for being spam email relays.

That means the SMTP exploits that are most useful to a penetration tester are typically associated with a specific vulnerable SMTP server version. Thus, if you encounter an SMTP server, connecting to it and gathering banner information may provide enough of a clue to determine if it is a vulnerable service.

STMP servers can also be used for information gathering by connecting them and using the EXPN and VRFY commands. To do this, simply telnet to the SMPT server (telnet example.server.com 25) and when connected, type VRFY [username] or EXPN [user_alias]. As you might guess, Metasploit includes an SMTP enumeration tool as part of its list of auxiliary scanners; auxiliary/scanner/smtp/smtp_enum will provide a list of users quickly and easily.

SMTP servers can be useful if you have access to them from a trusted system or network. Sending email that appears to be from a trusted sender through a valid email server can make social engineering attacks more likely to succeed, even with an aware and alert group of end users at the target organization. While probing SMTP servers may not seem terribly useful at fi rst glance, this trust means that scanning for and testing SMTP servers can be useful.

FTP Exploits

File Transfer Protocol (FTP) has been around since 1971, and it remains a plaintext, unencrypted protocol that operates on TCP port 21 as well as higher ephemeral TCP ports for passive transfers**. From that description, you might expect that it would have been com-** **pletely replaced by now by secure services and HTTP-based fi le transfers. Fortunately for penetration testers, that isn't always the case, and FTP servers remain in use around the world.**

Alternatives to unencrypted FTP include SFTP (SSH File Transfer Protocol) and FTPS (FTP Secure), which are both secure file transfer methods. SFTP transfers files via SSH on TCP port 22, while FTPS extends FTP itself to use Transport Layer Security (TLS) and uses TCP ports 21 and 990.

Exploiting FTP is quite simple if you can gain access to FTP network traffic. Since the protocol is unencrypted, the simplest attack is to capture usernames and passwords on the wire and use them to log into the target system or other target systems!

FTP servers themselves may also be vulnerable. Critical vulnerabilities in many major FTP servers have been discovered over time, and since FTP is an increasingly forgotten service, administrators may not have paid attention to FTP services they run. ***FTP has historically been built into many embedded devices, including network devices, printers,***

and other similar machines. Embedded FTP services are often difficult, if not impossible, to update and may also be forgotten, creating an opportunity for attack.244

A fi nal avenue for FTP service exploitation is via the configuration of the FTP service itself. **Poorly or improperly configured FTP servers** may allow navigation outside their own base d*irectories. This makes exploring the directory structure that is exposed by an FTP server useful once you have usable credentials. Since many FTP servers historically supported a public login, you may even be able to navigate some of the directory structure without specific credentials being required*. Those publicly accessible directories **can sometimes be treasure troves of organizational data.**

Years ago, one of the authors of this book discovered an FTP server during a security assessment that had what he considered the worst-case misconfiguration of an FTP server. It was configured to share the root directory of the server it was on, allowing attackers to navigate to and download almost any file on the system or to upload files to sensitive directories—possibly allowing attackers to cause the system to run files of their choosing!

Samba Exploits

Much like the Microsoft implementation of SMB, the Linux Samba server has proven to have a variety of security flaws. 2017's SambaCry exploit was discovered to allow remote code execution in all SMB versions newer than Samba 3.5.0—a 2010 code release!

Because Samba and Microsoft SMB operate on the same ports and protocols, fi nger-printing the operating system before attempting an exploit is important to ensure that you are using the right exploit for the OS and server service.

Metasploitable 2 includes a vulnerable SMB server you can use to practice SMB exploits.

SSH Exploits

Secure Shell (SSH) is used for secure command-line access to systems, typically via TCP port 22, and is found on devices and systems of all types. Because SSH is so common, attacking systems that provide an SSH service is a very attractive option for a penetration tester. This also means that most organizations will patch SSH quickly if they are able to.

Unfortunately for many organizations, SSH is embedded in de

snmpwalk(1) - Linux man page

Name

snmpwalk - retrieve a subtree of management values using SNMP GETNEXT requests

Synopsis

snmpwalk [APPLICATION OPTIONS] [COMMON OPTIONS] [OID]

Description

snmpwalk is an SNMP application that uses SNMP GETNEXT requests to query a network entity for a tree of information.

An object identifier (OID) may be given on the command line. This OID specifies which portion of the object identifier space will be searched using GETNEXT requests. ***All variables in the subtree below the given OID are queried and their values presented to the user. Each variable name is given in the format specified in variables(5).***

If no OID argument is present, snmpwalk will search the subtree rooted at SNMPv2-SMI::mib-2 (including any MIB object values from other MIB modules, that are defined as lying within this subtree). If the network entity has an error processing the request packet, an error packet will be returned and a message will be shown, helping to pinpoint why the request was malformed.

If the tree search causes attempts to search beyond the end of the MIB, the message "End of MIB" will be displayed.
Options

-Cc

Do not check whether the returned OIDs are increasing. Some agents (LaserJets are an example) return OIDs out of order, but can complete the walk anyway. Other agents return OIDs that are out of order and can cause snmpwalk to loop indefinitely. By default, snmpwalk tries to detect this behavior and warns you when it hits an agent acting illegally. Use -Cc to turn off this check.
-CE {OID}
End the walk at the specified OID, rather than a simple subtree. This can be used to walk a partial subtree, selected columns of a table, or even two or more tables within a single command.
-Ci

Include the given OID in the search range. Normally snmpwalk uses GETNEXT requests starting with the OID you specified and returns all results in the MIB subtree rooted at that OID. Sometimes, you may wish to include the OID specified on the command line in the printed results if it is a valid OID in the tree itself. This option lets you do this explicitly.

-CI

In fact, the given OID will be retrieved automatically if the main subtree walk returns no useable values. This allows a walk of a single instance to behave as generally expected, and return the specified instance value. This option turns off this final GET request, so a walk of a single instance will return nothing.

-Cp

Upon completion of the walk, print the number of variables found.

-Ct

Upon completion of the walk, print the total wall-clock time it took to collect the data (in seconds). Note that the timer is started just before the beginning of the data request series and stopped just after it finishes. Most importantly, this means that it does not include snmp library initialization, shutdown, argument processing, and any other overhead.
In addition to these options, snmpwalk takes the common options described in the snmpcmd(1) manual page.

Examples

The command:

snmpwalk -Os -c public -v 1 zeus system

will retrieve all of the variables under system:

sysDescr.0 = STRING: "SunOS zeus.net.cmu.edu 4.1.3_U1 1 sun4m"
sysObjectID.0 = OID: enterprises.hp.nm.hpsystem.10.1.1
sysUpTime.0 = Timeticks: (155274552) 17 days, 23:19:05
sysContact.0 = STRING: ""
sysName.0 = STRING: "zeus.net.cmu.edu"
sysLocation.0 = STRING: ""
sysServices.0 = INTEGER: 72
(plus the contents of the sysORTable).

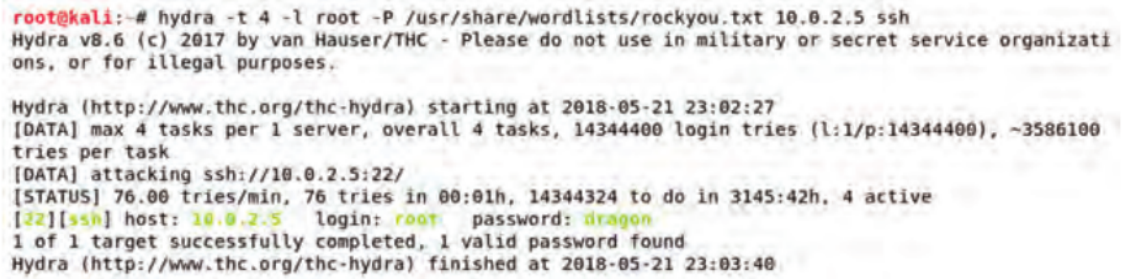
The command:

snmpwalk -Os -c public -v 1 -CE sysORTable zeus system

will retrieve the scalar values, but omit the sysORTable.x

vices of all descriptions, and
updating SSH throughout their infrastructure may be difficult. Thus, penetration testers should validate both SSH and operating system versions when reviewing vulnerability scan results to determine if a vulnerable version of SSH is running.
Another method of attacking services like SSH is to use a brute-forcing tool like THC ***Hydra (or an equivalent Metasploit module). Hydra is a brute-forcing tool that can crack systems using password guessing***. In the example shown in Figure 7.12 , Hydra is run

FIGURE 7.12 THC Hydra SSH brute-force attack



against a Metasploitable system's root account using the rockyou password list that Kali includes by default. Note the -t flag, setting the number of parallel threads for the target. By default, Hydra uses 16, but this example uses 4.

THC Hydra SSH brute-force attack

Once you have credentials, additional Metasploit modules like the ssh_login and ssh_login_pubkey modules can allow you to test them across an entire network range or list of possible target systems.

While wireless and wired networks share many of the same functions, protocols, and behaviors, there are a number of attack methods that are specifically used for wireless networks, access points, and wireless clients. These attacks focus on the way that wireless devices connect to networks, how they authenticate, and other features and capabilities specific to wireless networks.

Evil Twins and Wireless MITM

Evil twin attacks work by creating bogus access points that unsuspecting users connect to. This makes them useful for man-in-the-middle attacks like those discussed earlier in this chapter. While it is possible to create an evil twin of a secured access point, more sophisticated users are likely to notice differences like having to accept new security certificates or other changes.

KARMA Attacks

KARMA (KARMA Attacks Radio Machines Automatically) uses attacker devices that listen for probe requests for WiFi networks. When they receive the probe request, they pretend to be the access point to which the connecting system tried to connect. This allows the KARMA device to act as a man-in-the-middle device. For more details, see https://insights.sei.cmu.edu/cert/2015/08/instant-karma-might-still-get-you.html246

Evil t*wins can also be used for downgrade attacks, which trick clients into using a less-secure protocol or encryption scheme. Downgrade attacks aren't limited to 802.11-based protocols; researchers used a downgrade attack to defeat protections built into the Z-Wave protocol used by many home automation and Internet of Things (IoT) devices, causing them to downgrade from the modern and more secure S2 security standards to the S0 standard that many devices also support.*

You can read more about the Z-Shave attack at https://thehackernews .com/2018/05/z-wave-wireless-hacking.html.

Penetration testers can use Aircrack-ng to create an evil twin using the airbase-ng tool.

The process is relatively simple:

1.Capture traffic to determine the SSID and MAC addresses of a legitimate access point.

2.Clone that access point using airbase-ng.

3.Conduct a de-authentication attack

|||||||||||}]]]]}***have successfully conducted a man-in-the-middle attack, you can also work on credential harvesting by capturing unencrypted traffic between the client and remote systems and services. The same techniques that are used for a wired connection will work here, and the same challenges exist: most authentication traffic on modern networks is encrypted, making sniffi ng credentials “on the wire”—in this case via wireless connections—much harder.

Attacking WPS

WiFi Protected Setup (WPS) has been a known issue for years, but it remains in use for ease of setup, particularly for consumer wireless devices. Setting up a printer with theWireless Exploits 247

push of a button, rather than entering a pre-shared key or password, can seem attrac- tive. Unfortunately, one WPS setup mode requires an 8-digit PIN, which is easily cracked because WPS uses an insecure method of validating PINs. WPS passwords can be attacked

using a pixie dust attack, a type of attack that brute-forces the keyhale house augusta phonehale house augusta phone for WPS. Vulnerable routers can simply be attacked by leveraging the fact that many have poor selection algorithms for their pre-shared key random numbers.

You can read about how to conduct a pixie dust attack in Kali Linux at https://www.hackingtutorials.org/WiFi-hacking-tutorials/pixie-dust-attack-wps-in-kali-linux-with-reaver/.

Bluetooth Attacks

Bluetooth attacks can be useful for penetration testers who have physical access to a local network, or who can get into range of a target's computer, phone, vehicle, or other Bluetooth-enabled device. There are two common Bluetooth attack methods you need to be aware of for the PenTest+ exam:

■✓

■✓

Bluesnarfing, the theft of information from Bluetooth-enabled devices. Kali includes the bluesnarfer package, which allows phonebook contact theft via Bluetooth, given a device ID or address.

Bluejacking, which sends unsolicited messages over Bluetooth devices.

While discovering Bluetooth devices may be part of a penetration test, the broad fears about wide-scale exploits of Bluetooth-enabled devices have not resulted in significant real-world issues. **Bluetooth is a potential path into systems and should be documented, but it's unlikely to be a primary exploit method for most penetration tests.**

Other Wireless Protocols and Systems

While the PenTest+ exam doesn't currently include wireless standards other than those we have discussed here, you should make sure to review any information that you fi nd about an organization's wireless capabilities. It is relatively common to discover proprietary or open-standard wireless devices operating in an environment that may provide either interesting information or even a path into a network. The methods to capture and interpret those protocols are well beyond the scope of this book, but there are many groups and individuals that focus on this type of reverse engineering. You can fi nd a treasure trove of projects related to this type of work at https://hackaday.com/tag/hackrf/, as well as presentations like the 2018 Blackhat "Bringing Software Defi ned Radio to the Penetration Testing Community," found at https://www.blackhat.com/docs/us-14/materials/us-14-Picod-Bringing-Software-Defined-Radio-To-The-Penetration-Testing-Community-WP.pdf248

Wireless Security Tools

Some of the most common open-source wireless security assessment tools are **Aircrack-ng, Kismet, and WiFite.**

Aircrack-ng provides the ability to conduct replay and deauthentication attacks and to act as a fake access point. It also provides the ability to crack WPA PSK, in addition to the normal packet capture and injection capabilities built into most wireless security tools.

You can read more at https://www.aircrack-ng.org/.

Kismet provides wireless packet capture and sniffing features and can also be used as a wireless intrusion detection system. Kismet can be found at https://www.kismetwireless.net/.

WiFite, or more accurately WiFite2, is a wireless network auditing tool. It includes WPA handshake capture capabilities, support for pixie dust attacks, support for identification of hidden access points, and WPA handshake cracking, among other auditing- and penetration-testing–friendly capabilities.

If you're exploring Kali Linux, you'll find a number of other tools designed to execute specific attacks, and each of those tools can be very useful in specific circumstances. In most cases, however, one of these three tools will be your starting place for penetration tests.

RFID Cloning

Access cards, ID cards, and similar tokens are often used to provide access control to facilities. This makes cloning RFID cards a useful tool for penetration testers. While each of the technologies relies on radio frequency (RF), there are three primary types of card or device that you are likely to encounter:

- ***Low frequency 125–134.2 KHz RFID cards, which can be cloned to other cards using a readily available cloning tool.***
- ***High frequency 13.56 MHz tags and cards. Many phones now support this near-field communication (NFC) capability, making it possible to clone cards with phones.***
- ***Ultra high frequency tags vary in range from 865 to 928 MHz, and they vary around*** **the world because there is not an accepted international standard**.

Figure 7.13 shows an inexpensive low frequency RFID cloning device and tags. Devices like these can make cloning RFID-based access cards trivial, and since RFID cards are often generic in appearance, using a cloned card or even a fob like those shown in the image can make a physical penetration test far simpler.Wireless Exploits

Jamming

Wireless DoS can also be a legitimate technique for penetration testers, but it isn't a common technique. It may be used to prevent access to a wireless device or to prevent a device from communicating with a controller or monitoring system, as may be required as part of a penetration test. As wireless IoT devices become increasingly common, **blocking them from communicating upstream may allow you to avoid detection or prevent an alarm from being sent.**

Jamming may not be legal in the jurisdiction you are in or for the type of device or system you want to block. In the United States, the Federal Communications Commission (FCC) specifically prohibits the use of jammers that block authorized radio communication. You can ==***read more at https://www.fcc.gov/general/jammer-enforcement,***== ***and there is a complete FAQ on GPS, WiFi, and cell phone jammers at*** ==***https://transition.fcc.gov/eb/jammerenforcement/jamfaq.pdf.***==

Repeating

Repeating traffic, or relaying traffic, can be useful for a penetration tester who needs access to a wireless network but cannot remain in range of the network. While *directional antennas can help, adding a concealed repeater to a remote network can allow traffic to be relayed over longer distances. Commercial devices like the Pwnie Express Pwnplug provide the ability to deploy a device to a local organization and then connect to that device to relay attack traffic to local networks.*