Exploiting Wired and Wireless Network

Introduction

There are a wide range of vulnerabilities that can be exploited on wired and wireless LANs. It is important to have a good working knowledge of them so you can devise strategies for exploiting them. **Go through and refresh your knowledge of network attacks**. We will try out some really powerful tools in simulated attacks to help you get your skills up-to-speed.

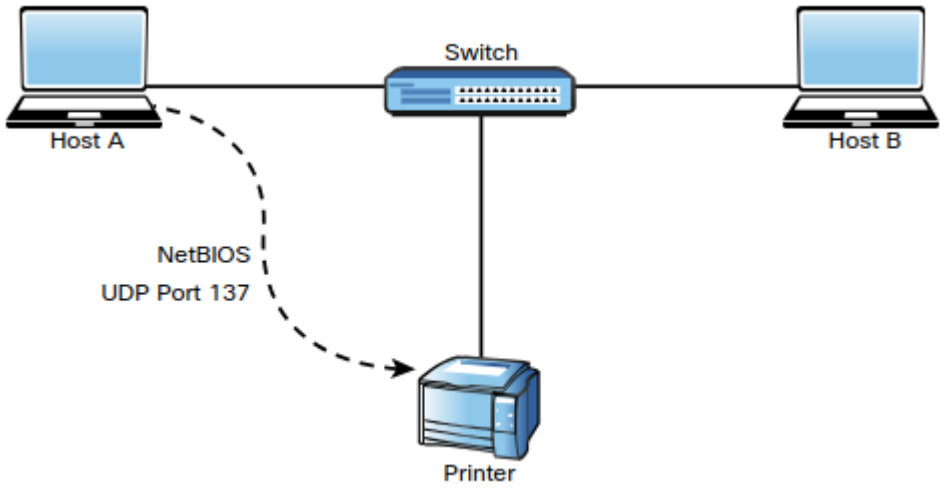
Exploiting Network-Based Vulnerabilities

Network-based vulnerabilities and exploits can be catastrophic because of the types of damage and impact they can cause in an organization. The following are some examples of network-based attacks and exploits:

- Windows name resolution-based attacks and exploits
- DNS cache poisoning attacks
- Attacks and exploits against Server Message Block (SMB) implementations
- Simple Network Management Protocol (SNMP) vulnerabilities and exploits
- Simple Mail Transfer Protocol (SMTP) vulnerabilities and exploits
- File Transfer Protocol (FTP) vulnerabilities and exploits
- Pass-the-hash attacks
- On-path attacks (previously known as man-in-the-middle [MITM] attacks)
- SSL stripping attacks
- Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks
- Network access control (NAC) bypass
- Virtual local area network (VLAN) hopping attacks

1- Windows name resolution-based attacks and exploits

Name resolution is one of the most fundamental aspects of networking, operating systems and applications. There are several name-to-IP address resolution technologies and protocols, including Network Basic Input/Output System (NetBIOS), **Link-Local Multicast Name Resolution (LLMNR)** and Domain Name System (DNS). The sections that follow cover vulnerabilities and exploits related to these protocols.



NetBIOS Name Service and LLMNR

NetBIOS and LLMNR are protocols that are used primarily by Microsoft Windows for host identification. LLMNR, which is based on the DNS protocol format, allows hosts on the same local link to perform name resolution for other hosts. For example, a Windows host trying to communicate to a printer or to a network shared folder may use NetBIOS

- NetBIOS Name Service (NetBIOS-NS) for name registration and resolution
- Datagram Service (NetBIOS-DGM) for connectionless communication
- Session Service (NetBIOS-SSN) for connection-oriented communication

Used ports and protocols :

- UDP port 137: NetBIOS Name Service
- UDP port 138: NetBIOS Datagram Service
- TCP port 139: NetBIOS Session Service
- TCP port 445: SMB protocol, used for sharing files between different operating systems, including Windows and Unix-based systems

SMB Exploits:
As you learned in the previous section, SMB has historically suffered from numerous catastrophic vulnerabilities. You can easily see this by just exploiting the dozens of well-known exploits in the Exploit Database (exploit-db.com) by using the **searchsploit** command

```
┌──(root㉿Kali)
└─# searchsploit smb
```

Detailed information about how to install SearchSploit is available at *https://www.exploit-db.com/searchsploit/*.

One of the most commonly used SMB exploits in recent times has been the EternalBlue exploit, which was leaked by an entity called the Shadow Brokers that allegedly stole numerous exploits from the U.S. National Security Agency (NSA). Successful exploitation of EternalBlue allows an unauthenticated remote attacker to compromise an affected system and execute arbitrary code. This exploit has been used in ransomware such as WannaCry and Nyeta. This exploit has been ported to many different tools, including Metasploit.

Scanning for SMB Vulnerabilities with enum4linux

Server Message Block (SMB) is a Microsoft protocol that is available on non-Microsoft networks through the open-source Samba service. SMB makes it easy to set up and access network shares on LANs. However, many vulnerabilities have been found in it, and a number of high-profile

exploits of it have appeared, such as WannaCry, Conficker, and EternalBlue. In this lab you will get familiar with a popular tool that is used to discover network shares and other sensitive information through the exploitation of SMB. You will also conduct a simulated exploit in which you transfer an unauthorized and potentially malicious file to an unprotected share.

```
┌──(root㉿Kali)
└─# enum4linux -help
```

Samba Utilities : rpcclient, net, nmblookup and smbclient.

Terms associated with SMB functions.

Relative Identifier (RID): Uniquely identifies a user, group, system, or domain.
Security Identifier (SID): Uniquely identifies users and groups within the local domain. Globally unique so can also work between domains.
Domain Controller (DC) : Domain controller is a server that manages network and identity security requests. It authenticates users and determines whether the users are allowed to access IT resources in the domain.
Lightweight Directory Access Protocol (LDAP): Used for authentication and authorization purpose in enterprise environments, particularly with directory services like Microsoft Active Directory
Workgroup: a group of standalone computers that are independently administered.

Use Nmap to find SMB Services

List all hosts

```
┌──(root㉿Kali)
└─# nmap -sN 172.17.0.0/24
```

list only hosts with opened ports

```
┌──(root㉿Kali)
└─# nmap -sN  -p-  --open 172.17.0.0/24
```

Use enum4linux to enumerate users and network file shares.

Use the **enum4linux -U** option to list the users configured on the **target 172.17.0.2**. after **nmap** scan we find this target **host 172.17.0.2** with SMB opened **ports139** and **445**

-U find configured users

-S get a list of file shares

-G get a list of the groups and their members

-P list the password policies

-i get a list of printers

```
┌──(root㉿Kali)
└─# enum4linux -U  172.17.0.2
```

Use smbclient to transfer files between systems

Smbclient is a component of **samba** that can store and retrieve files, similar to an FTP client. You will use **smbclient** to transfer a file to the target system at 172.17.0.2. This simulates exploiting a network host with malware through an SMB vulnerability.

Create a text file using **cat** command

```
┌──(root㉿Kali)
```

```
└─# cat >> badfile.txt

┌──(root㉿Kali)
└─# smbclient --help
```

Using **-L** command to list shares on the target host

```
┌──(root㉿Kali)
└─# smbclient -L //172.17.0.2/
```

Connect to the tmp share

```
┌──(root㉿Kali)
└─# smbclient  //172.17.0.2/tmp
```

prompt changed to **smb:>**

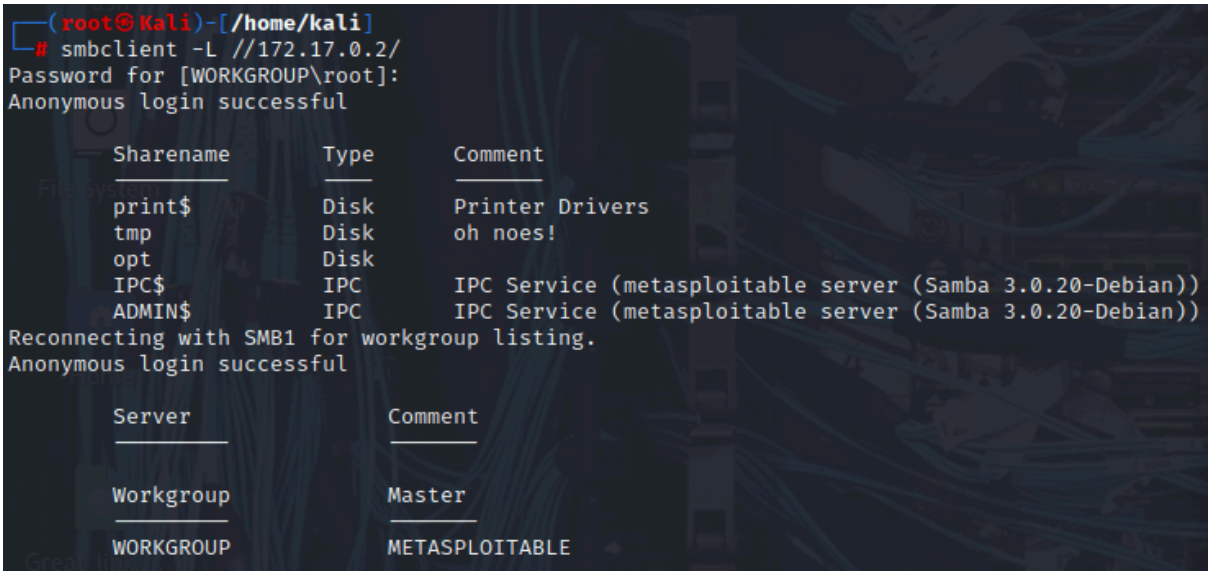
```
smb: > dir
```

Upload the badfile.txt to the target server using the put command. The syntax for the command is: ***put local-file-name remote-file-name***

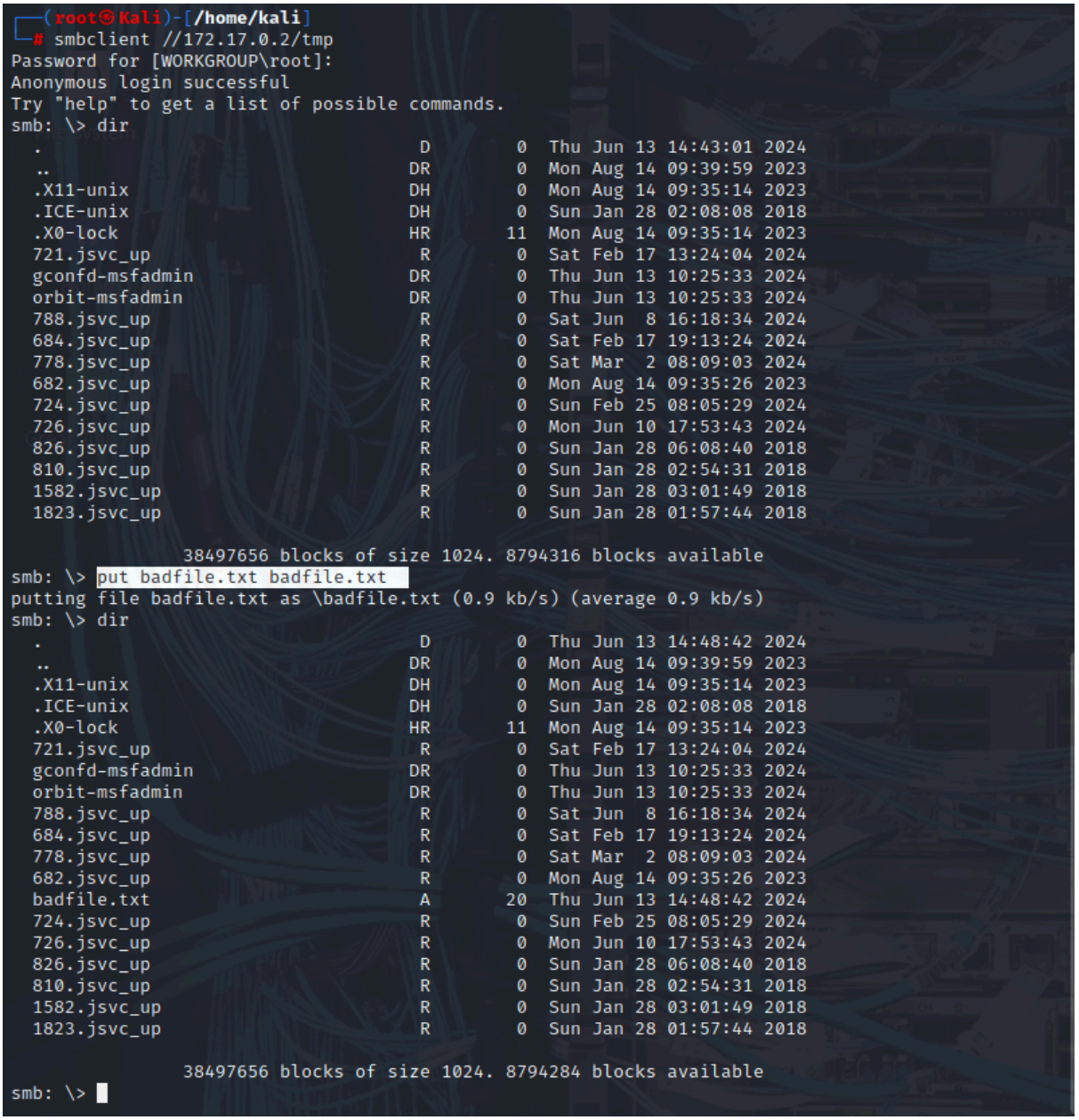
```
smb: > put badfile.txt badfile.txt
```

```
quit
```

Use the **smbclient** -L command to list the shares on the target host. This command produces a similar output to what the **enum4linx** command did in Part 3. When asked for a password, press enter. The double / character before the IP address and the / following it are necessary if the target is a Windows computer.



Connect to the tmp share using the **smbclient** command by specifying the share name and IP address.

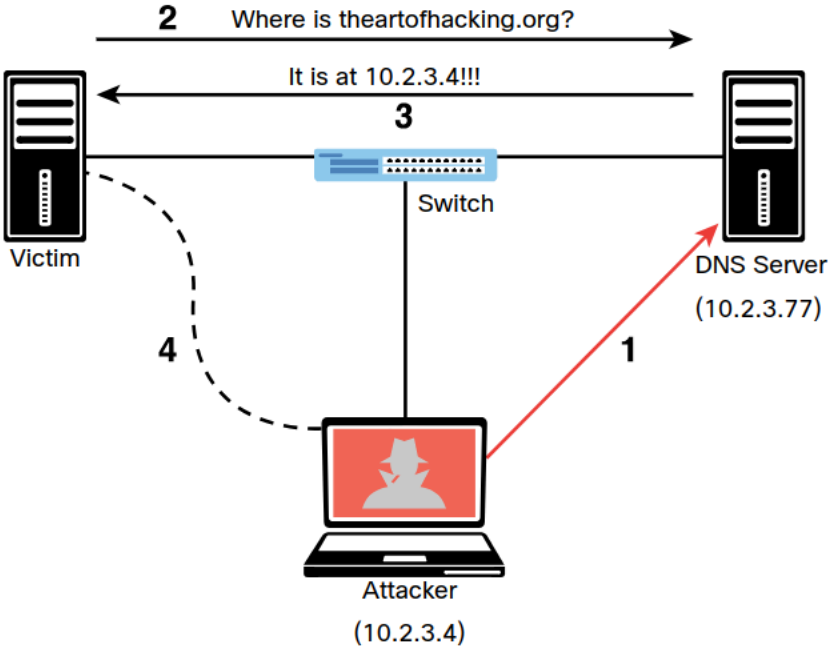


You are conducting a penetration test of a client network. You have gained access to an internal network by social engineering the username and password of an ad hoc webserver that is not behind the firewall. You can remotely access the network from a Kali VM configured with the enum4linux tool.

DNS Cache Poisoning

DNS cache poisoning is another popular attack leveraged by threat actors. In short, DNS cache poisoning involves the manipulation of the DNS resolver cache through

the injection of corrupted DNS data. This is done to force the DNS server to send the wrong IP address to the victim and redirect the victim to the attacker's system.



DNS cache Poisoning Example 5-3

Step 1. The attacker corrupts the data of the DNS server cache to impersonate the website theartofhacking.org. Before the attacker executes the DNS poisoning attack, the DNS server successfully resolves the IP address of the theartofhacking.org to the correct address (104.27.176.154) by using the nslookup command,.	nslookup theartofhacking.org Server: 10.2.3.77 Address: 10.2.3.77#53 Non-authoritative answer: Name: theartofhacking.org Address: 104.27.176.154

Step 2. After the attacker executes the DNS poisoning attack, the DNS server resolves the theartofhacking.org to the	$ nslookup theartofhacking.org Server: 10.2.3.77 Address: 10.2.3.77#53

IP address of the attacker's system (10.2.3.4), as shown in Example 5-4.	`Non-authoritative answer:`
	`Name: theartofhacking.org`
	`Address: 10.2.3.4`

Step 3. The victim sends a request to the DNS server to obtain the IP address of the domain theartofhacking.org.

Step 4. The DNS server replies with the IP address of the attacker's system.

Step 5. The victim sends an HTTP GET to the attacker's system, and the attacker impersonates the domain theartofhacking.org.

DNS cache poisoning attacks can also combine elements of social engineering to manipulate victims into downloading malware or to ask a victim to enter sensitive data into forms and spoofed applications.

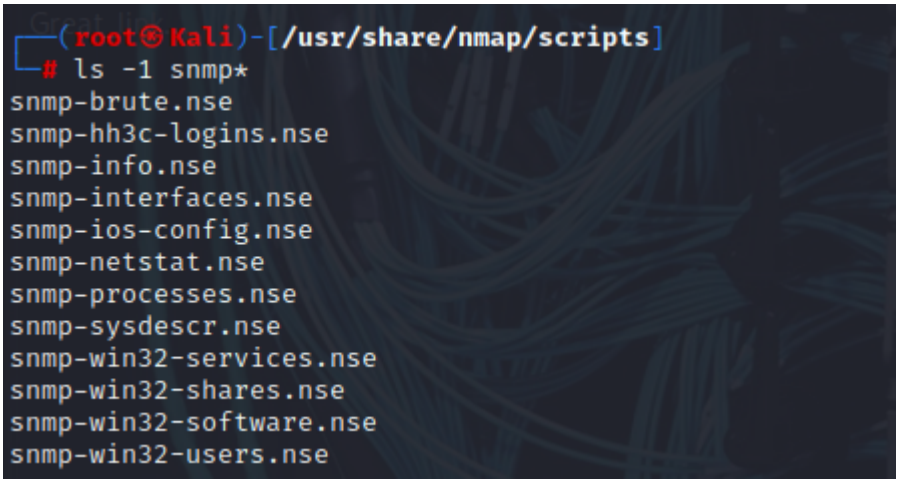
SNMP Exploits

Simple Network Management Protocol (SNMP) is a protocol that many individuals and organizations use to manage network devices. SNMP uses UDP port 161. In SNMP implementations, every network device contains an SNMP agent that connects with an independent SNMP server (also known as the SNMP manager). An administrator can use SNMP to obtain health information and the configuration of a networking device, to change the configuration and to perform other administrative

tasks. As you can imagine, this is very attractive to attackers because they can leverage SNMP vulnerabilities to perform similar actions in a malicious way.

There are several versions of SNMP. The two most popular versions today are **SNMPv2c** and **SNMPv3**. SNMPv2c uses community strings, which are passwords that are applied to a networking device to allow an administrator to restrict access to the device in two ways: by providing read-only or read/write access.

Example below shows the available SNMP-related NSE scripts in kali Linux system.



In addition to NSE(**N**map **S**cripting **E**ngine) scripts, you can use the **snmp-check** tool to perform an SNMP walk in order to gather information on devices configured for SNMP.

Attackers may leverage insecure SMTP servers to send spam and conduct phishing and other email-based attacks. SMTP is a server-to-server protocol, which is different from client/server protocols such as POP3 or IMAP.

Different email ports :

- **TCP port 25:** Used for SMTP for non-encrypted communications.

- **TCP port 465:** The port registered by the Internet Assigned Numbers Authority (IANA) for SMTP over SSL (SMTPS). SMTPS has been deprecated in favor of STARTTLS.
- **TCP port 587:** The Secure SMTP (SSMTP) protocol for encrypted communications, as defined in RFC 2487, using STARTTLS. Mail user agents (MUAs) use TCP port 587 for email submission. STARTTLS can also be used over TCP port 25 in some implementations.
- **TCP port 110:** The default port used by the POP3 protocol in non-encrypted communications.
- **TCP port 995:** The default port used by the POP3 protocol in encrypted communications.
- **TCP port 143:** The default port used by the IMAP protocol in non-encrypted communications.
- **TCP port 993:** The default port used by the IMAP protocol in encrypted (SSL/TLS) communications.

SMTP Open Relays

SMTP open relay is the term used for an email server that accepts and relays (that is, sends) emails from any user. It is possible to abuse these configurations to send spoofed emails, spam, phishing, and other email-related scams. Nmap has an NSE script to test for open relay configurations. The details about the script are available at https://svn.nmap.org/nmap/scripts/smtp-open-relay.nse.

```
root@kali:/usr/share/nmap/scripts# nmap --script smtp-open-relay.nse
10.1.2.14

Starting Nmap 7.60 ( https://nmap.org ) at 2018-04-15 13:32 EDT
Nmap scan report for 10.1.2.14
Host is up (0.00022s latency).
PORT STATE SERVICE
25/tcp open smtp
|_smtp-open-relay: Server is an open relay (16/16 tests)
Nmap done: 1 IP address (1 host up) scanned in 6.82 seconds
root@kali:/usr/share/nmap/scripts#
```

```
omar@kali:~$ telnet 192.168.78.8 25
```

The **smtp-user-enum** tool(which is installed by default in kali Linux) enables you to automate these information-gathering steps.

```
Examples:

$ smtp-user-enum -M VRFY -U users.txt -t 10.0.0.1
$ smtp-user-enum -M EXPN -u admin1 -t 10.0.0.1
$ smtp-user-enum -M RCPT -U users.txt -T mail-server-ips.txt
$ smtp-user-enum -M EXPN -D example.com -U users.txt -t 10.0.0.1
```

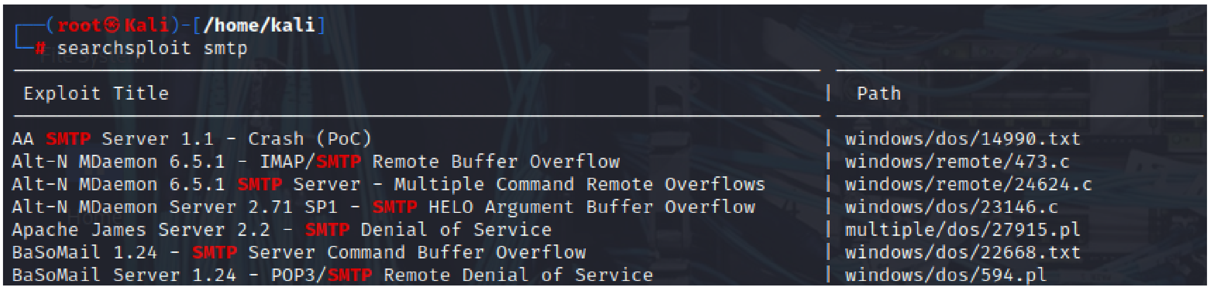
the smtp-user-enum command to verify whether the user omar exists in the server. Most modern email servers disable the VRFY and EXPN commands. It is highly

recommended that you disable these SMTP commands. Modern firewalls also help protect and block any attempts at SMTP connections using these commands.

Known SMTP Server Exploits
It is possible to take advantage of exploits that have been created to leverage known SMTP-related vulnerabilities.

Using **searchsploit** to find Known SMTP Exploits

```
┌──(root㉿Kali)-[/home/kali]
└─# searchsploit smtp
```



```
┌──(root㉿Kali)-[/home/kali]
└─# searchsploit smtp
 Exploit Title                                                          |  Path
AA SMTP Server 1.1 - Crash (PoC)                                        | windows/dos/14990.txt
Alt-N MDaemon 6.5.1 - IMAP/SMTP Remote Buffer Overflow                  | windows/remote/473.c
Alt-N MDaemon 6.5.1 SMTP Server - Multiple Command Remote Overflows     | windows/remote/24624.c
Alt-N MDaemon Server 2.71 SP1 - SMTP HELO Argument Buffer Overflow      | windows/dos/23146.c
Apache James Server 2.2 - SMTP Denial of Service                        | multiple/dos/27915.pl
BaSoMail 1.24 - SMTP Server Command Buffer Overflow                     | windows/dos/22668.txt
BaSoMail Server 1.24 - POP3/SMTP Remote Denial of Service               | windows/dos/594.pl
```

FTP Exploits

Attackers often abuse FTP servers to steal information. The legacy FTP protocol doesn't use encryption or perform any kind of integrity validation. Recommended practice dictates that you implement a more secure alternative, such as File Transfer Protocol Secure (FTPS) or Secure File Transfer Protocol (SFTP).

The SFTP and FTPS protocols use encryption to protect data; however, some implementations – such as Blowfish and DES – offer weak encryption ciphers

(encryption algorithms). You should use stronger algorithms, such as AES. Similarly, SFTP and FTPS servers use hashing algorithms to verify the integrity of file transmission. SFTP uses SSH, and FTPS uses FTP over TLS. Best practice calls for disabling weak hashing protocols such as MD5 or SHA-1 and using stronger algorithms in the SHA-2 family (such as SHA-2 or SHA-512).

In addition, FTP servers often enable anonymous user authentication, which an attacker may abuse to store unwanted files in your server, potentially for exfiltration. For example, an attacker who compromises a system and extracts sensitive information can store that information (as a stepping stone) to any FTP server that may be available and allows any user to connect using the anonymous account.

UsingNmap to Scan an FTP Server

```
root@kali:~# nmap -sV 172.16.20.136
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-05 22:37 EDT
Nmap scan report for 172.16.20.136
Host is up (0.00081s latency).
Not shown: 997 closed ports
PORT STATE SERVICE VERSION
21/tcp open ftp    vsftpd 3.0.3
22/tcp open ssh    OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux;
protocol 2.0)
```

Shows a scan (using Nmap) against a server with IP address 172.16.20.136 Nmap can determine the type and version of the FTP server (in this case, **vsftpd version 3.0.3).**

The following are several additional best practices for mitigating FTP server abuse and attacks:

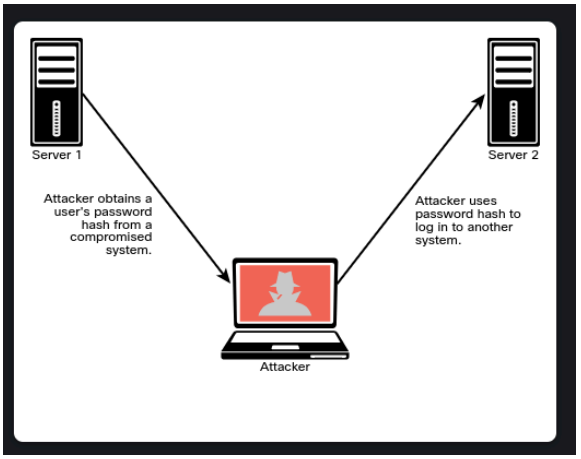
- Use strong passwords and multifactor authentication. A best practice is to use good credential management and strong passwords. When possible, use two-factor authentication for any critical service or server.
- Implement file and folder security, making sure that users have access to *only* the files they are entitled to access.
- Use encryption at rest – that is, encrypt all files stored in the FTP server.
- Lock down administration accounts. You should restrict administrator privileges to a limited number of users and require them to use multifactor authentication. In addition, do not use common administrator usernames such as root or admin.
- Keep the FTPS or SFTP server software up-to-date.
- Use the U.S. government FIPS 140-2 validated encryption ciphers for general guidance on what encryption algorithms to use.
- Keep any back-end databases on a different server than the FTP server.
- Require re-authentication of inactive sessions.

Pass-the -Hash Attacks

All versions of Windows store passwords as hashes in a file called the Security Accounts Manager (SAM) file. The operating system does not know what the actual password is because it stores only a hash of the password. Instead of using a well-known hashing algorithm, Microsoft created its own implementation that has developed over the years.

Microsoft also has a suite of security protocols for authentication, called this New **Technology LAN Manager (NTLM).** NTLM had two versions: NTLMv1 and NTLMv2. Since Windows 2000, Microsoft has used Kerberos in Windows domains. However, NTLM may still be used when the client is authenticating to a server via IP address or if a client is authenticating to a server in a different Active Directory (AD) forest configured for NTLM trust instead of a transitive inter-forest trust. In addition, NTLM might also still be used if the client is authenticating to a server that doesn't belong to a domain or if the Kerberos communication is blocked by a firewall.

So, what is a pass-the-hash attack? Because password hashes cannot be reversed, instead of trying to figure out what the user's password is, an attacker can just use a password hash collected from a compromised system and then use the same hash to log in to another client or server system. Figure 5-3 illustrates a pass-the-hash attack.



The Windows operating system and Windows applications ask users to enter their passwords when they log in. The system then converts the passwords into hashes (in most cases, using an API called **LsaLogonUser**). A pass-the-hash attack goes around this process and just sends the hash to the system to authenticate.

TIP Mimikatz is a tool used by many penetration testers, attackers, and even malware that can be useful for retrieving password hashes from memory; it is a very useful post-exploitation tool. You can download the Mimikatz tool from https://github.com/gentilkiwi/mimikatz. Metasploit also includes Mimikatz as a Meterpreter script to facilitate exploitation without the need to upload any files to the disk of the compromised host. You can find more information about Mimikatz/Metasploit integration at https://www.offensive-security.com/metasploit-unleashed/mimikatz/.

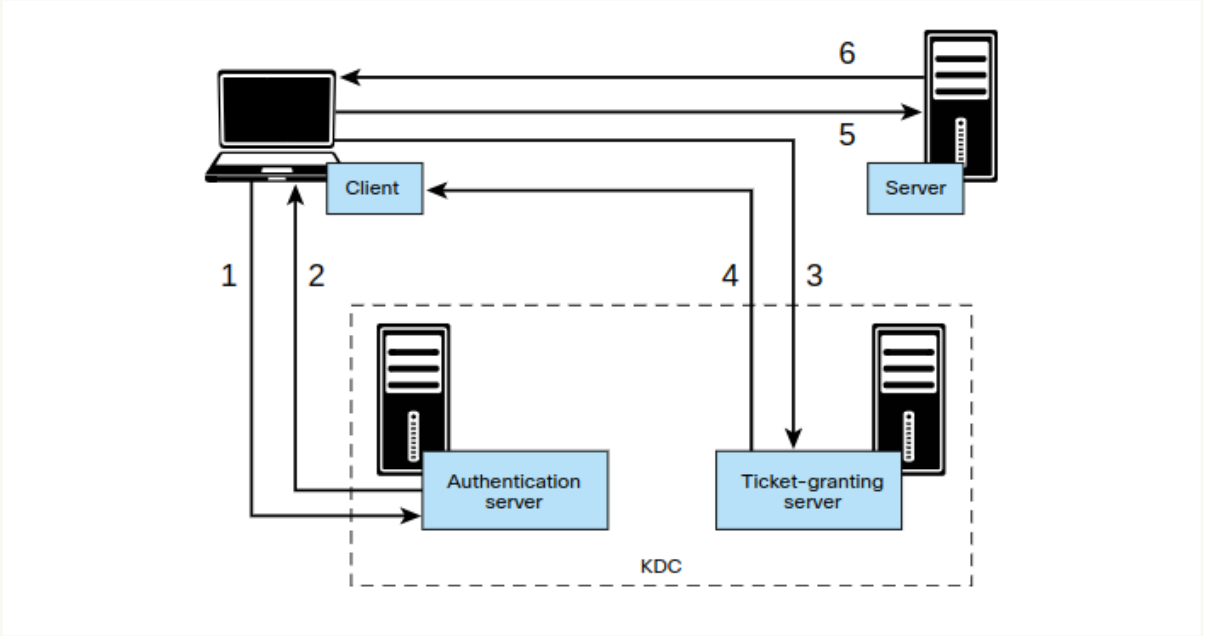
Kerberos and LDAP-Based Attacks

Kerberos is an authentication protocol defined in RFC 4120 that has been used by Windows for a number of years. Kerberos is also used by numerous applications and other operating systems. The Kerberos Consortium's website provides detailed information about Kerberos at https://www.kerberos.org. A Kerberos implementation contains three basic elements:

- **Client**
- **Server**

- **Key distribution center (KDC)**, including the authentication server and the ticket-granting server

Steps in kerberos Authentication



Step 1. The client sends a request to the authentication server within the KDC.

Step 2. The authentication server sends a session key and a ticket-granting ticket (TGT) that is used to verify the client's identity.

Step 3. The client sends the TGT to the ticket-granting server.

Step 4. The ticket-granting server generates and sends a ticket to the client.

Step 5. The client presents the ticket to the server.

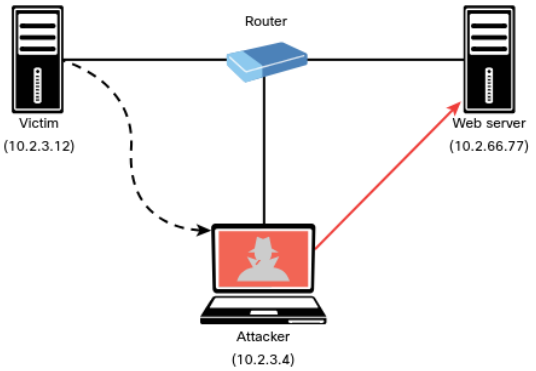
Step 6. The server grants access to the client.

Kerberoasting

Another attack against Kerberos-based deployments is Kerberoasting. Kerberoasting is a post-exploitation activity that is used by an attacker to extract service account credential hashes from Active Directory for offline cracking. It is a pervasive attack that exploits a combination of weak encryption implementations and improper password practices. Kerberoasting can be an effective attack because the threat actor can extract service account credential hashes **w**ithout sending any IP packets to the victim and without having domain admin credentials.

On Path Attacks (man-in-the-middle [MITM] attack)

In an on-path attack (**previously known as a man-in-the-middle [MITM] attack**), an attacker places himself or herself in-line between two devices or individuals that are communicating in order to eavesdrop (that is, steal sensitive data) or manipulate the data being transferred (such as by performing data corruption or data modification). On-path attacks can happen at Layer 2 or Layer 3. Figure 5-5 illustrates an on-path attack.



ARP Spoofing and ARP Cache Poisoning

ARP cache poisoning (also known as ARP spoofing) is an example of an attack that leads to an on-path attack scenario. An ARP spoofing attack can target hosts, switches, and routers connected to a Layer 2 network by poisoning the ARP caches of systems connected to the subnet and intercepting traffic intended for other hosts on the subnet. In Figure 5-5, the attacker spoofs Layer 2 MAC addresses to make the victim believe that the Layer 2 address of the attacker is the Layer 2 address of its default gateway (10.2.3.4). The packets that are supposed to go to the default gateway are forwarded by the switch to the Layer 2 address of the attacker on the same network. The attacker can forward the IP packets to the correct destination in order to allow the client to access the web server (10.2.66.77).

Media Access Control (MAC) spoofing is an attack in which a threat actor impersonates the MAC address of another device (typically an infrastructure device such as a router). The MAC address is typically a hard-coded address on a network interface controller. In virtual environments, the MAC address could be a virtual address (that is, not assigned to a physical adapter). An attacker could spoof the MAC address of physical or virtual systems to either circumvent access control measures or perform an on-path attack.

NOTE A common mitigation for ARP cache poisoning attacks is to use Dynamic Address Resolution Protocol (ARP) Inspection (DAI) on switches to prevent spoofing of the Layer 2 addresses.

Another example of a Layer 2 on-path attack involves placing a switch in the network and manipulating Spanning Tree Protocol (STP) to make it the root switch. This type of attack can allow an attacker to see any traffic that needs to be sent through the root switch.

An attacker can carry out an on-path attack at Layer 3 by placing a rogue router on the network and then tricking the other routers into believing that this new router has a better path than other routers. It is also possible to perform an on-path attack by compromising the victim's system and installing malware that can intercept the packets sent by the victim. The malware can capture packets before they are encrypted if the victim is using SSL/TLS/HTTPS or any other mechanism. An attack tool called SSLStrip uses on-path functionality to transparently look at HTTPS traffic, hijack it, and return non-encrypted HTTP links to the user in response. This tool was created by a security researcher called Moxie Marlinspike. You can download the tool from *https://github.com/moxie0/sslstrip*.

The following are some additional Layer 2 security best practices for securing your infrastructure:

- Select an unused VLAN (other than VLAN 1) and use it as the native VLAN for all your trunks. Do not use this native VLAN for any of your enabled access ports. Avoid using VLAN 1 anywhere because it is the default.
- Administratively configure switch ports as access ports so that users cannot negotiate a trunk; also disable the negotiation of trunking (that is, do not allow Dynamic Trunking Protocol [DTP]).
- Limit the number of MAC addresses learned on a given port by using the port security feature.
- Control Spanning Tree to stop users or unknown devices from manipulating it. You can do so by using the BPDU Guard and Root Guard features.
- Turn off Cisco Discovery Protocol (CDP) on ports facing untrusted or unknown networks that do not require CDP for anything positive. (CDP operates at Layer 2 and might provide attackers information you would rather not disclose.)

- On a new switch, shut down all ports and assign them to a VLAN that is not used for anything other than a parking lot. Then bring up the ports and assign correct VLANs as the ports are allocated and needed.
- Use Root Guard to control which ports are not allowed to become root ports to remote switches.
- Use DAI.
- Use IP Source Guard to prevent spoofing of Layer 3 information by hosts.
- Implement 802.1X when possible to authenticate and authorize users before allowing them to communicate to the rest of the network.
- Use Dynamic Host Configuration Protocol (DHCP) snooping to prevent rogue DHCP servers from impacting the network.
- Use storm control to limit the amount of broadcast or multicast traffic flowing through a switch. An attacker could perform a *packet storm* (or broadcast storm) attack to cause a DoS condition. The attacker does this by sending excessive transmissions of IP packets (often broadcast traffic) in a network.
- Deploy access control lists (ACLs), such as Layer 3 and Layer 2 ACLs, for traffic control and policy enforcement.

Downgrade Attacks

In a downgrade attack, an attacker forces a system to favor a weak encryption protocol or hashing algorithm that may be susceptible to other vulnerabilities. An example of a downgrade vulnerability and attack is the Padding Oracle on Downgraded Legacy Encryption (POODLE) vulnerability in OpenSSL, which allowed the attacker to negotiate the use of a lower version of TLS between the client and server. You can find more information about the POODLE vulnerability at *https://www.openssl.org/~bodo/ssl-poodle.pdf*.

POODLE was an OpenSSL-specific vulnerability and has been patched since 2014. However, in practice, removing backward compatibility is often the only way to prevent any other downgrade attacks or flaws.

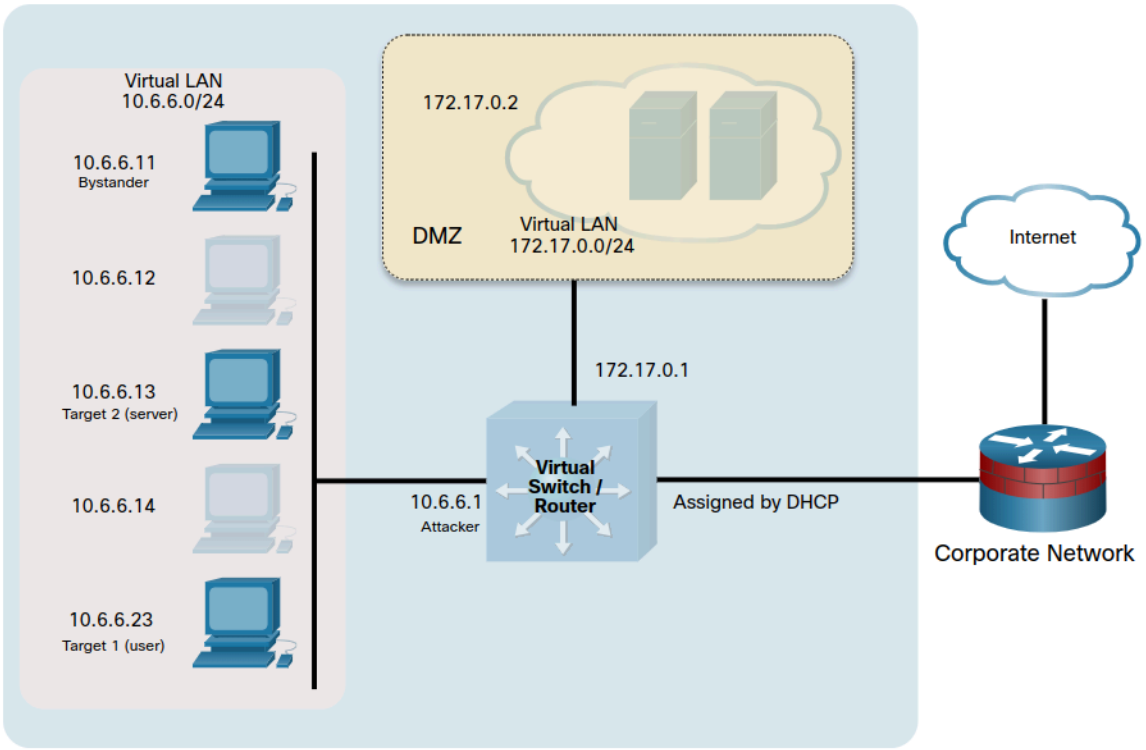
Kerberos golden ticket	In this type of attack, an attacker can manipulate Kerberos tickets based on available hashes by compromising a vulnerable system and obtaining the local user credentials and password hashes.
Kerberos silver ticket	In this type of attack, the attacker uses a forged service authentication ticket for a given service on a particular server to compromise accounts on the server.
Pass-the-Hash	In this type of attack, an attacker can use a password hash collected from a compromised system to log into another client or server.
On-Path	In this type of attack, the attacker intercepts communications between two or more communicating devices in order to steal or modify sensitive data.
Kerberoasting	In this attack, an attacker exploits a combination of weak encryption implementations and improper password practices to extract service account credential hashes from Active Directory for offline cracking.

On-Path Attacks with Ettercap

Once we have gained access to a client's LAN, we often will use a tool like **Ettercap** to conduct an on-path attack. **On-path,** or **MITM**, attacks can be very damaging because they enable manipulation or theft of any unencrypted data that is sent between a host and a destination. It is especially damaging when an attacker masquerades as the default gateway for a LAN segment, because the attacker can intercept all traffic that is destined for remote networks, such as the internet, because it must pass through the default gateway. On a large LAN, the chances of intercepting sensitive communications are quite high.

In this lab, you will practice a common form of on-path attack using a Kali tool. In doing so, you can observe how the exploit occurs, which will help you to understand the nature of on-path attacks.

Topology



Objectives

- Part 1: Launch Ettercap and Explore Its Capabilities
- Part 2: Perform the On-Path (MITM) Attack
- Part 3: Use Wireshark to observe the ARP Spoofing Attack

On-path attacks are very powerful ways to steal data that is travelling on a network. Without end-to-end encryption, as with much data travelling on local LANs, it is easy to capture clear text information, and even complete files, using on-path attack methods.

Note: On-path is replacing man-in-the-middle (MITM) as the name of this type of attack. The replacement process is incomplete; however, so you may still see MITM in many places, including some exam questions. Just be aware that the two terms are currently interchangeable.

Launch Ettercap and Explore its Capabilities

Ettercap is used to perform on-path (MITM) attacks. The goal of an on-path attack is to intercept traffic between devices to obtain information that can be used to impersonate the target or to alter data being transmitted. The attacker is situated "between " two communicating hosts. In on-path attacks, the hacker doesn't need to compromise the target device, but can just sniff traffic passing back and forth between the target and destination. **Ettercap** is used as an on-path tool, and the attack machine is on the same IP network as the victim.

Set up an ARP spoofing attack

In this attack, you will use **ARP spoofing** to redirect traffic on the local virtual network to your kali Linux system at **10.6.6.1**. ARP spoofing is often used to impersonate the default gateway router to capture all traffic entering or leaving the local IP network. Because your lab environment uses an internal virtual network, instead of spoofing the default gateway, you will use ARP spoofing to redirect traffic that is destined for a local server with the address **10.6.6.13**.

The target host in this lab is the Linux device at 10.6.6.23. To view the network from the target perspective, and initiate traffic between the target and the server, use **SSH** to log in to this host. The username is **labuser** and the password is **Cisco123**.

The user of the 10.6.6.23 host is communicating with the server at 10.6.6.13. The on-path attacker at 10.6.6.1(your kali VM) will intercept and relay traffic between these hosts.

```
┌──(kali㉿Kali)-[~]
└─$ ssh -l labuser 10.6.6.23
The authenticity of host '10.6.6.23 (10.6.6.23)' can't be established.
ED25519 key fingerprint is SHA256:u3Yjj1imvIGFFU6uLfJlAyM+BC1AXhLyO45oPedjNk8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? Y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.6.6.23' (ED25519) to the list of known hosts.
labuser@10.6.6.23's password:
Linux gravemind 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023-06-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
labuser@gravemind:~$
```

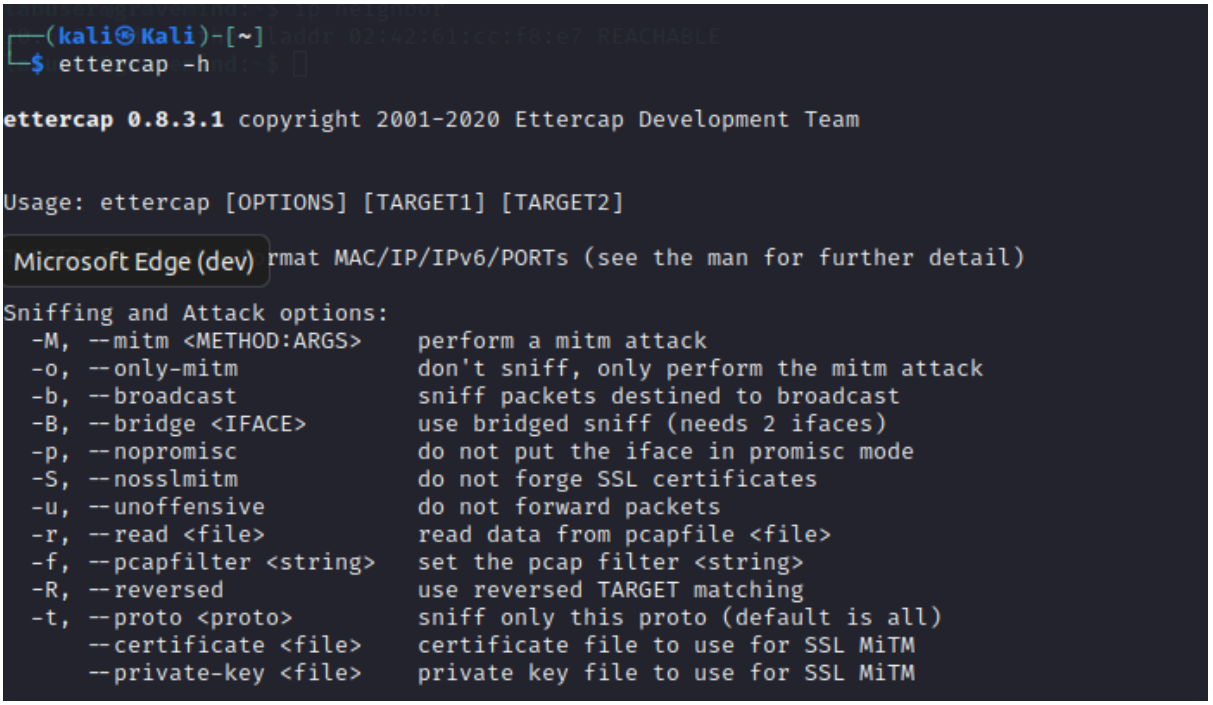
Because you are creating an on-path attack that uses ARP spoofing, you will be monitoring the ARP mappings on the victim host. The attack will cause changes to those mappings.
Use the command ip neighbor to view the current ARP cache on the target computer. Note: The hostname **gravemind** maybe different for your Kali VM environment.

```
labuser@gravemind:~$ ip neighbor
10.6.6.1 dev eth0 lladdr 02:42:61:cc:f8:e7 REACHABLE
labuser@gravemind:~$
```

Note: If you are using the ARM CPUs (Apple M1/M2) version of the VM, you will need to switch to use the root user with the password Cisco123 and use the command arp -a in place of ip neighbor to view the current ARP cache throughout this activity.

Load Ettercap GUI Interface to begin scanning

Use the **ettercap -h** command to view the help file for the Ettercap application

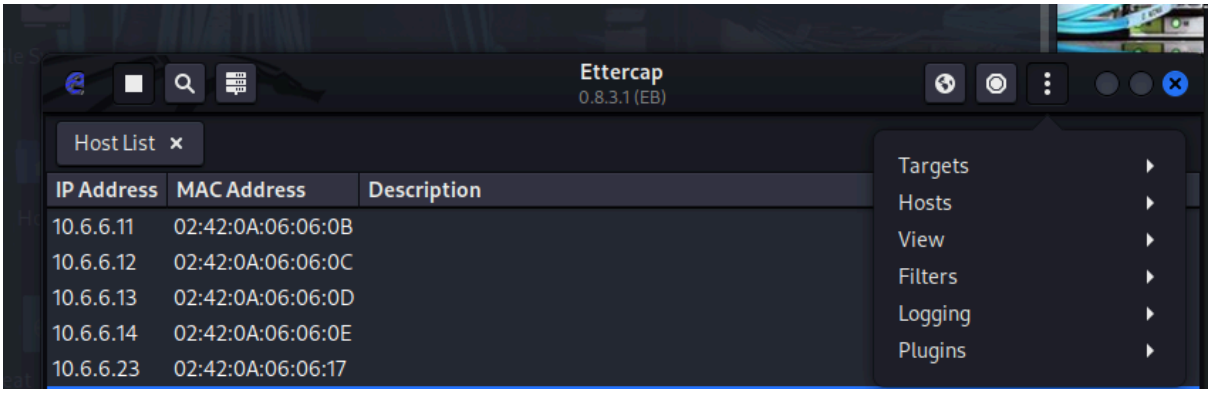


The Ettercap GUI opens in a new window. You are sniffing traffic on an internal, virtual network. The default setup is to scan using interface eth0. Change the sniffing interface to br-internal, which is the interface that is configured on the 10.6.6.0/24 virtual network, by changing the value in the Setup > Primary Interface dropdown. Click the checkbox icon at the top right of the Ettercap screen to continue. A message appears at the bottom of the screen indicating that Unified sniffing has started.

Perform the on-Path(MITM) Attack

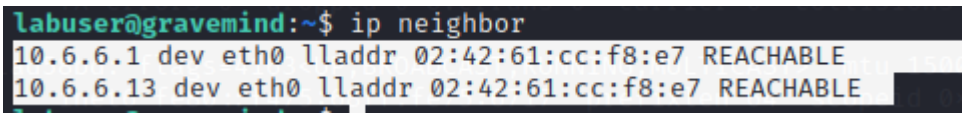
Select the target Devices

In the ettercap GUI window, open the Hosts List window by clicking the ettercap menu (**three dots icon**) Select the Hosts entry and then **Hosts List**. Click the **Scan for Hosts icon** (magnifying glass) at top left in the menu bar. A list of the hosts that were discovered on the 10.6.6.0/24 network appears in the Host List window.



Perform the ARP spoofing attack

Return to the terminal window that is running the SSH session with the target user host at 10.6.6.23. Repeat the ping to 10.6.6.13.



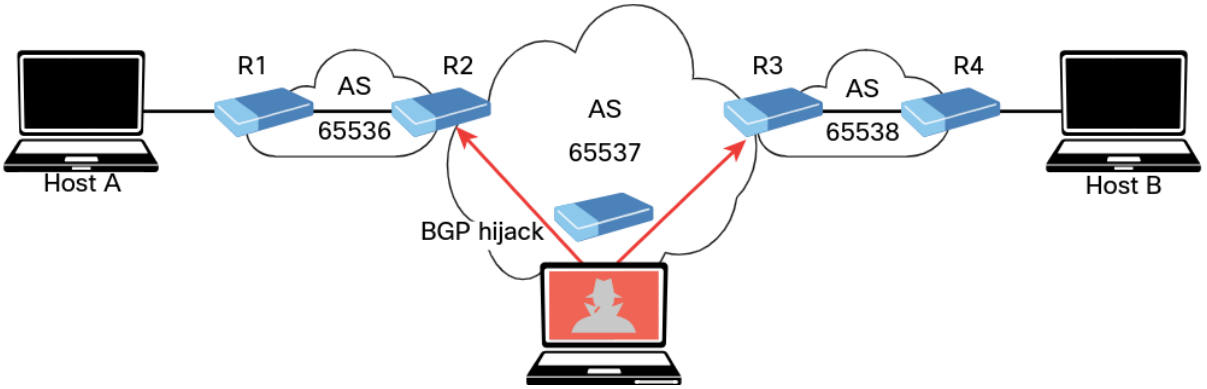
Use Wireshark to Observe the ARP Spoofing Attack

Select the target Devices and Platform the MITM attack using the CLI

In this step, you will use the command line interface in Ettercap to perform ARP spoofing and write a .pcap file that can be opened in wireshark. Refer to the help information for Ettercap to interpret the options used in the commands.

Route Manipulation Attacks

Although many different route manipulation attacks exist, one of the most common is the BGP hijacking attack. **Border Gateway Protocol (BGP)** is a dynamic routing protocol used to route Internet traffic. An attacker can launch a BGP hijacking attack by configuring or compromising an edge router to announce prefixes that have not been assigned to his or her organization. If the malicious announcement contains a route that is more specific than the legitimate advertisement or that presents a shorter path, the victim's traffic could be redirected to the attacker. In the past, threat actors have leveraged unused prefixes for BGP hijacking in order to avoid attention from the legitimate user or organization. Figure below illustrates a **BGP hijacking** route manipulation attack. The attacker compromises a router and performs a BGP hijack attack to intercept traffic between Host A and Host B.



Dos and DDos Attacks

Denial-of-service (DoS) and distributed Dos (DDos) attacks have been around for quite some time, but there has been heightened awareness of them over the past few years. **Dos attacks can generally be divided into three categories, described in the following sections:**

- Direct
- Botnet
- Reflected
- Amplification

Direct DoS Attacks

A direct DoS attack occurs when the source of the attack generates the packets, regardless of protocol, application, and so on, that are sent directly to the victim of the attack. Figure -1 illustrates a direct DoS attack.

In Figure-1 the attacker launches a direct DoS attack to a web server (the victim) by sending numerous TCP SYN packets. This type of attack is aimed at flooding the victim with an overwhelming number of packets in order to oversaturate its connection bandwidth or deplete the target's system resources. This type of attack is also known as a *SYN flood attack*.

Cybercriminals can also use DoS and DDoS attacks to produce added costs for the victim when the victim is using cloud services. In most cases, when you use a cloud service such as Amazon Web Services (AWS), Microsoft Azure, or Digital Ocean, you pay per usage. Attackers can launch DDoS attacks to cause you to pay more for usage and resources.

Another type of DoS attack involves exploiting vulnerabilities such as buffer overflows to cause a server or even a network infrastructure device to crash, subsequently causing a DoS condition.

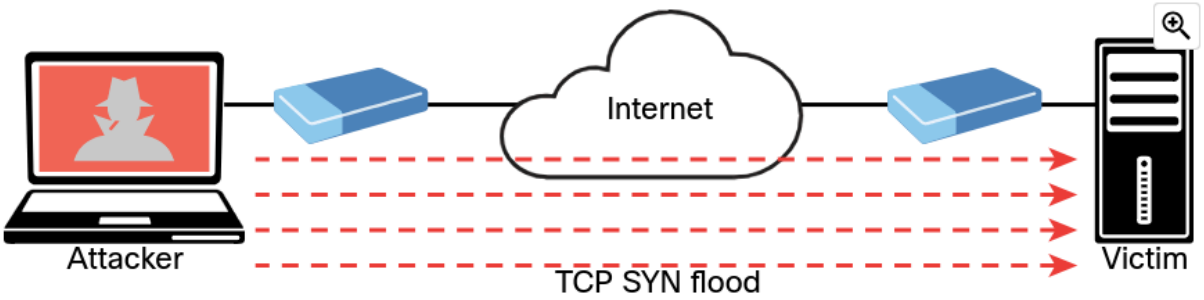


Figure-1

Botnet

Many attackers use botnets to launch DDoS attacks. A _botnet _is a collection of compromised machines that the attacker can manipulate from a command and control (CnC, or C2) system to participate in a DDoS attack, send spam emails, and perform other illicit activities. Figure-2 shows how an attacker may use a botnet to launch a DDoS attack. The botnet is composed of compromised user endpoints (laptops), home wireless routers, and Internet of Things (IoT) devices such as IP cameras.

In Figure 5-8, the attacker sends instructions to the C2; subsequently, the C2 sends instructions to the bots within the botnet to launch the DDoS attack against the victim server.

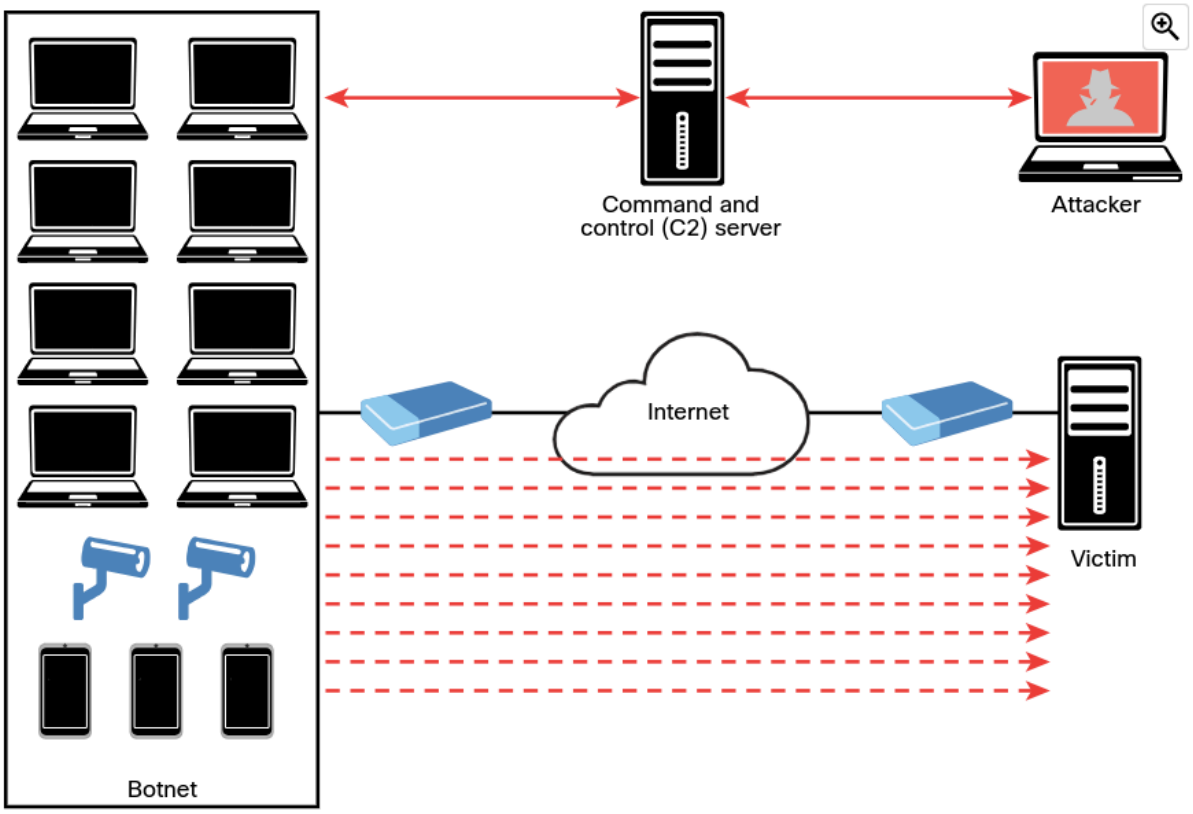


Figure-2

Reflected DoS and DDoS Attacks

In reflected DoS and DDoS attacks, attackers send spoofed packets to sources, making them believe the packets are from the victim. The sources then unknowingly send response traffic to the victim. UDP is commonly used due to its lack of a three-way handshake. For example, an attacker can send spoofed NTP requests to a source, which then replies to the victim, causing an unwanted flood of traffic. This

process, illustrated in Figure-3, shows the attacker sending packets with the victim's IP address to Host A, which then sends the response traffic to the victim, consuming bandwidth and resources.

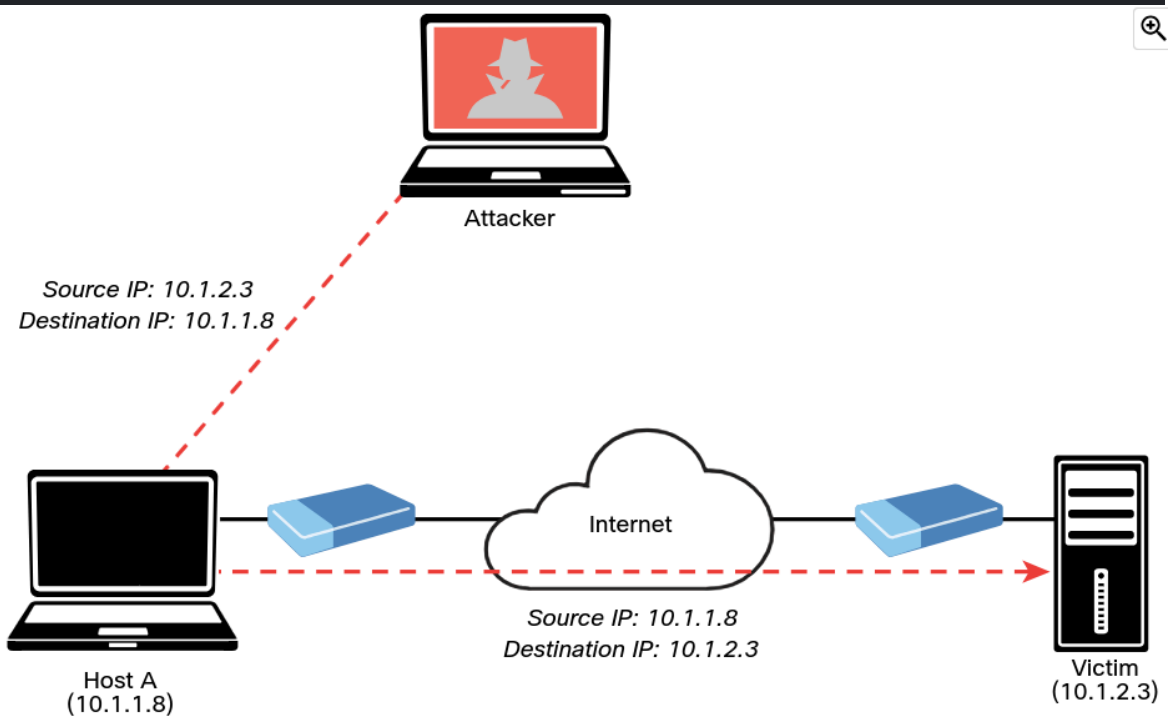


Figure-3

Amplification DDoS Attacks

An amplification attack is a form of reflected DoS attack in which the response traffic (sent by the unwitting participant) is made up of packets that are much larger than those that were initially sent by the attacker (spoofing the victim). An example of this type of attack is an attacker sending DNS queries to a DNS server configured as an open resolver. Then the DNS server (open resolver) replies with responses much larger in packet size than the initial query packets. The end result is that the victim's machine gets flooded by large packets for which it never actually issued queries. Figure-4 shows an example.

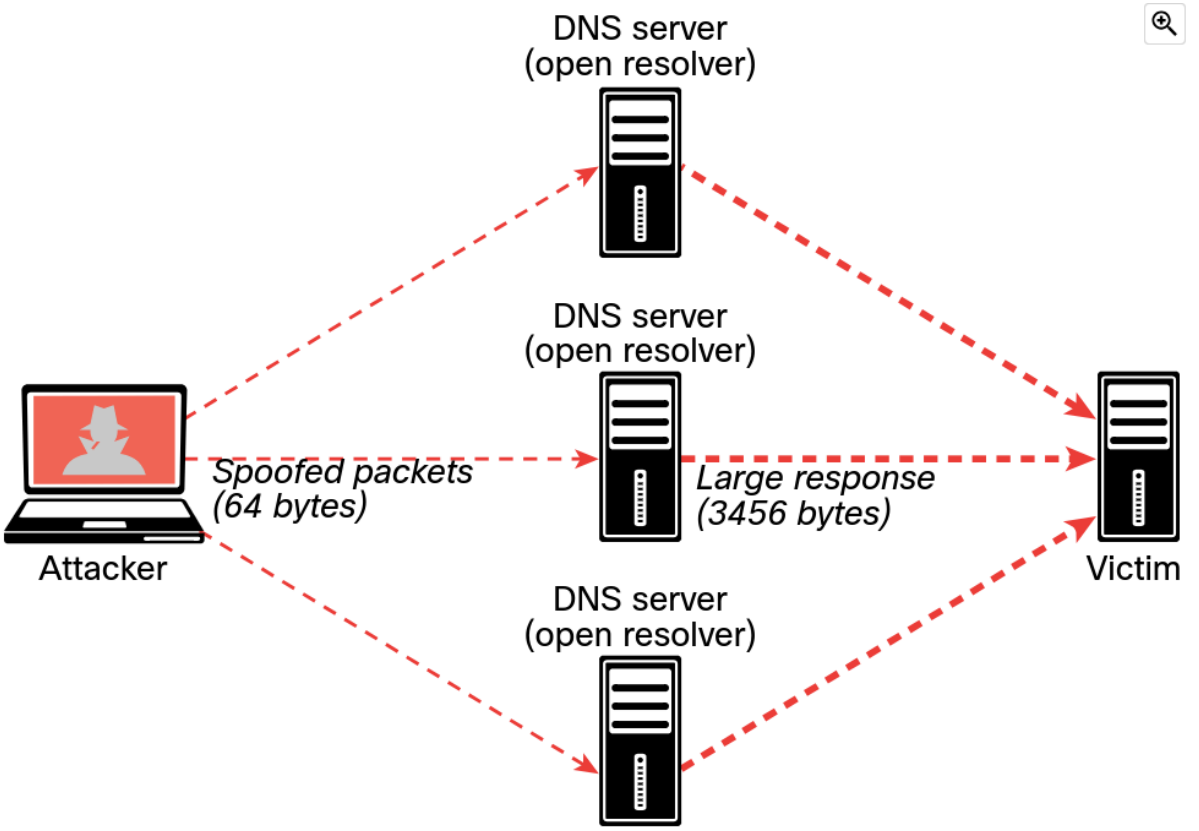


Figure-3

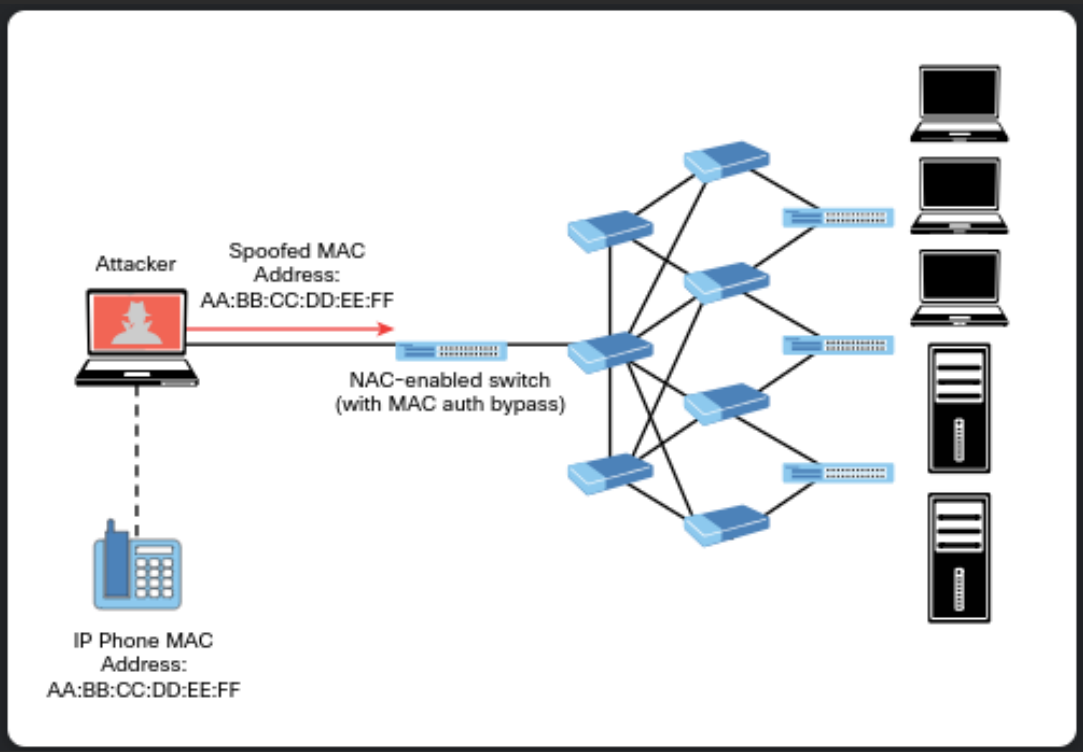
As a penetration tester, you might be tasked with performing different types of stress testing for availability and demonstrating how a DDos attack can potentially affect a system or a network. In most cases, those types of stress tests are performed in a controlled environment and are typically out of scope in production systems.

Network Access Control(NAC) Bypass

NAC (Network Access Control) is a technology that checks the security status of devices before they join a wired or wireless network. It uses 802.1X for identity management and ensures endpoints have necessary security software and updates. NAC-enabled devices intercept network traffic like DHCP and ARP requests to detect connecting devices and may use client-based agents for security assessments. They can also perform MAC authentication bypass for specific devices like printers and IP phones, allowing them to join the network using a whitelist of MAC addresses. This process can be manually configured by network administrators for specific VLAN access.

An attacker could easily spoof an authorized MAC address (in a process called *MAC address spoofing*) and bypass a NAC configuration. For example, it is possible to spoof the MAC address of an IP phone and use it to connect to a network. This is because a port for which MAC auth bypass is enabled can be dynamically enabled or disabled based on the MAC address of the device that connects to it. Figure 5-11 illustrates this scenario.

Figure - Abusing MAC Auth Bypass Implementations



VLAN Hopping

One way to identify a LAN is to say that all the devices in the same LAN have a common Layer 3 IP network address and that they also are all located in the same Layer 2 broadcast domain. A virtual LAN (VLAN) is another name for a Layer 2 broadcast domain. A VLAN is controlled by a switch. The switch also controls which ports are associated with which VLANs. In Figure 5-12, if the switches are in their default configuration, all ports by default are assigned to VLAN 1, which means all the devices, including the two users and the router, are in the same broadcast domain, or VLAN.

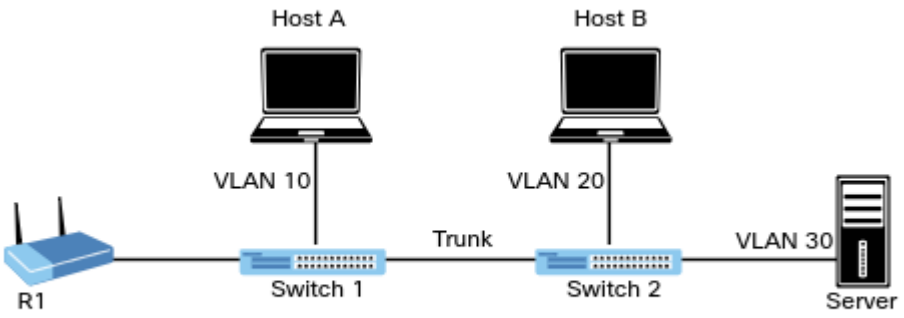


Figure 5-12. Understanding VLANs

To manage large numbers of users, assign them to VLANs based on subnet groups using switch port configurations. VLANs enable devices in the same group to communicate via shared IP addresses. Inter-switch links are set as trunks to tag frames with VLAN IDs, ensuring correct transmission across switches. VLANs allow devices to communicate locally; routing between VLANs requires a router with distinct interfaces. VLAN hopping, a security concern, involves methods like switch spoofing and double tagging, which can be mitigated by best practices such as avoiding VLAN 1 and securing trunk negotiations.

DHCP Starvation Attacks and Rogue DHCP Servers

Most organizations run DHCP servers. The two most popular attacks against DHCP servers and infrastructure are DHCP starvation and DHCP spoofing (which involves rogue DHCP servers). In a DHCP starvation attack, an attacker broadcasts a large number of DHCP request messages with spoofed source MAC addresses, as illustrated in Figure-1 below.

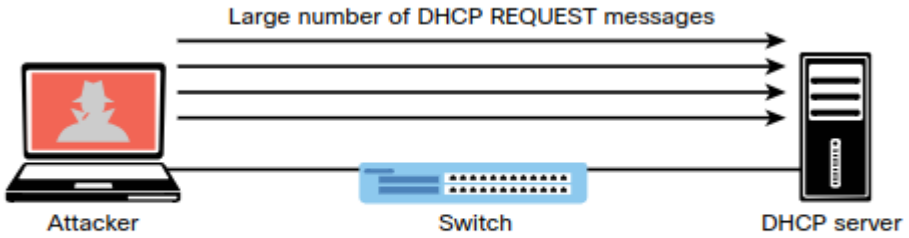


Figure-1 **DHCP Starvation Attack**

If the DHCP server responds to all these fake DHCP REQUEST messages, available IP addresses in the DHCP server scope are depleted within a few minutes or seconds. After the available number of IP addresses in the DHCP server is depleted, the attacker can then set up a rogue DHCP server and respond to new DHCP requests from network DHCP clients, as shown in Figure-2.

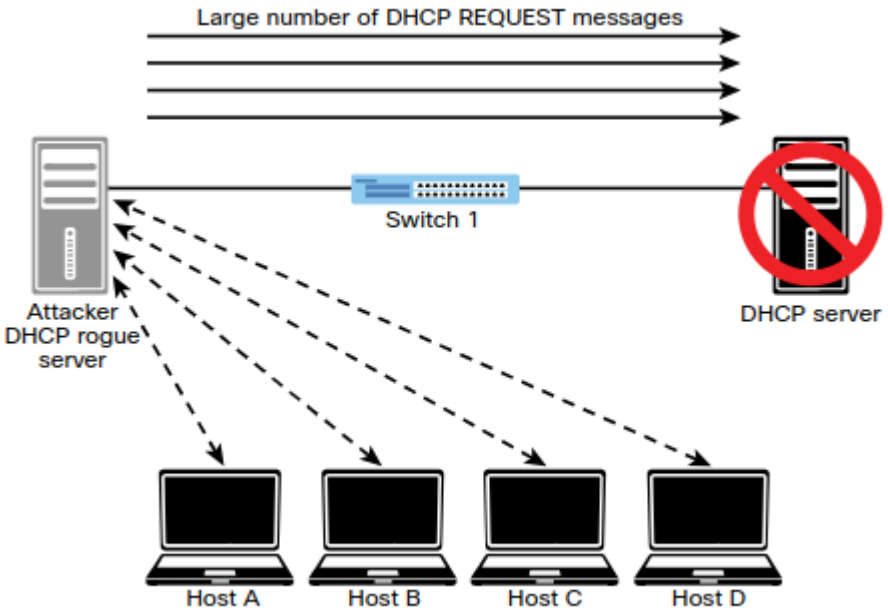


Figure-2 **Rogue DHCP Servers and DHCP Spoofing Attacks**

The attacker in Figure-2 sets up a rogue DHCP server to launch a DHCP spoofing attack. The attacker can set the IP address of the default gateway and DNS server to itself so that it can intercept the traffic from the network hosts.

Figure-3 shows an example of a tool called **Yersenia** that can be used to create a rogue DHCP server and launch DHCP starvation and spoofing attacks.

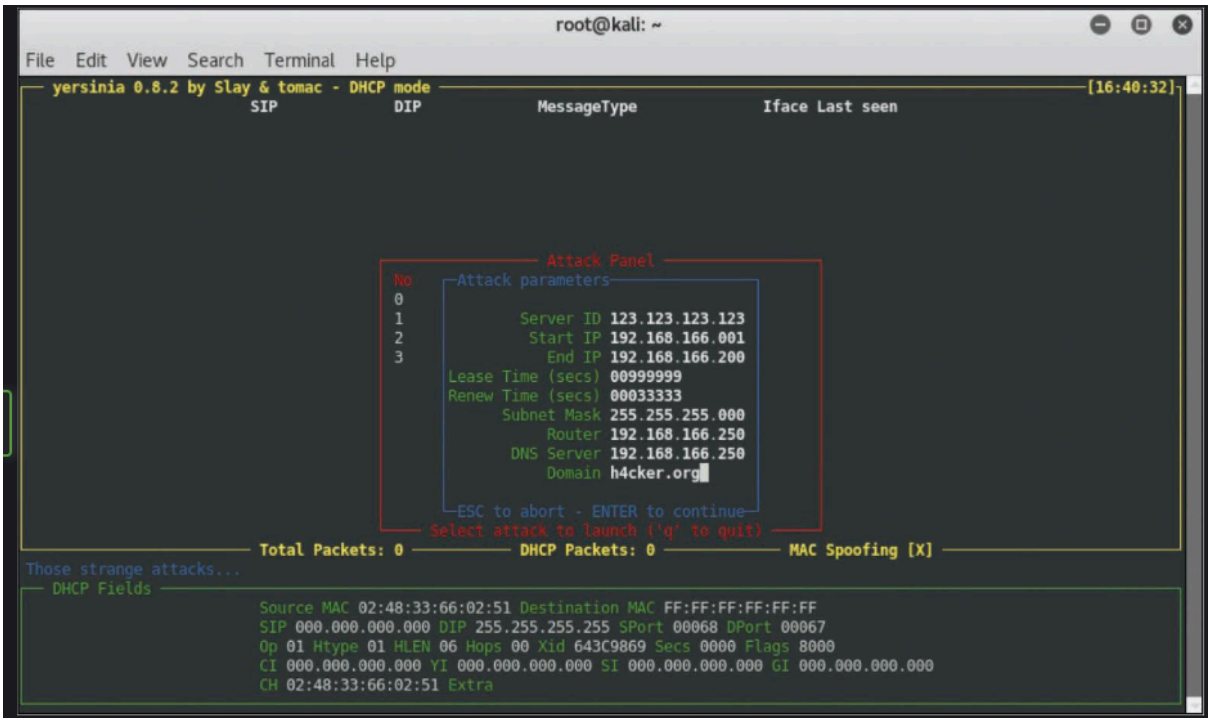


Figure-3 **Setting Up a Rogue DHCP Server in Yersenia**

Exploiting Wireless Vulnerabilities

Rogue Access Points

One of the most simplistic wireless attacks involves an attacker installing a **rogue AP** in a network to fool users to connect to that AP. Basically, the attacker can use the **rogue AP to create a backdoor** and obtain access to the network and its systems, as illustrated in Figure-1

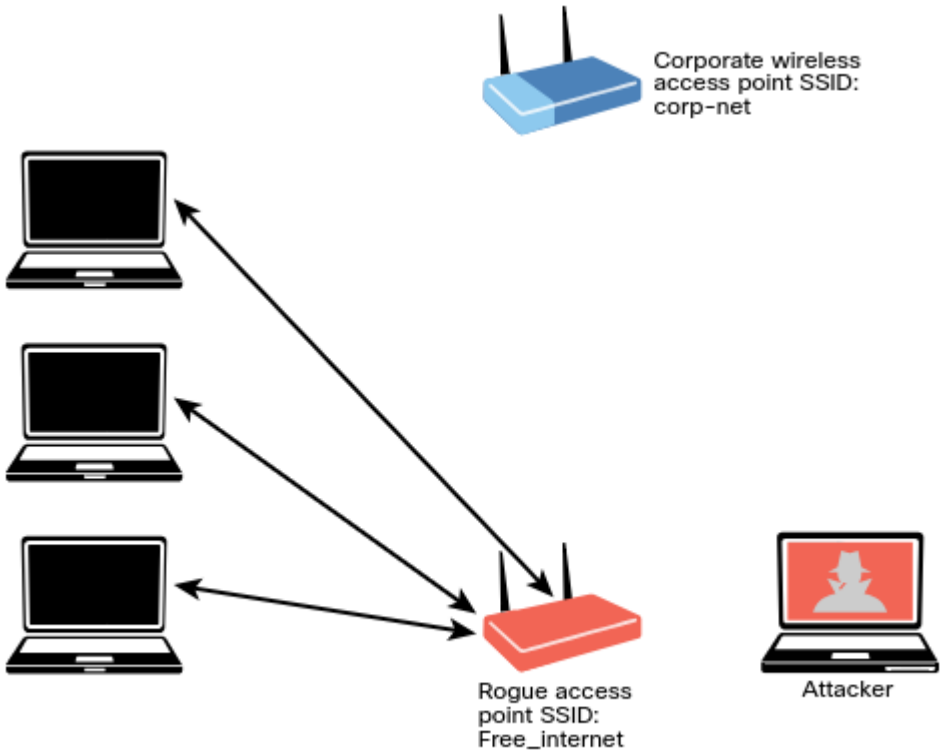


Figure-1 **Rogue Wireless Access Point**

Evil Twin Attacks

In an evil twin attack, the attacker creates a rogue access point and configures it exactly the same as the existing corporate network, as illustrated in Figure-2.

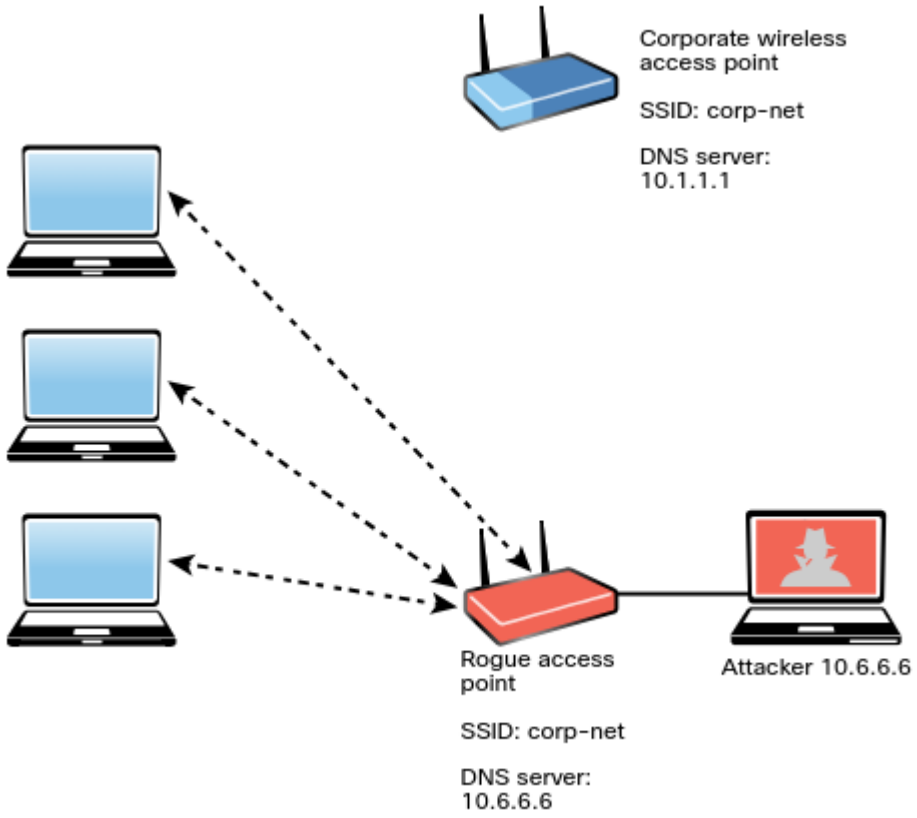


Figure-2 **Evil Twin Attack**

Typically, the attacker uses DNS spoofing to redirect the victim to a cloned captive portal or a website. When users are logged on to the evil twin, a hacker can easily inject a spoofed DNS record into the DNS cache, changing the DNS record for all users on the fake network. Any user who logs in to the evil twin will be redirected by the spoofed DNS record injected into the cache. An attacker who performs a DNS cache poisoning attack wants to get the DNS cache to accept a spoofed record. Some ways to defend against DNS spoofing are using packet filtering, cryptographic protocols, and spoofing detection features provided by modern wireless implementations.

Disassociation(or Deauthentication) Attacks

An attacker can cause legitimate wireless clients to deauthenticate from legitimate wireless APs or wireless routers to either perform a DoS condition or to make those clients connect to an evil twin. This type of attack is also known as a disassociation attack because the attacker disassociates (tries to disconnect) the user from the

authenticating wireless AP and then carries out another attack to obtain the user's valid credentials.
A service set identifier (SSID) is the name or identifier associated with an 802.11 wireless local area network (WLAN). SSID names are included in plaintext in many wireless packets and beacons. A wireless client needs to know the SSID in order to associate with a wireless AP. It is possible to configure wireless passive tools like Kismet or KisMAC to listen to and capture SSIDs and any other wireless network traffic. In addition, tools such as *Airmon-ng* (which is part of the *Aircrack-ng suite*) can perform this reconnaissance. The Aircrack-ng suite of tools can be downloaded from https://www.aircrack-ng.org. Example 5-14 shows the Airmon-ng tool. The system in this example has five different wireless network adapters, and the adapter wlan1 is used for monitoring.

The **Airodump-ng tool** (which is also part of the Aircrack-ng suite) can be used to sniff and analyze wireless network traffic, as shown in Example.

```
┌──(root㉿Kali)-[/home/kali]
└─# airodump-ng wlan1mon
nl80211 not found.
Interface wlan1mon:
ioctl(SIOCGIFINDEX) failed: No such device
Failed initializing wireless card(s): wlan1mon
```

You can use the **Airodump-ng tool** to sniff wireless networks and obtain their SSIDs, along with the channels they are operating.

NOTE Many wireless adapters do not allow you to inject packets into a wireless network. For a list of wireless adapters and their specifications that can help you build your wireless lab, see https://theartofhacking.org/github.

Preferred Network List Attacks

Operating systems and wireless supplicants (clients), in many cases, maintain a list of trusted or preferred wireless networks. This is also referred to as the preferred network list (PNL). A PNL includes the wireless network SSID, plaintext passwords, or WEP or WPA passwords. Clients use these preferred networks to automatically

associate to wireless networks when they are not connected to an AP or a wireless router.

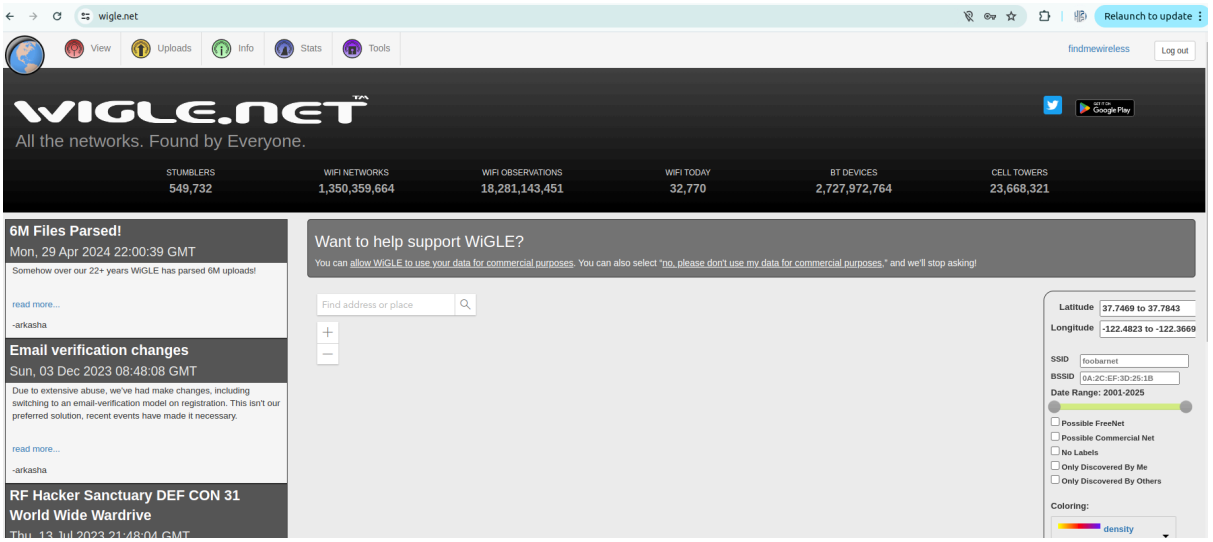
It is possible for attackers to listen to these client requests and impersonate the wireless networks in order to make the clients connect to the attackers' wireless devices and eavesdrop on their conversation or manipulate their communication.

Wireless Signal Jamming and Interference

The purpose of **jamming** wireless signals or causing wireless network interference is to create a full or partial DoS condition in the wireless network. Such a condition, if successful, is very disruptive. Most modern wireless implementations provide built-in features that can help immediately detect such attacks. In order to jam a Wi-Fi signal or any other type of radio communication, an attacker basically generates random noise on the frequencies that wireless networks use. With the appropriate tools and wireless adapters that support packet injection, an attacker can cause legitimate clients to disconnect from wireless infrastructure devices.

War Driving

War driving is a method attackers use to find wireless access points wherever they might be. By just driving (or walking) around, an attacker can obtain a significant amount of information over a very short period of time. Another similar attack is **war flying**, which involves using a portable computer or other mobile device to search for wireless networks from an aircraft, such as a drone or another unmanned aerial vehicle (UAV).



TIP A popular site among war drivers is **WiGLE** (*https://wigle.net*). The site allows users to detect Wi-Fi networks and upload information about the networks by using a mobile app.

Initialization Vector(IV) Attacks and Unsecured Wireless Protocols

An attacker can cause some modification on the initialization vector(IV) of a wireless packet that is encrypted during transmission. The goal of the attacker is to obtain a lot of information about the plaintext of a single packet and generate another encryption key that can then be used to decrypt other packets using the same IV. WEP is susceptible to many different attacks, including IV attacks.

Attacks against WEP

Because WEP is susceptible to many different attacks, it is considered an obsolete wireless protocol. WEP must be avoided, and many wireless network devices no longer support it. WEP keys exist in two sizes: 40-bit (5-byte) and 104-bit (13-byte) keys. In addition, WEP uses a 24-bit IV, which is prepended to the pre-shared key (PSK). When you configure a wireless infrastructure device with WEP, the IVs are sent in plaintext.

WEP has been defeated for decades. WEP uses RC4 in a manner that allows an attacker to crack the PSK with little effort. The problem is related to how WEP uses the IVs in each packet. When WEP uses RC4 to encrypt a packet, it prepends the IV to the secret key before including the key in RC4. Subsequently, an attacker has the first 3 bytes of an allegedly “secret” key used on every packet. In order to recover the PSK, an attacker just needs to collect enough data from the air. An attacker can

accelerate this type of attack by just injecting ARP packets (because the length is predictable), which allows the attacker to recover the PSK much faster. After recovering the WEP key, the attacker can use it to access the wireless network.

An attacker can also use the Aircrack-ng set of tools to crack (recover) the WEP PSK. To perform this attack using the Aircrack-ng suite, an attacker first launches Airmon-ng, as shown in Example 5-16.

Example 5-16 - Using Airmon-ng to Monitor a Wireless Network