

Practical Assignment - Nmap & Scapy Labs

1.Nmap

1.1 Port Scan (-sS)

A *port scan* is an active scan in which the scanning tool sends various types of probes to the target IP address and then examines the responses to determine whether the service is listening.

nmap -sS 10.0.2.15

```
root@labvm:/home/cisco
File Edit View Search Terminal Help
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 20 bytes 2020 (2.0 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 20 bytes 2020 (2.0 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@labvm:/home/cisco# nmap -sS 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 09:19 UTC
Nmap scan report for labvm (10.0.2.15)
Host is up (0.0000020s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
631/tcp   open  ipp

Nmap done: 1 IP address (1 host up) scanned in 1.47 seconds
root@labvm:/home/cisco#
```

1.2 TCP Connect Scan (-sT)

A TCP connect scan makes use of the underlying operating system's networking mechanism to establish a full TCP connection with the target device being scanned.

nmap -sT 10.0.2.15

```
root@labvm:/home/cisco
File Edit View Search Terminal Help
root@labvm:/home/cisco# nmap -sF -p 80 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 11:53 UTC
Nmap scan report for 10.0.2.15
Host is up (0.00017s latency).

PORT      STATE SERVICE
80/tcp    closed http

Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds
root@labvm:/home/cisco# nmap -sT 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 11:57 UTC
Nmap scan report for 10.0.2.15
Host is up (0.000045s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
631/tcp   open  ipp

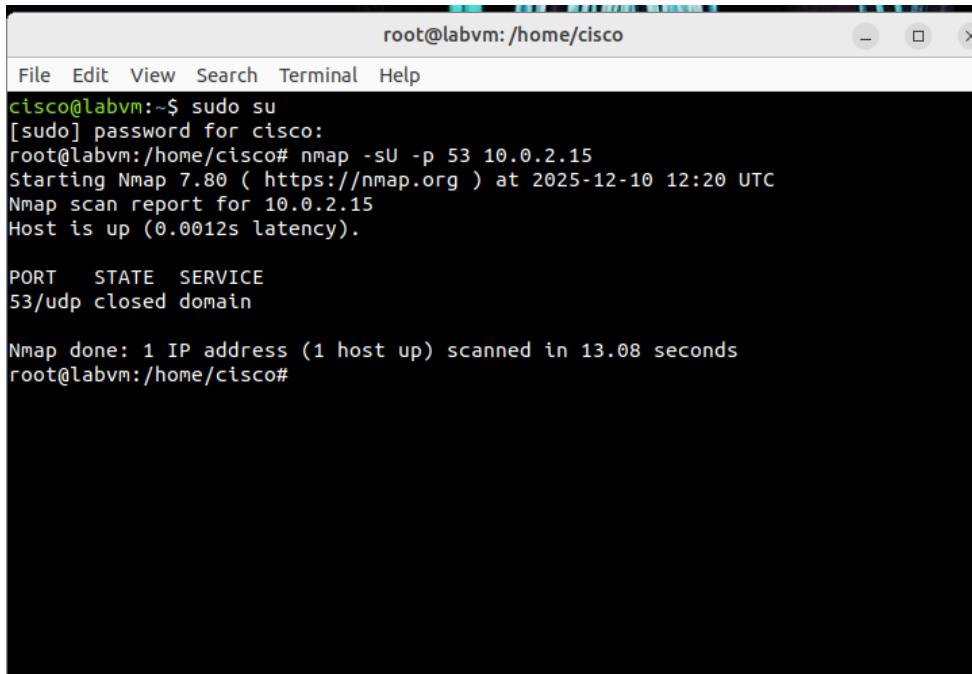
Nmap done: 1 IP address (1 host up) scanned in 13.07 seconds
root@labvm:/home/cisco#
```

1.3 UDP Scan (-sU)

A UDP scan checks which **UDP ports** on a target system may be open.

Unlike TCP, UDP is *connectionless*, so discovering open ports is slower and less reliable.

nmap -sU -p 53 10.0.2.15



```
root@labvm:/home/cisco
File Edit View Search Terminal Help
cisco@labvm:~$ sudo su
[sudo] password for cisco:
root@labvm:/home/cisco# nmap -sU -p 53 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 12:20 UTC
Nmap scan report for 10.0.2.15
Host is up (0.0012s latency).

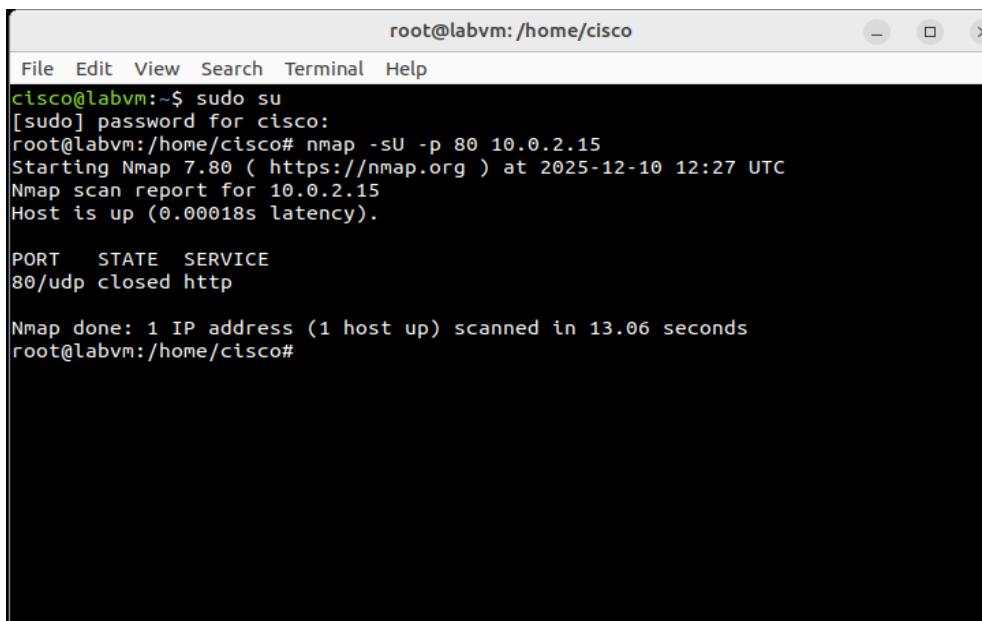
PORT      STATE SERVICE
53/udp    closed domain

Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds
root@labvm:/home/cisco#
```

1.4 TCP Fin Scan (-sF)

A TCP FIN scan is a type of stealth scan that sends TCP packets with the FIN flag set to probe the target's response.

nmap -sU -p 80 10.0.2.15



```
root@labvm:/home/cisco
File Edit View Search Terminal Help
cisco@labvm:~$ sudo su
[sudo] password for cisco:
root@labvm:/home/cisco# nmap -sU -p 80 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 12:27 UTC
Nmap scan report for 10.0.2.15
Host is up (0.00018s latency).

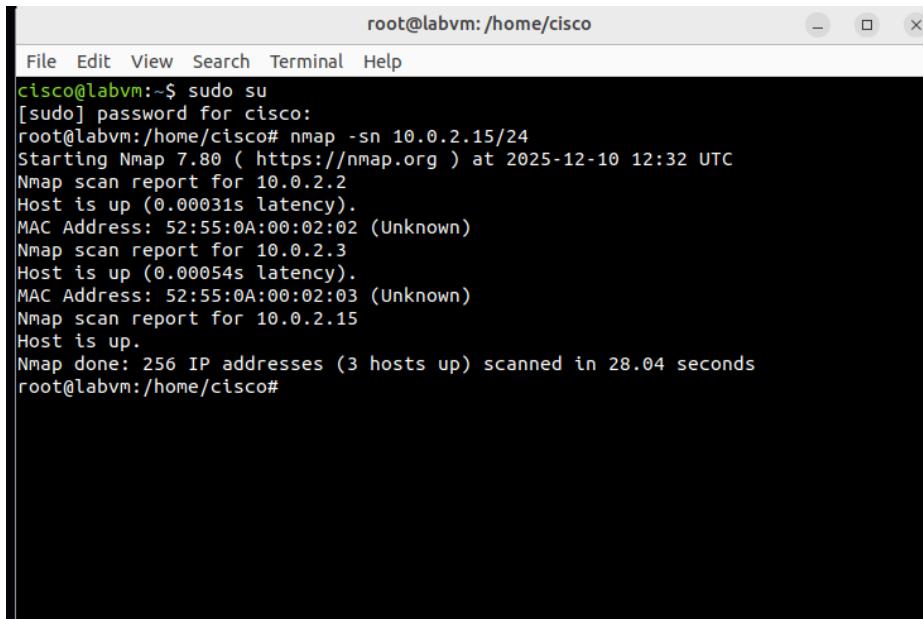
PORT      STATE SERVICE
80/udp    closed http

Nmap done: 1 IP address (1 host up) scanned in 13.06 seconds
root@labvm:/home/cisco#
```

1.5 Host Discovery Scan (-sn)

A host discovery scan is one of the most common types of scans used to enumerate hosts on a network because it can use several types of ICMP messages to determine whether a host is online and responding on a network.

nmap -sn 10.0.2.15/24

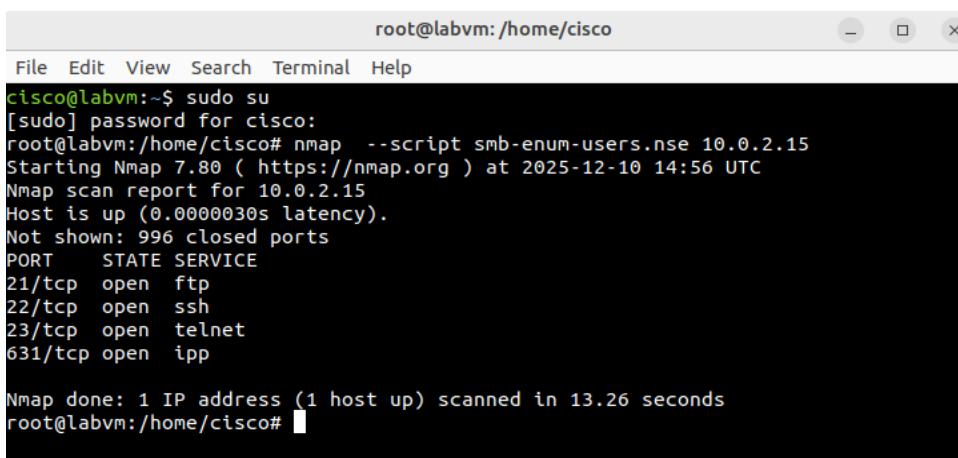


```
root@labvm:/home/cisco
File Edit View Search Terminal Help
cisco@labvm:~$ sudo su
[sudo] password for cisco:
root@labvm:/home/cisco# nmap -sn 10.0.2.15/24
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 12:32 UTC
Nmap scan report for 10.0.2.2
Host is up (0.00031s latency).
MAC Address: 52:55:0A:00:02:02 (Unknown)
Nmap scan report for 10.0.2.3
Host is up (0.00054s latency).
MAC Address: 52:55:0A:00:02:03 (Unknown)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 28.04 seconds
root@labvm:/home/cisco#
```

1.6 User Enumeration

User Enumeration is where an attacker can discover whether a specific username exists on a system.

nmap --script smb-enum-users.nse 10.0.2.15



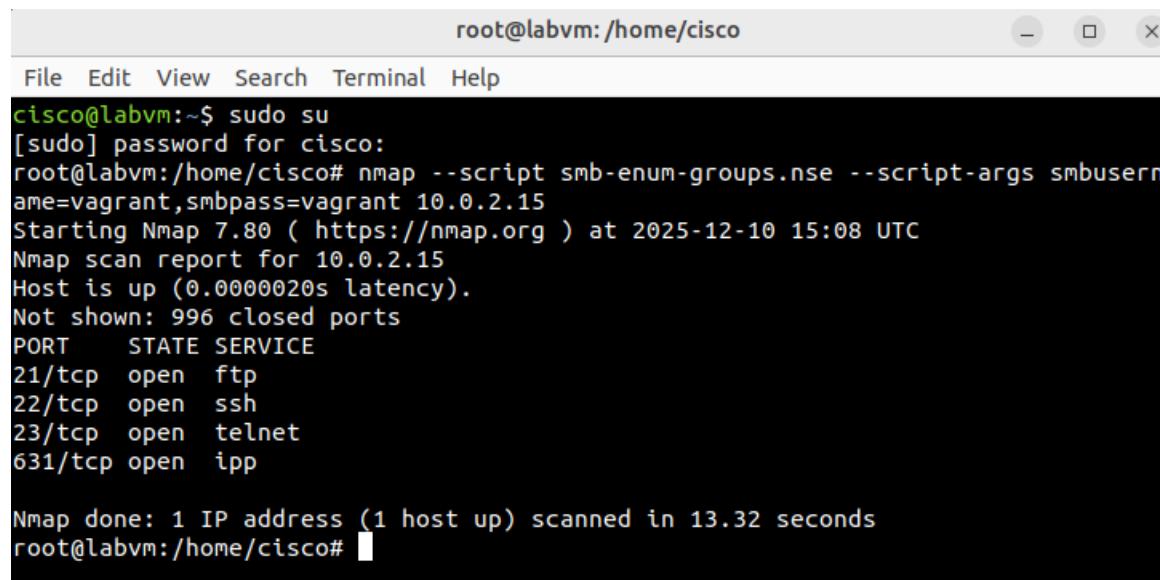
```
root@labvm:/home/cisco
File Edit View Search Terminal Help
cisco@labvm:~$ sudo su
[sudo] password for cisco:
root@labvm:/home/cisco# nmap --script smb-enum-users.nse 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 14:56 UTC
Nmap scan report for 10.0.2.15
Host is up (0.0000030s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
631/tcp   open  ipp

Nmap done: 1 IP address (1 host up) scanned in 13.26 seconds
root@labvm:/home/cisco#
```

1.7 Group Enumeration

Group Enumeration is like *user enumeration*, but instead of identifying valid users, an attacker tries to discover groups, roles, or permission sets that exist inside a system or directory service.

```
nmap --script smb-enum-groups.nse --script-args  
smbusername=vagrant,smbpass=vagrant 10.0.2.15
```



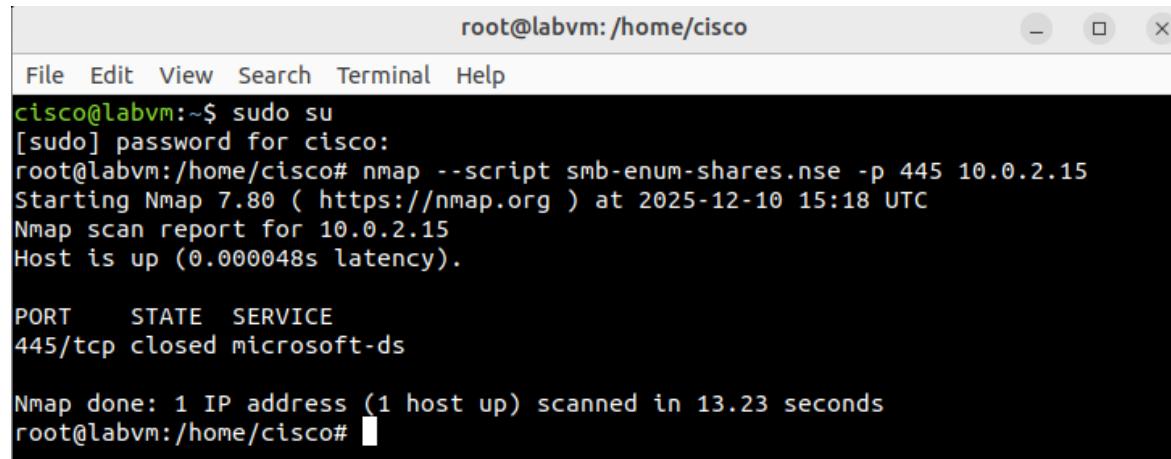
A terminal window titled "root@labvm: /home/cisco". The command "nmap --script smb-enum-groups.nse --script-args smbusername=vagrant,smbpass=vagrant 10.0.2.15" is run. The output shows the host is up and provides a table of open ports and services. The table includes rows for port 21/tcp (ftp), 22/tcp (ssh), 23/tcp (telnet), and 631/tcp (ipp). The scan took 13.32 seconds.

```
cisco@labvm:~$ sudo su  
[sudo] password for cisco:  
root@labvm:/home/cisco# nmap --script smb-enum-groups.nse --script-args smbusername=vagrant,smbpass=vagrant 10.0.2.15  
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 15:08 UTC  
Nmap scan report for 10.0.2.15  
Host is up (0.0000020s latency).  
Not shown: 996 closed ports  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
23/tcp    open  telnet  
631/tcp   open  ipp  
  
Nmap done: 1 IP address (1 host up) scanned in 13.32 seconds  
root@labvm:/home/cisco#
```

1.8 Network Share Enumeration

Network Share Enumeration refers to discovering shared folders, drives, or resources exposed on a network—usually via SMB, NFS, or other file-sharing protocols. From a defensive perspective, understanding this is crucial because misconfigured or overly permissive shares are one of the most common ways attackers gain unauthorized access, spread malware, or escalate privileges.

```
nmap --script smb-enum-shares.nse -p 445 10.0.2.15
```



A terminal window titled "root@labvm: /home/cisco". The command "nmap --script smb-enum-shares.nse -p 445 10.0.2.15" is run. The output shows the host is up and provides a table of open ports and services. The table includes a row for port 445/tcp (microsoft-ds). The scan took 13.23 seconds.

```
cisco@labvm:~$ sudo su  
[sudo] password for cisco:  
root@labvm:/home/cisco# nmap --script smb-enum-shares.nse -p 445 10.0.2.15  
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 15:18 UTC  
Nmap scan report for 10.0.2.15  
Host is up (0.000048s latency).  
  
PORT      STATE SERVICE  
445/tcp   closed microsoft-ds  
  
Nmap done: 1 IP address (1 host up) scanned in 13.23 seconds  
root@labvm:/home/cisco#
```

1.9 OS Footprinting

Nmap OS Footprinting (also called OS fingerprinting) is the process of using Nmap to identify the operating system running on a target machine. Nmap does this by analyzing how a system responds to specially crafted network packets.

nmap -O 10.0.2.15

```
root@labvm:/home/cisco
File Edit View Search Terminal Help
TCP/IP fingerprinting (for OS scan) requires root privileges.
QUITTING!
cisco@labvm:~$ sudo su
[sudo] password for cisco:
root@labvm:/home/cisco# nmap -O 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-11 08:07 UTC
Nmap scan report for labvm (10.0.2.15)
Host is up (0.000055s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
631/tcp   open  ipp
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
root@labvm:/home/cisco#
```

1.10 Service Detection

Nmap service detection is the process of identifying which services and versions are running on a target's open ports. This helps you determine what software (e.g., Apache, SSH, MySQL) is running and sometimes the exact version.

nmap -sV 10.0.2.15

```
root@labvm:/home/cisco
File Edit View Search Terminal Help
cisco@labvm:~$ sudo su
[sudo] password for cisco:
root@labvm:/home/cisco# nmap -sV 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-11 08:15 UTC
Nmap scan report for labvm (10.0.2.15)
Host is up (0.0000050s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
23/tcp    open  telnet   Linux telnetd
631/tcp   open  ipp?
Service Info: Host: Welcome; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 188.83 seconds
root@labvm:/home/cisco#
```

2. Scapy

2.1 Scapy sniff () command

The Scapy sniff () command is a function used in the Scapy Python library to capture network packets from an interface in real time. It is one of Scapy's most powerful features for network analysis, penetration testing, and packet manipulation.

Scapy v2.4.4

```
File Edit View Search Terminal Help
<frozen importlib._bootstrap>:914: ImportWarning: _SixMetaPathImporter.find_spec()
() not found; falling back to find_module()

          aSPY//YASa
          apyyyyCY/////////YCa
          sY//////YSpcs  scpCY//Pp
          ayp ayyyyyyyySCP//Pp      syY//C
          AYAsAYYYYYYYYY//Ps      cY//S
          pCCCCY//p      cSSps y//Y
          SPPPP//a      pP///AC//Y
          A//A      cyP///C
          p///Ac      sC//a
          P///YCpc      A//A
          scccccp///pSP//p      p//Y
          sY/////////y caa      S//P
          cayCyayP//Ya      pY/Ya
          sY/PsY///YCc      aC//Yp
          sc  sccaCY//PCyapaCP//YSS
          spCPY//////YPSPs
          ccaacs

          using IPython 7.31.1

>>> sniff()
^C<Sniffed: TCP:0 UDP:4 ICMP:10 Other:5>
>>> 
```

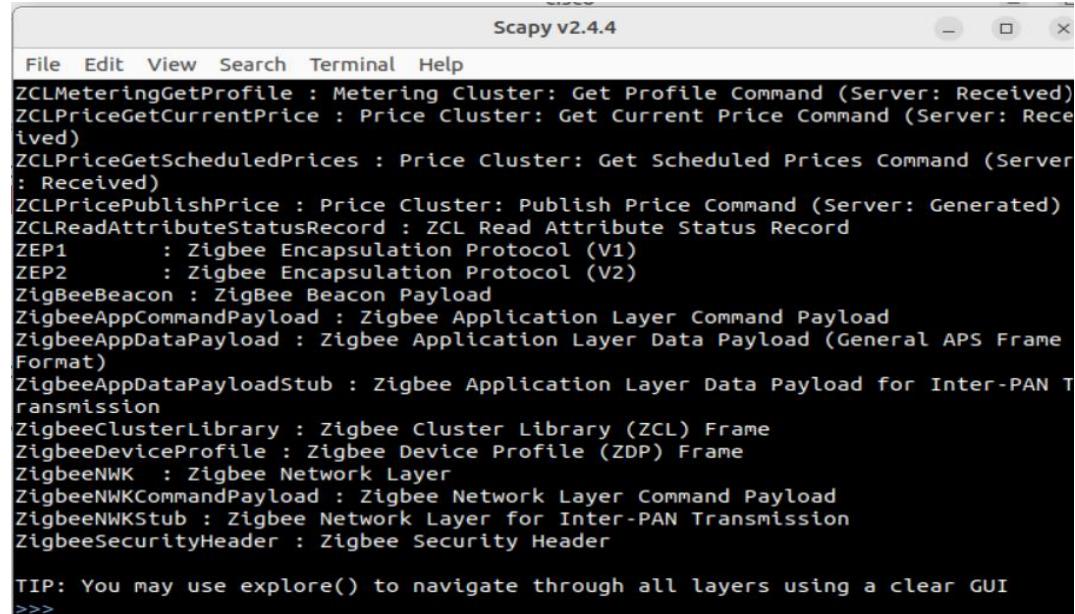
cisco@labvm:~

```
File Edit View Search Terminal Help
cisco@labvm:~$ ping -c 5 www.cisco.com
PING e2867.dsca.akamaiedge.net (2.22.192.103) 56(84) bytes of data.
64 bytes from a2-22-192-103.deploy.static.akamaitechnologies.com (2.22.192.103):
  icmp_seq=1 ttl=255 time=44.4 ms
64 bytes from a2-22-192-103.deploy.static.akamaitechnologies.com (2.22.192.103):
  icmp_seq=2 ttl=255 time=43.8 ms
64 bytes from a2-22-192-103.deploy.static.akamaitechnologies.com (2.22.192.103):
  icmp_seq=3 ttl=255 time=44.9 ms
64 bytes from a2-22-192-103.deploy.static.akamaitechnologies.com (2.22.192.103):
  icmp_seq=4 ttl=255 time=223 ms
64 bytes from a2-22-192-103.deploy.static.akamaitechnologies.com (2.22.192.103):
  icmp_seq=5 ttl=255 time=36.6 ms

--- e2867.dsca.akamaiedge.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 36.635/78.543/222.943/72.263 ms
cisco@labvm:~$ 
```

2.2 Scapy ls() Function

The **Scapy ls()** function is a built-in helper used to **list fields, options, and layers** available in Scapy. It is extremely useful for exploring protocols, packet layers, or specific packet objects.



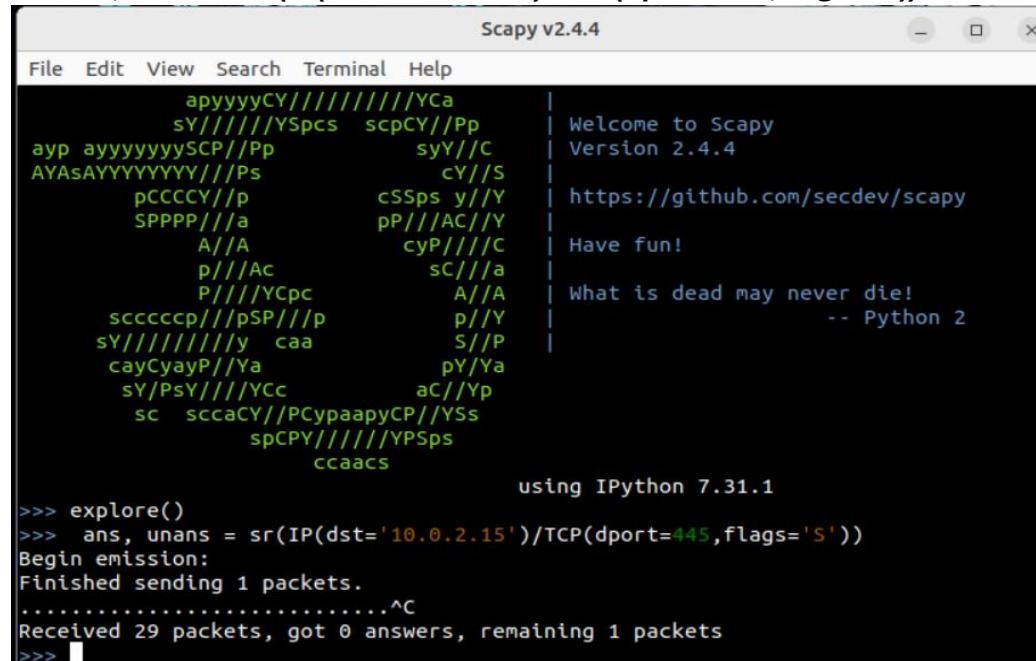
```
Scapy v2.4.4
File Edit View Search Terminal Help
ZCLMeteringGetProfile : Metering Cluster: Get Profile Command (Server: Received)
ZCLPriceGetCurrentPrice : Price Cluster: Get Current Price Command (Server: Received)
ZCLPriceGetScheduledPrices : Price Cluster: Get Scheduled Prices Command (Server: Received)
ZCLPricePublishPrice : Price Cluster: Publish Price Command (Server: Generated)
ZCLReadAttributeStatusRecord : ZCL Read Attribute Status Record
ZEP1      : Zigbee Encapsulation Protocol (V1)
ZEP2      : Zigbee Encapsulation Protocol (V2)
ZigBeeBeacon : ZigBee Beacon Payload
ZigbeeAppCommandPayload : Zigbee Application Layer Command Payload
ZigbeeAppDataPayload : Zigbee Application Layer Data Payload (General APS Frame Format)
ZigbeeAppDataPayloadStub : Zigbee Application Layer Data Payload for Inter-PAN Transmission
ZigbeeClusterLibrary : Zigbee Cluster Library (ZCL) Frame
ZigbeeDeviceProfile : Zigbee Device Profile (ZDP) Frame
ZigbeeNWK : Zigbee Network Layer
ZigbeeNWKCommandPayload : Zigbee Network Layer Command Payload
ZigbeeNWKStub : Zigbee Network Layer for Inter-PAN Transmission
ZigbeeSecurityHeader : Zigbee Security Header

TIP: You may use explore() to navigate through all layers using a clear GUI
>>>
```

2.3 Scapy Packet Sending

Sending a TCP SYN packet using Scapy means crafting a custom TCP packet with the SYN flag set (the first step in the TCP 3-way handshake) and transmitting it to a target host. This is commonly used for testing, research, and learning how TCP works.

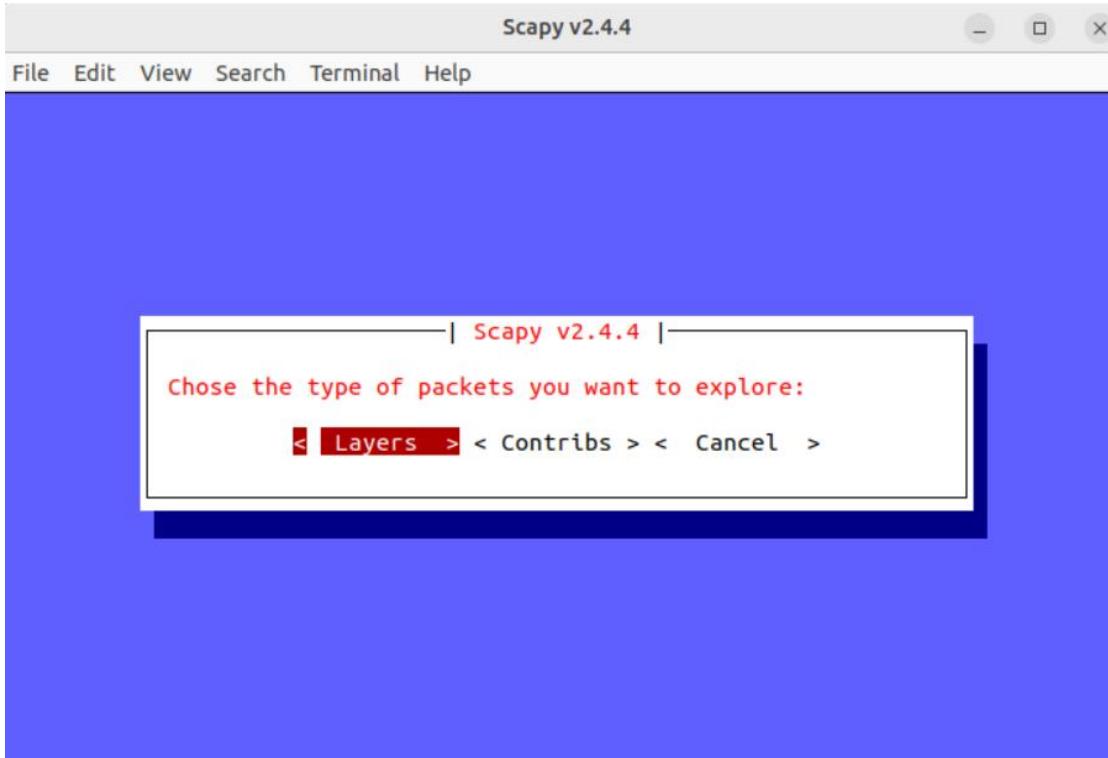
```
>>>ans, unans = sr(IP(dst='10.0.2.15')/TCP(dport=445,flags='S'))
```



```
Scapy v2.4.4
File Edit View Search Terminal Help
apyyyyCY//////////YCa
sY//////YSpes    scpCY//Pp
ayp ayyyyyyySCP//Pp          syY//C
AYAsYYYYYYYYYY//Ps          CY//S
pCCCCY//p          cSSps y//Y
SPPPP///a          pP///AC//Y
A//A              cyP///C
p///Ac            sC///a
P///YCpc          A//A
scccccp///pSP///p          p//Y
sY/////////y caa          S//P
cayCayP//Ya          pY/Ya
sY/PsY///YCc          aC//Yp
sc  sccaCY//PCyapaCP//YSS
spCPY//////YPSps
ccaaacs
using IPython 7.31.1
>>> explore()
>>> ans, unans = sr(IP(dst='10.0.2.15')/TCP(dport=445,flags='S'))
Begin emission:
Finished sending 1 packets.
.....^C
Received 29 packets, got 0 answers, remaining 1 packets
>>>
```

2.4 Scapy Explore Function

The `explore()` function to navigate the Scapy layers and protocols.



A screenshot of the Scapy v2.4.4 application window. The title bar says "Scapy v2.4.4". The menu bar includes File, Edit, View, Search, Terminal, and Help. A central dialog box titled "Scapy v2.4.4" contains the text "Please select a fileamong the following, to see all layers contained in it:" followed by a list of layer options. At the bottom is a navigation bar with buttons for < Ok > < Cancel >.

```
Please select a fileamong the following, to see all layers contained in it:  
> scapy.layers  
  
(*) Packet class  
( ) ASN.1 Packet  
( ) Bluetooth layers, sockets and send/receive functions.  
( ) CACE Per-Packet Information (PPI) header.  
( ) Bluetooth 4LE layer  
( ) Classes and functions for layer 2 protocols.  
( ) IPv4 (Internet Protocol v4).  
( ) IPv6 (Internet Protocol v6).  
( ) DHCP (Dynamic Host Configuration Protocol) and BOOTP  
( ) DNS: Domain Name System.  
( ) DHCPv6: Dynamic Host Configuration Protocol for IPv6. [RFC 3315,8415]  
( ) Wireless LAN according to IEEE 802.11.  
( ) Wireless MAC according to IEEE 802.15.4.  
( ) Extensible Authentication Protocol (EAP)  
( ) GPRS (General Packet Radio Service) for mobile data communication.
```

2.6 Scapy Basic Protocol Analysis

Scapy Basic Protocol Analysis refers to using Scapy to inspect, capture, and analyze network packets across various protocols (IP, TCP, UDP, ICMP, ARP, etc.). Scapy makes it easy to look “under the hood” of network traffic by letting you interact with packets at all layers of the OSI model.

```
>>> pkts = sniff(count=5)
```

```
>>> pkts.summary()
```

The screenshot shows the Scapy v2.4.4 interface running in a terminal window. The window title is "Scapy v2.4.4". The menu bar includes File, Edit, View, Search, Terminal, and Help. The main area displays captured network traffic and a welcome message from Scapy. The welcome message includes:

- Welcome to Scapy
- Version 2.4.4
- <https://github.com/secdev/scapy>
- Have fun!
- We are in France, we say Skappee.
- OK? Merci.
- Sebastien Chabal

The captured traffic is listed as follows:

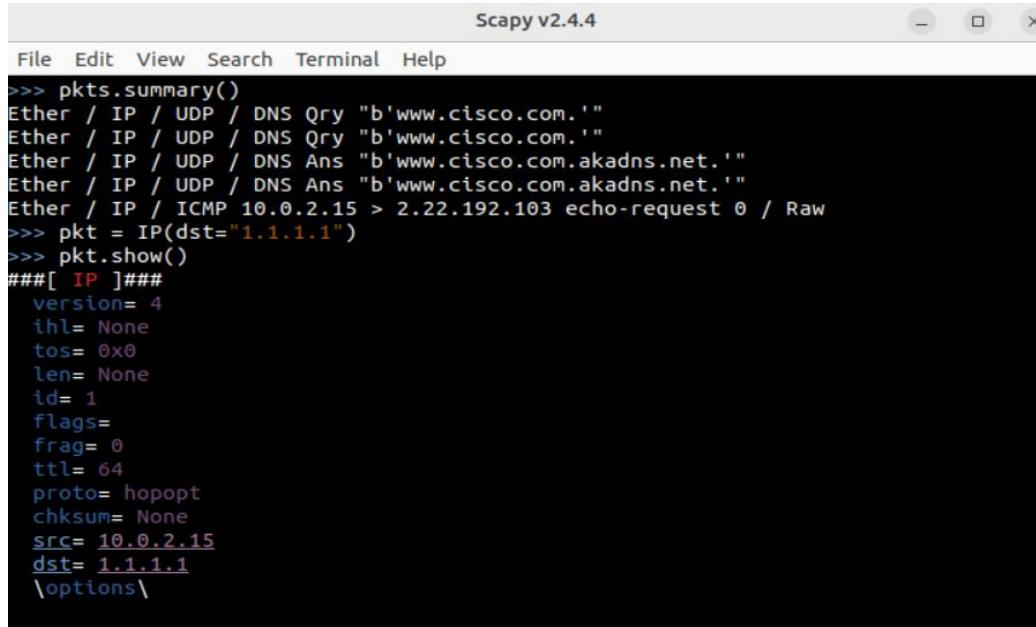
```
sY/////////YSpcs  scpCY//Pp | Welcome to Scapy
ayp ayyyyyyySCP//Pp      syY//C
AYAsAYYYYYYYYY///Ps      cY//S
    pCCCCY//p          cSSps y//Y
    SPPPP//a          pP//AC//Y
        A//A          cyP///C
        p///Ac          sc///a
        P///YCpc         A//A
    scccccp///pSP///p      p//Y
sY/////////y  caa          S//P
    cayCyayP//Ya        pY/Ya
    sY/PsY///YCcc       aC//Yp
    sc  sccaCY//PCypaapyCP//YSs
                    spCPY//////YPSPs
                    ccaacs
                                         using IPython 7.31.1
>>> pkts = sniff(count=5)
>>> pkts.summary()
Ether / IP / UDP / DNS Qry "b'www.cisco.com.'"
Ether / IP / UDP / DNS Qry "b'www.cisco.com.'"
Ether / IP / UDP / DNS Ans "b'www.cisco.com.akadns.net.'"
Ether / IP / UDP / DNS Ans "b'www.cisco.com.akadns.net.'"
Ether / IP / ICMP 10.0.2.15 > 2.22.192.103 echo-request 0 / Raw
>>> █
```

2.7 Scapy packet crafting

Scapy packet crafting means using Scapy to **build your own custom network packets** by stacking protocol layers (Ethernet, IP, TCP, UDP, ICMP, ARP, DNS, etc.) and filling in their fields manually. It allows deep learning, testing, debugging, and research into how network protocols behave.

```
>>> pkt = IP(dst="1.1.1.1")
```

```
>>> pkt.show()
```



The screenshot shows a terminal window titled "Scapy v2.4.4". The menu bar includes File, Edit, View, Search, Terminal, and Help. The terminal window displays Python code for crafting an IP packet:

```
>>> pkts.summary()
Ether / IP / UDP / DNS Qry "b'www.cisco.com.'"
Ether / IP / UDP / DNS Qry "b'www.cisco.com.'"
Ether / IP / UDP / DNS Ans "b'www.cisco.com.akadns.net.'"
Ether / IP / UDP / DNS Ans "b'www.cisco.com.akadns.net.'"
Ether / IP / ICMP 10.0.2.15 > 2.22.192.103 echo-request 0 / Raw
>>> pkt = IP(dst="1.1.1.1")
>>> pkt.show()
###[ IP ]###
version= 4
ihl= None
tos= 0x0
len= None
id= 1
flags=
frag= 0
ttl= 64
proto= hopopt
chksum= None
src= 10.0.2.15
dst= 1.1.1.1
\options\
```

Submitted By : Kamohelo Motaung(kmstitches777@gmail.com)

Lecturer : Ronald Mawuli Awuku

Institute/s : ParoCyber & Netacad

Submission Date : 12/12/2025