INETMS
INET Managed Services Co., Ltd.

Basic Computer Networking for Cyber Security

ชื่อ: กฤช อินทรวิชา (ปาย)

ฝ่าย: Managed Security Services

ตำแหน่ง: Senior Security Engineer

CompTIA CSAP Security Analytics Professional

ITIL® 4 FOUNDATION

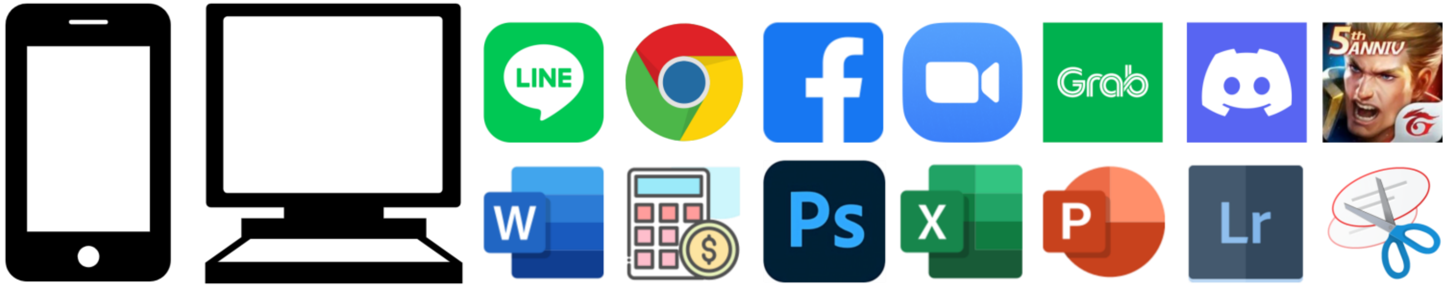
McAfee™ Product Specialist CERTIFIED

Software

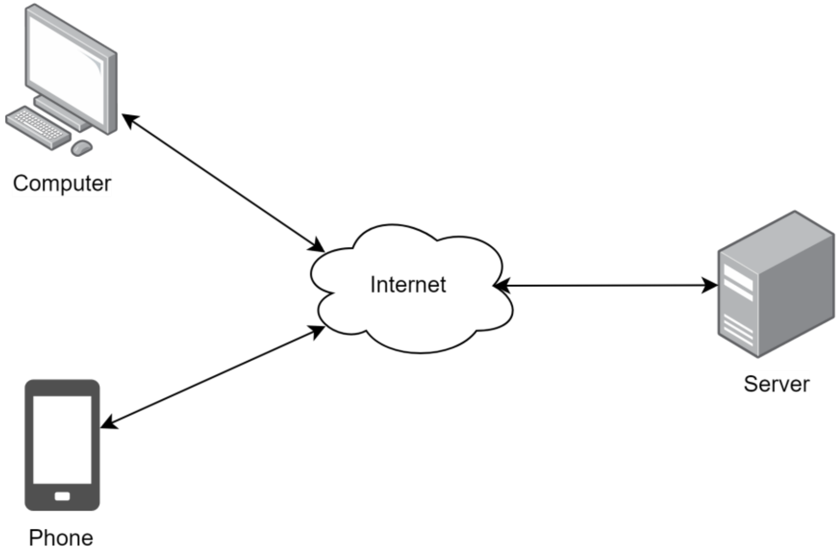
Software is anything that is created to run on digital devices.

Digital devices such as Computer, Laptop, Server, Smart Phone, Smart Watch, Smart Television.

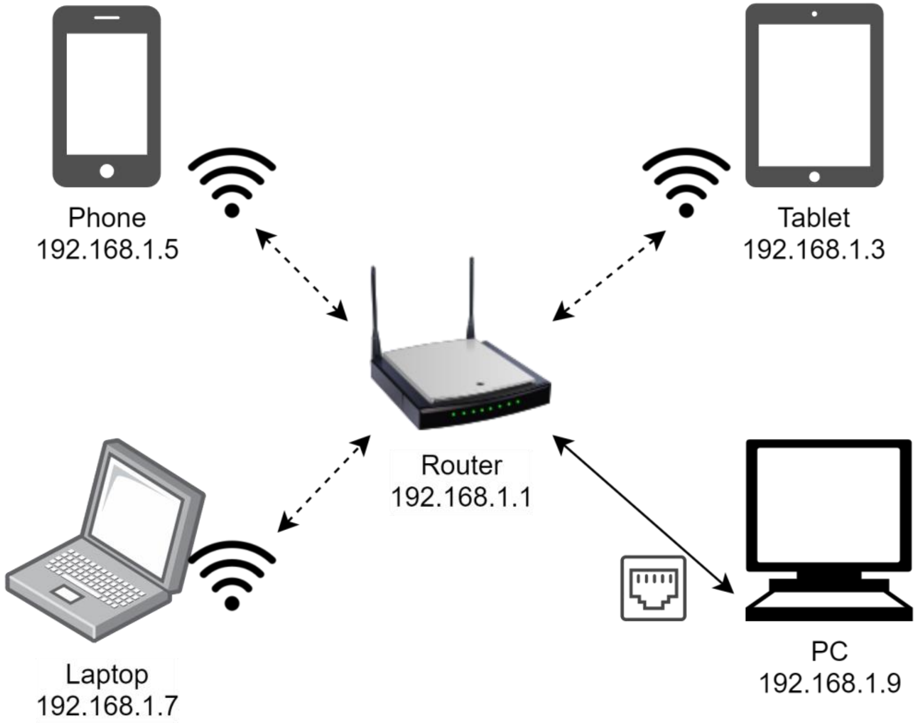
Software that require network connection

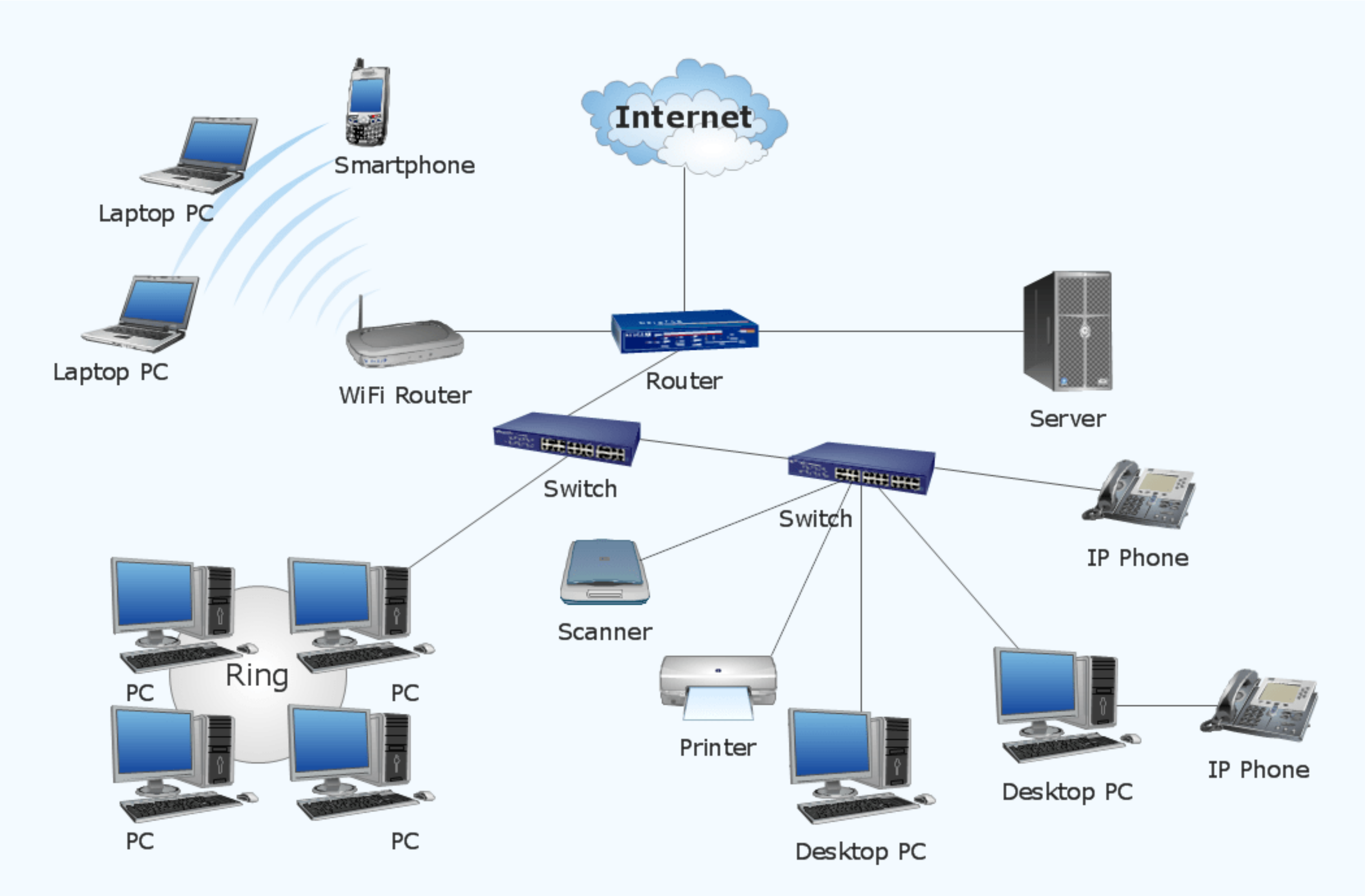


Software that not require network connection



Device Communication (IP: Internet Protocol)





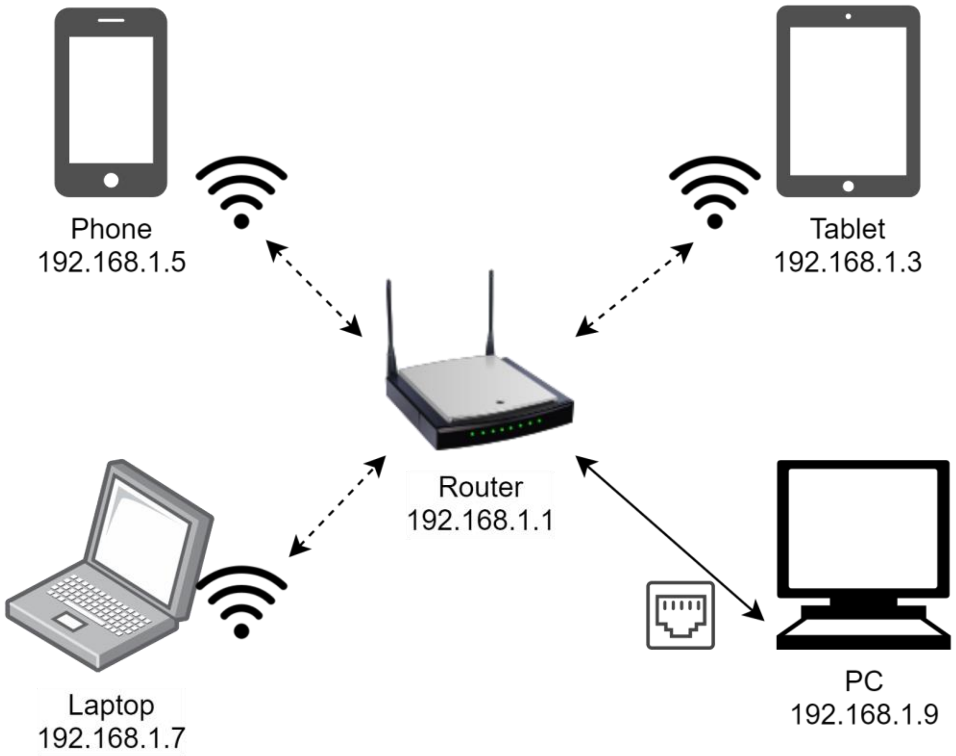
IP Address check command

Windows: ipconfig /all

Linux: ip addr

Device Communication (IP: Internet Protocol)

Tablet send data to Laptop : “Hello Laptop!” and Laptop send data back to Tablet: “Hello Tablet!”



IP Address version 4 (IPv4): 000.000.000.000

Range: 0.0.0.0 – 255.255.255.255

Example: 65.18.3.154, 192.168.1.1, 127.0.0.1, 10.45.2.78

IP Address version 6 (IPv6): 0000:0000:0000:0000:0000:0000:0000:0000

Range: 0000:0000:0000:0000:0000:0000:0000:0000 – ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Example: 2001:0db8:0000:0000:34f4:0000:0000:f3dd

Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f

	Data Package		
	Source IP	Destination IP	Data
Tablet send data to Laptop	192.168.1.3	192.168.1.7	Hello Laptop!
Laptop send data back to Tablet	192.168.1.7	192.168.1.3	Hello Tablet!

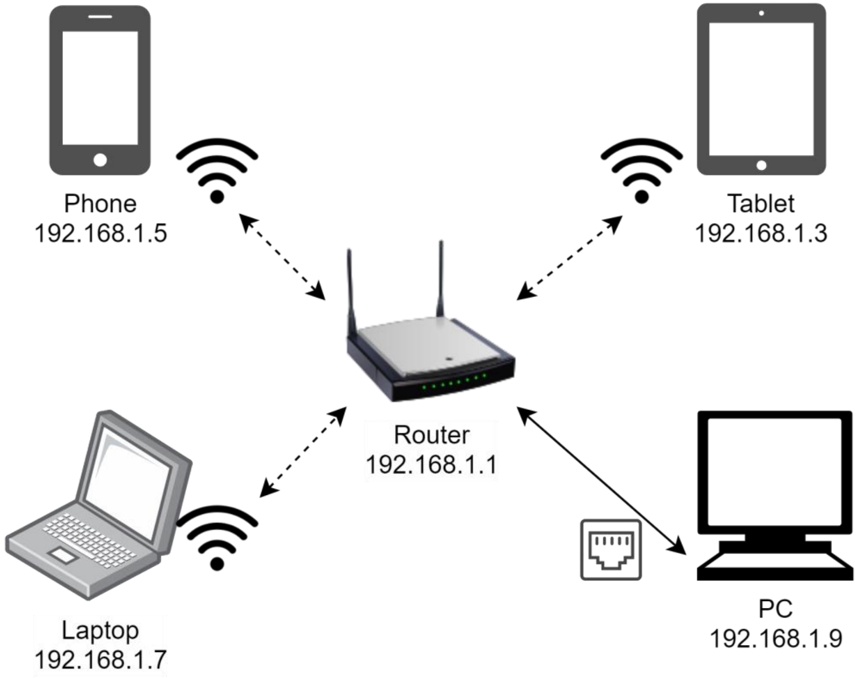
Transport Protocol: TCP/UDP

	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Detail	การขนส่งข้อมูลโดยที่ผู้รับและผู้ส่งมีการติดต่อสื่อสารกันอย่างสม่ำเสมอตลอดการขนส่งข้อมูลเพื่อคอยตรวจสอบว่าข้อมูลที่ขนส่งนั้นถึงปลายทางครบถูกต้องทุกข้อมูลและไม่มีข้อมูลส่วนไหนเสียหายหรือถูกเปลี่ยนแปลงระหว่างทาง เหมาะสำหรับการขนส่งข้อมูลที่มีขนาดใหญ่, ข้อมูลที่ต้องการความถูกต้องและแม่นยำสูง	การขนส่งข้อมูลที่ผู้ส่งไม่ได้ทำการติดต่อสื่อสารกับผู้รับ โดยผู้ส่งสามารถเริ่มต้นการส่งข้อมูลไปหาผู้รับได้ทันที โดยไม่ได้ทำการตรวจสอบสถานะผู้รับ
Pros	รับประกันการส่งข้อมูลที่ถูกต้องและครบถ้วนจากต้นทางถึงปลายทาง	มีความสะดวกและรวดเร็วในการขนส่งข้อมูล
Cons	ใช้เวลาในการขนส่งข้อมูลมากกว่าการส่งข้อมูลแบบ UDP	ไม่รับประกันการขนส่งข้อมูลว่าจะถึงปลายทางและข้อมูลถูกต้องครบถ้วน
Example Application Protocol	HTTP (Web Application Protocol, Mobile Application Protocol)	DNS (Resolve Domain Protocol)

Transport Protocol: TCP/UDP

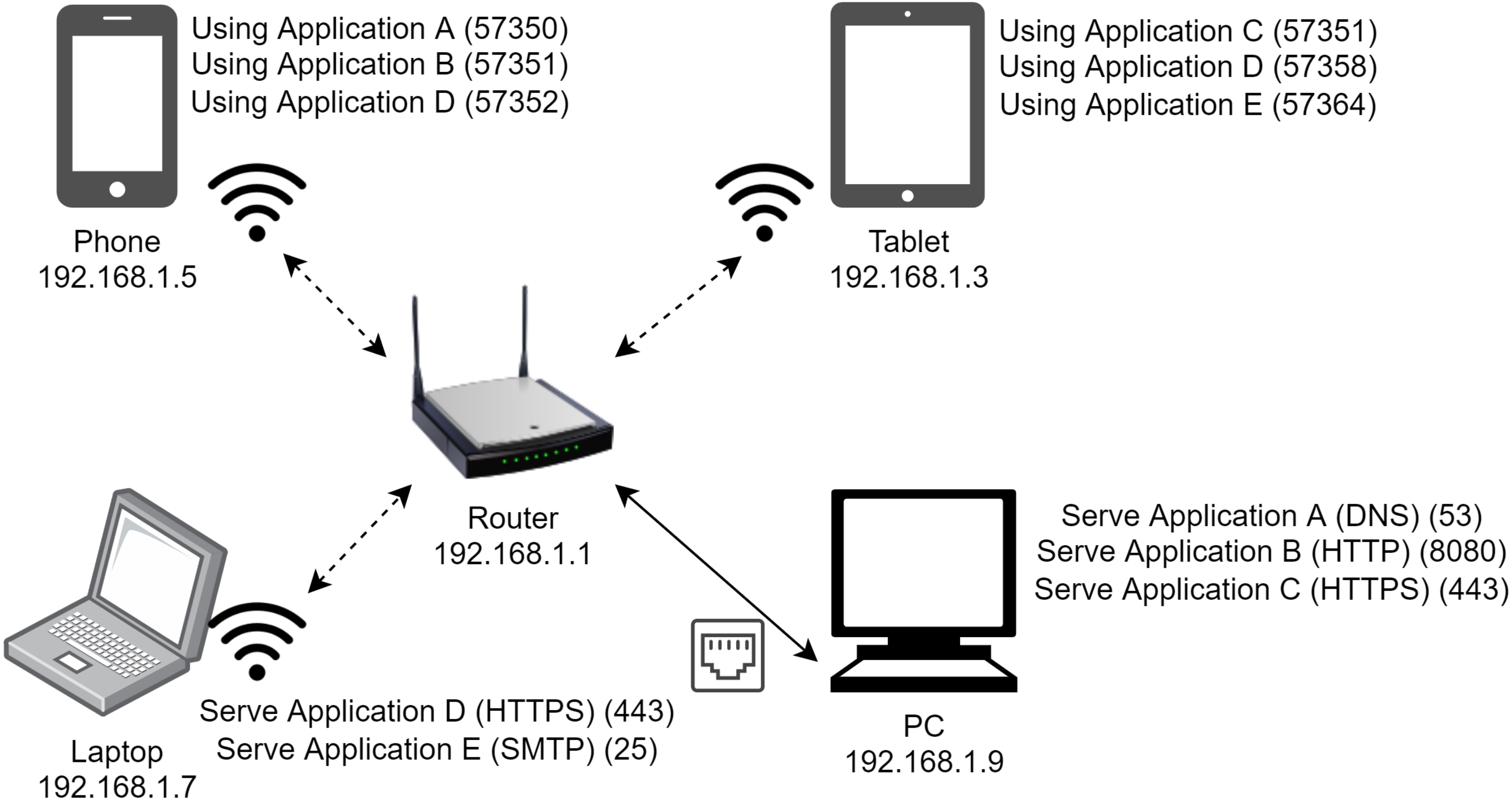
Phone Access Website on PC with data: “GET /index.html HTTP/1.1”

and PC send data back to phone: “<!DOCTYPE html><html>Hello Web Page</html>”



	Data Package			
	Source IP	Destination IP	Transport Protocol	Data
Phone send data to PC			TCP	GET /index.html HTTP/1.1
PC send data back to phone			TCP	<!DOCTYPE html><html>Hello Web Page</html>

Port



Phone
192.168.1.5

Using Application A (57350)
Using Application B (57351)
Using Application D (57352)

Tablet
192.168.1.3

Using Application C (57351)
Using Application D (57358)
Using Application E (57364)

Router
192.168.1.1

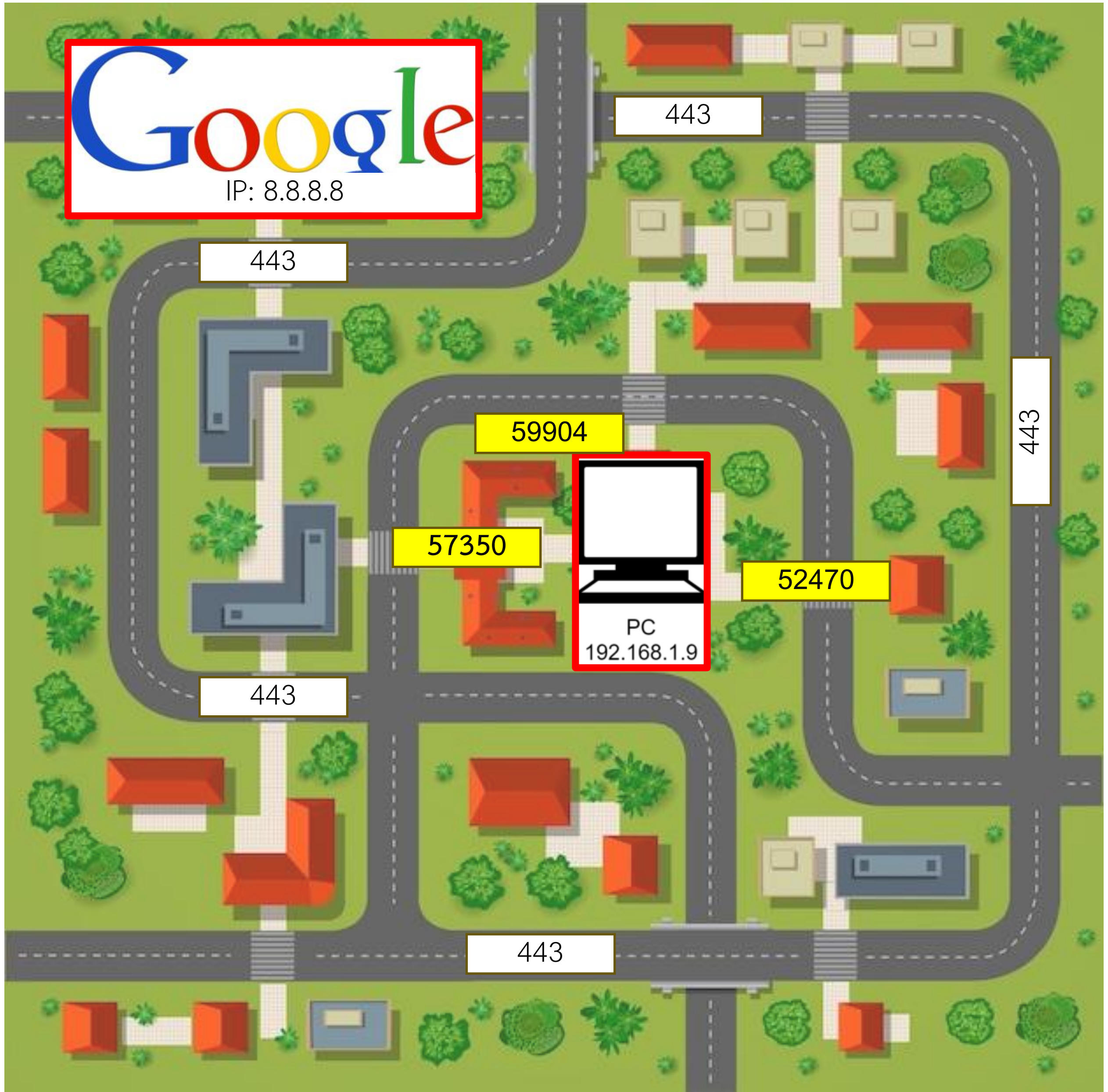
Laptop
192.168.1.7

Serve Application D (HTTPS) (443)
Serve Application E (SMTP) (25)

PC
192.168.1.9

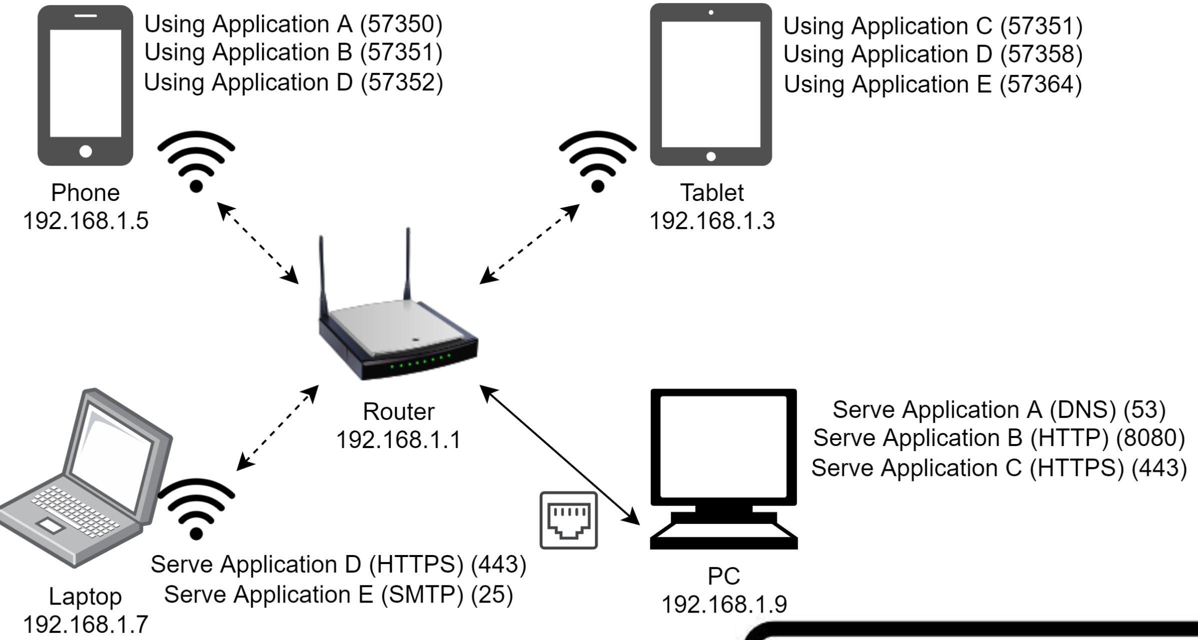
Serve Application A (DNS) (53)
Serve Application B (HTTP) (8080)
Serve Application C (HTTPS) (443)

Port



Src IP	Dest IP	Trans Pro	Src Port	Dest Port	App Pro
		TCP			

Port



Using Application A (57350)
Using Application B (57351)
Using Application D (57352)

Phone
192.168.1.5

Using Application C (57351)
Using Application D (57358)
Using Application E (57364)

Tablet
192.168.1.3

Router
192.168.1.1

Serve Application A (DNS) (53)
Serve Application B (HTTP) (8080)
Serve Application C (HTTPS) (443)

PC
192.168.1.9

Serve Application D (HTTPS) (443)
Serve Application E (SMTP) (25)

Laptop
192.168.1.7

Src IP	Src IP	Dest IP	Trans Pro	Src Port	Dest Port	App Pro
Ph(A) > PC(A)	192.168.1.5	192.168.1.9	UDP	57350	53	DNS
PC(A) > Ph(A)	192.168.1.9	192.168.1.5	UDP	53	57350	DNS
Ph(B) > PC(B)	192.168.1.5	192.168.1.9	TCP	57351	8080	HTTP
PC(B) > Ph(B)	192.168.1.9	192.168.1.5	TCP	8080	57351	HTTP
Ph(D) > Lap(D)	192.168.1.5	192.168.1.7	TCP	57352	443	HTTPS
Lap(D) > Ph(D)	192.168.1.7	192.168.1.5	TCP	443	57352	HTTPS
	1					
	2					
	3					
	4					
	5					
	6					

COMMON WELL-KNOWN PORTS

Service	Port	Function
HTTP	tcp/80	Web
HTTPS	tcp/443	Web (secure)
FTP	tcp/20,21	File transfer
SNMP	udp/161,162	System monitoring
DNS	udp/53	Find IP address
SMTP	tcp/25	Internet mail
SSH	tcp/22	Remote login (secure)
RDP	3389	Remote Desktop

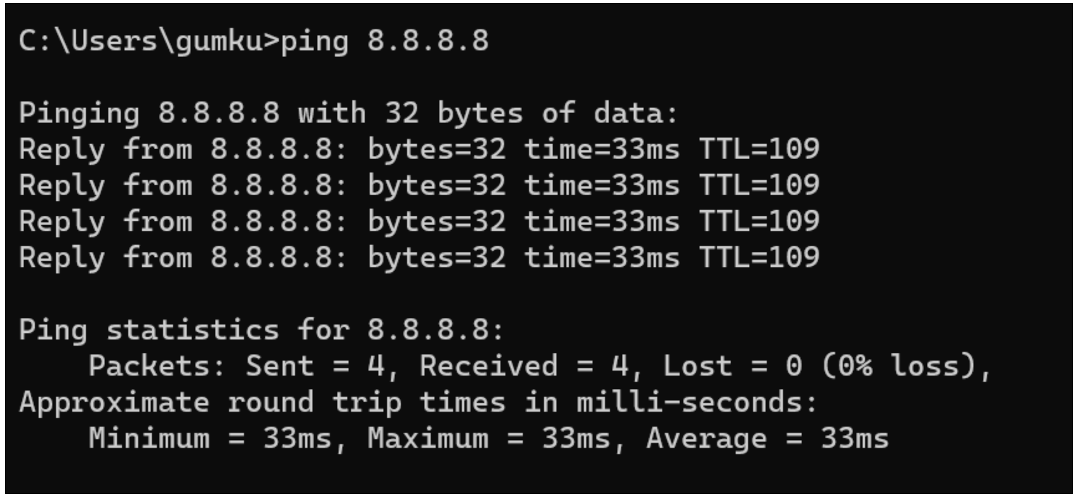
NETWORK PORTS

Well-known Ports	0 - 1023
Registered Ports	1024 - 49151
Dynamic Ports	49152 - 65565

Testing Network Connection with Ping (ICMP Protocol)

- We can use the Ping command to verify the connection between two machines.
- Available on Windows, Linux, MacOS
- Ping <DestinationIPAddress>

Connection Success



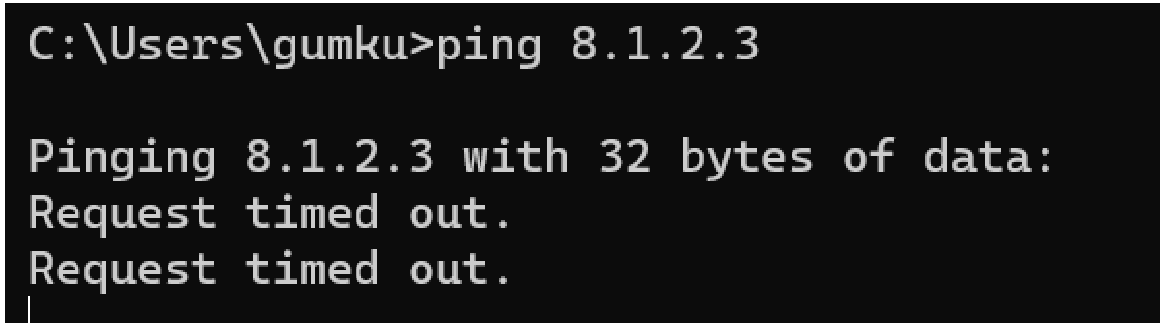
```
C:\Users\gumku>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=33ms TTL=109
Reply from 8.8.8.8: bytes=32 time=33ms TTL=109
Reply from 8.8.8.8: bytes=32 time=33ms TTL=109
Reply from 8.8.8.8: bytes=32 time=33ms TTL=109

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 33ms, Maximum = 33ms, Average = 33ms
```

Connection fail or

connection success but destination host disable ICMP reply



```
C:\Users\gumku>ping 8.1.2.3

Pinging 8.1.2.3 with 32 bytes of data:
Request timed out.
Request timed out.
```

Testing Network Connection with open TCP port

Windows (PowerShell only)

tnc <Server> -port <PortNumber>

Test-NetConnection <Server> -port <PortNumber>

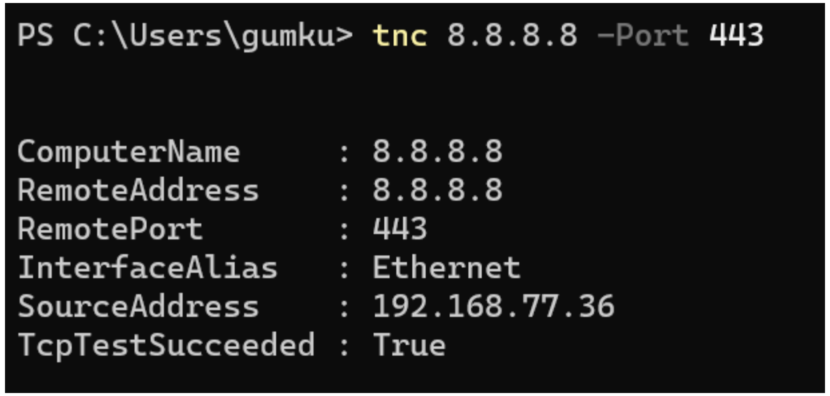
Linux

telnet <Server> <port>

nc -zv <Server> <port>

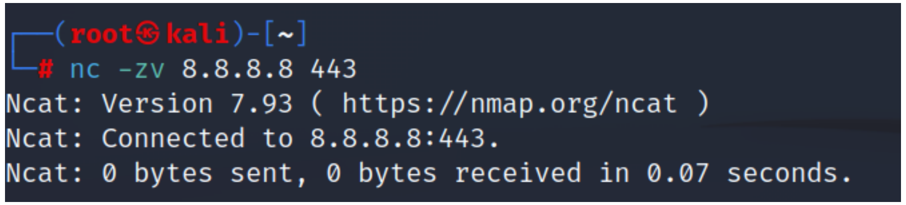
Connection

Success

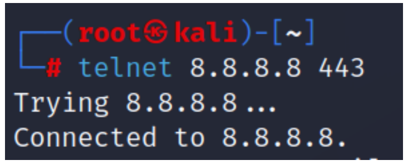


```
PS C:\Users\gumku> tnc 8.8.8.8 -Port 443


ComputerName     : 8.8.8.8
RemoteAddress    : 8.8.8.8
RemotePort       : 443
InterfaceAlias   : Ethernet
SourceAddress    : 192.168.77.36
TcpTestSucceeded : True
```



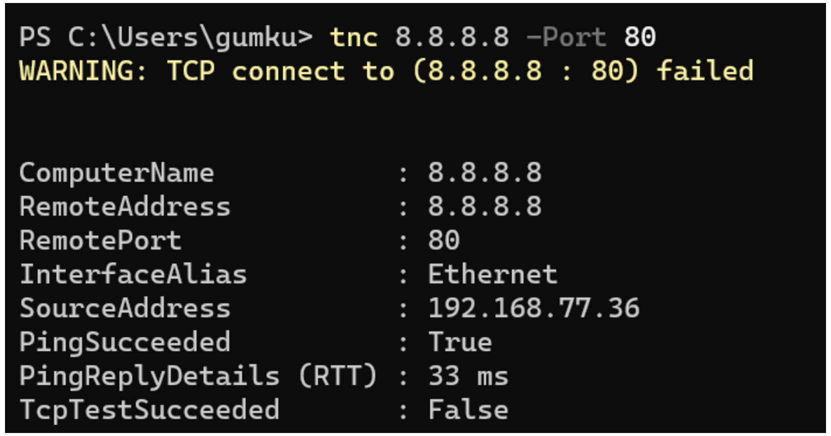
```
┌──(root㉿kali)-[~]
└─# nc -zv 8.8.8.8 443
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Connected to 8.8.8.8:443.
Ncat: 0 bytes sent, 0 bytes received in 0.07 seconds.
```



```
┌──(root㉿kali)-[~]
└─# telnet 8.8.8.8 443
Trying 8.8.8.8...
Connected to 8.8.8.8.
```

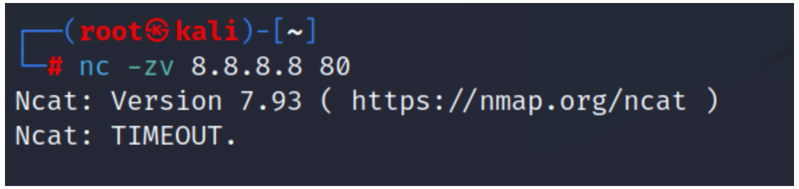
Connection

Fail

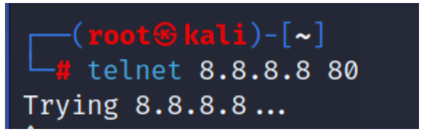


```
PS C:\Users\gumku> tnc 8.8.8.8 -Port 80
WARNING: TCP connect to (8.8.8.8 : 80) failed


ComputerName           : 8.8.8.8
RemoteAddress          : 8.8.8.8
RemotePort             : 80
InterfaceAlias         : Ethernet
SourceAddress          : 192.168.77.36
PingSucceeded          : True
PingReplyDetails (RTT) : 33 ms
TcpTestSucceeded       : False
```



```
┌──(root㉿kali)-[~]
└─# nc -zv 8.8.8.8 80
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: TIMEOUT.
```

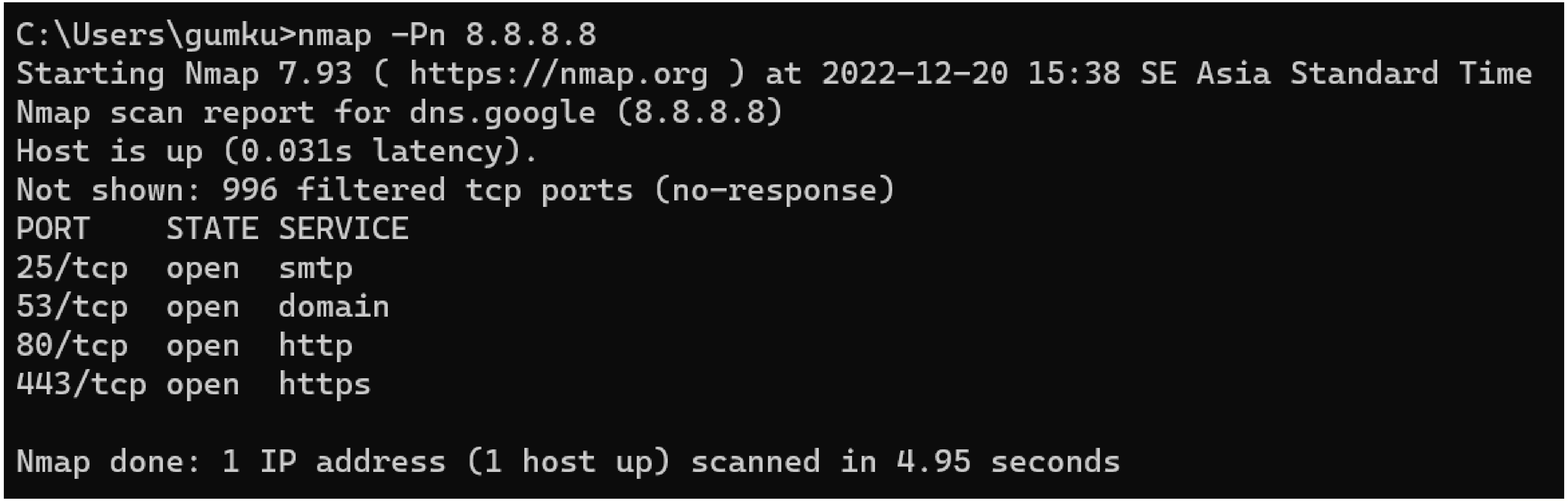


```
┌──(root㉿kali)-[~]
└─# telnet 8.8.8.8 80
Trying 8.8.8.8...
```

https://nmap.org/download.html

Network discovery with Nmap

- We can use the Nmap to check the connection between two machines. Also, we can use Nmap to discover listening Ports and services on the remote machines.
- Need to download and install from Nmap page.
- Nmap –Pn –p- --min-rate 10000 –vvv <DestinationIPAddress>

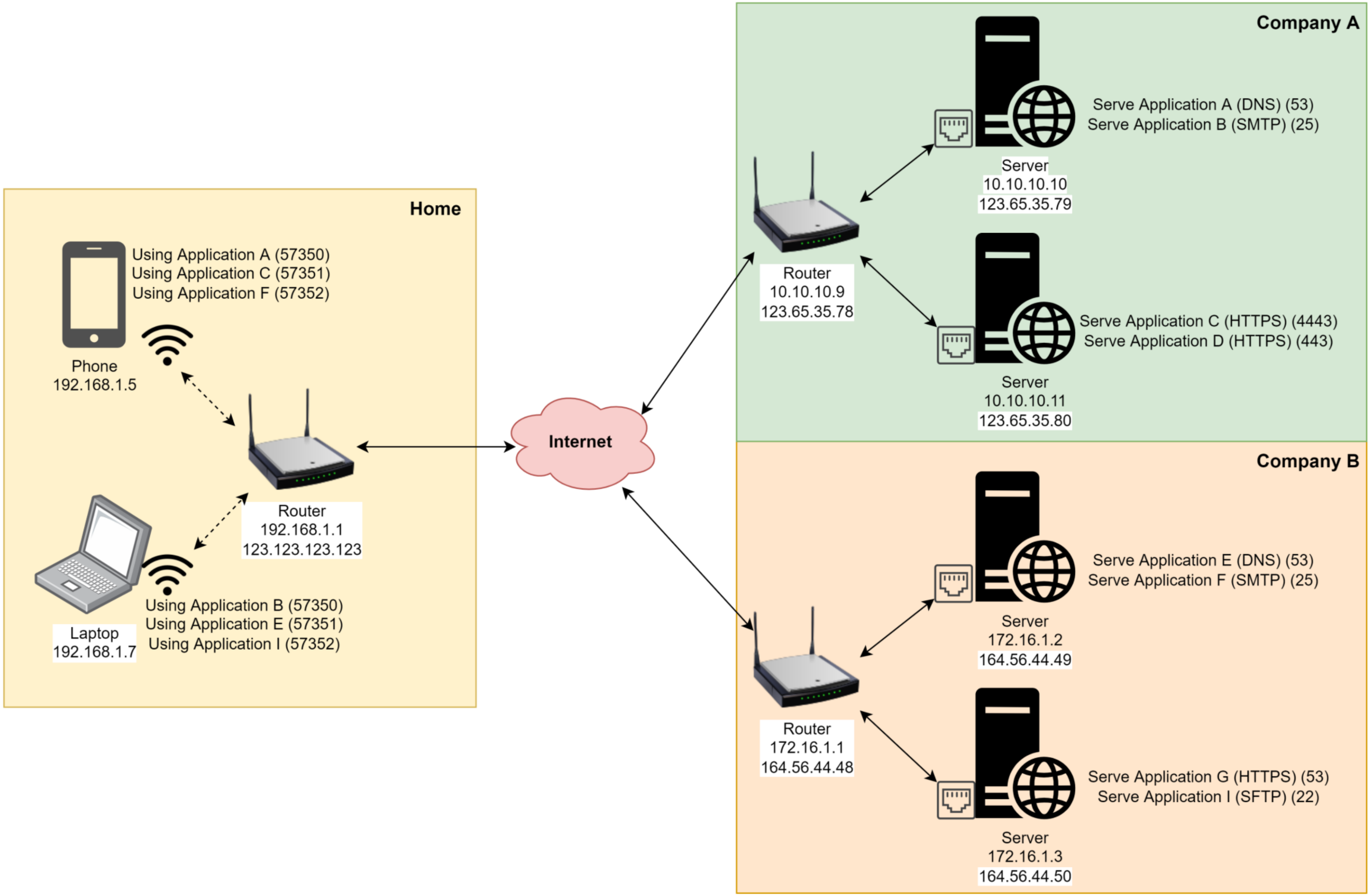


```
C:\Users\gumku>nmap -Pn 8.8.8.8
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-20 15:38 SE Asia Standard Time
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.031s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE
25/tcp  open  smtp
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.95 seconds
```

https://nmap.org/download.html

Public Network (WAN: Wide Area Network: Internet)
Private Network (LAN: Local Area Network)

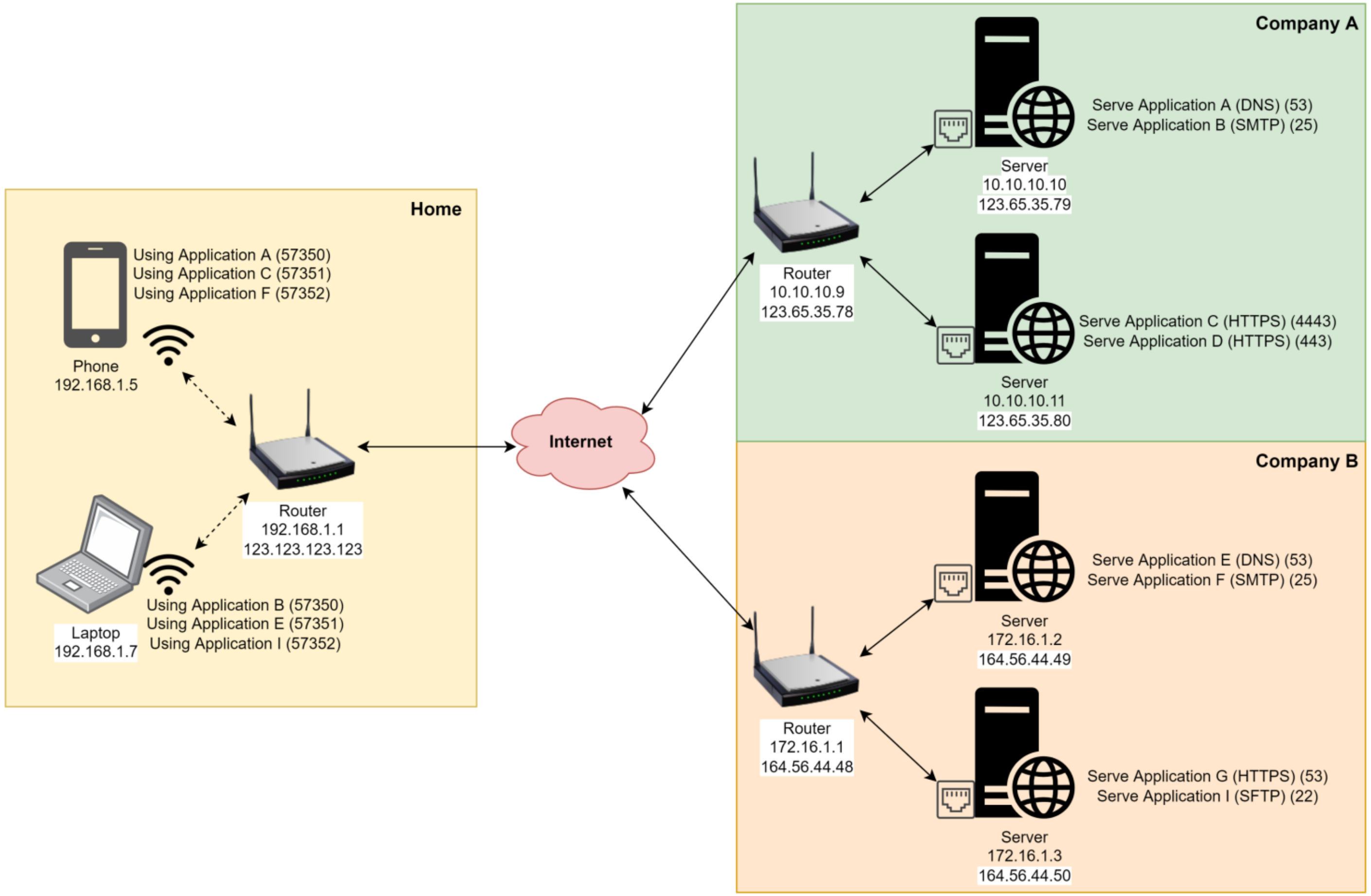

Home
Using Application A (57350)
Using Application C (57351)
Using Application F (57352)
Phone
192.168.1.5
Router
192.168.1.1
123.123.123.123
Laptop
192.168.1.7
Using Application B (57350)
Using Application E (57351)
Using Application I (57352)
Internet
Company A
Router
10.10.10.9
123.65.35.78
Serve Application A (DNS) (53)
Serve Application B (SMTP) (25)
Server
10.10.10.10
123.65.35.79
Serve Application C (HTTPS) (4443)
Serve Application D (HTTPS) (443)
Server
10.10.10.11
123.65.35.80
Company B
Router
172.16.1.1
164.56.44.48
Serve Application E (DNS) (53)
Serve Application F (SMTP) (25)
Server
172.16.1.2
164.56.44.49
Serve Application G (HTTPS) (53)
Serve Application I (SFTP) (22)
Server
172.16.1.3
164.56.44.50

Class	Private Address Ranges
Class A	10.0.0.0 – 10.255.255.255
Class B	172.16.0.0 – 172.31.255.255
Class C	192.168.0.0 – 192.168.255.255
Loopback	127.0.0.0 – 127.255.255.255 (127.0.0.1)

NAT: Network Address Translation

- SNAT: Source Network Address Translation
  (Private -> Public)
- DNAT: Destination Network Address Translation
  (Public -> Private)

Internet



Home

Using Application A (57350)
Using Application C (57351)
Using Application F (57352)

Phone
192.168.1.5

Router
192.168.1.1
123.123.123.123

Laptop
192.168.1.7

Using Application B (57350)
Using Application E (57351)
Using Application I (57352)

Internet

Company A

Router
10.10.10.9
123.65.35.78

Server
10.10.10.10
123.65.35.79

Serve Application A (DNS) (53)
Serve Application B (SMTP) (25)

Server
10.10.10.11
123.65.35.80

Serve Application C (HTTPS) (4443)
Serve Application D (HTTPS) (443)

Company B

Router
172.16.1.1
164.56.44.48

Server
172.16.1.2
164.56.44.49

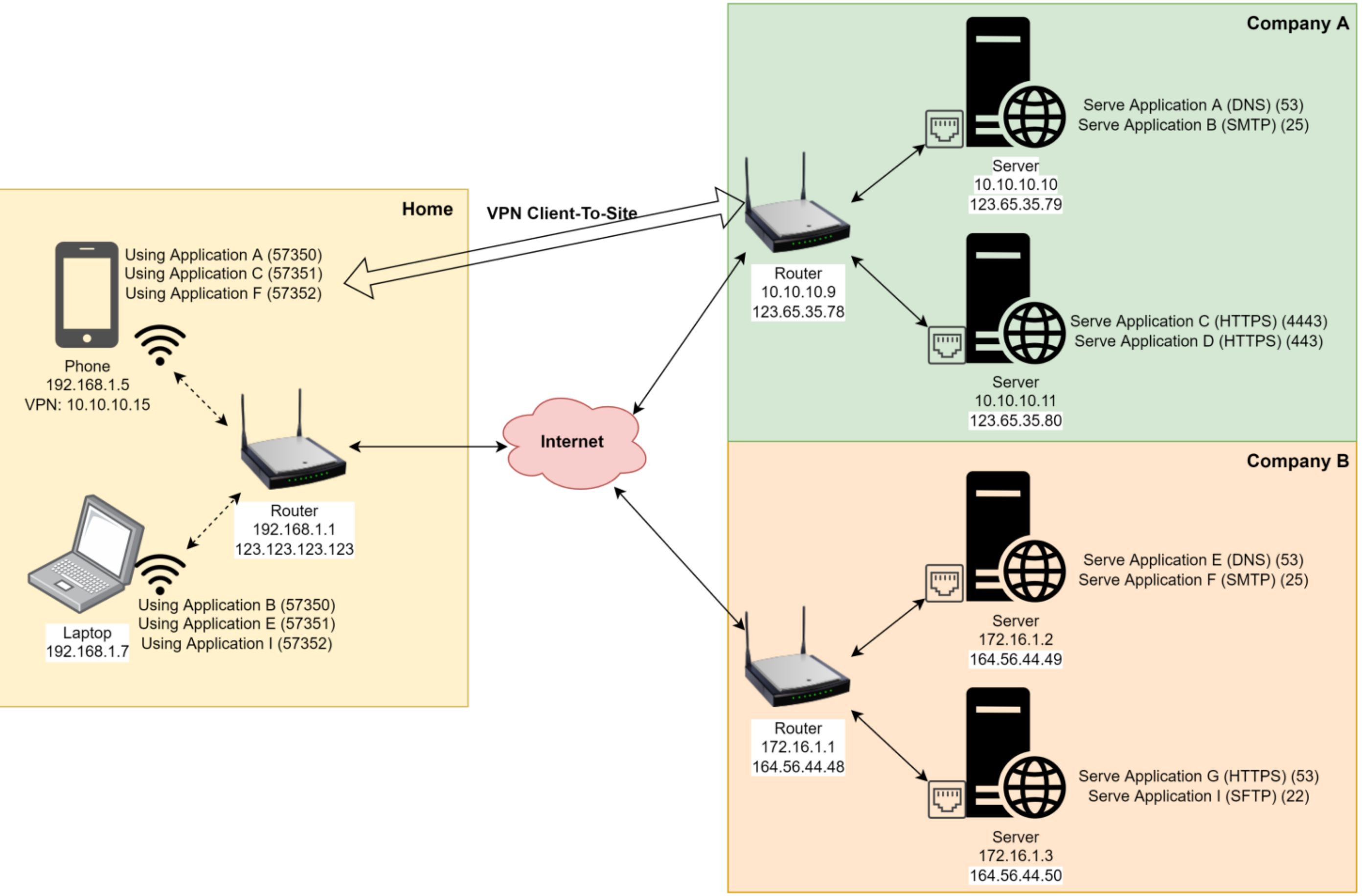
Serve Application E (DNS) (53)
Serve Application F (SMTP) (25)

Server
172.16.1.3
164.56.44.50

Serve Application G (HTTPS) (53)
Serve Application I (SFTP) (22)

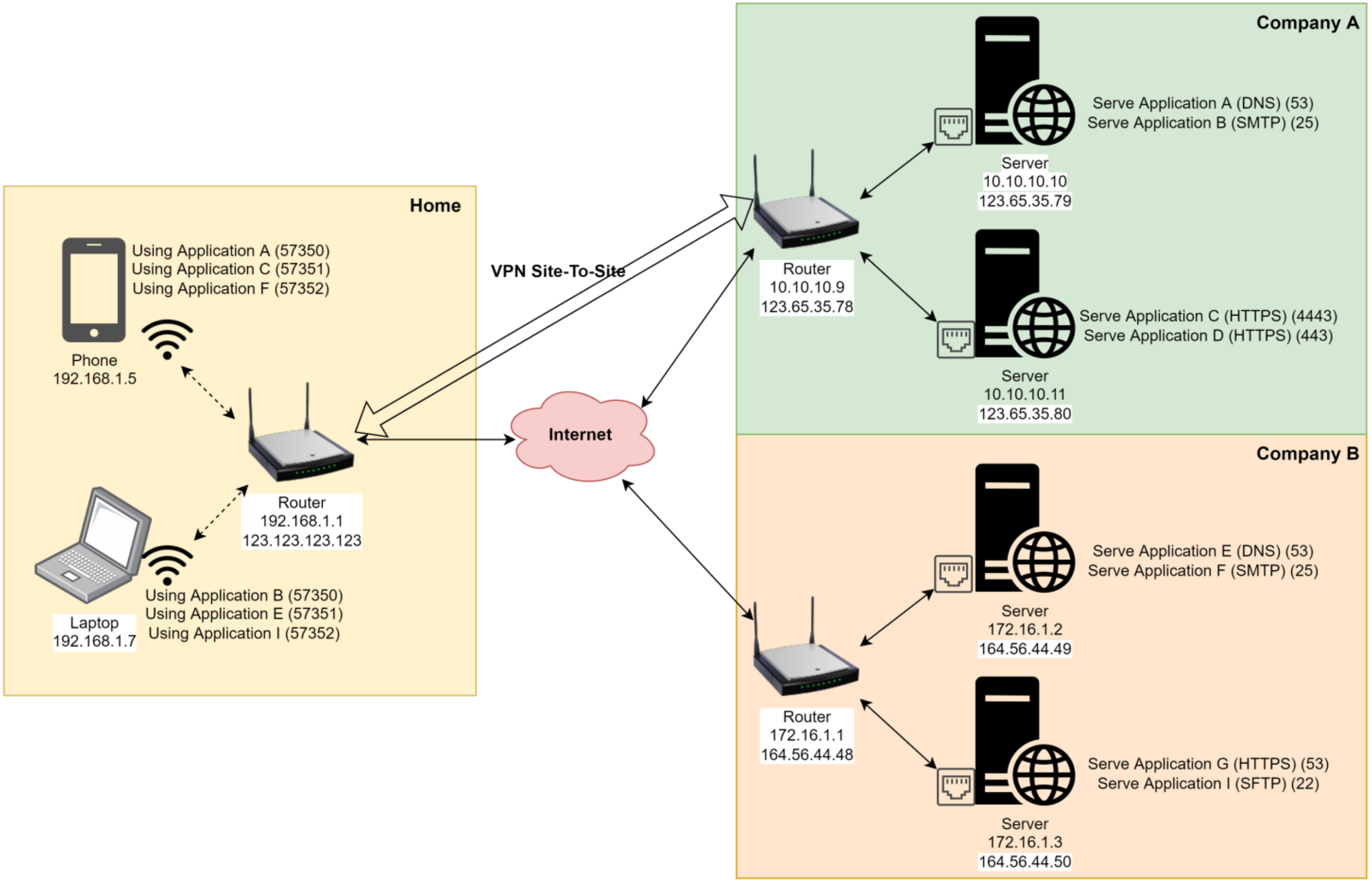
Src IP	Dest IP	Trns	Src Port	Dest Port	Data	App Pro
123.123.123.123	123.65.35.79	UDP	57350	53	XXX	DNS
123.65.35.79	123.123.123.123	UDP	53	57350	XXX	DNS
123.123.123.123	123.65.35.80	TCP	57351	4443	XXX	HTTPS
123.65.35.80	123.123.123.123	TCP	4443	57351	XXX	HTTPS
123.123.123.123	164.56.44.49	TCP	57352	25	XXX	SMTP
164.56.44.49	123.123.123.123	TCP	25	57352	XXX	SMTP
123.123.123.123	123.65.35.79	TCP	57350	25	XXX	SMTP
123.65.35.79	123.123.123.123	TCP	25	57350	XXX	SMTP
123.123.123.123	164.56.44.49	UDP	57351	53	XXX	DNS
164.56.44.49	123.123.123.123	UDP	53	57351	XXX	DNS
123.123.123.123	164.56.44.50	TCP	57352	22	XXX	SFTP
164.56.44.50	123.123.123.123	TCP	22	57352	XXX	SFTP

VPN: Virtual Private Network : Client-To-Site


Company A
Server
10.10.10.10
123.65.35.79
Serve Application A (DNS) (53)
Serve Application B (SMTP) (25)
Router
10.10.10.9
123.65.35.78
Server
10.10.10.11
123.65.35.80
Serve Application C (HTTPS) (4443)
Serve Application D (HTTPS) (443)
Home
VPN Client-To-Site
Using Application A (57350)
Using Application C (57351)
Using Application F (57352)
Phone
192.168.1.5
VPN: 10.10.10.15
Router
192.168.1.1
123.123.123.123
Internet
Laptop
192.168.1.7
Using Application B (57350)
Using Application E (57351)
Using Application I (57352)
Company B
Server
172.16.1.2
164.56.44.49
Serve Application E (DNS) (53)
Serve Application F (SMTP) (25)
Router
172.16.1.1
164.56.44.48
Server
172.16.1.3
164.56.44.50
Serve Application G (HTTPS) (53)
Serve Application I (SFTP) (22)

Src IP	Dest IP	Trns	Src Port	Dest Port	Data	App Pro
123.123.123.123	123.65.35.79	UDP	57350	53	XXX	DNS
123.65.35.79	123.123.123.123	UDP	53	57350	XXX	DNS
123.123.123.123	123.65.35.80	TCP	57351	4443	XXX	HTTPS
123.65.35.80	123.123.123.123	TCP	4443	57351	XXX	HTTPS
123.123.123.123	164.56.44.49	TCP	57352	25	XXX	SMTP
164.56.44.49	123.123.123.123	TCP	25	57352	XXX	SMTP
123.123.123.123	123.65.35.79	TCP	57350	25	XXX	SMTP
123.65.35.79	123.123.123.123	TCP	25	57350	XXX	SMTP
123.123.123.123	164.56.44.49	UDP	57351	53	XXX	DNS
164.56.44.49	123.123.123.123	UDP	53	57351	XXX	DNS
123.123.123.123	164.56.44.50	TCP	57352	22	XXX	SFTP
164.56.44.50	123.123.123.123	TCP	22	57352	XXX	SFTP
10.10.10.15	10.10.10.10	UDP	57350	53	XXX	DNS
10.10.10.10	10.10.10.15	UDP	53	57350	XXX	DNS
10.10.10.15	10.10.10.11	TCP	57351	4443	XXX	HTTPS
10.10.10.11	10.10.10.15	TCP	4443	57351	XXX	HTTPS

VPN: Virtual Private Network : Site-To-Site

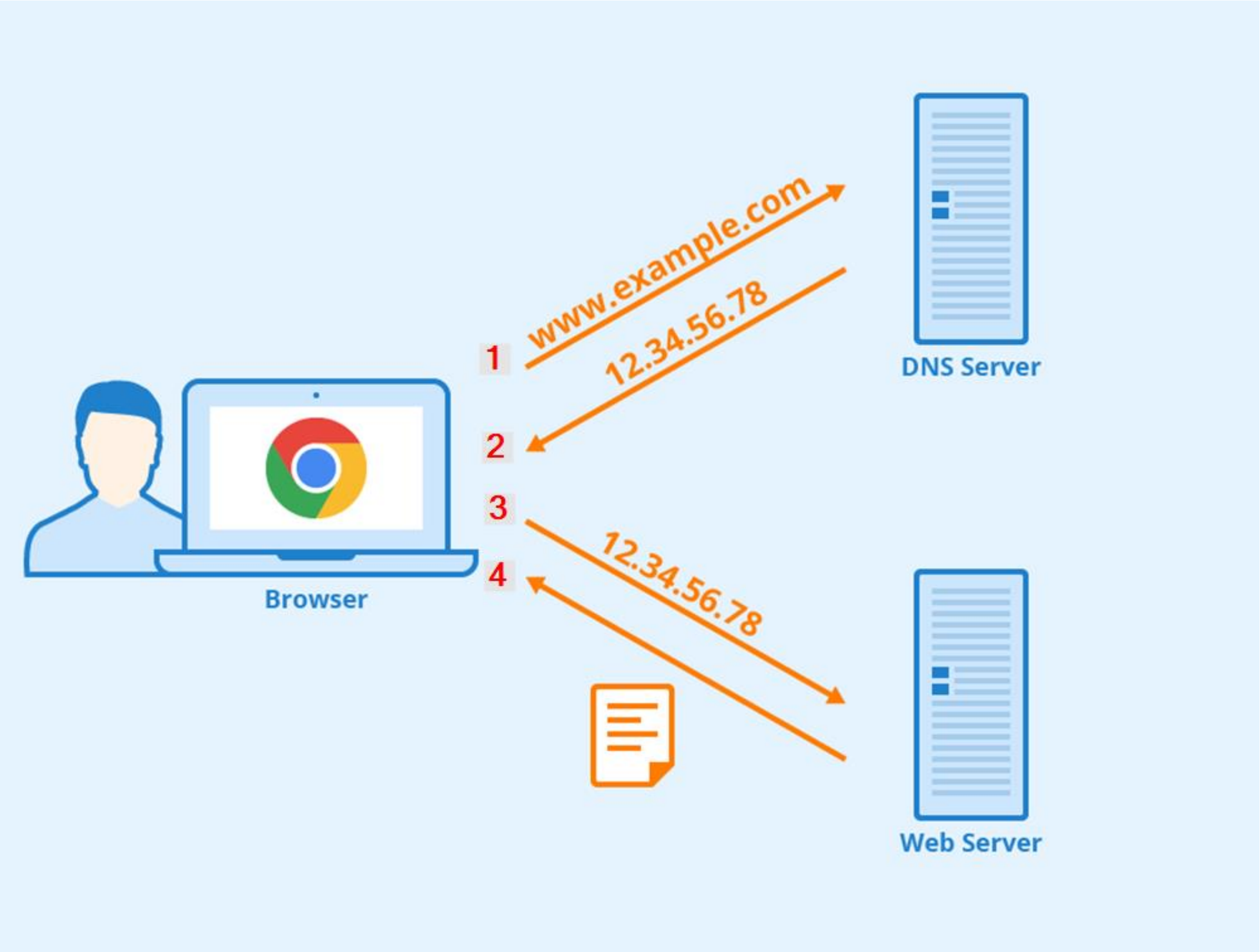

Home
Using Application A (57350)
Using Application C (57351)
Using Application F (57352)
Phone
192.168.1.5
Router
192.168.1.1
123.123.123.123
Laptop
192.168.1.7
Using Application B (57350)
Using Application E (57351)
Using Application I (57352)
VPN Site-To-Site
Internet
Company A
Serve Application A (DNS) (53)
Serve Application B (SMTP) (25)
Server
10.10.10.10
123.65.35.79
Router
10.10.10.9
123.65.35.78
Serve Application C (HTTPS) (4443)
Serve Application D (HTTPS) (443)
Server
10.10.10.11
123.65.35.80
Company B
Serve Application E (DNS) (53)
Serve Application F (SMTP) (25)
Server
172.16.1.2
164.56.44.49
Router
172.16.1.1
164.56.44.48
Serve Application G (HTTPS) (53)
Serve Application I (SFTP) (22)
Server
172.16.1.3
164.56.44.50

Src IP	Dest IP	Trns	Src Port	Dest Port	Data	App Pro
123.123.123.123	123.65.35.79	UDP	57350	53	XXX	DNS
123.65.35.79	123.123.123.123	UDP	53	57350	XXX	DNS
123.123.123.123	123.65.35.80	TCP	57351	4443	XXX	HTTPS
123.65.35.80	123.123.123.123	TCP	4443	57351	XXX	HTTPS
123.123.123.123	164.56.44.49	TCP	57352	25	XXX	SMTP
164.56.44.49	123.123.123.123	TCP	25	57352	XXX	SMTP
123.123.123.123	123.65.35.79	TCP	57350	25	XXX	SMTP
123.65.35.79	123.123.123.123	TCP	25	57350	XXX	SMTP
123.123.123.123	164.56.44.49	UDP	57351	53	XXX	DNS
164.56.44.49	123.123.123.123	UDP	53	57351	XXX	DNS
123.123.123.123	164.56.44.50	TCP	57352	22	XXX	SFTP
164.56.44.50	123.123.123.123	TCP	22	57352	XXX	SFTP
192.168.1.5	10.10.10.10	UDP	57350	53	XXX	DNS
10.10.10.10	192.168.1.5	UDP	53	57350	XXX	DNS
192.168.1.5	10.10.10.11	TCP	57351	4443	XXX	HTTPS
10.10.10.11	192.168.1.5	TCP	4443	57351	XXX	HTTPS
192.168.1.7	10.10.10.10	TCP	57350	25	XXX	SMTP
10.10.10.10	192.168.1.7	TCP	25	57350	XXX	SMTP

15
Coffee
Est. 2016

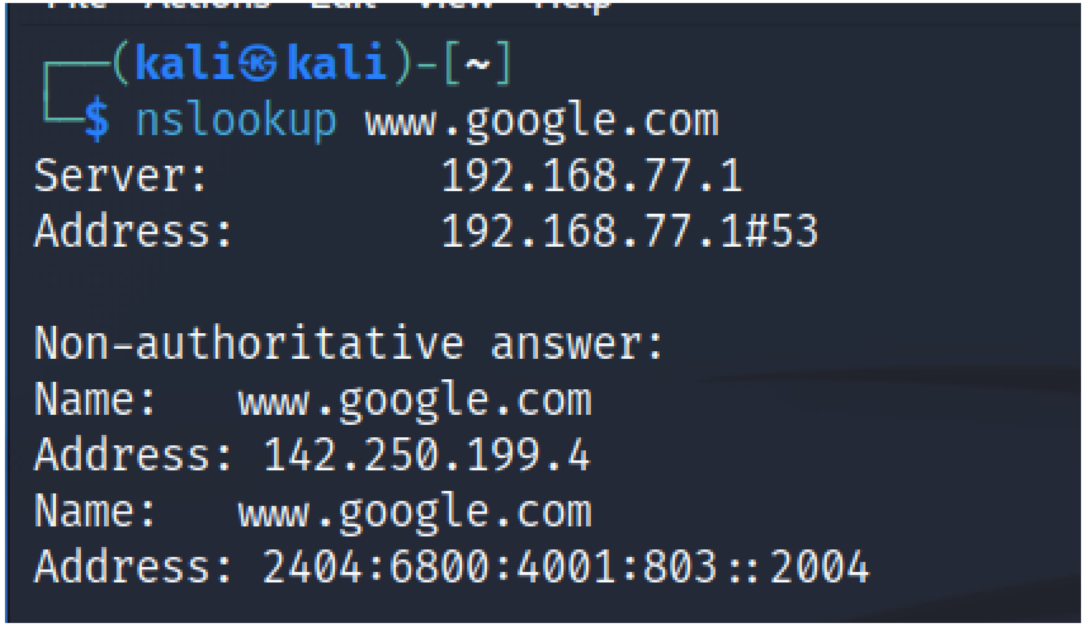
Domain Name System (DNS)

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 12.34.56.78)



Domain name resolve with nslookup

- We can use nslookup command to resolve the domain name to IP address
- nslookup <domainName>

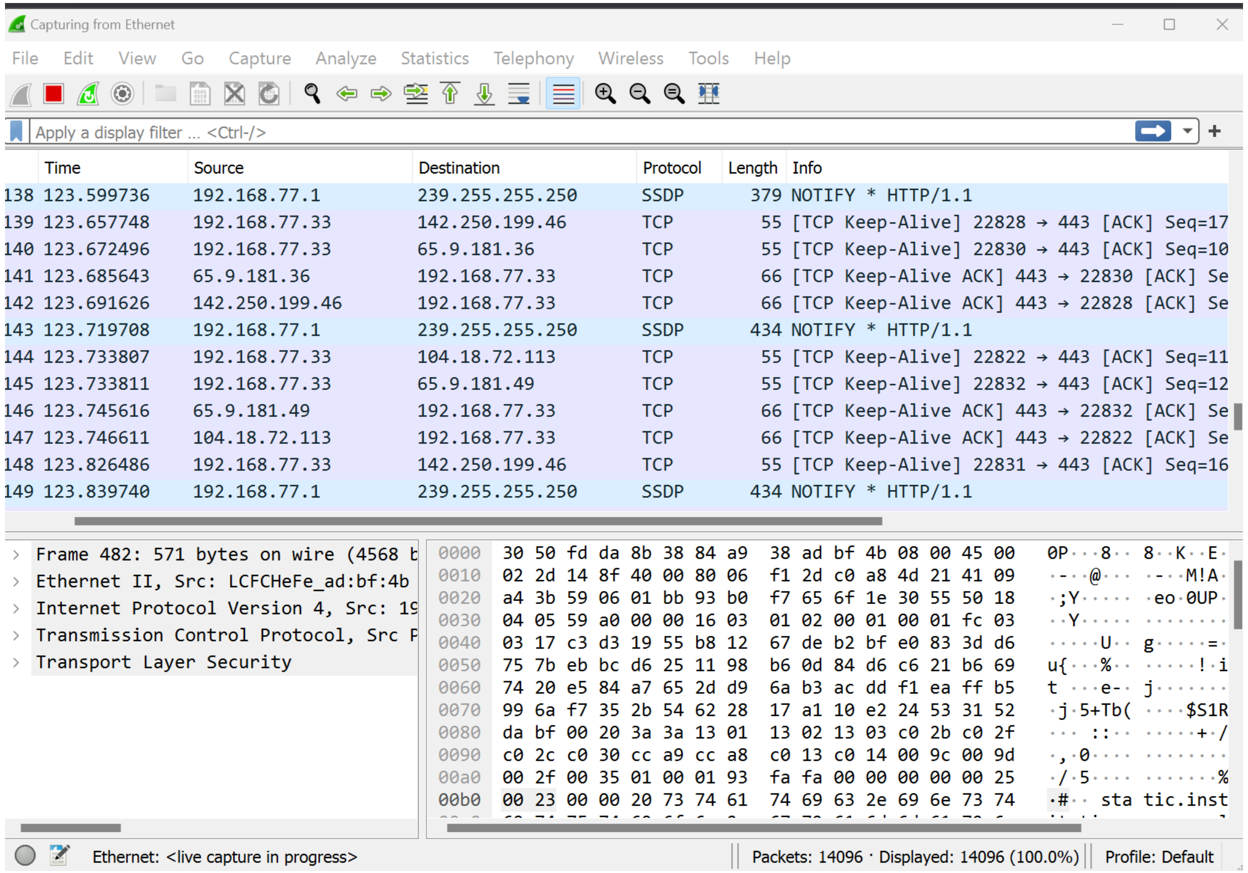


```
┌──(kali㉿kali)-[~]
└─$ nslookup www.google.com
Server:         192.168.77.1
Address:        192.168.77.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.199.4
Name:   www.google.com
Address: 2404:6800:4001:803::2004
```

Packet Sniffing with Wireshark

- We can use the Wireshark to monitor the incoming and outgoing traffic from our machines.
- Need to download and install from Wireshark page.



```
    Time        Source          Destination      Protocol  Length Info
138 123.599736  192.168.77.1    239.255.255.250  SSDP         379 NOTIFY * HTTP/1.1
139 123.657748  192.168.77.33   142.250.199.46   TCP           55 [TCP Keep-Alive] 22828 → 443 [ACK] Seq=17
140 123.672496  192.168.77.33   65.9.181.36      TCP           55 [TCP Keep-Alive] 22830 → 443 [ACK] Seq=10
141 123.685643  65.9.181.36     192.168.77.33    TCP           66 [TCP Keep-Alive ACK] 443 → 22830 [ACK] Se
142 123.691626  142.250.199.46  192.168.77.33    TCP           66 [TCP Keep-Alive ACK] 443 → 22828 [ACK] Se
143 123.719708  192.168.77.1    239.255.255.250  SSDP         434 NOTIFY * HTTP/1.1
144 123.733807  192.168.77.33   104.18.72.113    TCP           55 [TCP Keep-Alive] 22822 → 443 [ACK] Seq=11
145 123.733811  192.168.77.33   65.9.181.49      TCP           55 [TCP Keep-Alive] 22832 → 443 [ACK] Seq=12
146 123.745616  65.9.181.49     192.168.77.33    TCP           66 [TCP Keep-Alive ACK] 443 → 22832 [ACK] Se
147 123.746611  104.18.72.113   192.168.77.33    TCP           66 [TCP Keep-Alive ACK] 443 → 22822 [ACK] Se
148 123.826486  192.168.77.33   142.250.199.46   TCP           55 [TCP Keep-Alive] 22831 → 443 [ACK] Seq=16
149 123.839740  192.168.77.1    239.255.255.250  SSDP         434 NOTIFY * HTTP/1.1
```

```
> Frame 482: 571 bytes on wire (4568 b
> Ethernet II, Src: LCFCHeFe_ad:bf:4b
> Internet Protocol Version 4, Src: 19
> Transmission Control Protocol, Src P
> Transport Layer Security
```

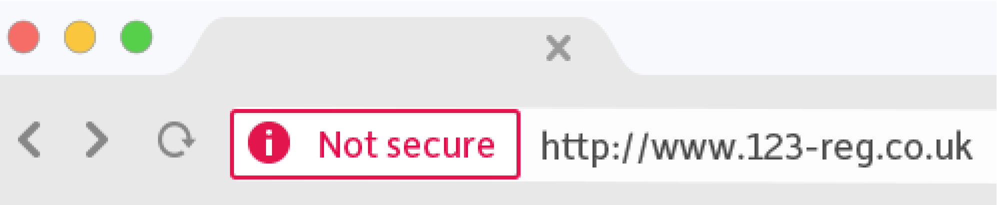
```
0000  30 50 fd da 8b 38 84 a9  38 ad bf 4b 08 00 45 00   0P···8·· 8··K··E·
0010  02 2d 14 8f 40 00 80 06  f1 2d c0 a8 4d 21 41 09   ·-··@··· ·-··M!A·
0020  a4 3b 59 06 01 bb 93 b0  f7 65 6f 1e 30 55 50 18   ·;Y····· ·eo·0UP·
0030  04 05 59 a0 00 00 16 03  01 02 00 01 00 01 fc 03   ··Y····· ········
0040  03 17 c3 d3 19 55 b8 12  67 de b2 bf e0 83 3d d6   ·····U·· g·····=·
0050  75 7b eb bc d6 25 11 98  b6 0d 84 d6 c6 21 b6 69   u{···%·· ·····!·i
0060  74 20 e5 84 a7 65 2d d9  6a b3 ac dd f1 ea ff b5   t ···e-· j·······
0070  99 6a f7 35 2b 54 62 28  17 a1 10 e2 24 53 31 52   ·j·5+Tb( ····$S1R
0080  da bf 00 20 3a 3a 13 01  13 02 13 03 c0 2b c0 2f   ··· ::·· ·····+·/
0090  c0 2c c0 30 cc a9 cc a8  c0 13 c0 14 00 9c 00 9d   ·,·0···· ········
00a0  00 2f 00 35 01 00 01 93  fa fa 00 00 00 00 00 25   ·/·5···· ·······%
00b0  00 23 00 00 20 73 74 61  74 69 63 2e 69 6e 73 74   ·#·· sta tic.inst
```

Ethernet: <live capture in progress>

Packets: 14096 · Displayed: 14096 (100.0%)

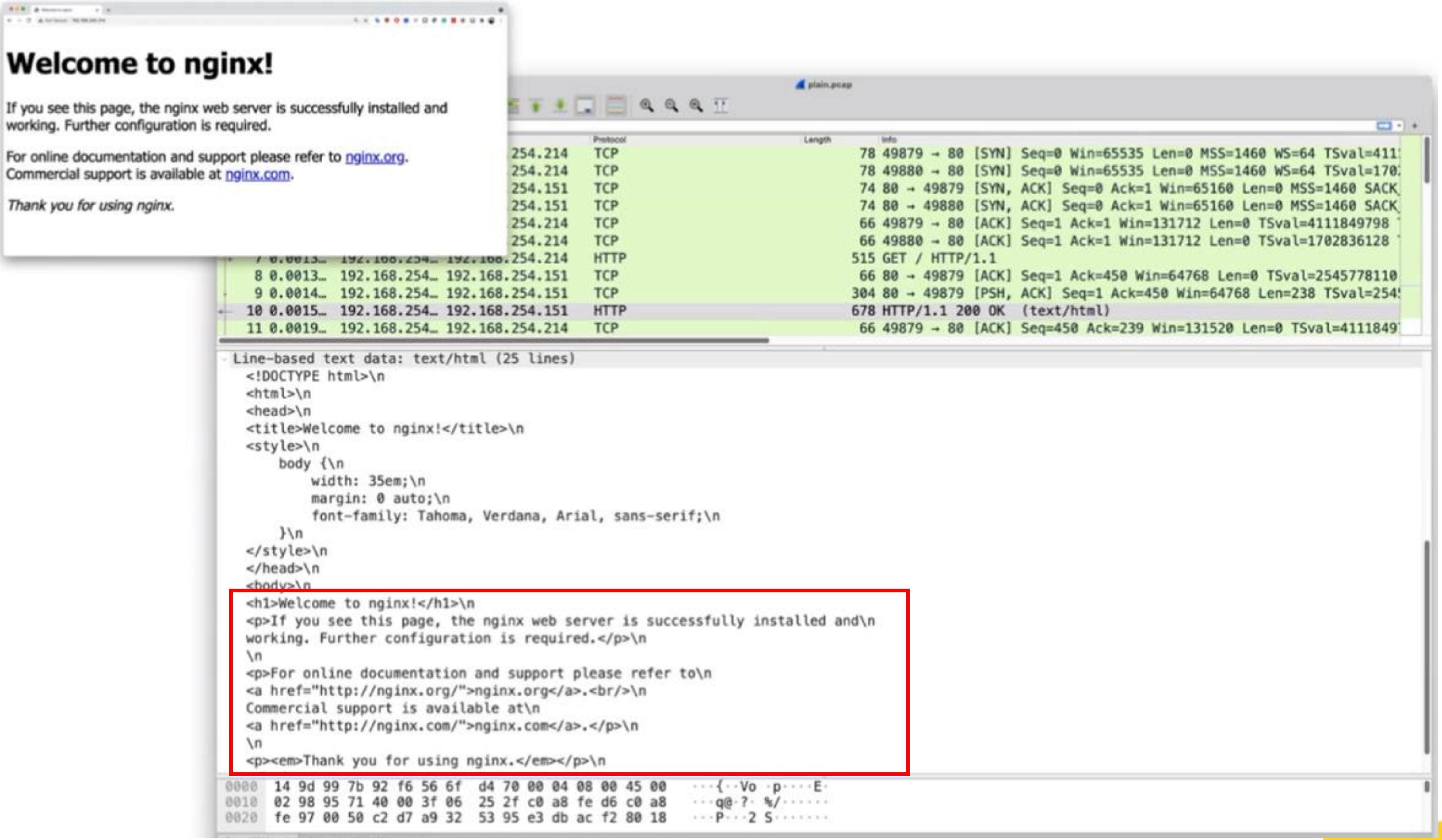
Profile: Default

https://www.wireshark.org/download.html



Not secure http://www.123-reg.co.uk ✗

🔒 | https:// ✓



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

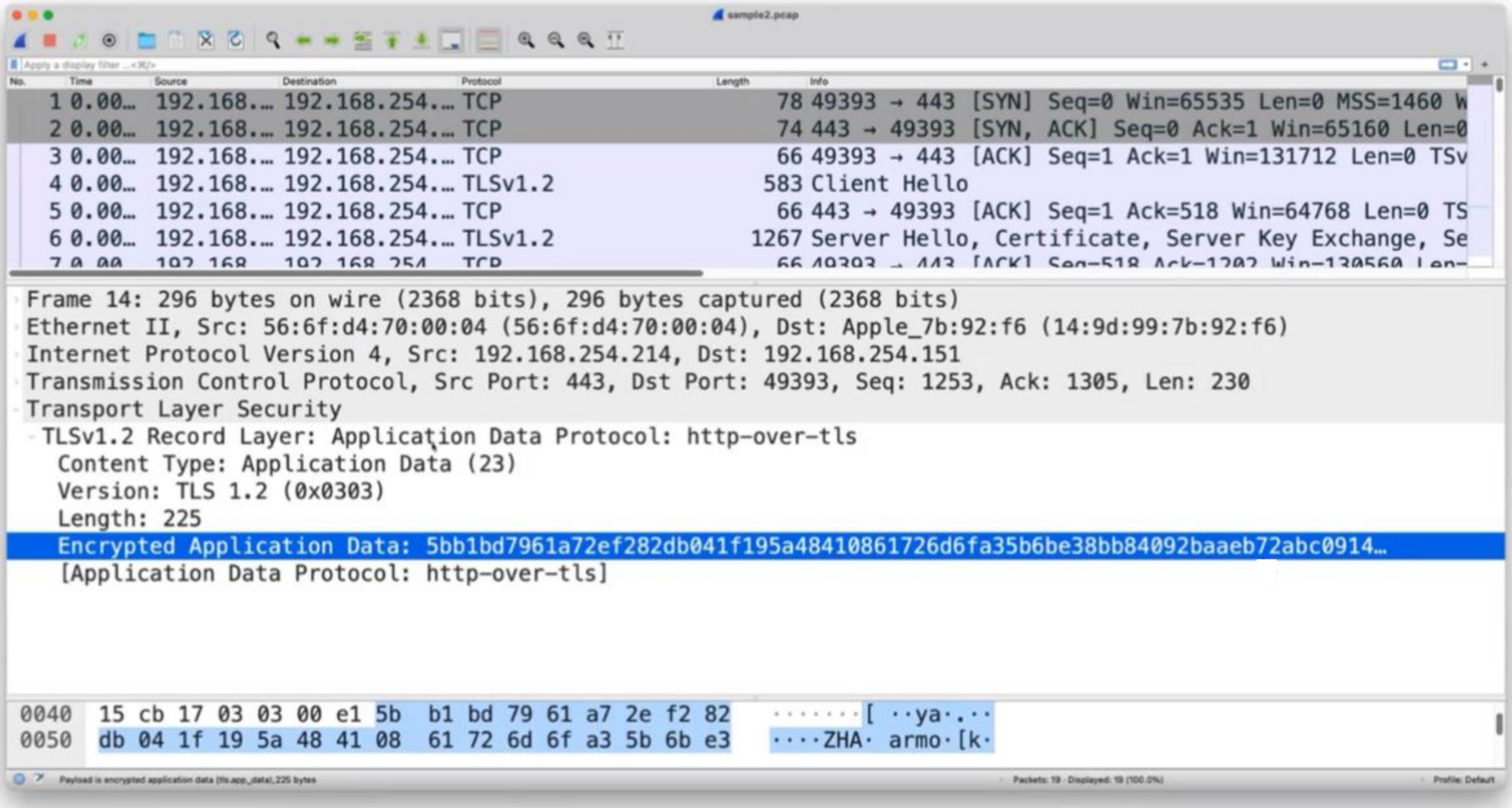
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

```
                                   Protocol               Length  Info
     254.214                       TCP                        78 49879 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=411
     254.214                       TCP                        78 49880 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=170
     254.151                       TCP                        74 80 → 49879 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK
     254.151                       TCP                        74 80 → 49880 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK
     254.214                       TCP                        66 49879 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=4111849798
     254.214                       TCP                        66 49880 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=1702836128
 7 0.0013… 192.168.254… 192.168.254.214 HTTP                     515 GET / HTTP/1.1
 8 0.0013… 192.168.254… 192.168.254.151 TCP                       66 80 → 49879 [ACK] Seq=1 Ack=450 Win=64768 Len=0 TSval=2545778110
 9 0.0014… 192.168.254… 192.168.254.151 TCP                      304 80 → 49879 [PSH, ACK] Seq=1 Ack=450 Win=64768 Len=238 TSval=254
10 0.0015… 192.168.254… 192.168.254.151 HTTP                     678 HTTP/1.1 200 OK  (text/html)
11 0.0019… 192.168.254… 192.168.254.214 TCP                       66 49879 → 80 [ACK] Seq=450 Ack=239 Win=131520 Len=0 TSval=4111849
```

```
Line-based text data: text/html (25 lines)
  <!DOCTYPE html>\n
  <html>\n
  <head>\n
  <title>Welcome to nginx!</title>\n
  <style>\n
      body {\n
          width: 35em;\n
          margin: 0 auto;\n
          font-family: Tahoma, Verdana, Arial, sans-serif;\n
      }\n
  </style>\n
  </head>\n
  <body>\n
  <h1>Welcome to nginx!</h1>\n
  <p>If you see this page, the nginx web server is successfully installed and\n
  working. Further configuration is required.</p>\n
  \n
  <p>For online documentation and support please refer to\n
  <a href="http://nginx.org/">nginx.org</a>.<br/>\n
  Commercial support is available at\n
  <a href="http://nginx.com/">nginx.com</a>.</p>\n
  \n
  <p><em>Thank you for using nginx.</em></p>\n
```

```
0000  14 9d 99 7b 92 f6 56 6f  d4 70 00 04 08 00 45 00   ···{··Vo ·p····E·
0010  02 98 95 71 40 00 3f 06  25 2f c0 a8 fe d6 c0 a8   ···q@·?· %/······
0020  fe 97 00 50 c2 d7 a9 32  53 95 e3 db ac f2 80 18   ···P···2 S·······
```



```
No. Time    Source       Destination      Protocol          Length Info
  1 0.00…   192.168.…    192.168.254.…    TCP                   78 49393 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 W
  2 0.00…   192.168.…    192.168.254.…    TCP                   74 443 → 49393 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0
  3 0.00…   192.168.…    192.168.254.…    TCP                   66 49393 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSv
  4 0.00…   192.168.…    192.168.254.…    TLSv1.2              583 Client Hello
  5 0.00…   192.168.…    192.168.254.…    TCP                   66 443 → 49393 [ACK] Seq=1 Ack=518 Win=64768 Len=0 TS
  6 0.00…   192.168.…    192.168.254.…    TLSv1.2             1267 Server Hello, Certificate, Server Key Exchange, Se
  7 0.00…   192.168.…    192.168.254.…    TCP                   66 49393 → 443 [ACK] Seq=518 Ack=1202 Win=130560 Len=
```

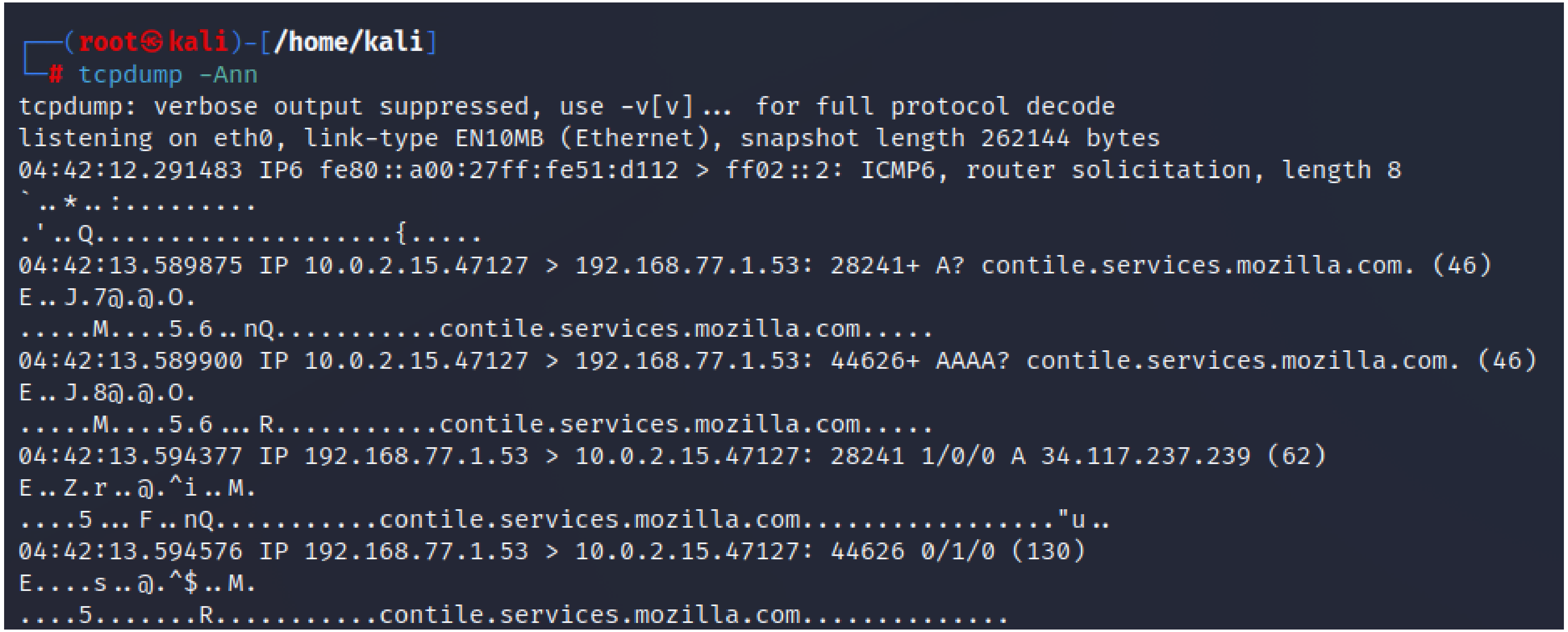
```
Frame 14: 296 bytes on wire (2368 bits), 296 bytes captured (2368 bits)
Ethernet II, Src: 56:6f:d4:70:00:04 (56:6f:d4:70:00:04), Dst: Apple_7b:92:f6 (14:9d:99:7b:92:f6)
Internet Protocol Version 4, Src: 192.168.254.214, Dst: 192.168.254.151
Transmission Control Protocol, Src Port: 443, Dst Port: 49393, Seq: 1253, Ack: 1305, Len: 230
Transport Layer Security
  TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 225
    Encrypted Application Data: 5bb1bd7961a72ef282db041f195a48410861726d6fa35b6be38bb84092baaeb72abc0914…
    [Application Data Protocol: http-over-tls]
```

```
0040  15 cb 17 03 03 00 e1 5b  b1 bd 79 61 a7 2e f2 82   ·······[ ··ya·.··
0050  db 04 1f 19 5a 48 41 08  61 72 6d 6f a3 5b 6b e3   ····ZHA· armo·[k·
```

Payload is encrypted application data (tls.app_data), 225 bytes   Packets: 19 · Displayed: 19 (100.0%)   Profile: Default

Packet Sniffing with TCPDump

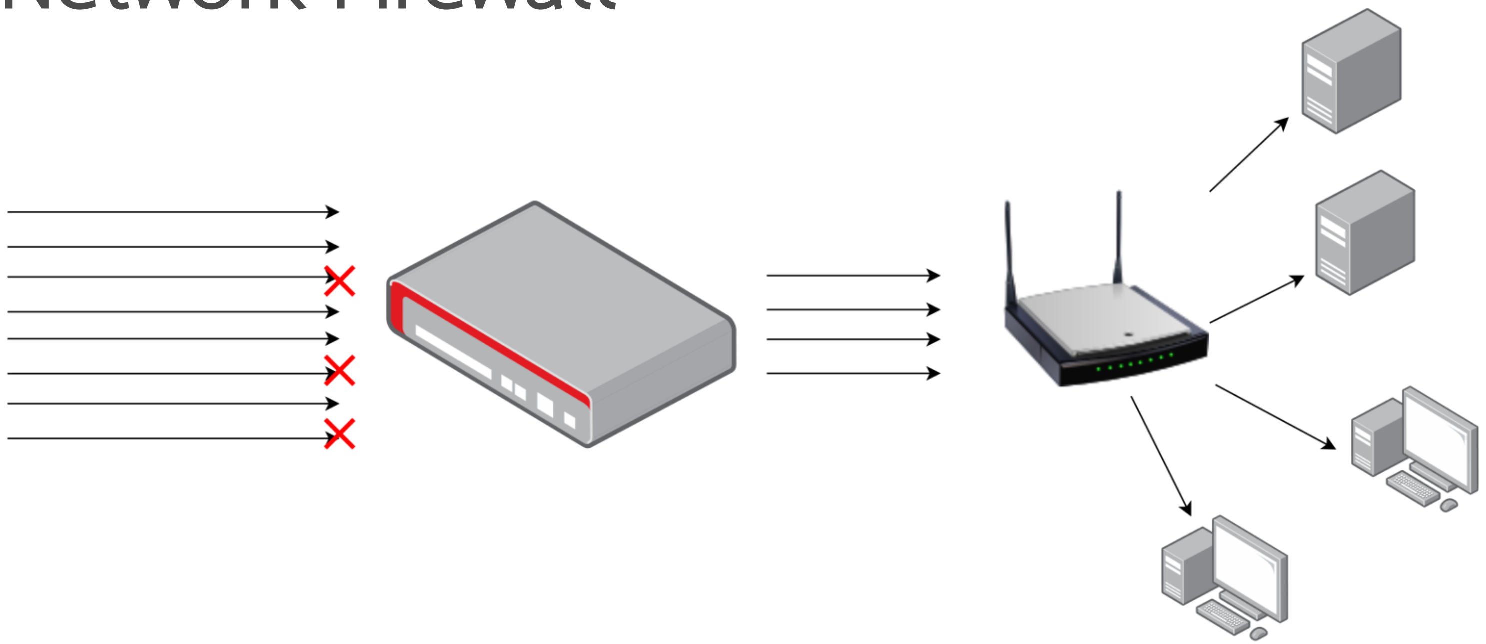
- We can use the TCPDump to monitor the incoming and outgoing traffic from our machines.
- tcpdump –i <Interface> -Ann



```
┌──(root㉿kali)-[/home/kali]
└─# tcpdump -Ann
tcpdump: verbose output suppressed, use -v[v]...  for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:42:12.291483 IP6 fe80::a00:27ff:fe51:d112 > ff02::2: ICMP6, router solicitation, length 8
`..*..:.........
.'..Q....................{.....
04:42:13.589875 IP 10.0.2.15.47127 > 192.168.77.1.53: 28241+ A? contile.services.mozilla.com. (46)
E..J.7@.@.O.
.....M....5.6..nQ...........contile.services.mozilla.com.....
04:42:13.589900 IP 10.0.2.15.47127 > 192.168.77.1.53: 44626+ AAAA? contile.services.mozilla.com. (46)
E..J.8@.@.O.
.....M....5.6...R...........contile.services.mozilla.com.....
04:42:13.594377 IP 192.168.77.1.53 > 10.0.2.15.47127: 28241 1/0/0 A 34.117.237.239 (62)
E..Z.r..@.^i..M.
....5...F..nQ...........contile.services.mozilla.com................"u..
04:42:13.594576 IP 192.168.77.1.53 > 10.0.2.15.47127: 44626 0/1/0 (130)
E....s..@.^$..M.
....5.......R...........contile.services.mozilla.com.............
```

INET MS
INET Managed Services Co., Ltd.

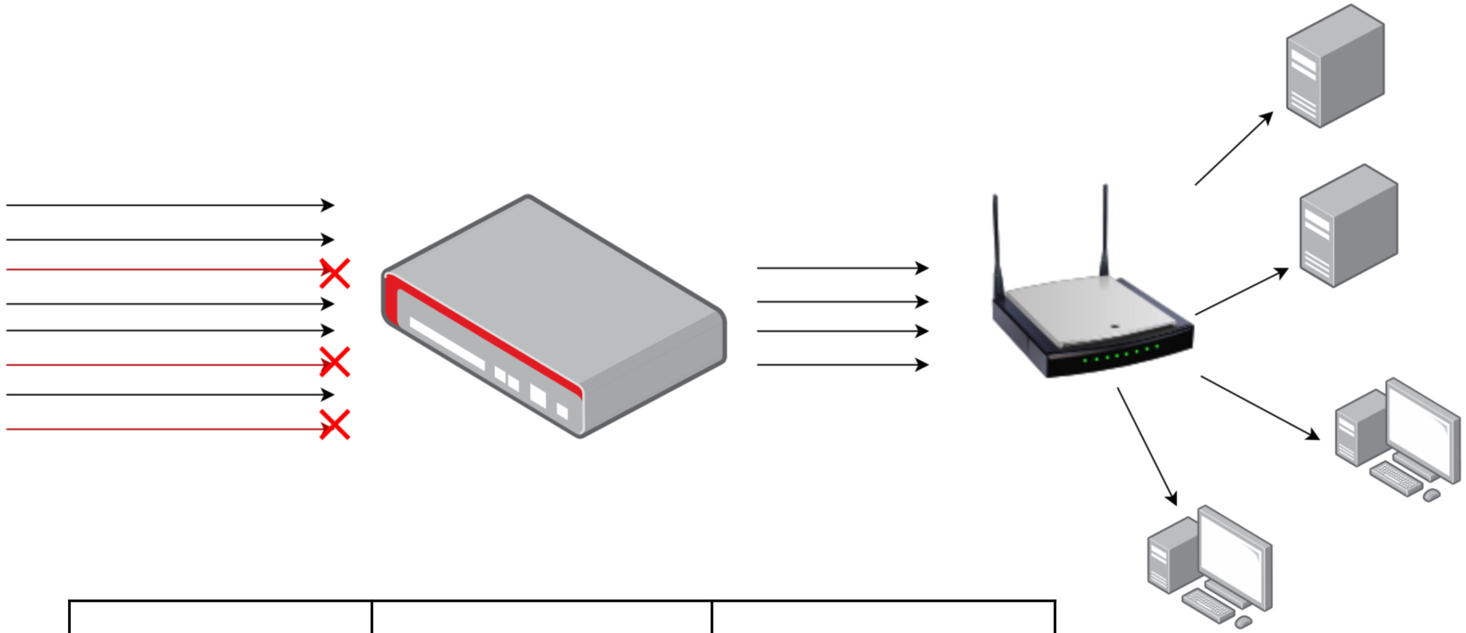
Network Firewall



Rule Number	Rule Name	Source IP	Destination IP	Source Port	Destination Port	Protocol	Action
1	Allow HTTP	Any	192.168.1.100	Any	80	TCP	Allow
2	Allow SSH	203.0.113.10	192.168.1.200	Any	22	TCP	Allow
3	Block SMTP	Any	Any	Any	25	TCP	Block
4	Allow DNS	Any	Any	Any	53	UDP	Allow
5	Block ICMP	Any	Any	Any	Any	ICMP	Block
6	Custom Rule	192.168.1.50	Any	Any	Any	Any	Allow

Intrusion Detection System (IDS)

Intrusion Prevention System (IPS)



Rule Number	Intrusion Name	Action
1	Port Scan	Block
2	DOS	Block
3	DDOS	Block

Network Firewall + IDS/IPS + other feature = UTM (Unified Threat Management)

Workshop

https://forms.gle/2npjZiS1QfJ2KuAE8

INETMS
INET Managed Services Co., Ltd.

Thank You

Innovation is our Business

INET Manged Services is a Leading Service Provider.