Scanning
Networks

@mmar

Network Scanning refers to the set of procedures adopted for identifying a network's hosts, ports and services. It is one of the key components of intelligence gathering that attackers use to create a profile of the target organization

It has the following main objectives:

- ✓ Discover live hosts, IP addresses and open ports of all live hosts
- ✓ Discover OS and system architecture
- ✓ Discover services running on hosts
- ✓ Discover vulnerabilities on live hosts

NMAP

- Nmap is a free and open-source network scanner. Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses
- Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection. These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features. Nmap can adapt to network conditions including latency and congestion during a scan

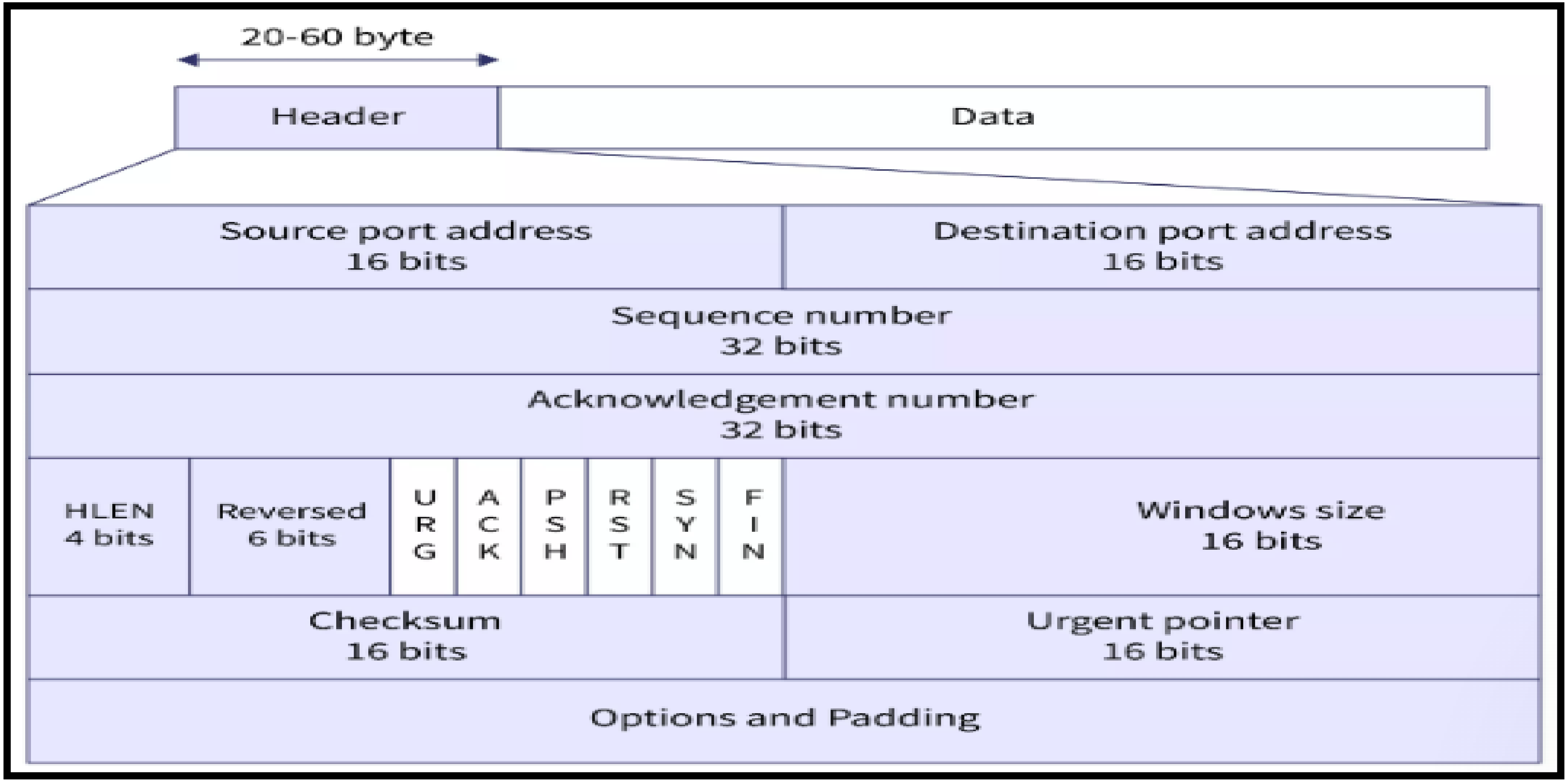
Port States

- While many port scanners have traditionally labelled all ports into the **open** or **closed** states, Nmap is much more granular.
- It divides ports into six states: **open**, **closed**, **filtered**, **unfiltered**, **open|filtered**, or **closed|filtered**
- These states are not intrinsic properties of the port itself, but describe how Nmap sees them
- For example, an Nmap scan from the same network as the target may show port 135/tcp as **open**, while a scan at the same time with the same options from across the Internet might show that port as **filtered**

Port States

- **open** – indicates that an application is listening for connections on the port. The primary goal of port scanning is to find these.
- **closed** – indicates that the probes were received but there is no application listening on the port.
- **filtered** – indicates that the probes were not received, and the state could not be established.
- **unfiltered** – indicates that the probes were received but a state could not be established. In other words, a port is accessible, but Nmap is unable to determine whether it is open or closed.
- **open/filtered**– indicates that the port was filtered or open, but Nmap couldn't establish the state.
- **closed/filtered** – indicates that Nmap is unable to determine whether a port is closed or filtered.

TCP Header



20-60 byte

Header

Data

Source port address
16 bits

Destination port address
16 bits

Sequence number
32 bits

Acknowledgement number
32 bits

HLEN
4 bits

Reversed
6 bits

URG

ACK

PSH

RST

SYN

FIN

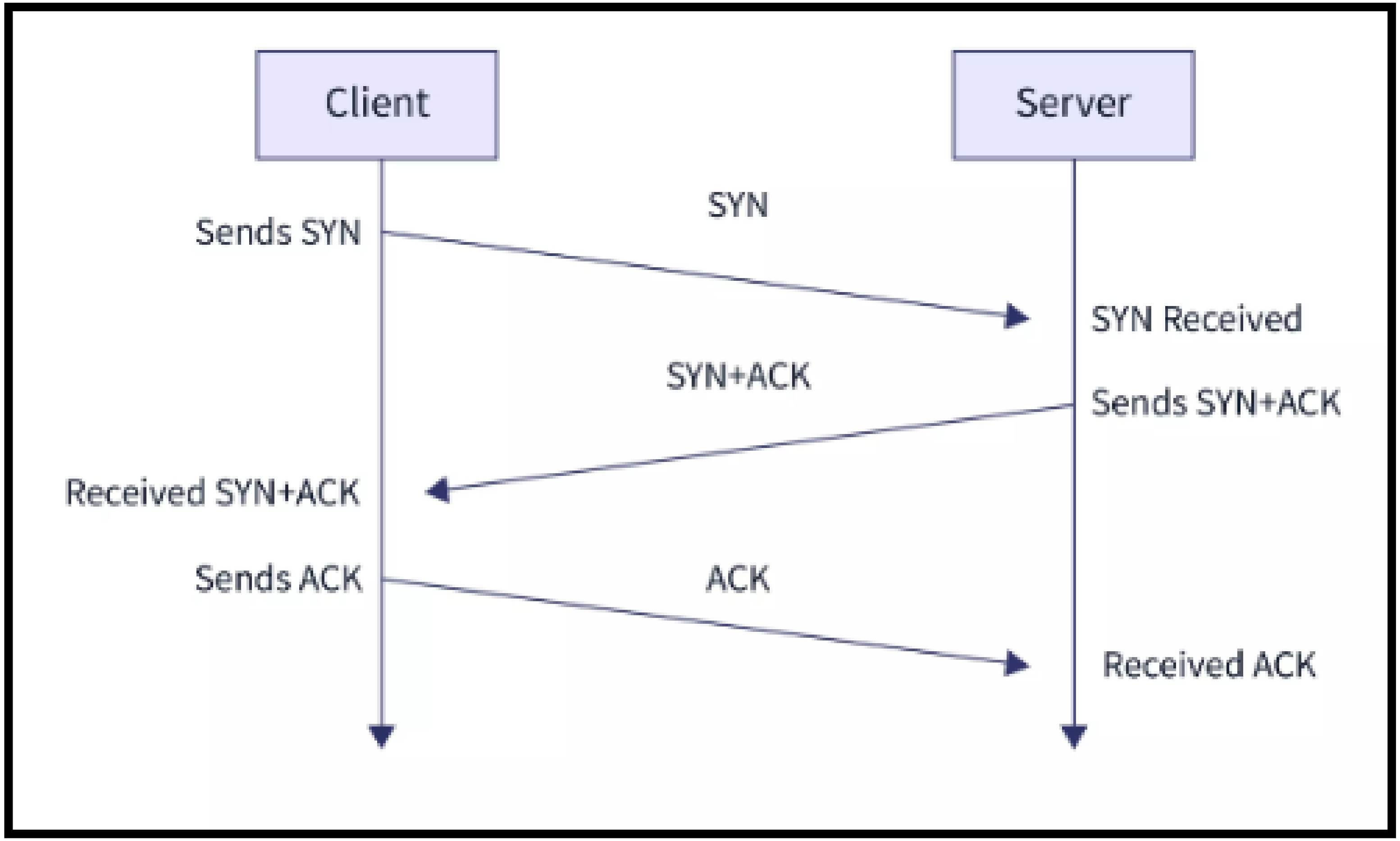
Windows size
16 bits

Checksum
16 bits

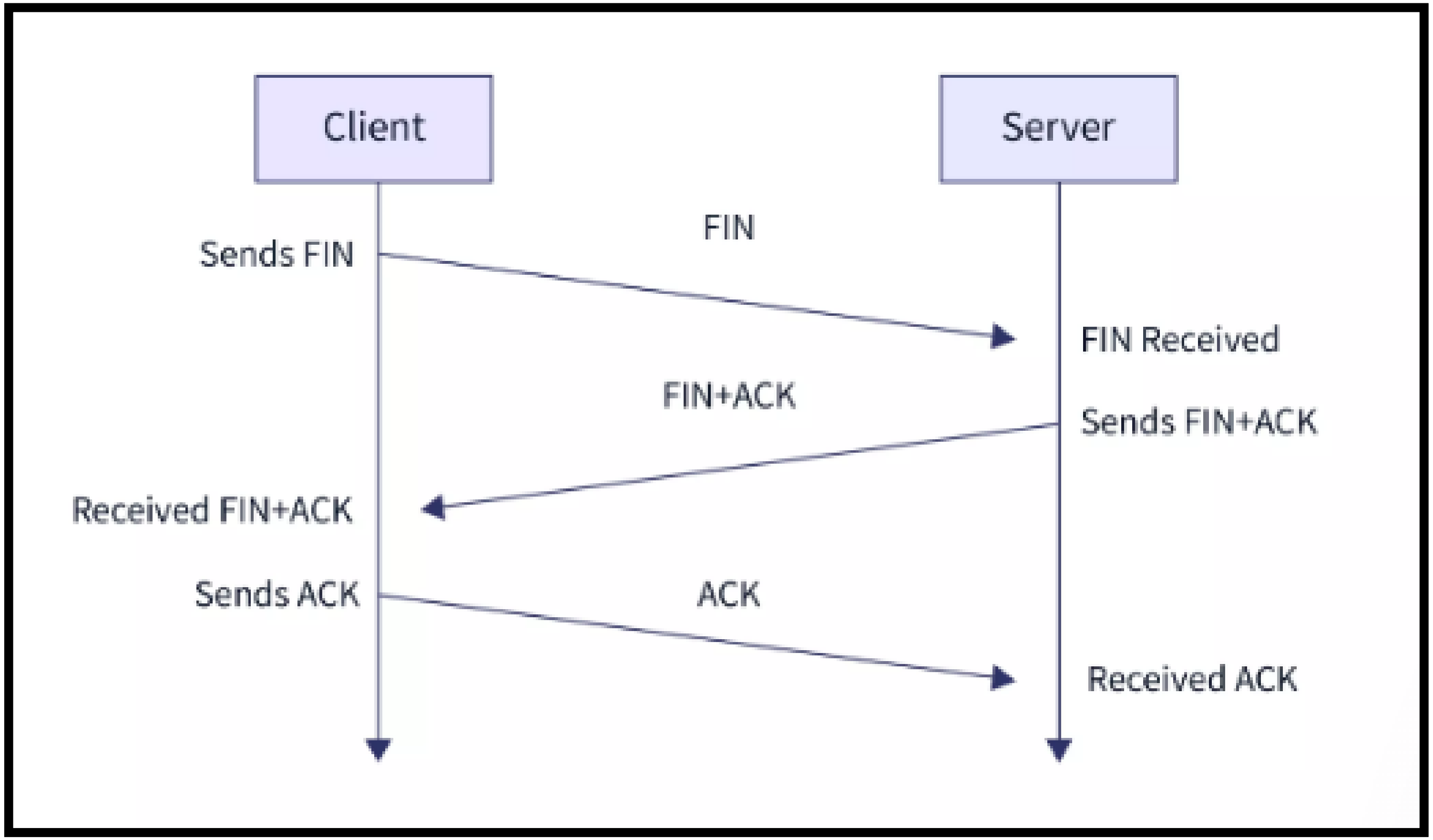
Urgent pointer
16 bits

Options and Padding

TCP Handshake

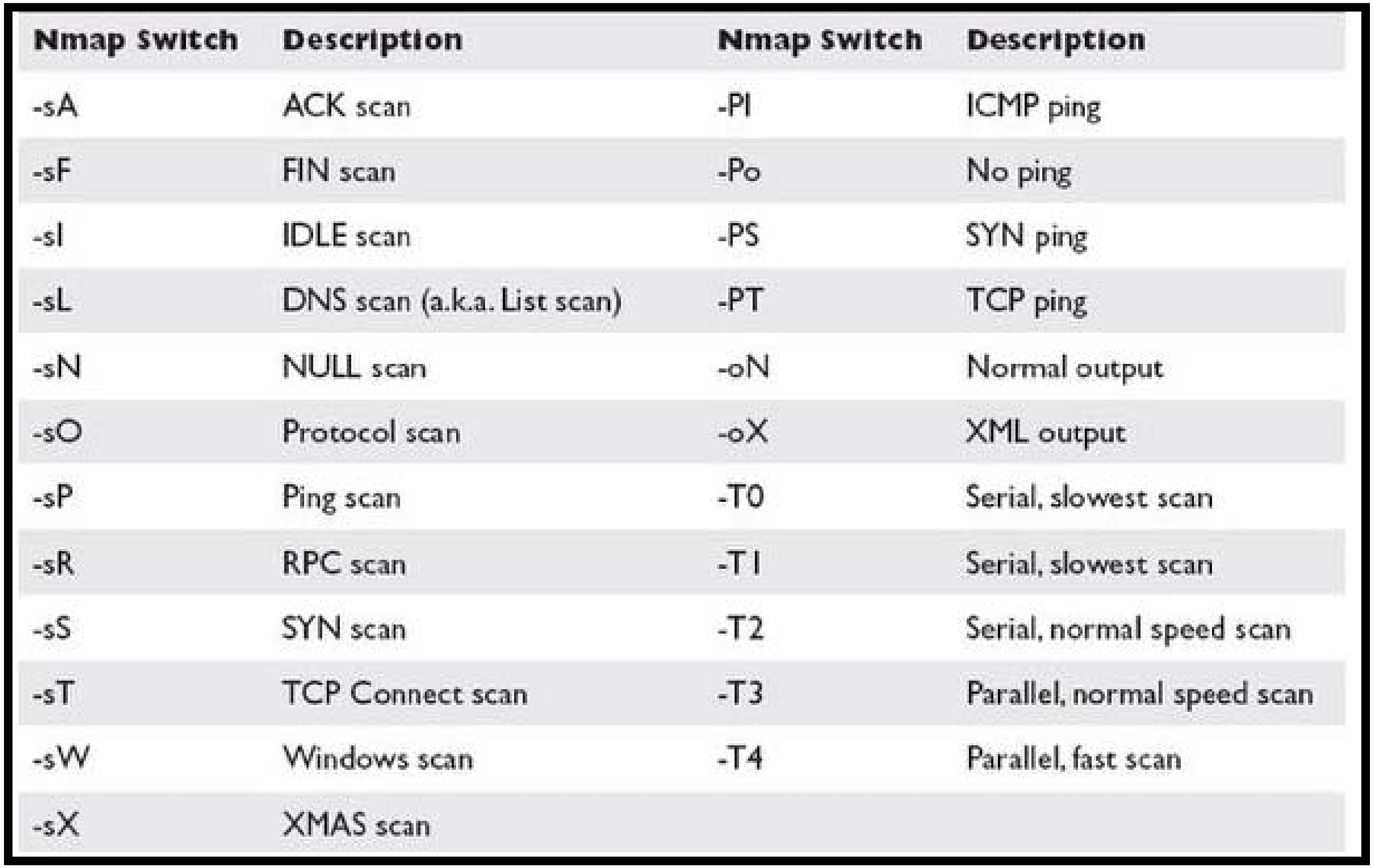


Connection termination



SCAN TYPES

Nmap Scan types

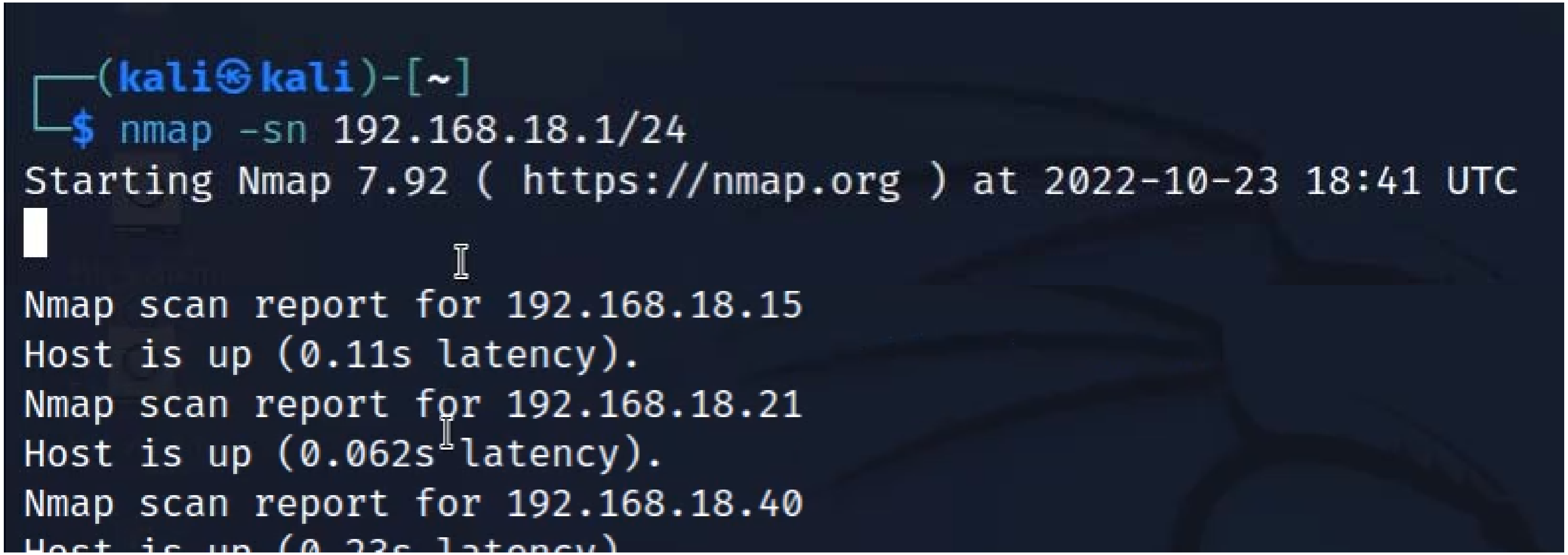


Nmap Switch	Description	Nmap Switch	Description
-sA	ACK scan	-PI	ICMP ping
-sF	FIN scan	-Po	No ping
-sI	IDLE scan	-PS	SYN ping
-sL	DNS scan (a.k.a. List scan)	-PT	TCP ping
-sN	NULL scan	-oN	Normal output
-sO	Protocol scan	-oX	XML output
-sP	Ping scan	-T0	Serial, slowest scan
-sR	RPC scan	-T1	Serial, slowest scan
-sS	SYN scan	-T2	Serial, normal speed scan
-sT	TCP Connect scan	-T3	Parallel, normal speed scan
-sW	Windows scan	-T4	Parallel, fast scan
-sX	XMAS scan		

Ping Scan

- Ping scan is used to scan for the live hosts on the network

>nmap –sn 192.168.18.1/24



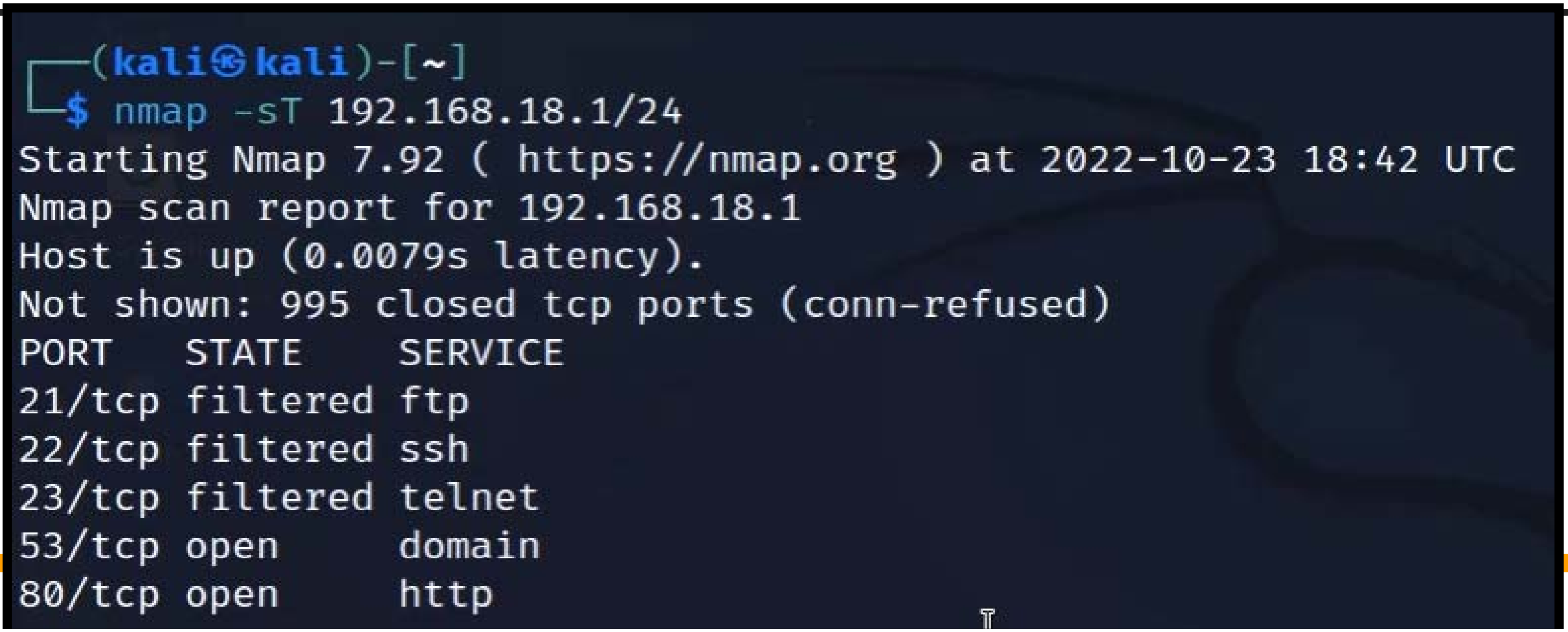
```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 192.168.18.1/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 18:41 UTC


Nmap scan report for 192.168.18.15
Host is up (0.11s latency).
Nmap scan report for 192.168.18.21
Host is up (0.062s latency).
Nmap scan report for 192.168.18.40
Host is up (0.23s latency).
```

TCP Connect Scan

- TCP scan will scan for TCP ports and ensure for listening port (open) through a 3-way handshake connection between the source and destination port

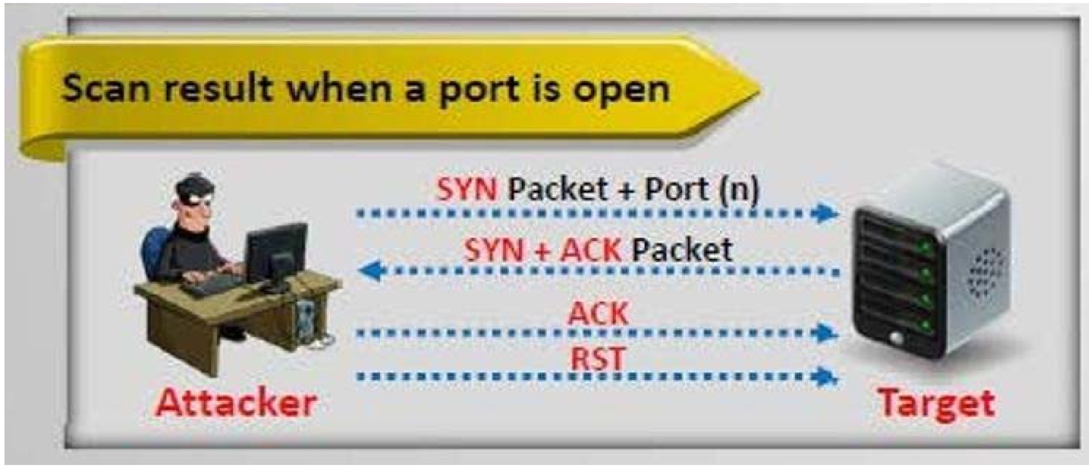
```
>nmap –sT 192.168.18.1
```

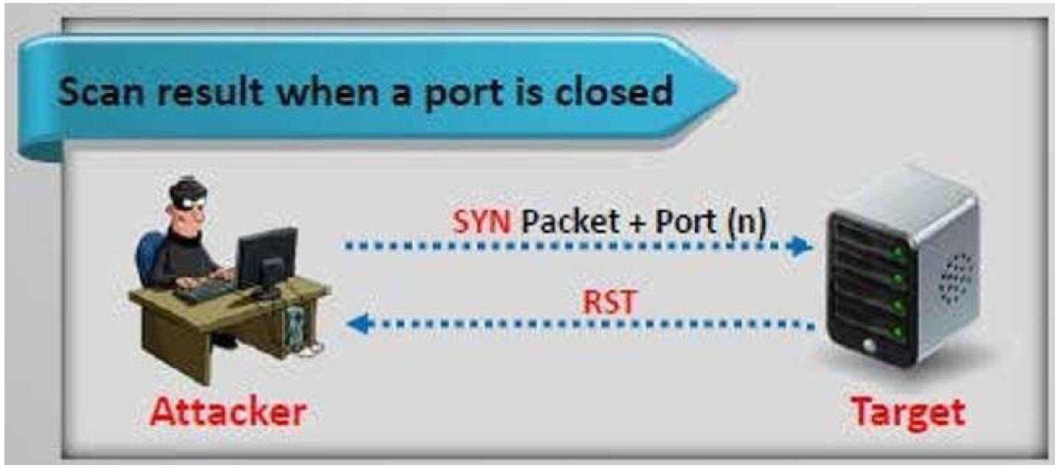


```
┌──(kali㉿kali)-[~]
└─$ nmap -sT 192.168.18.1/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 18:42 UTC
Nmap scan report for 192.168.18.1
Host is up (0.0079s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT    STATE    SERVICE
21/tcp filtered ftp
22/tcp filtered ssh
23/tcp filtered telnet
53/tcp open     domain
80/tcp open     http
```

TCP Connect Scan/Full Open Scan

- The Scan does this take longer and require more packets to obtain the same information, but target machines are more likely to log the connection
- If the port is open then source made request with **SYN** packet, a response destination sent **SYN, ACK** packet and then source sent **ACK** packets, at last source again sent **RST, ACK** packets

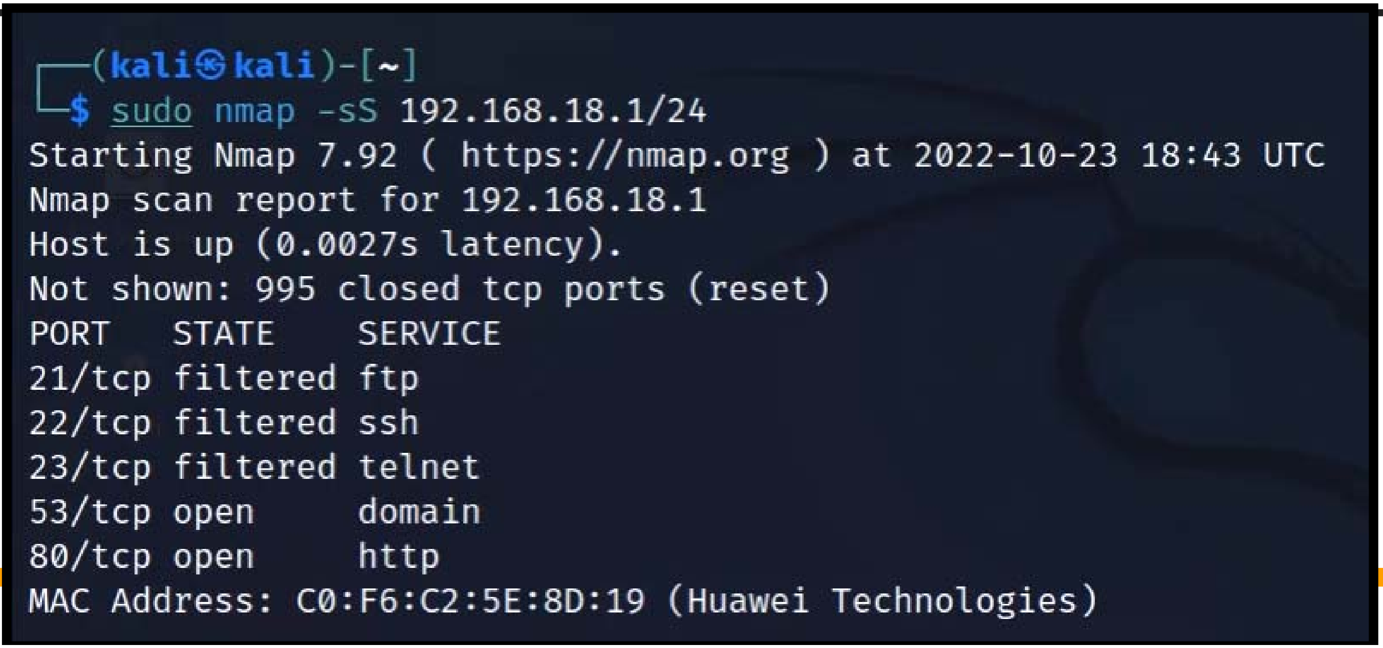




TCP Syn Scan

- This scan is often referred to as half-open scanning because you don't open a full TCP connection. You send an SYN packet, as if you are going to open a real connection and then wait for a response

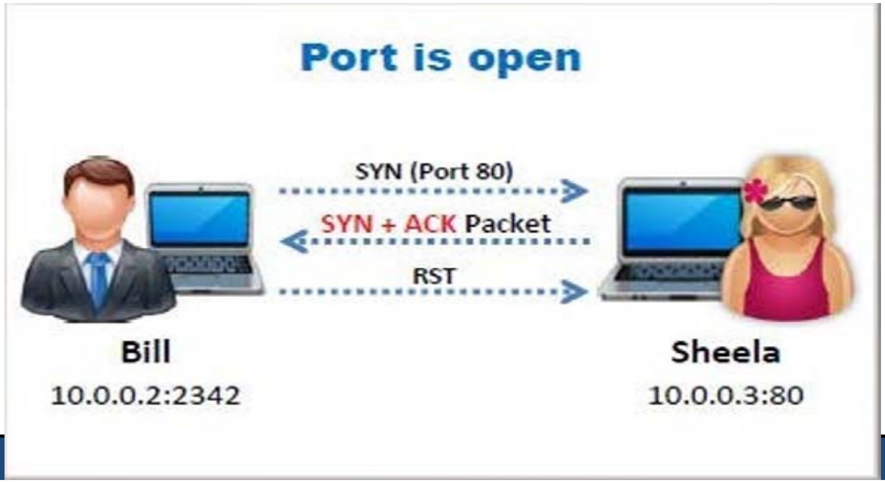
```
>nmap –sS 192.168.18.1
```

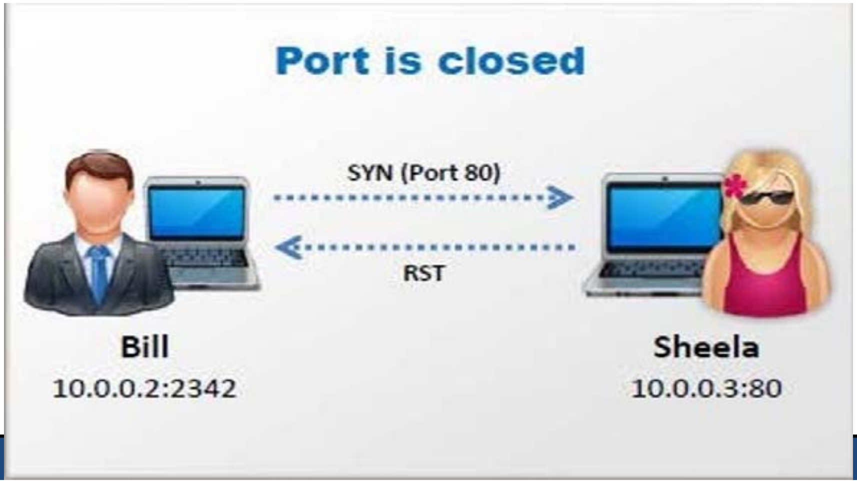


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS 192.168.18.1/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 18:43 UTC
Nmap scan report for 192.168.18.1
Host is up (0.0027s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE    SERVICE
21/tcp filtered ftp
22/tcp filtered ssh
23/tcp filtered telnet
53/tcp open     domain
80/tcp open     http
MAC Address: C0:F6:C2:5E:8D:19 (Huawei Technologies)
```

TCP SYN Scan/Stealth Scan

- SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls
- A SYN/ACK indicates the port is listening (open), while RST (reset) is indicative of a non-listener. If no response is received after several retransmissions, the port is marked as filtered
- The port is also marked filtered if an ICMP unreachable error (type 3, code 0, 1, 2, 3, 9, 10, or 13) is received

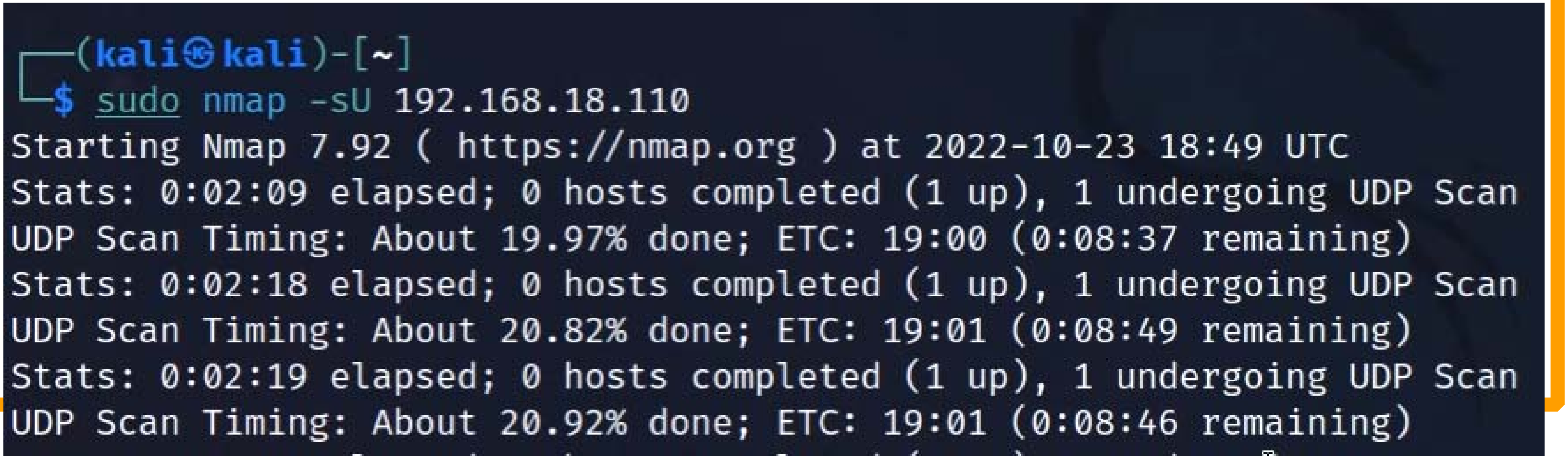




UDP Scan

- UDP scan works by sending a UDP packet to every targeted port. For most ports, this packet will be empty (no payload), but for a few of the more common ports a protocol-specific payload will be sent

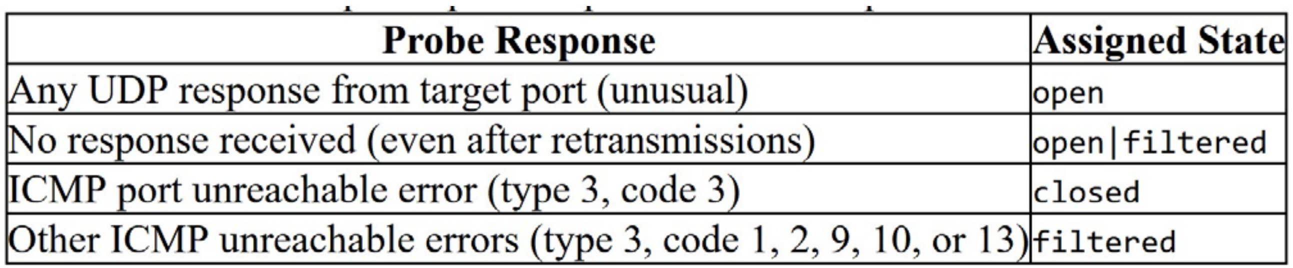
```
>nmap –sU 192.168.18.110
```



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sU 192.168.18.110
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 18:49 UTC
Stats: 0:02:09 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 19.97% done; ETC: 19:00 (0:08:37 remaining)
Stats: 0:02:18 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 20.82% done; ETC: 19:01 (0:08:49 remaining)
Stats: 0:02:19 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 20.92% done; ETC: 19:01 (0:08:46 remaining)
```

UDP Scan

- UDP is a connectionless protocol and there's no protocol-defined relationship between packets in either direction
- However, most OS TCP/IP stacks will return an ICMP “Port Unreachable” packet if a UDP packet is sent to a closed UDP port
- Thus, a port that doesn't return an ICMP packet can be assumed open
- Neither the probe-packet nor its potential ICMP packet are guaranteed to arrive

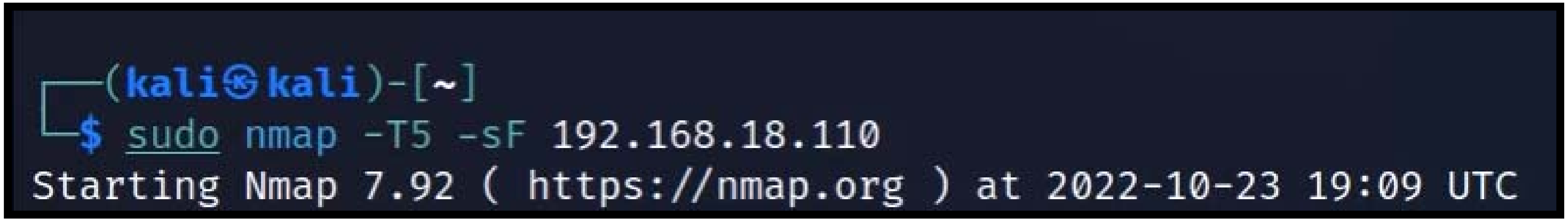


Probe Response	Assigned State
Any UDP response from target port (unusual)	`open`
No response received (even after retransmissions)	`open|filtered`
ICMP port unreachable error (type 3, code 3)	`closed`
Other ICMP unreachable errors (type 3, code 1, 2, 9, 10, or 13)	`filtered`

FIN Scan

- FIN SCAN is one of the port scanning methods in Nmap, which uses the sheer stupidity of old and stateless firewalls. In fact, when it comes to FIN Scan, our Port Scanner software sends a packet with a flag in the form of FIN meaning the end of the session to the destination firewall or host. If no response is received, it means that the port is open, and if the return is RST//ACK, it means that the server port is closed

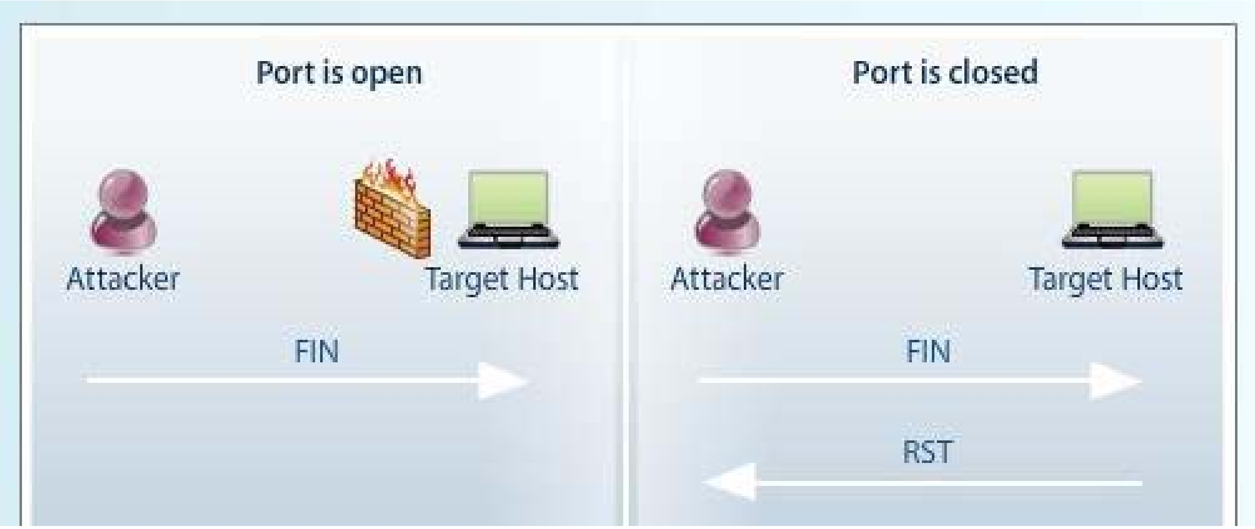
```
>nmap –sF 192.168.18.110
```



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -T5 -sF 192.168.18.110
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 19:09 UTC
```

FIN Scan

- A FIN bit is used to terminate the TCP connection between the source and destination port typically after the data transfer is complete
- Here, rather than even pretending to initiate a standard TCP connection, nmap sends a single FIN (final) packet
- If the target's TCP/IP stack is RFC-793-compliant then open ports will drop the packet and closed ports will send an RST



NULL and XMAS Scans

- NULL and XMAS scan types are exactly the same in behavior except for the TCP flags set in probe packets. If a RST packet is received, the port is considered closed, while no response means it is open|filtered.
- The port is marked filtered if an ICMP unreachable error (type 3, code 0, 1, 2, 3, 9, 10, or 13) is received
- XMAS scans are designed to manipulate the PSH, URG and FIN flags of the TCP header, Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree. When source sent FIN, PUSH, and URG packet to a specific port and if the port is open then destination will discard the packets and will not send any reply to the source
- A Null Scan is a series of TCP packets which hold a sequence number of “zeros” (0000000). since there are none flags set, the destination will not know how to reply the request.It will discard the packet and no reply will be sent, which indicate that the port is open

Nmap Cheat Sheet

Target Specification

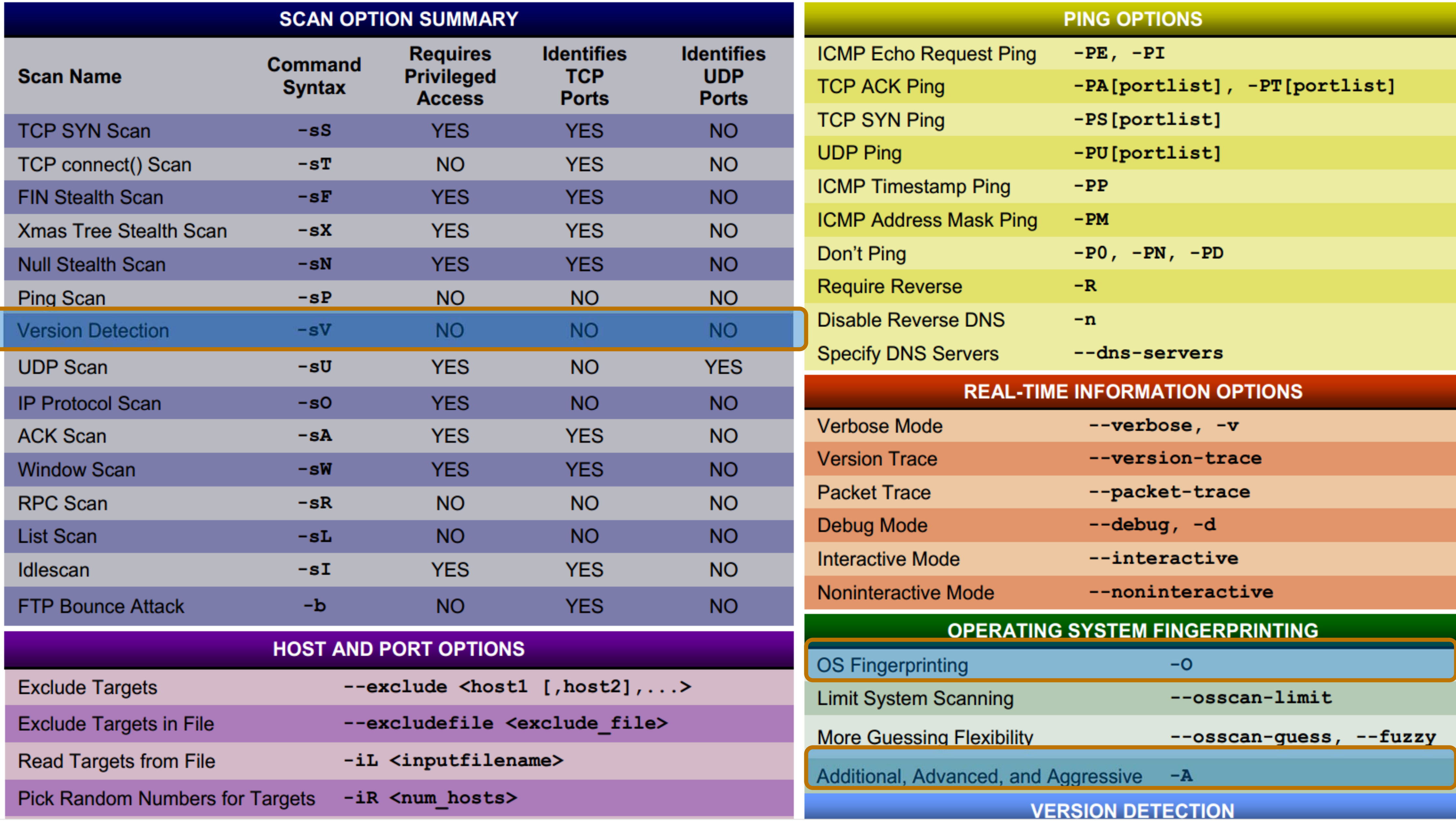
Switch	Example	Description
	nmap 192.168.1.1	Scan a single IP
	nmap 192.168.1.1 192.168.2.1	Scan specific IPs
	nmap 192.168.1.1-254	Scan a range
	nmap scanme.nmap.org	Scan a domain
	nmap 192.168.1.0/24	Scan using CIDR notation
-iL	nmap -iL targets.txt	Scan targets from a file
-iR	nmap -iR 100	Scan 100 random hosts
--exclude	nmap --exclude 192.168.1.1	Exclude listed hosts

Scan Techniques

Switch	Example	Description
-sS	nmap 192.168.1.1 -sS	TCP SYN port scan (Default)
-sT	nmap 192.168.1.1 -sT	TCP connect port scan (Default without root privilege)
-sU	nmap 192.168.1.1 -sU	UDP port scan
-sA	nmap 192.168.1.1 -sA	TCP ACK port scan
-sW	nmap 192.168.1.1 -sW	TCP Window port scan
-sM	nmap 192.168.1.1 -sM	TCP Maimon port scan

Host Discovery

Switch	Example	Description
-sL	nmap 192.168.1.1-3 -sL	No Scan. List targets only
-sn	nmap 192.168.1.1/24 -sn	Disable port scanning
-Pn	nmap 192.168.1.1-5 -Pn	Disable host discovery. Port scan only
-PS	nmap 192.168.1.1-5 -PS22-25,80	TCP SYN discovery on port x. Port 80 by default
-PA	nmap 192.168.1.1-5 -PA22-25,80	TCP ACK discovery on port x. Port 80 by default
-PU	nmap 192.168.1.1-5 -PU53	UDP discovery on port x. Port 40125 by default
-PR	nmap 192.168.1.1-1/24 -PR	ARP discovery on local network
-n	nmap 192.168.1.1 -n	Never do DNS resolution



SCAN OPTION SUMMARY

Scan Name	Command Syntax	Requires Privileged Access	Identifies TCP Ports	Identifies UDP Ports
TCP SYN Scan	`-sS`	YES	YES	NO
TCP connect() Scan	`-sT`	NO	YES	NO
FIN Stealth Scan	`-sF`	YES	YES	NO
Xmas Tree Stealth Scan	`-sX`	YES	YES	NO
Null Stealth Scan	`-sN`	YES	YES	NO
Ping Scan	`-sP`	NO	NO	NO
Version Detection	`-sV`	NO	NO	NO
UDP Scan	`-sU`	YES	NO	YES
IP Protocol Scan	`-sO`	YES	NO	NO
ACK Scan	`-sA`	YES	YES	NO
Window Scan	`-sW`	YES	YES	NO
RPC Scan	`-sR`	NO	NO	NO
List Scan	`-sL`	NO	NO	NO
Idlescan	`-sI`	YES	YES	NO
FTP Bounce Attack	`-b`	NO	YES	NO

HOST AND PORT OPTIONS

Exclude Targets	`--exclude <host1 [,host2],...>`
Exclude Targets in File	`--excludefile <exclude_file>`
Read Targets from File	`-iL <inputfilename>`
Pick Random Numbers for Targets	`-iR <num_hosts>`

PING OPTIONS

ICMP Echo Request Ping	`-PE, -PI`
TCP ACK Ping	`-PA[portlist], -PT[portlist]`
TCP SYN Ping	`-PS[portlist]`
UDP Ping	`-PU[portlist]`
ICMP Timestamp Ping	`-PP`
ICMP Address Mask Ping	`-PM`
Don't Ping	`-P0, -PN, -PD`
Require Reverse	`-R`
Disable Reverse DNS	`-n`
Specify DNS Servers	`--dns-servers`

REAL-TIME INFORMATION OPTIONS

Verbose Mode	`--verbose, -v`
Version Trace	`--version-trace`
Packet Trace	`--packet-trace`
Debug Mode	`--debug, -d`
Interactive Mode	`--interactive`
Noninteractive Mode	`--noninteractive`

OPERATING SYSTEM FINGERPRINTING

OS Fingerprinting	`-O`
Limit System Scanning	`--osscan-limit`
More Guessing Flexibility	`--osscan-guess, --fuzzy`
Additional, Advanced, and Aggressive	`-A`

VERSION DETECTION

THANKS