Wireshark : Traffic Analysis

NMAP Scans

Nmap is an industry-standard tool for mapping network identifying live hosts and discovering the services.

Most common Nmap scan types:
- TCP connect scans (Transmission Control Protocol) → Data Transfer (uses TLS)
- SYN scans (Synchronize) → SYN, SYN-ACK, ACK
- UDP scans (User Datagram Protocol) connectionless (rely on fast queries, real time)

What are TCP flags
- In every TCP packet there is a field bit that say what kind of control message it is
- SYN: start a connection
- ACK: acknowledge received data
- RST: reset or abort a connection
- FIN: politely end a connection (normal close)

Wireshark's tcp.flags is often treat as an integer bitmask

FIN = 1 SYN = 2 RST = 4 PSH = 8 ACK = 16
URG = 32

PSH: tells the receiver (don't wait to fill buffers - deliver to data as soon as possible)

Basically, this is to deliver to app now

URG (Urgent): indicating there is urgent data in the stream that the receiver should treat specially

So:

- tcp.flags == 2 means only SYN (2)
- tcp.flags == 16 means only ACK (16)
- tcp.flags == 18 means SYN+ACK (2+16)
- tcp.flags == 20 means RST+ACK (4+16=20)

But, if we want set (every TCP packet where the SYN Bit is set)

Use: - tcp.flags.syn == 1 (one is indicating Boolean)

This will apply all of them

- tcp.flags.ack == 1
- tcp.flags.reset == 1
- tcp.flags.fin == 1

If we want only SYN+ACK packet:

- tcp.flags.syn == 1 && tcp.flags.ack

and so on.

TCP Connect Scans

TCP Connect scans in nutshells:

- Relies on three way handshake
- Usually conduct with nmap -sT
- Used by non-priviledged users
- Usually has a windows size larger than 1024 bytes as the request expect some data due to the nature of the protocol

which means:

- In TCP, each side advertises a receive window
  - bytes we are allowed to send before we must stop and wait for ACK
- For some protocols, when we make a request, we expect a non-trivial response (a chunk of data)
- Because of that, the requester or receiver will typically advertise a bigger receive window to avoid slowing things down

Open TCP ports

- SYN →
- ← SYN, ACK
- ACK →

Open TCP port

- SYN →
- ← SYN, ACK
- ACK →
- RST, ACK →

Closed TCP port

- SYN →
- ← RST, ACK

SYN scans

TCP SYN scan in a nutshell:

- Doesn't rely on the three way handshake
- Usually conduct with nmap -sS
- Used by priviledged users
- Usually has a size less than or equal to 1024 bytes as the request is not finished and it doesn't expect to receive data

Open TCP port	Close TCP port
• SYN →	SYN →
• ← SYN, ACK	← RST, ACK
• RST →	

UDP Scans

UDP scan in a nutshell:

- Doesn't require a handshake process
- No prompt for open ports
- ICMP error message for close ports
- Usually conducted with nmap -sU

Open UDP port	Closed UDP Port
- UDP packet →	• UDP packet →
	• ICMP Type 3, Code 3 message

UDP scan patters in a capture file.

icmp.type == 3 and icmp.code == 3

TCP SYN scan patterns in capture file

tcp.flags.syn == 1 and tcp.flags.ack == 0 and tcp.window_size <= 1024

Same but tcp.window-size > 1024 in TCP Connect Scan

Question:

- what is the numbes of the TCP Connect scans?

tcp.flags.syn == 1 and tcp.flags.ack == 0 and tcp.window-size > 1024

- Which scan type is used to scan the ~~port~~ TCP port 80?

tcp.port == 80 (and followed the pattern)

-

- How many 'UDP close port' messages are there?

icmp.type == 3 and icmp.code == 3

- which UDP port in the 55-70 port range is open?

udp && udp.dstport >= 55 && udp.dstport <= 70

67 and 69 were closed