

THM_Wireshark 101 [Task 9 TCP]

I learned that RST packets indicate a closed port and that one should look at a stream of TCP packets, not a few.

TCP or Transmission Control Protocol handles the delivery of packets including sequencing and errors. You should already have an understanding of how TCP works, if you need a refresher check out the [IETF TCP Documentation](#).

Below you can see a sample of a Nmap scan, scanning port 80 and 443. We can tell that the port is closed due to the RST, ACK packet in red.

53	38.899808	192.168.227.128	192.168.227.131	TCP	74	47800 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=749056 TSecr=0 WS=128
54	38.899873	192.168.227.128	0.0.0.80	TCP	74	35032 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1615245101 TSecr=0 WS=128
55	38.899907	192.168.227.128	192.168.227.131	TCP	74	48720 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=749056 TSecr=0 WS=128
56	38.899938	192.168.227.128	0.0.0.80	TCP	74	34510 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1615245101 TSecr=0 WS=128
57	38.899940	192.168.227.131	192.168.227.128	TCP	60	80 → 47800 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
58	38.899971	192.168.227.131	192.168.227.128	TCP	60	443 → 48720 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

For analyzing TCP packets we will not go into the details of each individual detail of the packets; however, look at a few of the behaviors and structures that the packets have.

Below we see packet details for an SYN packet. The main thing that we want to look for when looking at a TCP packet is the sequence number and acknowledgment number.

Wireshark · Packet 53 · VMware Network Adapter VMnet8

- > Frame 53: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{16998130-78EE-4040-89D6-92BC3748DE1F}, id 0
- > Ethernet II, Src: VMware_d3:93:f5 (00:0c:29:d3:93:f5), Dst: VMware_bb:69:77 (00:0c:29:bb:69:77)
- > Internet Protocol Version 4, Src: 192.168.227.128, Dst: 192.168.227.131
- ▼ Transmission Control Protocol, Src Port: 47800, Dst Port: 80, Seq: 0, Len: 0
 - Source Port: 47800
 - Destination Port: 80
 - [Stream index: 1]
 - [TCP Segment Len: 0]
 - Sequence number: 0 (relative sequence number)
 - Sequence number (raw): 238988457
 - [Next sequence number: 1 (relative sequence number)]
 - Acknowledgment number: 0
 - Acknowledgment number (raw): 0
 - 1010 = Header Length: 40 bytes (10)
 - > Flags: 0x002 (SYN)
 - Window size value: 64240
 - [Calculated window size: 64240]
 - Checksum: 0x20be [unverified]
 - [Checksum Status: Unverified]
 - Urgent pointer: 0
 - > Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
 - > [Timestamps]

In this case, we see that the port was not open because the acknowledgment number is 0.

Within Wireshark, we can also see the original sequence number by navigating to edit > preferences > protocols > TCP > relative sequence numbers (uncheck boxes).

Wireshark · Preferences

- Steam IHS Di ^
- STP
- STT
- STUN
- SUA
- SV
- SYNC
- SYNCHROPH
- Synergy
- Syslog
- T.38
- TACACS
- TACACS+
- TALI
- TAPA
- TCAP
- TCP
- TCPENCAP
- TCPROS
- TDMoE
- TDMoP
- TDS
- TeamSpeak2
- TELNET

Transmission Control Protocol

- ☒ Show TCP summary in protocol tree
- ☐ Validate the TCP checksum if possible
- ☒ Allow subdissector to reassemble TCP streams
- ☐ Reassemble out-of-order segments
- ☒ Analyze TCP sequence numbers
- ☐ Relative sequence numbers (Requires "Analyze TCP sequence numbers")
- Scaling factor to use when not available from capture: Not known
- ☒ Track number of bytes in flight
- ☒ Calculate conversation timestamps
- ☐ Try heuristic sub-dissectors first
- ☐ Ignore TCP Timestamps in summary
- ☒ Do not call subdissectors for error packets
- ☒ TCP Experimental Options with a Magic Number
- ☐ Display process information via IPFIX
- TCP UDP port: 0

OK Cancel Help

```
> Frame 53: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{16998130-78EE-4040-89D6-92BC3748DE1F}, id 0
> Ethernet II, Src: VMware_d3:93:f5 (00:0c:29:d3:93:f5), Dst: VMware_bb:69:77 (00:0c:29:bb:69:77)
> Internet Protocol Version 4, Src: 192.168.227.128, Dst: 192.168.227.131
▼ Transmission Control Protocol, Src Port: 47800, Dst Port: 80, Seq: 238988457, Len: 0
    Source Port: 47800
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 238988457
    [Next sequence number: 238988458]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    1010 .... = Header Length: 40 bytes (10)
> Flags: 0x002 (SYN)
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0x20be [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
> Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
> [Timestamps]
```

Typically TCP packets need to be looked at as a whole to tell a story rather than one by one at the details.

Answer the questions below

Read the above and move into Task 10.

Boohoo!