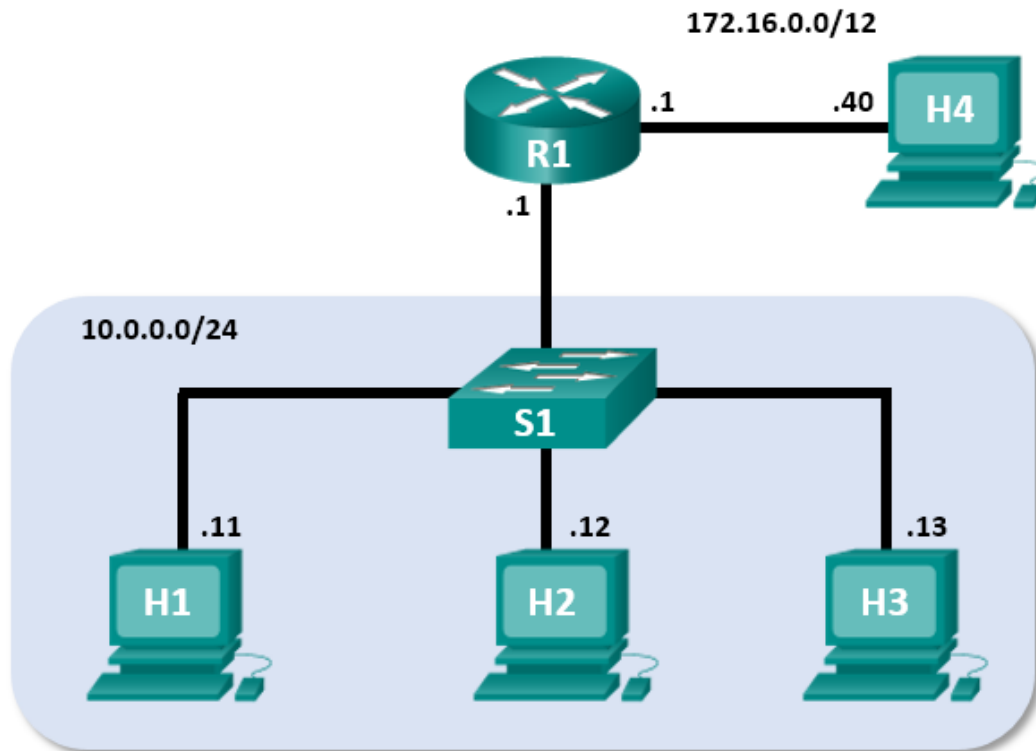# Lab - Using Wireshark to Observe the TCP 3-Way Handshake

Mininet Topology

Objectives

Part 1: Prepare the Hosts to Capture the Traffic

Part 2: Analyze the Packets using Wireshark

Part 3: View the Packets using tcpdump

Background / Scenario

In this lab, you will use Wireshark to capture and examine packets generated between the PC browser using the HyperText Transfer Protocol (HTTP) and a web server, such as www.google.com. When an application, such as HTTP or File Transfer Protocol (FTP) first starts on a host, TCP uses the three-way handshake to establish a reliable TCP session between the two hosts. For example, when a PC uses a web browser to surf the Internet, a three-way handshake is initiated, and a session is established between the PC host and web server. A PC can have multiple, simultaneous, active TCP sessions with various web sites.

Required Resources

- CyberOps Workstation Virtual Machine
- Internet access

Part 1: Prepare the Hosts to Capture the Traffic

a. Start the CyberOps VM. Log in with username **analyst** and the password **cyberops**.

b. Start Mininet.

```
[analyst@secOps ~]$ sudo lab.support.files/scripts/cyberops_topo.py
```

c. Start host H1 and H4 in Mininet.

```
*** Starting CLI:
mininet> xterm H1
mininet> xterm H4
```

d. Start the web server on H4.

```
[root@secOps analyst]#
/home/analyst/lab.support.files/scripts/reg_server_start.sh
```

e. Start the web browser on H1. This will take a few moments.

```
[root@secOps analyst]# firefox &
```

f. After the Firefox window opens, start a tcpdump session in the terminal **Node: H1** and send the output to a file called **capture.pcap**. With the -v option, you can watch the progress. This capture will stop after capturing 50 packets, as it is configured with the option -c 50.

```
[root@secOps analyst]# tcpdump -i H1-eth0 -v -c 50 -w
/home/analyst/capture.pcap
```

g. After the tcpdump starts, quickly navigate to 172.16.0.40 in the Firefox web browser.

Part 2: Analyze the Packets using Wireshark

Step 1: Apply a filter to the saved capture.

a. Press ENTER to see the prompt. Start Wireshark on **Node: H1**. Click **OK** when prompted by the warning regarding running Wireshark as superuser.

```
[root@secOps analyst]# wireshark-gtk &
```

b. In Wireshark, click **File** > **Open**. Select the saved pcap file located at /home/analyst/capture.pcap.

c. Apply a **tcp** filter to the capture. In this example, the first 3 frames are the interested traffic.

Filter:	tcp				▼	Expression...	Clear	Apply	Save
No.	**Time**	**Source**	**Destination**	**Protocol**	**Length**	**Info**			
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PER			
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=14			
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645			
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1			

Step 2: Examine the information within packets including IP addresses, TCP port numbers, and TCP control flags.

a. In this example, frame 1 is the start of the three-way handshake between the PC and the server on H4. In the packet list pane (top section of the main window), select the first packet, if necessary.

b. Click the **arrow** to the left of the Transmission Control Protocol in the packet details pane to expand the and examine the TCP information. Locate the source and destination port information.

c. Click the **arrow** to the left of the Flags. A value of 1 means that flag is set. Locate the flag that is set in this packet.

Note: You may have to adjust the top and middle windows sizes within Wireshark to display the necessary information.

```
No.   Time        Source        Destination   Protocol  Length  Info
   1 0.000000    10.0.0.11     172.16.0.40   TCP           74 58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PEF
   2 0.000081    172.16.0.40   10.0.0.11     TCP           74 80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=14
   3 0.000082    10.0.0.11     172.16.0.40   TCP           66 58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
   4 0.000194    10.0.0.11     172.16.0.40   HTTP         356 GET /favicon.ico HTTP/1.1

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▶ Ethernet II, Src: a6:a1:15:2c:d8:de (a6:a1:15:2c:d8:de), Dst: a2:86:17:7c:c3:65 (a2:86:17:7c:c3:65)
▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40
▼ Transmission Control Protocol, Src Port: 58716, Dst Port: 80, Seq: 0, Len: 0
      Source Port: 58716
      Destination Port: 80
      [Stream index: 0]
      [TCP Segment Len: 0]
      Sequence number: 0    (relative sequence number)
      Acknowledgment number: 0
      Header Length: 40 bytes
    ▶ Flags: 0x002 (SYN)
      Window size value: 29200
      [Calculated window size: 29200]
      Checksum: 0xb671 [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
    ▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
```

What is the TCP source port number? _____

How would you classify the source port? _____

What is the TCP destination port number? _____

How would you classify the destination port? _____

Which flag (or flags) is set? _____

What is the relative sequence number set to? _____

d. Select the next packet in the three-way handshake. In this example, this is frame 2. This is the web server replying to the initial request to start a session.

```
Filter:  tcp                                      ▼  Expression...  Clear   Apply    Save

No.   Time        Source        Destination   Protocol  Length  Info
   1 0.000000    10.0.0.11     172.16.0.40   TCP           74 58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PEF
   2 0.000081    172.16.0.40   10.0.0.11     TCP           74 80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=14
   3 0.000082    10.0.0.11     172.16.0.40   TCP           66 58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
   4 0.000194    10.0.0.11     172.16.0.40   HTTP         356 GET /favicon.ico HTTP/1.1

▶ Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▶ Ethernet II, Src: a2:86:17:7c:c3:65 (a2:86:17:7c:c3:65), Dst: a6:a1:15:2c:d8:de (a6:a1:15:2c:d8:de)
▶ Internet Protocol Version 4, Src: 172.16.0.40, Dst: 10.0.0.11
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 58716, Seq: 0, Ack: 1, Len: 0
      Source Port: 80
      Destination Port: 58716
      [Stream index: 0]
      [TCP Segment Len: 0]
      Sequence number: 0    (relative sequence number)
      Acknowledgment number: 1    (relative ack number)
      Header Length: 40 bytes
    ▶ Flags: 0x012 (SYN, ACK)
      Window size value: 28960
      [Calculated window size: 28960]
      Checksum: 0xc85a [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
    ▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
```
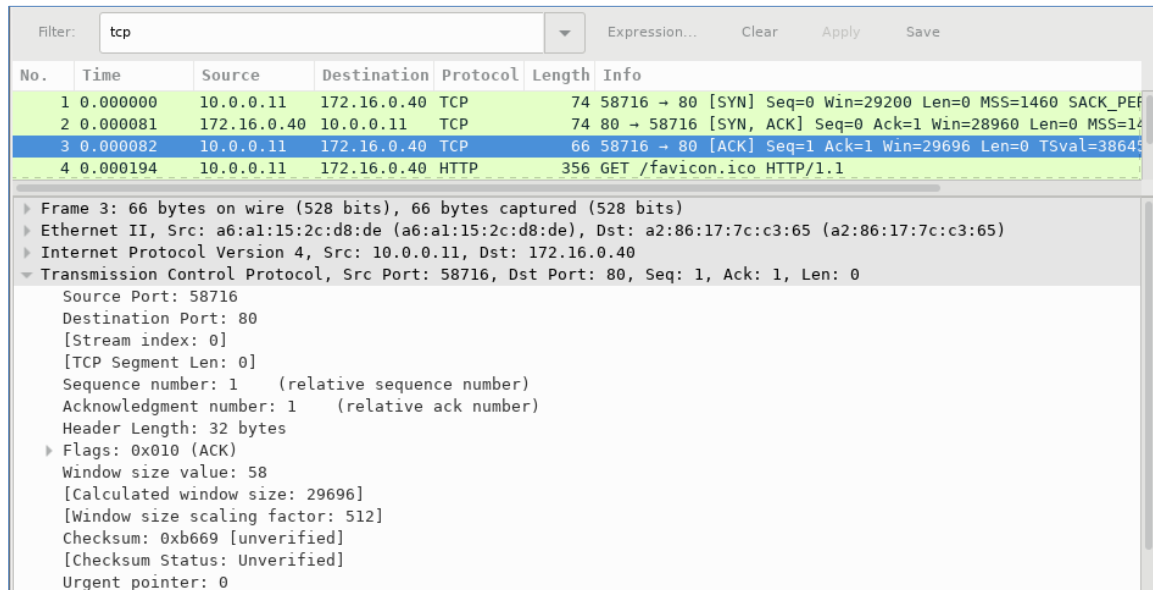
What are the values of the source and destination ports? _____

Which flags are set? _____

What are the relative sequence and acknowledgment numbers set to?

_____

e. Finally, select the third packet in the three-way handshake.

Examine the third and final packet of the handshake.

Which flag (or flags) is set? _____

The relative sequence and acknowledgment numbers are set to 1 as a starting point. The TCP connection is established and communication between the source computer and the web server can begin.

Part 3: View the packets using tcpdump

You can also view the pcap file and filter for the desired information.

a. Open a new terminal window, enter **man tcpdump**. **Note**: You may need to press ENTER to see the prompt.

Using the manual pages available with the Linux operating system, you read or search through the manual pages for options for selecting the desired information from the pcap file.

```
[analyst@secOps ~]# man tcpdump
TCPDUMP(1)                    General Commands Manual                    TCPDUMP(1)


NAME
       tcpdump - dump traffic on a network


SYNOPSIS
       tcpdump [ -AbdDefhHIJKlLnNOpqStuUvxX# ] [ -B buffer_size ]
               [ -c count ]
               [ -C file_size ] [ -G rotate_seconds ] [ -F file ]
               [ -i interface ] [ -j tstamp_type ] [ -m module ] [ -M secret ]
```

```
                      [ --number ] [ -Q in|out|inout ]
                      [ -r file ] [ -V file ] [ -s snaplen ] [ -T type ] [ -w file ]
                      [ -W filecount ]
                      [ -E spi@ipaddr algo:secret,...  ]
                      [ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]
                      [ --time-stamp-precision=tstamp_precision ]
                      [ --immediate-mode ] [ --version ]
                      [ expression ]
```

`<some output omitted>`

To search through the man pages, you can use / (searching forward) or ? (searching backward) to find specific terms, and **n** to forward to the next match and **q** to quit. For example, search for the information on the switch -r, type **/-r**. Type **n** to move to the next match. What does the switch **-r** do?

_____

_____

_____

b. In the same terminal, open the capture file using the following command to view the first 3 TCP packets captured:

```
[analyst@secOps ~]# tcpdump -r /home/analyst/capture.pcap tcp -c 3
reading from file capture.pcap, link-type EN10MB (Ethernet)
13:58:30.647462 IP 10.0.0.11.58716 > 172.16.0.40.http: Flags [S], seq
2432755549, win 29200, options [mss 1460,sackOK,TS val 3864513189 ecr
0,nop,wscale 9], length 0
13:58:30.647543 IP 172.16.0.40.http > 10.0.0.11.58716: Flags [S.], seq
1766419191, ack 2432755550, win 28960, options [mss 1460,sackOK,TS val
50557410 ecr 3864513189,nop,wscale 9], length 0
13:58:30.647544 IP 10.0.0.11.58716 > 172.16.0.40.http: Flags [.], ack 1, win
58, options [nop,nop,TS val 3864513189 ecr 50557410], length 0
```

To view the 3-way handshake, you may need to increase the number of lines after the **-c** option.

c. Navigate to the terminal used to start Mininet. Terminate the Mininet by entering quit in the main CyberOps VM terminal window.

```
mininet> quit
*** Stopping 0 controllers

*** Stopping 2 terms
*** Stopping 5 links
.....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
[analyst@secOps ~]$
```

d. After quitting Mininet, enter **sudo mn -c** to clean up the processes started by Mininet. Enter the password **cyberops** when prompted.

```
[analyst@secOps scripts]$ sudo mn -c
[sudo] password for analyst:
```

Reflection

1. There are hundreds of filters available in Wireshark. A large network could have numerous filters and many different types of traffic. List three filters that might be useful to a network administrator.

   _____

   _____

   _____

2. What other ways could Wireshark be used in a production network?

   _____

   _____

   _____