Lab 3: Packet Inspection

Aim:
The aim of this lab is to provide a foundation in understanding Ethernet, IP and TCP, and to practise packet inspection using Wireshark and Snort.

Time to Complete:
4 hours (two supervised hours in the lab, and two additional unsupervised hours).

Activities:

- **Complete lab 3**: Practising packet inspection using Wireshark and Snort. Please feel free to choose Bodhi_Linux_Private or Windows_10_DMZ to carry out the practices below.
- All network trace files that you need for this lab are located in "**Week 4. Security Information and Event Management (SIEM)**" > "**Lab**" > "**Week 4 Network Traces**" on Moodle.

Ethernet, IP and TCP

L3.1 Download the following file webpage.zip from Moodle, and open it up in Wireshark.

In this case, a host connects to a Web server. Determine the following:

Host src IP address (Hint: Examine the Source IP on Packet 3):
Server src IP address (Hint: Examine the Destination IP on Packet 3):

Host src TCP port (Hint: Examine the Source Port on Packet 3):
Server src TCP port (Hint: Examine the Destination Port on Packet 3):

What is the MAC address of the server (Hint: Examine the reply for Packet 2), and **which is the manufacturer of the network card?**

What is the MAC address of the host contacting the server, and **which is the manufacturer of the network card?**

Identify the packets used for the SYN, SYN/ACK and ACK sequence. Which packets are these?

In Packet 1, which is the destination MAC address used in the ARP request?

Using the filter of `tcp.flags.syn==1`**, find all the packets that involve a SYN flag. What are their IDs?**

What does the filter of `tcp.flags.syn==1 && tcp.flags.ack==0` **do?**

What does the filter of `tcp.flags.syn==1 && tcp.flags.ack==1` **do?**

Which flags are set at the end of a connection?

L3.2 Download the following file googleWeb.zip from Moodle, and open it up in Wireshark

In this case, a host connects to the Google Web server (i.e., www.google.co.uk). Determine the following:

Hint: using the filter of `http.host contains "www.google.co.uk"` **to find Get requests sent from the Host to Google servers**

Host src IP address:
Server src IP address of the Web server:

Host src TCP port:
Server src TCP port:

Can you determine the MAC address of the server? If no, why you cannot?

What is the MAC address of the host contacting the server, and **which is the manufacturer of the network card?**

What is the IP address of the local gateway? (Hint: using the filter of `eth.src == 00:18:4d:b0:d6:8c && arp`**)**

What is the MAC address of the local gateway, and **which is the manufacturer of the network card?**

Identify the packets used for the SYN, SYN/ACK and ACK sequence. Which packets are these?

L3.3 Start capturing network packets on your main network adapter. Next go to intel.com, and access the page. Stop the network capture, and then from your network traffic, determine:

Your MAC address (and its manufacturer):
Your IP address:

The MAC address of the gateway:
The IP address of intel.com

The source TCP port of your connection:
The destination TCP port used by the server:

Apart from your network traffic, can you see other traffic from other hosts on the network? If so, which type of network traffic do you see?

HTTP, DNS and FTP

Aim: To provide a foundation in understanding HTTP, DNS and FTP.

L3.4 Download the following file webpage.zip from Moodle, and open it up in Wireshark.

In this case, a host connects to a Web server. Determine the following:

Using the filter of `http.request.method=="GET"`, and identifying the files that the host gets from the Web server:

Using the filter of `http.response`, and determining the response codes. Which files have transferred and which have been unsuccessful?

Which is the default file name on the server when the user accesses the top levels of the domain?

Which types of image files does the client want to accept?

Which language/character set is used by the client?

Which Web browser is the client using?

Which Web server technology is the server using?

On which date were the pages accessed?

L3.5 Download the following file, googleWeb.zip, from Moodle, and open it up in Wireshark.

In this case, a host connects to the Google Web server (i.e., www.google.co.uk). Determine the following:

Using the filter of `http.request.method=="GET" && http.host contains "www.google.co.uk"`, and identifying the files that the host gets from the Web server:

Using the filter of `http.response`, and determining the response codes. Which files have transferred and which have been unsuccessful?

Which is the default file name on the server when the user accesses the top levels of the domain?

Which types of image files does the client want to accept?

Which language/character set is used by the client?

Which Web browser is the client using?

Which Web server technology is the server using?

On which date were the pages accessed?

L3.6 Start capturing network packets on *your main network adapter*. Next go to intel.com, and access the page. Stop the network capture, and then from your network traffic, determine:

Using the filter of `http.request.method=="GET"`, and identifying the files that the host gets from the Web server:

Using the filter of `http.response`, and determining the response codes. Which files have transferred and which have been unsuccessful?

Which is the default file name on the server when the user accesses the top levels of the domain?

Which type of image files does the client want to accept?

Which language/character set is used by the client?

Which Web browser is the client using?

Which Web server technology is the server using?

L3.7 Download the following file, dnslookup.zip, from Moodle and open it up in Wireshark.

For this trace, determine the following:

Which is the domain being searched for?

Which are the IP addresses of the domain being searched for?

The first request is of class of PTR. What is the PTR?

The second request is of class for A. What is the A class?

The last request is for class of AAAA. What is the AAAA class?

Does the domain have an IPv6 address?

L3.8 Download the following file, ftp2.zip, from Moodle and open it up in Wireshark.

For this trace, determine the following:

Using the filter of `ftp.request.command`, and determining the FTP commands that the user has used:

Using the filter of `ftp.response.code`, and determining the FTP codes that have been returned:

What are the username and password for the access to the FTP server?
(Hint: using the filter of `ftp.response.code==230` to look for successful login)

What is the name of the file, which is uploaded?
(Hint: using the filter of `ftp.request.command == "STOR"`)

What is the name of the file, which is downloaded?
(Hint: using the filter of `ftp.request.command == "RETR"`)

Using the filter of `ftp.request.command=="LIST"`, and determining the first packet number which performs a "LIST":

In performing in this "LIST" of the files on the FTP server, which TCP port is used on the server for the transfer?

From the final "LIST" command, which are the files on the server?

What does the filter `ftp.response.code==227`, identify in terms of the ports that are used for the transfer?

ARP and ICMP

Aim: To provide a foundation in understanding ARP and ICMP.

L3.9 Download the following file, webpage.zip, from Moodle and open it up in Wireshark.

In this case, a host connects to a Web server. Determine the following:

By examining the ARP request and reply, what are the IP and MAC addresses of the server for the host?
(Hint: using the filter of `arp`)

Why does the host not go through a gateway?

L3.10 Download the following file, googleWeb.zip, from Moodle and open it up in Wireshark.

In this case, a host connects to the Google Web server (i.e., www.google.co.uk). Determine the following:

By examining the ARP request and reply, what are the IP and MAC addresses of the gateway for the host?
(Hint: using the filter of `arp`)

Can we determine the MAC address of the Google Web server?
(Hint: using the filter of `http.request.method == "GET" && http.host contains "www.google.co.uk"`. Then, examine the filtered packets and see if the destination MAC address is the gateway's address or the Google Web server's one)

L3.11 Download the following file, arp_scan.zip, from Moodle and open it up in Wireshark.

Determine the following:

This was generated by an intruder.

- **What can you say about the aim of the scan?**

- **What can say about whether this is an inside intruder or an external one?**

- **Which nodes did the intruder find where connected to the network?**

(Hint: using the filter of `arp.opcode ==2`)

SMTP, POP-3 and IMAP

Aim: To provide a foundation in understanding SNMP, POP-3 and IMAP.

L3.12 Download the following file, smtp.zip, from Moodle and open it up in Wireshark.

Determine the following:

The IP address and TCP port used by the host, which is sending the email:

(Hint: using the filter of `smtp.req.command`)

The IP address and the TCP port used by the SMTP server:
(Hint: using the filter of `smtp.response.code`)

Follow the TCP Stream of this connection to answer the following questions
- Who is sending the email?

- Who is receiving the email?

- When was the email sent?

- When was the email client used to send the email?

- What was the message, and what was the subject of the email?

- With SMTP, which character sequence is used to end the message?

L3.13 Download the following file, pop3.zip, from Moodle and open it up in Wireshark:

(Hints: POP protocol commands - https://tools.ietf.org/html/rfc1939)

Command	**Description**
USER [username]	1st login command
PASS [password]	2nd login command
QUIT	Logs out and saves any changes
STAT	Returns total number of messages and total size
LIST	Lists all messages
RETR [message]	Retrieves the whole message
DELE [message]	Deletes the specified message
NOOP	The POP3 server does nothing, it merely replies with a positive response.

RSET	Undelete the message if any marked for deletion
TOP [message] [number]	Returns the headers and number of lines from the message

Determine the following:

The IP address and TCP port used by the host, which is sending the email:
(Hint: using the filter of `pop.request.command`)

The IP address and the TCP port used by the POP-3 server:
(Hint: using the filter of `pop.response`)

Whose mail box is being accessed?
(Hints: using the filter of `imf`. Then, follow the "TCP stream" and look for the email address after the key word mailbox)

How many email messages are in the Inbox?

The messages are listed as:

1 5565
2 8412
3 xxxx

What is the ID for message 3?

For Message 1, who sent the message and what is the subject and outline the content of the message?

For Message 2, who sent the message and what is the subject and outline the content of the message?

For Message 3, who sent the message and what is the subject and outline the content of the message?

Which command does POP-3 use to get a specific message?

L3.14 Download the following file, imap.zip, from Moodle and open it up in Wireshark.

Simple Mail Transfer Protocol (SMTP) is the standard protocol for sending emails across the Internet.
Internet Message Access Protocol (IMAP) is a mail protocol used for accessing email on a remote web server from a local client. IMAP is used to retrieving emails.

Hints: IMAP commands - https://tools.ietf.org/html/rfc3501#section-8

Command	**Description**
<n> EXISTS	The number of messages in the mailbox.
<n> RECENT	The number of messages with the \Recent flag set.

Determine the following:

The IP address and TCP port used by the host, which is sending the email:
(Hint: using the filter of `imap.request.command || smtp.req.command`)

The IP address(es) and the TCP ports used by the SMTP and the IMAP server:
(Hint: using the filter of `imap.request.command || smtp.req.command`)

Whose mail box is being accessed?

How many email messages are in the Inbox?
(Hint: Follow the TCP stream of IMAP and read the EXISTS file.)

Trace the email message that has been sent for its basic details:

(Hint: Look at the SMTP)

Outline the details of email which are in the Inbox:
(Hint: Look at the IMAP)

Lab 3 (Part 2): Network Packet Analysis

A Find content using Wireshark

1. Using the following files from Moodle, perform searches and find the required content:

Obj	PCap file	Search filter	File name found
Find PNG	**with_png.zip**	`http contains "\x89\x50\x4E\x47"`	
Find PDF	**with_pdf.zip**	`http contains "%PDF"`	
Find GIF	**with_gif.zip**	`http contains "GIF89a"`	
Find ZIP	**with_zip.zip**	`http contains "\x50\x4B\x03\x04"`	
Find JPEG	**with_jpg.zip**	`http contains "\xff\xd8"`	
Find MP3	**with_mp3.zip**	`http contains "\x49\x44\x33"`	
Find RAR	**with_rar.zip**	`http contains "\x52\x61\x72\x21\x1A\x07\x00"`	
Find AVI	**with_avi.zip**	`http contains "\x52\x49\x46\x46"`	

2. Investigate the following files and display filters:

Obj	PCap file	Search filter	String that matches
Find Email Addresses	**email_cc2.zip**	`smtp matches "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9._%+-]"`	

Find and IP address	**webpage.zip**	`http matches "[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}.[0-9]{1,3}"`	
Find Credit card details	**email_cc2.zip**	`smtp matches "5\\d{3}(\\s|-)?\\d{4}(\\s|-)?\\d{4}(\\s|-)?\\d{4}"`	

3. The following file contains an FTP Hydra attack. Use a filter of: `ftp contains "530 User"` to investigate the following trace:

hydra_ftp.zip

Outline some of usernames and passwords which have been tried: Can you determine the username and password, which has been successful?

4. The following file contains an Telnet Hydra attack. Use a filter of: `telnet contains "unknown"` to investigate the following trace:

hydra_telnet.zip

Outline some of usernames and passwords which have been tried:

Can you determine the username and password, which has been successful?

5. The following file contains a TCP Syn flood DoS attack. Use a filter of: `tcp.flags.syn == 1 && tcp.flags.ack == 0` to investigate the following trace:

hping_syn.zip

Which is the IP address of the computer that is being attack?

Which is the IP address of the computer attacking?

B Snort analysis

Use your Windows_10_DMZ instance for this section. We can also use Snort to analyse network traces through an off-line filtering system. Download the following file from Moodle:

newtrace.zip

For this you can run Snort with a rules file and with a trace:

snort -c 1.rules -l log -r newtrace.pcap

You can then look in the log filter for the log file and alert.ids.

Some rules you can use are given in Appendix A.

Now test Snort to see if it can detect the same content that you found before:

Number of Bad FTP logins:

Number of Successful FTP logins:

Number of GIF files in the trace:

Number of PNG files in the trace:

Can you detect the port scan on a host?

C Snort analysis

Use the following PCAP files, and detect the activity:

Objective: Detect bad FTP login.

Trace: **hydra_ftp.zip.**

Rules used to detect:

Objective: Detect Telnet login.

Trace: **hydra_telnet.zip**.

Rules used to detect:

Objective: Detect port scan.

nmap.zip.

Rules used to detect:

Objective: Detect SYN flood.

hping_syn.zip.

Rules used to detect:

Objective: Detect FIN flood.

hping_fin.zip.

Rules used to detect:

Objective: Detect file attachments.

email_two_attachments.zip.

Rules used to detect:

Objective: Detect credit card details and email addresses.

email_cc2.zip.

Rules used to detect:

Objective: Detect ping sweep.

ping_sweep.zip.

Rules used to detect:

Can your extract the file and access it?

Objective: Detect PDF files.

with_pdf.zip.

Rules used to detect:

Can your extract the file and access it?

Objective: Detect SNMP connections

snmp.zip.

Rules used to detect:

Can your extract the file and access it?

Objective: Detect MP3 connections

with_mp3.zip

Rules used to detect:

Can your extract the files and access them?

What is the sound file and what are the graphics?

Objective: Detect and extract RAR files

with_rar.zip

Rules used to detect:

What is the name of the RAR file?

Can your extract the file and access it?

What are the contents of the file?

Objective: Detect and extract Zip files

with_zip.zip

Rules used to detect:

What is the name of the ZIP file?

Can your extract the file and access it?

Objective: Detect and extract GZip files

with_gzip.zip

Rules used to detect:

What is the name of the GZip file?

Can your extract the file and access it?

Objective: Detect and extract AVI files

with_avi.zip

Rules used to detect:

What is the name of the AVI file?

Can your extract the file and access it?

Objective: Detect BitTorrent

bit.zip

Rules used to detect:

Appendix A

Bad logins:

```
alert tcp any 21 -> any any (msg:"FTP Bad login"; content:"530 User "; nocase; flow:from_server,established; sid:491;
rev:5;)
```

Detecting email addresses:

```
alert tcp any any <> any 25 (pcre:"/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9._%+-]/"; \
msg:"Email in message";sid:9000000;rev:1;)
```

Detect DNS:

```
alert udp any any -> any 53 (msg: "DNS"; sid:10000;)
```

File types:

```
alert tcp any any -> any any (content:"GIF89a"; msg:"GIF";sid:10000)
alert tcp any any -> any any (content:"%PDF"; msg:"PDF";sid:10001)
alert tcp any any -> any any (content:"|89 50 4E 47|"; msg:"PNG";sid:10002)
alert tcp any any -> any any (content:"|50 4B 03 04|"; msg:"ZIP";sid:10003)
```

Telnet login:

```
alert tcp any any <> any 23 (flags:S; msg:"Telnet Login";sid:9000005;rev:1;)
```

Port scan:

```
preprocessor sfportscan:\
        proto { all } \
        scan_type { all } \
        sense_level { high } \
        logfile { portscan.log }
```

DoS on Web server:

```
alert tcp any any -> any 80 (msg:"DOS flood denial of service attempt";flow:to_server; \
detection_filter:track by_dst,  count 60, seconds 60; \
sid:25101; rev:1;)
```

Stealth scans:

```
alert tcp any any -> any any (msg:"SYN FIN Scan"; flags: SF;sid:9000000;)
alert tcp any any -> any any (msg:"FIN Scan"; flags: F;sid:9000001;)
alert tcp any any -> any any (msg:"NULL Scan"; flags: 0;sid:9000002;)
alert tcp any any -> any any (msg:"XMAS Scan"; flags: FPU;sid:9000003;)
alert tcp any any -> any any (msg:"Full XMAS Scan"; flags: SRAFPU;sid:9000004;)
alert tcp any any -> any any (msg:"URG Scan"; flags: U;sid:9000005;)
alert tcp any any -> any any (msg:"URG FIN Scan"; flags: FU;sid:9000006;)
alert tcp any any -> any any (msg:"PUSH FIN Scan"; flags: FP;sid:9000007;)
alert tcp any any -> any any (msg:"URG PUSH Scan"; flags: PU;sid:9000008;)
alert tcp any any -> any any (flags: A; ack: 0; msg:"NMAP TCP ping!";sid:9000009;)
```

ping sweep:

```
alert icmp any any -> any any (msg:"ICMP Packet found";sid:9000000;)
alert icmp any any -> any any (itype: 0; msg: "ICMP Echo Reply";sid:9000001;)
alert icmp any any -> any any (itype: 3; msg: "ICMP Destination Unreachable";sid:9000002;)
alert icmp any any -> any any (itype: 4; msg: "ICMP Source Quench Message received";sid:9000003;)
alert icmp any any -> any any (itype: 5; msg: "ICMP Redirect message";sid:9000004;)
alert icmp any any -> any any (itype: 8; msg: "ICMP Echo Request";sid:9000005;)
alert icmp any any -> any any (itype: 11; msg: "ICMP Time Exceeded";sid:9000006;)
```

Note you may have to add the following for the stream analysis:

```
preprocessor stream5_global: track_tcp yes, \
   track_udp yes, \
   track_icmp no, \
   max_tcp 262144, \
   max_udp 131072, \
   max_active_responses 2, \
   min_response_seconds 5
preprocessor stream5_tcp: policy windows, detect_anomalies, require_3whs 180, \
   overlap_limit 10, small_segments 3 bytes 150, timeout 180, \
    ports client 21 22 23 25 42 53 70 79 109 110 111 113 119 135 136 137 139 143 \
        161 445 513 514 587 593 691 1433 1521 1741 2100 3306 6070 6665 6666 6667 6668 6669 \
        7000 8181 32770 32771 32772 32773 32774 32775 32776 32777 32778 32779, \
    ports both 80 81 82 83 84 85 86 87 88 89 90 110 311 383 443 465 563 591 593 631 636 901 989 992 993 994 995 1220 1414 1830 2301 2381 2809 \
       3037 3057 3128 3443 3702 4343 4848 5250 6080 6988 7907 7000 7001 7144 7145 7510 7802 7777 7779 \
       7801 7900 7901 7902 7903 7904 7905 7906 7908 7909 7910 7911 7912 7913 7914 7915 7916 \
        7917 7918 7919 7920 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180 8222 8243 8280 8300 8500 8800 8888 8899 9000 9060 9080 9090 \
        9091 9443 9999 10000 11371 34443 34444 41080 50000 50002 55555
preprocessor stream5_udp: timeout 180
```