Andrew Ma

Lab 10

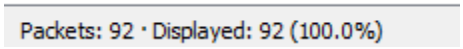
CPE 435

3/29/21

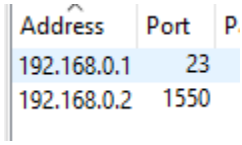
Observations and Answers

Subtask 1 (telnet-cooked.pcap)

1. How many packets are captured in the .pcap file that you loaded?
   92

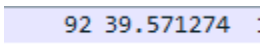


2. List all the communicating parties in the .pcap file? Can you also identify the ports being used by each of them?
   192.168.0.1 port 23, 192.168.0.2 port 1550

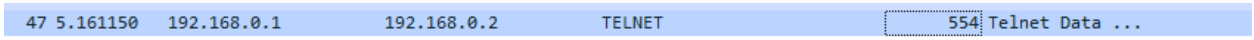


3. What protocols are used for communication by the communicating parties?
   Telnet (23), TCP (1550)

4. What is the total duration of the communication? (You may want to see the first and last frame)
   39.571274 seconds



5. What is the frame length and number of the longest frame transferred? Who is the source and destination of that packet?
   554 length, number 47
   Source 192.168.0.1, Dest 192.168.0.2



Subtask 2

6. Select frame number 8. Who is the sender and receiver of this frame?
Sender: 192.168.0.1, Receiver: 192.168.0.2

7. On the window that appears below the listing of all the frames (as shown below), expand Internet Protocol Version 4. What is the Time To Live of this frame? What does this mean?
The Time To Live is 64, and it means

The Time To Live or hop limit is a mechanism which limits the lifespan or lifetime of data in a computer or network. It is a counter, and once the count goes to 0, the data is discarded or revalidated, and this is to prevent a data packet from circulating indefinitely.

```
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x10 (DSCP: Unknown,
    Total Length: 52
    Identification: 0x5fca (24522)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 64
```

8. Select frame 8 again. Right click on it, and select Follow TCP Stream. What information can you see? What is the username and password that is transferred?

```
.......... ..!.."..'.....#..%..%.......... ..!.."."........P. ...."".....b........b....
B.
......................."......'.....#..&..&..$..&..&..$.. .....#.....'.......... .
9600,9600....#.bam.zing.org:0.0....'..DISPLAY.bam.zing.org:0.0......xterm-
color.............!.............".............
OpenBSD/i386 (oof) (ttyp2)

login: fake
......Password:user

......Last login: Sat Nov 27 20:11:43 on ttyp2 from bam.zing.org
Warning: no Kerberos tickets issued.
OpenBSD 2.6-beta (OOF) #4: Tue Oct 12 20:42:32 CDT 1999

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

$ /sbin/ping www.yahoo.com
PING www.yahoo.com (204.71.200.67): 56 data bytes
64 bytes from 204.71.200.67: icmp_seq=0 ttl=241 time=69.885 ms
64 bytes from 204.71.200.67: icmp_seq=1 ttl=241 time=73.591 ms
64 bytes from 204.71.200.67: icmp_seq=2 ttl=241 time=72.302 ms
64 bytes from 204.71.200.67: icmp_seq=3 ttl=241 time=73.493 ms
64 bytes from 204.71.200.67: icmp_seq=4 ttl=241 time=75.068 ms
64 bytes from 204.71.200.67: icmp_seq=5 ttl=241 time=70.239 ms
..........
.--- www.yahoo.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 69.885/72.429/75.068 ms
$ ls
$ ls -a
.        ..       .cshrc   .login   .mailrc  .profile .rhosts
$ exit
```

I see the commands running over telnet and their outputs.
The username is “fake”, and password is “user”

9. Repeat the same procedure in telnet-raw.pcap. Find the login information used to verify credentials. (Select frame 8 again)

```
.......... ..!.."..'.....#..%..%.......... ..!.."..".........P.
...."".....b........b....          B.
.......................".....'.....#..&..&..$..&..&..$.. .....#.....'..........
.9600,9600....#.bam.zing.org:0.0....'..DISPLAY.bam.zing.org:0.0......xterm-
color.............!..............".............
OpenBSD/i386 (oof) (ttyp1)

login: .."........"ffaakkee
.
Password:user
.
Last login: Thu Dec  2 21:32:59 on ttyp1 from bam.zing.org
Warning: no Kerberos tickets issued.
OpenBSD 2.6-beta (OOF) #4: Tue Oct 12 20:42:32 CDT 1999

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

$ llss
.
$ llss  --aa
.
.          ..         .cshrc     .login     .mailrc    .profile   .rhosts
$ //ssbbiinn//ppiinngg  wwwwww..yyaahhoooo..ccoomm
.
PING www.yahoo.com (204.71.200.74): 56 data bytes
64 bytes from 204.71.200.74: icmp_seq=0 ttl=239 time=73.569 ms
64 bytes from 204.71.200.74: icmp_seq=1 ttl=239 time=71.099 ms
64 bytes from 204.71.200.74: icmp_seq=2 ttl=239 time=68.728 ms
64 bytes from 204.71.200.74: icmp_seq=3 ttl=239 time=73.122 ms
64 bytes from 204.71.200.74: icmp_seq=4 ttl=239 time=71.276 ms
64 bytes from 204.71.200.74: icmp_seq=5 ttl=239 time=75.831 ms
64 bytes from 204.71.200.74: icmp_seq=6 ttl=239 time=70.101 ms
64 bytes from 204.71.200.74: icmp_seq=7 ttl=239 time=74.528 ms
64 bytes from 204.71.200.74: icmp_seq=9 ttl=239 time=74.514 ms
64 bytes from 204.71.200.74: icmp_seq=10 ttl=239 time=75.188 ms
64 bytes from 204.71.200.74: icmp_seq=11 ttl=239 time=72.925 ms
...^C
.--- www.yahoo.com ping statistics ---
13 packets transmitted, 11 packets received, 15% packet loss
round-trip min/avg/max = 68.728/72.807/75.831 ms
$ eexxiitt
.
```

The username is “fake”, and the password is “user”

10. What do you think is wrong with these two files that you analyzed? How can you not allow anyone to know your password that you send for authentication?

The problem is all the data including the login credentials are sent over plaintext and not over an encrypted tunnel. Someone who is a MITM can capture the plaintext data. You can't allow anyone to know your password because they can login as you and this violates confidentiality and nonrepudiation.

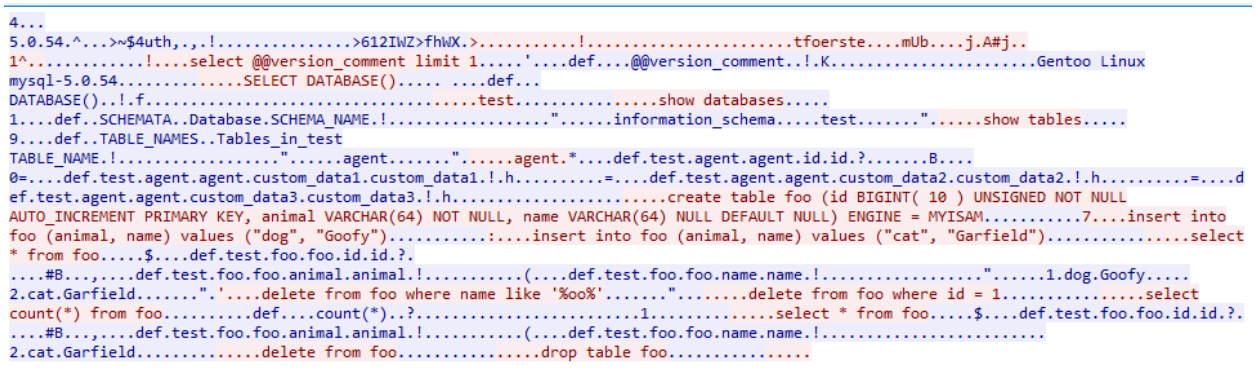
11. Load the file uftp_v3_transfer.pcapng. The protocol used is UFTP. What is UFTP? Can you identify two parties that are involved in file transfer. (Use your intelligent guessing)
UFTP is an encrypted multicast file transfer program, designed to securely, reliably, and efficiently transfer files to multiple receivers simultaneously. UFTP works over the UDP protocol.
The 2 parties are 10.0.0.1 and 230.5.5.25.

12. Write differences between TCP and UDP.
TCP establishes connection between sender and receiver with a 3 way handshake, and it is for reliable data transport. UDP is connectionless and the sender just sends and hopes the receiver gets the data. There is error checking and recovery (ensuring data is in order and not missing data) for TCP because there is a connection, but none for UDP. Because UDP is connectionless and doesn't do error checking and recovery, it is faster than TCP.

Subtask 3

12. What is the difference between https:// and http://? What is the encryption standard used by them, if any?
HTTPS is Hypertext Transfer Protocol Secure and encrypts data while http only sends plaintext data. HTTPS is used to secure data over HTTP using SSL or TLS.

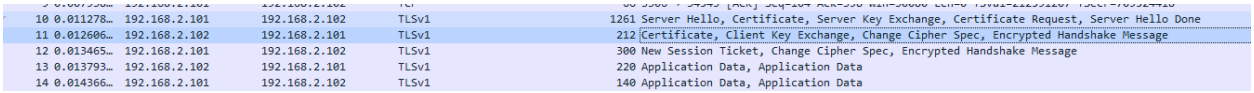
13. Download the file mysql_complete_pcap. Is it encrypted? Please justify.
No it is not encrypted.



Here we can see the plaintext SQL commands.

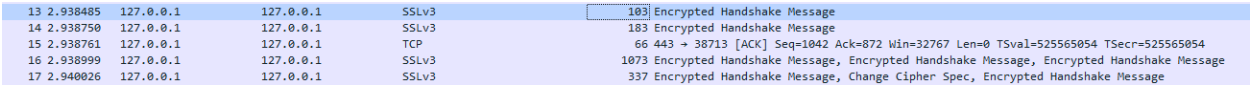
14. Download the file mysql-ssl-larger.pcapng. Is it encrypted? Please justify.
Yes it is encrypted, and the data is not decipherable.

It uses TLS to encrypt the message.



10	0.011278…	192.168.2.101	192.168.2.102	TLSv1	1261	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
11	0.012606…	192.168.2.102	192.168.2.101	TLSv1	212	Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	0.013465…	192.168.2.101	192.168.2.102	TLSv1	300	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
13	0.013793…	192.168.2.102	192.168.2.101	TLSv1	220	Application Data, Application Data
14	0.014366…	192.168.2.101	192.168.2.102	TLSv1	140	Application Data, Application Data

15. Download the zipped file snakeoil2_070531.tgz. Extract the content in your local folder. Load the .pcap file in wireshark. Is it encrypted?
Yes, it is encrypted with SSLv3.



13	2.938485	127.0.0.1	127.0.0.1	SSLv3	103	Encrypted Handshake Message
14	2.938750	127.0.0.1	127.0.0.1	SSLv3	183	Encrypted Handshake Message
15	2.938761	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1042 Ack=872 Win=32767 Len=0 TSval=525565054 TSecr=525565054
16	2.938999	127.0.0.1	127.0.0.1	SSLv3	1073	Encrypted Handshake Message, Encrypted Handshake Message, Encrypted Handshake Message
17	2.940026	127.0.0.1	127.0.0.1	SSLv3	337	Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message

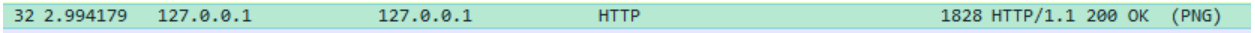
16. Perform the decryption of the .pcap file as demonstrated in class by the instructor.
    1. What frame number requests the image apache_pb.png?

31	2.993840	127.0.0.1	127.0.0.1	HTTP	471	GET /icons/apache_pb.png HTTP/1.1

Frame 31

    2. Does the server provide the image? What is the status code that implies the response has payload?
    This is apache_pb.png

The response for the GET /icons/apache_pb.png has status code 200, so it was successful.



32	2.994179	127.0.0.1	127.0.0.1	HTTP	1828	HTTP/1.1 200 OK (PNG)

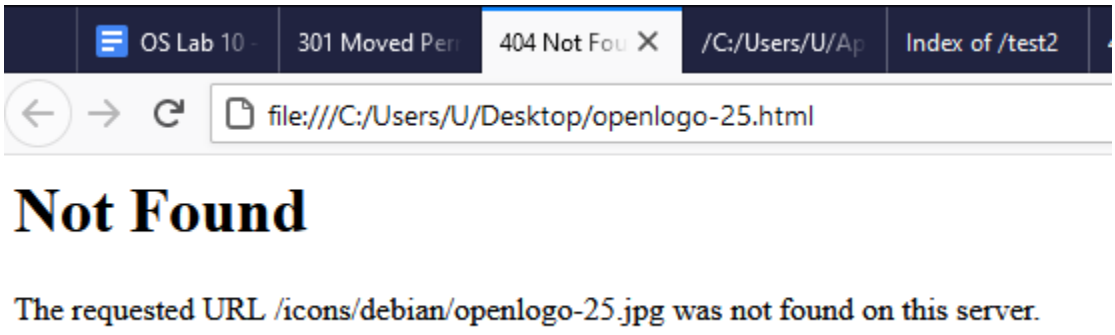
    3. Attach the image apache_pb.png to your report.

17. What is the response that the server provided when requested for openlogo-25.jpg?

29	2.992855	127.0.0.1	127.0.0.1	HTTP	562 GET /icons/debian/openlogo-25.jpg HTTP/1.1
30	2.993501	127.0.0.1	127.0.0.1	HTTP	596 HTTP/1.1 404 Not Found (text/html)

The response was 404 Not Found.

Can you see the html code sent as a response? If yes, copy and paste it in a .html file and load it in your favourite browser. Attach the screenshot of how the response looks like in the web browser.



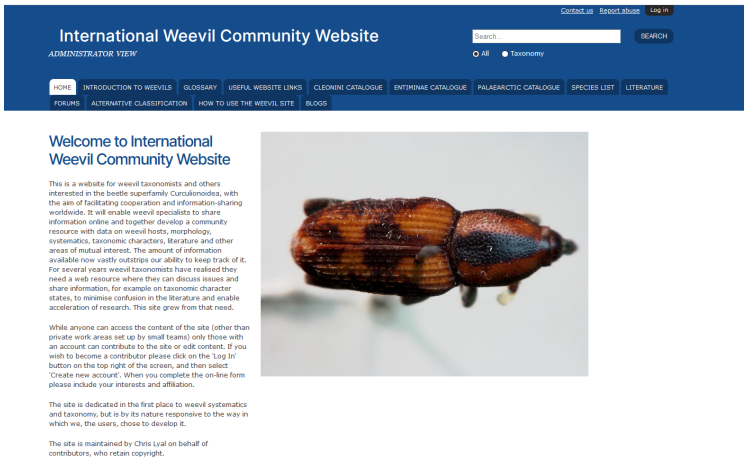
Subtask 4

18. The first thing that you will do is capture packets. You can use wireshark or tcpdump to capture packets. While you can capture packets from wireshark, I suggest you to use tcpdump so that you can be familiar with a new tool. Following are the procedures that you will follow:

1. Find the interface that is connected to the internet. Do ifconfig in the terminal and select the one which is connected to the internet. Wireshark should show you the interface in its GUI.

2. Start packet capture in tcpdump using tcpdump -i <interface> -s 65535 -w <filename>. Or select the bluefin below File menu in wireshark after you select the interface if you wish to use wireshark.

3. Please visit the website http://weevil.info/ What is wrong with this website?



The problem with this website is that it doesn't use HTTPS, just HTTP which sends data in plaintext.

4. After it is completely loaded, stop the capture. You can select the button in Wireshark GUI or kill the tcpdump process if you are using tcpdump.

5. Load the file in wireshark. If you are using wireshark, it is already loaded.

6. Try to find at least two images that are sent by the server to your machine and attach them to your report.

Pseudopoophagus%20longipes%20dorsal_0.jpg
Rhabdoscelus%20obscurus%20Australia%20male%20dorsal.jpg

7. What are the vulnerabilities of the website that you can see right away?
The main vulnerability is the site only uses HTTP and not HTTPS, so data will be sent in plaintext.

8. Repeat similar operation for https://www.foxnews.com/. Attach two images sent from the server to your machine if you can.

I could not find any pictures in the HTTP object list

The site uses HTTPS, so data is not sent in plaintext but encrypted using TLS.

3292	8.274352	146.229.246.26	35.186.224.25	TCP	54 62891 → 443 [ACK] Seq=1906 Ack=40 Win=1027 Len=0
3293	8.276105	35.186.224.25	146.229.246.26	TLSv1.2	124 Application Data
3294	8.276167	35.186.224.25	146.229.246.26	TLSv1.2	201 Application Data
3295	8.276187	146.229.246.26	35.186.224.25	TCP	54 62891 → 443 [ACK] Seq=1906 Ack=257 Win=1026 Len=0
3296	8.276199	35.186.224.25	146.229.246.26	TLSv1.2	122 Application Data
3297	8.276207	35.186.224.25	146.229.246.26	TLSv1.2	93 Application Data
3298	8.276213	146.229.246.26	35.186.224.25	TCP	54 62891 → 443 [ACK] Seq=1906 Ack=364 Win=1026 Len=0
3299	8.277090	146.229.246.26	35.186.224.25	TLSv1.2	93 Application Data
3300	8.288304	35.186.224.25	146.229.246.26	TCP	60 443 → 62891 [ACK] Seq=364 Ack=1945 Win=457 Len=0