

OLD DOMINION UNIVERSITY
CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

Assignment #2

Kilah

Summer 2020

Task A: Basic Wireshark and tshark practice

Wireshark 2.4.0 - Wi-Fi: en0

ip.addr == 128.82.112.29

No.	Time	Source	Destination	Protocol	Length	Info
5	0.081392	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=176/45056, ttl=64 (reply in 6)
6	0.136168	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0xa2f9, seq=176/45056, ttl=238 (request in 5)
31	1.084915	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=177/45312, ttl=64 (reply in 72)
72	1.144182	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0xa2f9, seq=177/45312, ttl=238 (request in 31)
159	2.089847	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=178/45568, ttl=64 (reply in 1729)
1729	2.155889	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0xa2f9, seq=178/45568, ttl=238 (request in 1595)
1731	3.091834	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=179/45824, ttl=64 (reply in 1732)
1732	3.163531	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0xa2f9, seq=179/45824, ttl=238 (request in 1731)
1733	4.096755	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=180/46080, ttl=64 (reply in 1735)
1735	4.184774	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0xa2f9, seq=180/46080, ttl=238 (request in 1734)
1737	5.099612	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=181/46336, ttl=64 (reply in 1738)
1738	5.144980	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0xa2f9, seq=181/46336, ttl=238 (request in 1737)
1741	6.099246	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=182/46592, ttl=64 (reply in 1742)
1742	6.164760	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0xa2f9, seq=182/46592, ttl=238 (request in 1741)
1745	7.099618	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=183/46848, ttl=64 (reply in 1746)
1746	7.174850	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0xa2f9, seq=183/46848, ttl=238 (request in 1745)
1747	8.104219	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0xa2f9, seq=184/47104, ttl=64 (reply in 1748)

Frame 5: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0

Ethernet II, Src: Apple_ec0:c0:08 (10:94:bbe:c0:08:08), Dst: fe:2a:9c:ba:34:64 (fe:2a:9c:ba:34:64)

Internet Protocol Version 4, Src: 172.20.10.4, Dst: 128.82.112.29

Internet Control Message Protocol

Wireshark 2.4.0 - Wi-Fi: en0

0000 fe 2a 9c ba 34 64 10 94 bb ec c0 08 08 00 45 00 ... * 4d · · · E · · ·

wireshark_Wi-Fi_20200712181719_vBC42W.pcapng

Packets: 17692 - Displayed: 188 (1.1%) Profile: Default

Mac OS X

Terminal Shell Edit View Window Help Wi-Fi: en0

Wireshark 2.4.0 - Wi-Fi: en0

Apply a display filter: <⌘>/

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0x652a, seq=4/1024, ttl=64 (reply in 2)
2	0.010552	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0x652a, seq=4/1024, ttl=238 (request in 1)
3	0.002788	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0x652a, seq=5/1280, ttl=64 (reply in 4)
4	1.166016	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0x652a, seq=5/1280, ttl=238 (request in 3)
5	1.913382	2600:1002:b46a:c26-	2607:fb08:4004:c09-	TCP	74	589696 → 5228 [ACK] Seq=1 Ack=1 Win=2048 Len=0
6	2.005510	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0x652a, seq=6/1536, ttl=64 (reply in 0)
7	2.035957	2607:fb08:4004:c09-	2600:1003:b46a:c26-	TCP	86	[TCP ACKed unseen segment] 5228 → 589696 [ACK] Seq=2 Ack=2 Win=249 Len=0 TSecr=2687689007 TSeqc=2985773
8	2.095975	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0x652a, seq=6/1536, ttl=238 (request in 6)
9	2.743863	172.20.10.1	224.0.0.251	MDNS	115	Standard query 0x0000 PTR _companion-link._tcp.local, "QU" question OPT
10	2.743869	fe80::151:aef8:72-	ff02::1b	MDNS	135	Standard query 0x0000 PTR _companion-link._tcp.local, "QU" question OPT
11	2.838471	172.20.10.4	172.20.10.1	MDNS	402	Standard query response 0x0000 PTR Kilah's MacBook Air._companion-link._tcp.local SRV, cache flush 0
12	2.910568	172.20.10.4	23.62.27.250	TCP	54	62020 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
13	2.910561	172.20.10.4	38.81.32.37	TCP	54	61983 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
14	2.945543	38.81.32.37	172.20.10.4	TCP	66	[TCP ACKed unseen segment] 443 → 61983 [ACK] Seq=1 Ack=2 Win=807 Len=0 TSecr=1656748352 TSeqc=29880541
15	2.945548	23.62.27.250	172.20.10.4	TCP	66	[TCP ACKed unseen segment] 443 → 62020 [ACK] Seq=2 Ack=2 Win=226 Len=0 TSecr=1656748352 TSeqc=29862471
16	3.006695	172.20.10.4	128.82.112.29	ICMP	98	Echo (ping) request id=0x652a, seq=7/1792, ttl=64 (reply in 17)
17	3.066687	128.82.112.29	172.20.10.4	ICMP	98	Echo (ping) reply id=0x652a, seq=7/1792, ttl=238 (request in 16)

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0

Ethernet II, Src: Apple_ec0:c0:08 (10:94:bbe:c0:08:08), Dst: fe:2a:9c:ba:34:64 (fe:2a:9c:ba:34:64)

Internet Protocol Version 4, Src: 172.20.10.4, Dst: 128.82.112.29

Internet Control Message Protocol

Wireshark 2.4.0 - Wi-Fi: en0

0000 fe 2a 9c ba 34 64 10 94 bb ec c0 08 08 00 45 00 ... * 4d · · · E · · ·

wireshark_Wi-Fi_20200712184223.vBpuJ2T.pcapng

Packets: 63 - Displayed: 63 (100%) Profile: Default

Mac OS X

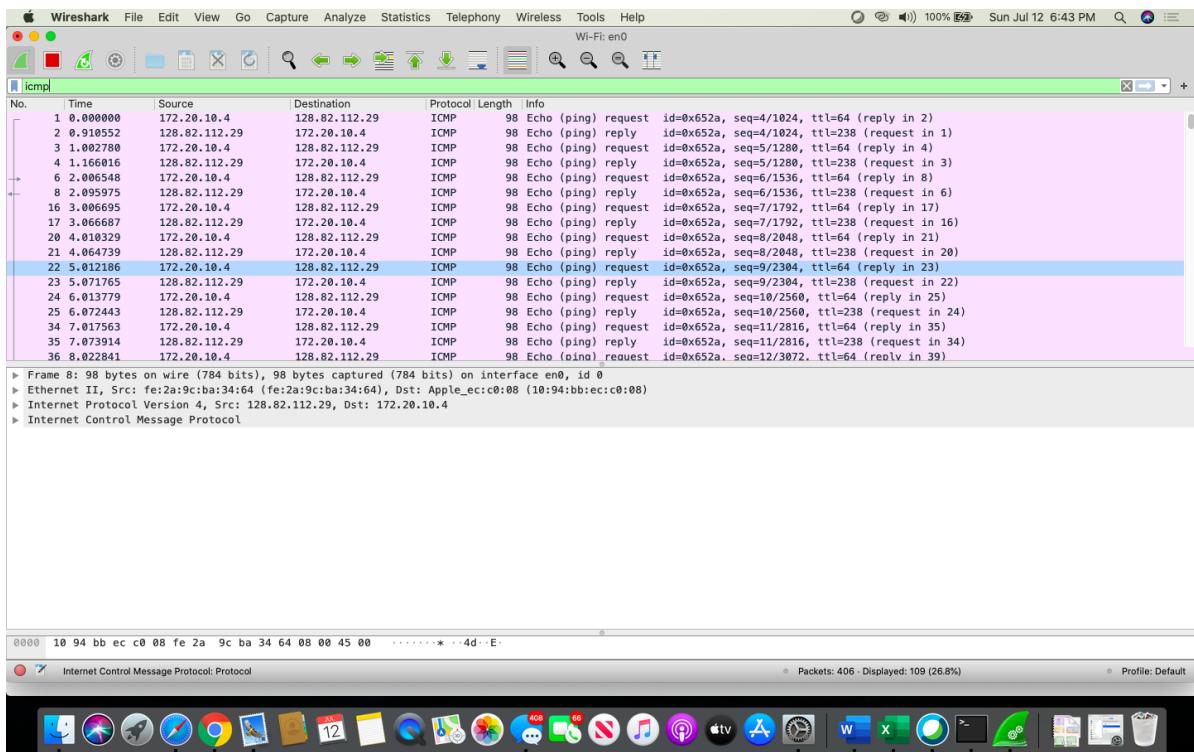


Figure 1 for Task A.1

1. Use **Wireshark** to find out all related packets when you ping the IP address "128.82.112.29"
 - When I ping the IP address 128.82.112.29, my device will send multiple ICMP packets to the target address.
 - I typed in icmp for the 128.82.112.29 ip address

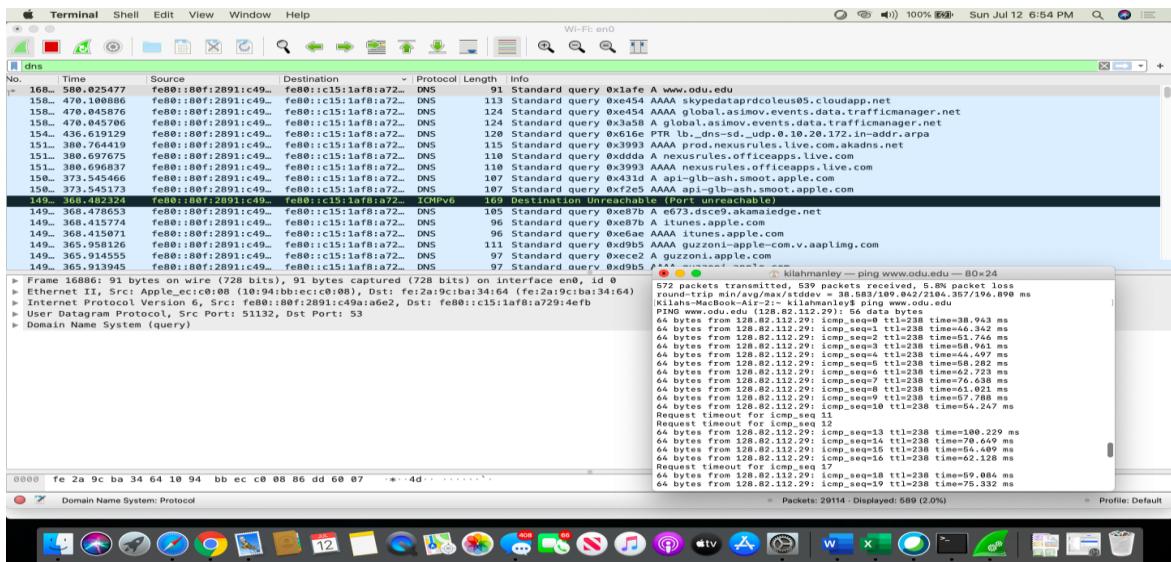
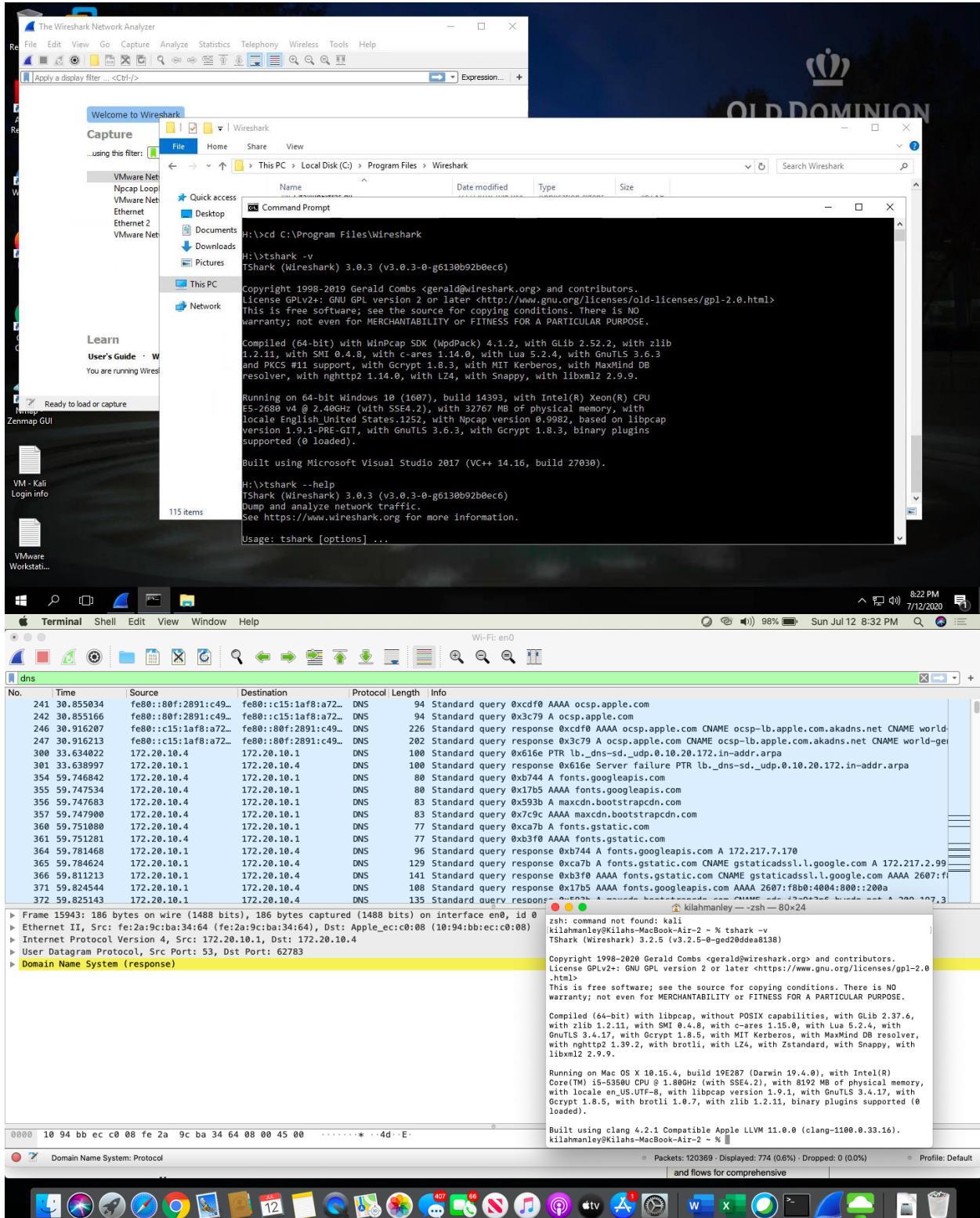


Figure 2 for Task A.2

2. Use **Wireshark** to find out all related packets when you ping the website "www.odu.edu". And mark the **new** packets generated in this traffic (compared with the result of **task 1** above). Hint: think about the difference between pinging an IP and pinging a URL.

- When I ping www.odu.edu , my device will send DNS packets to the DNS query to filter and look for IP address that may be alike/the same.



Figures 3 & 4 for Task A.3

3. Use **tshark** capture the DNS query for the ODU official websites. (Refer to Figure 21 in PLE Section 1.4.2)

- I had so much difficulty attempting to get tshark to work on my mac, and nothing worked**, so I used the Cybersecurity Environment and I was able to use tshark on Windows. I opened the command prompt, searched the program files for wireshark, copied the address at the top and used ‘cd’ in the command prompt to change directory and pasted the link address for wireshark to access tshark.
- (**EDIT: I figured out how to run tshark on my macbook)

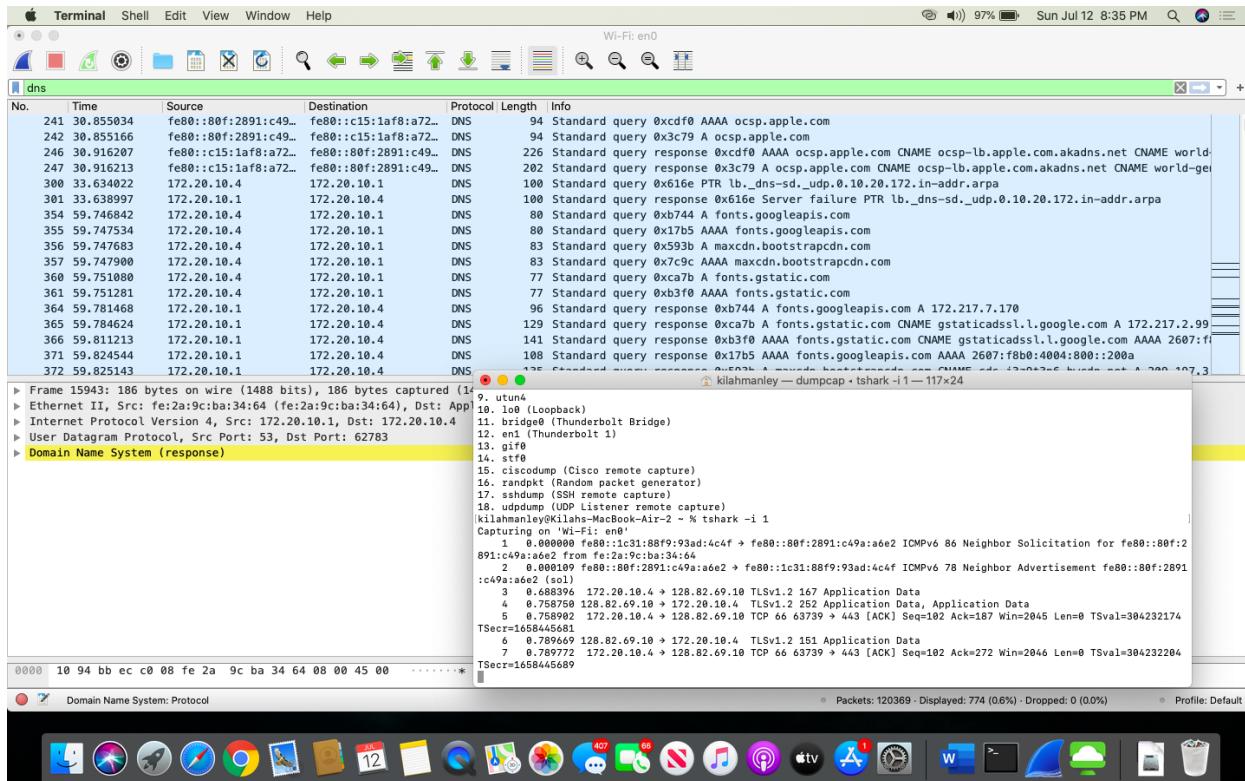


Figure 5 for Task A.3

- This figure shows me capturing data on the correct interface = Wifi

Task 3 (cont)

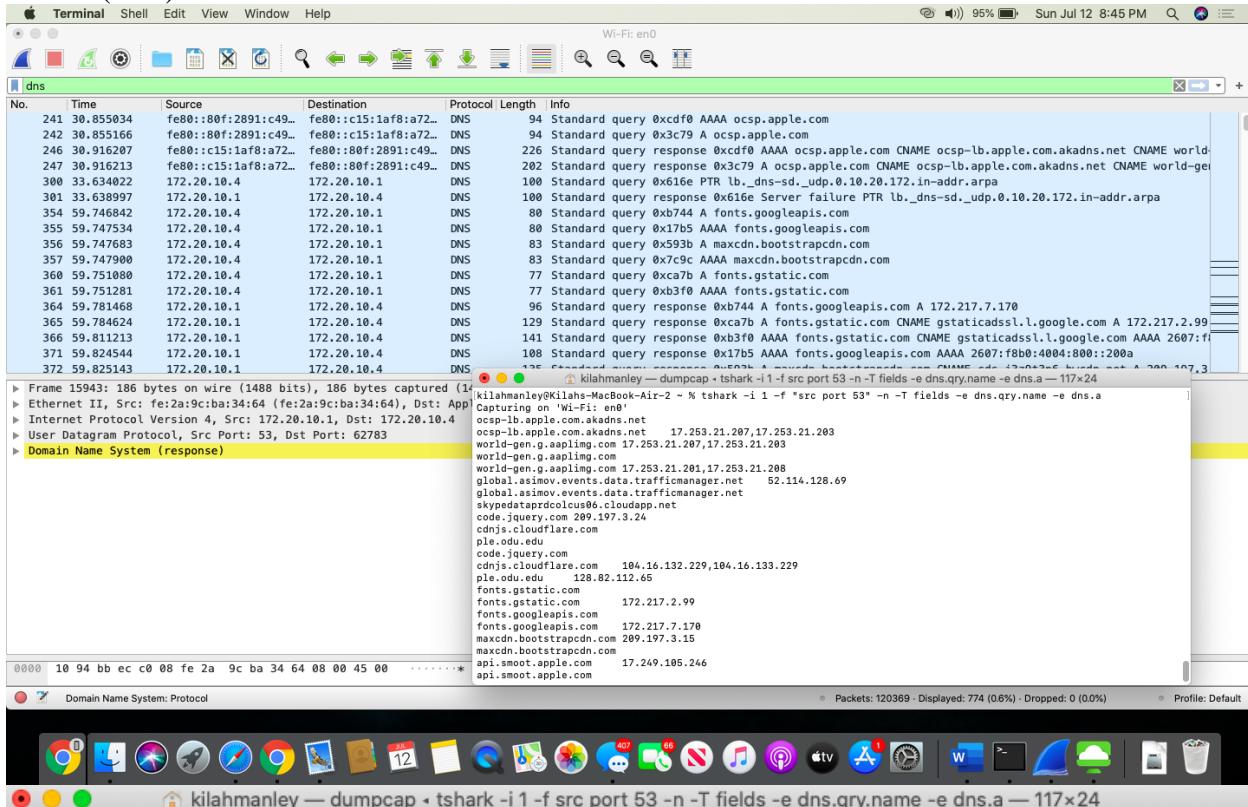


Figure 6 & 7 for Task A.3

- I used 'tshark -i 1 -f "src port 53" -n -T fields -e dns.qry.name -e dns.a' command to capture filters
- figure 7 shows 'www.odu.edu' with the ip address of 128.82.112.65

4. Visiting a safe HTTP site (the one does not ask you for the username and password, or run any scripts on your browser or local machine). And use **tshark** to capture the whole process. Filter and save the HTTP traffic to your hard drive.

- I went to tcc.edu

5. Use **Wireshark** to open the saved traffic file from your previous step, then analyze and **export** (**File -> Export Objects -> HTTP**) the HTTP objects from the traffic.

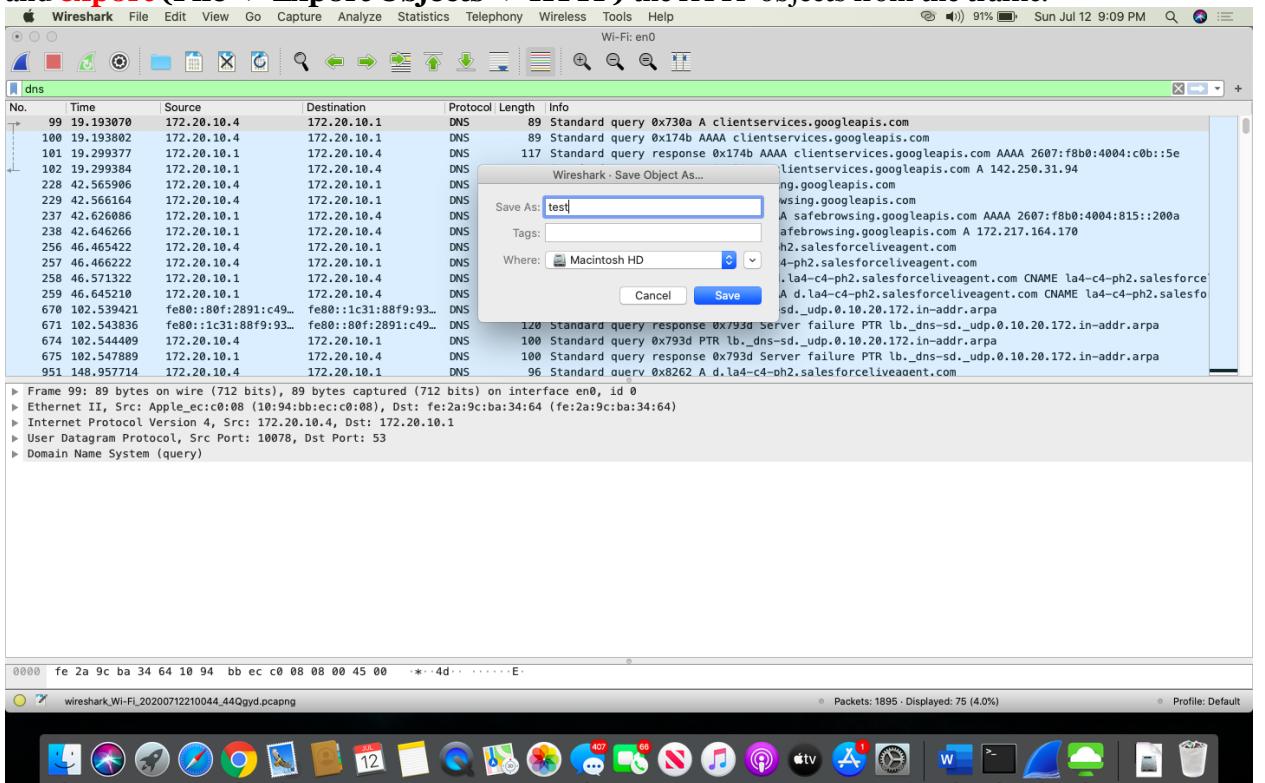


Figure 8 for A.5

Task B: Capture the FTP password

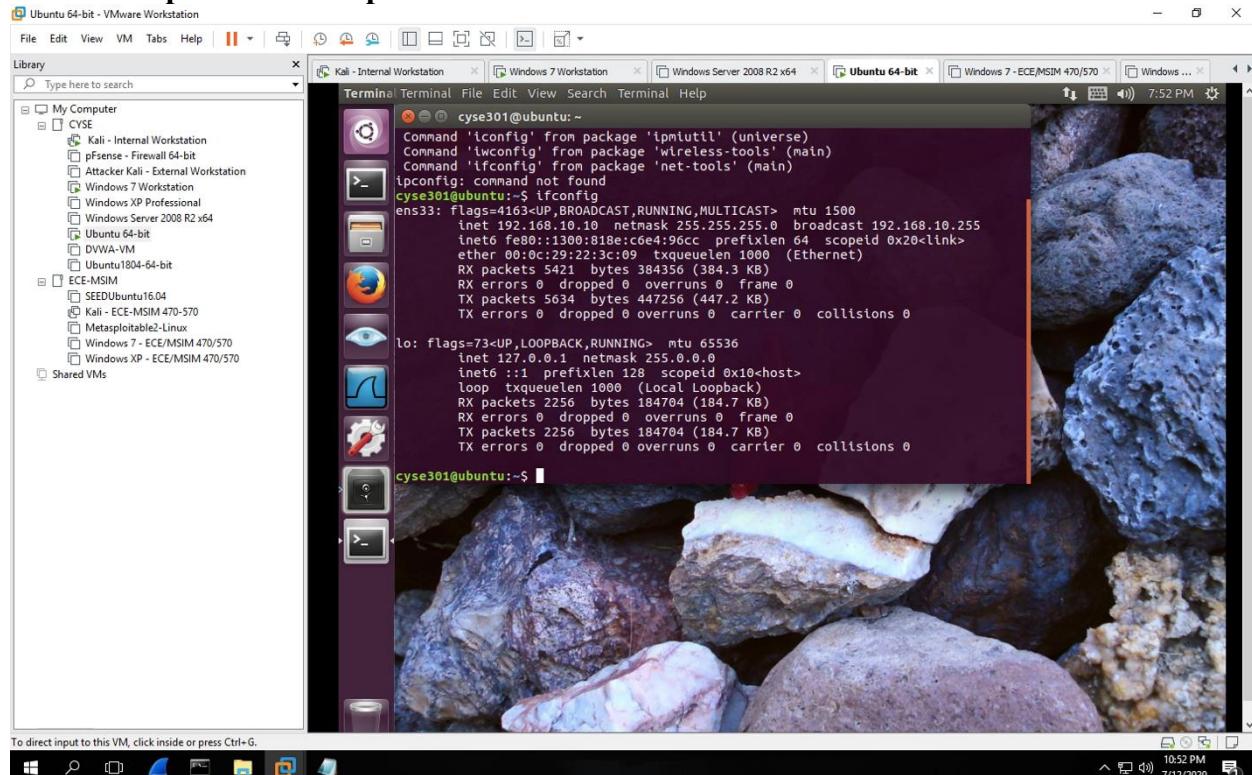


Figure 9 for B.1

- Here I opened up ubuntu and used 'ifconfig' to find the IP address (192.168.10.10)

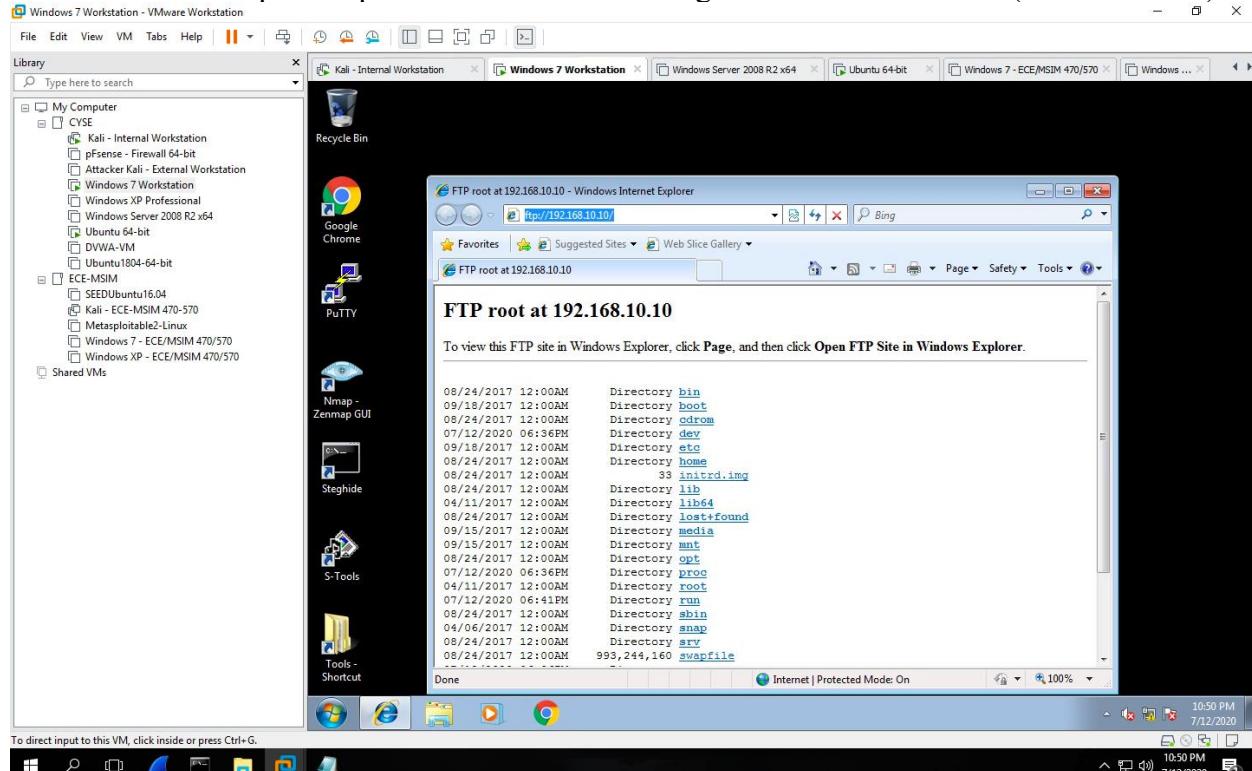
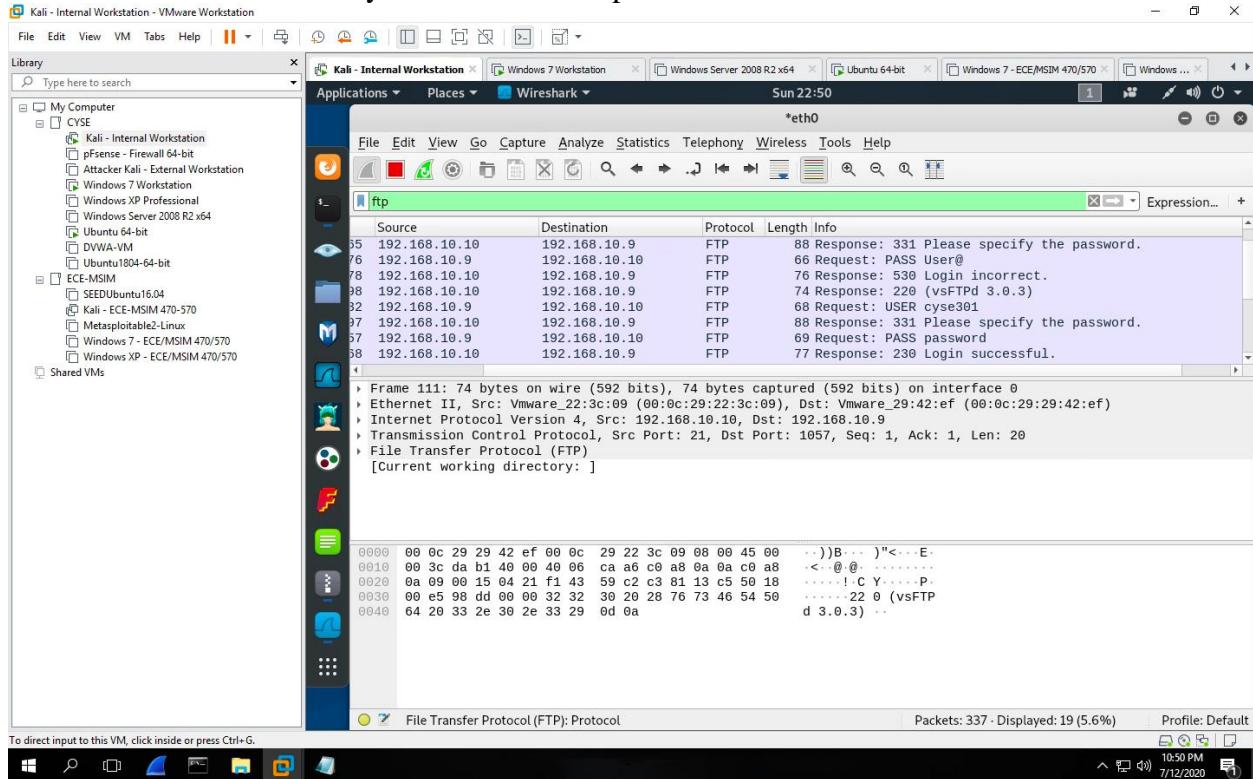


Figure 10 for Task B.

- I then opened up internal kali and ran wireshark
- Then I opened up internet explorer in Windows 7 and typed ‘ftp://192.168.10.10)
- I then went back to Internal Kali and filtered the search to ‘ftp’
- I found USER: cyse301 and PASS: password was in CLEARTEXT



Task C: Discussion

1. What is the difference between Task A-1 and Task A-1?
 - Task A-2, www.odu.edu is a domain name which will give you too many results and will take longer to process.
2. List a few websites nowadays that still use http, are they safe?
 - dictionary.com
 - Washington.edu
 - They are **not** 100% secure and safe.
3. Why does DNS use UDP Stream while HTTP uses TCP Stream?
 - As previously mentioned, in Task A-2, the response time took much longer than Task A-1. UDP is generally faster and does not require a connection (unreliable) because DNS requests are generally small.
 - TCP is connection based and must be ‘acknowledged’ and if not, it will resend the message. In UDP, the message could likely be lost and not resent (unreliable).