

CS 3339 N11

Lab 10

1.

The image displays two screenshots of the Wireshark network protocol analyzer. The top screenshot shows an HTTP GET request from 10.0.1.15 to 10.0.1.144 on port 80. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol layers. The packet bytes pane shows the raw data in hexadecimal and ASCII.

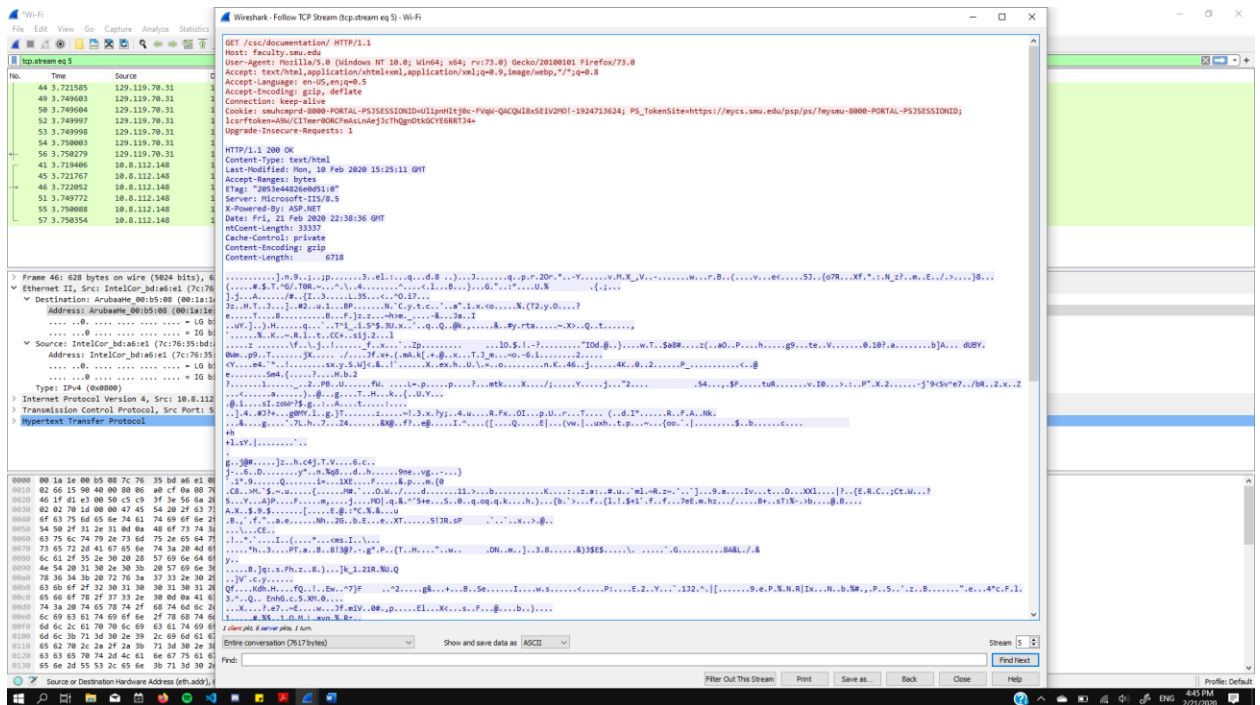
The bottom screenshot shows a DNS query from 10.0.1.15 to 10.0.1.144 on port 53. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and User Datagram Protocol layers. The packet bytes pane shows the raw data in hexadecimal and ASCII.

Packet 46: HTTP GET

No.	Time	Source	Destination	Protocol	Length	Info
46	3.720952	10.0.1.15	10.0.1.144	HTTP	628	GET /css/documentation/ HTTP/1.1

Packet 47: DNS Query

No.	Time	Source	Destination	Protocol	Length	Info
47	3.721584	10.0.1.15	10.0.1.144	DNS	282	Standard query response 0x0b0c: A faculty.smu.edu CNME sdrsl81.systems.smu.edu A 129.119.70.31 NS seas.smu.edu NS epony.emergency.smu.edu NS pony.cis.smu.edu NS xpony.its.smu.edu A 129.119.70.31



2. Packets highlighted in black are TCP packets that had a problem.
 3. To list http traffic, "http" is the command for the filter bar.
 4. DNS uses UDP because it is much faster than TCP and can fit DNS requests in each segment.
 5. To capture the password used to connect to the FTP server, I had Wireshark capture packets while logging into the server. I filtered through the traffic using "ftp" and found packets with the header USER that contained the username in plaintext. I found another packet with the header PASS with the password in plaintext.
- Sending credentials in plaintext could've been prevented by hashing them first. That hash could be checked against the hash of the correct login hash, and access could be denied or granted accordingly.