CENG435 THE1

Utku Güngör

2237477

PART 1

1) To find the number of legitimate users, I checked the source IP addresses. I went the following path from the toolbar:

   Statistics – IPv4 Statistics – Source and Destination Addresses

   There are 5 different IP addresses. So, the answer is 5.

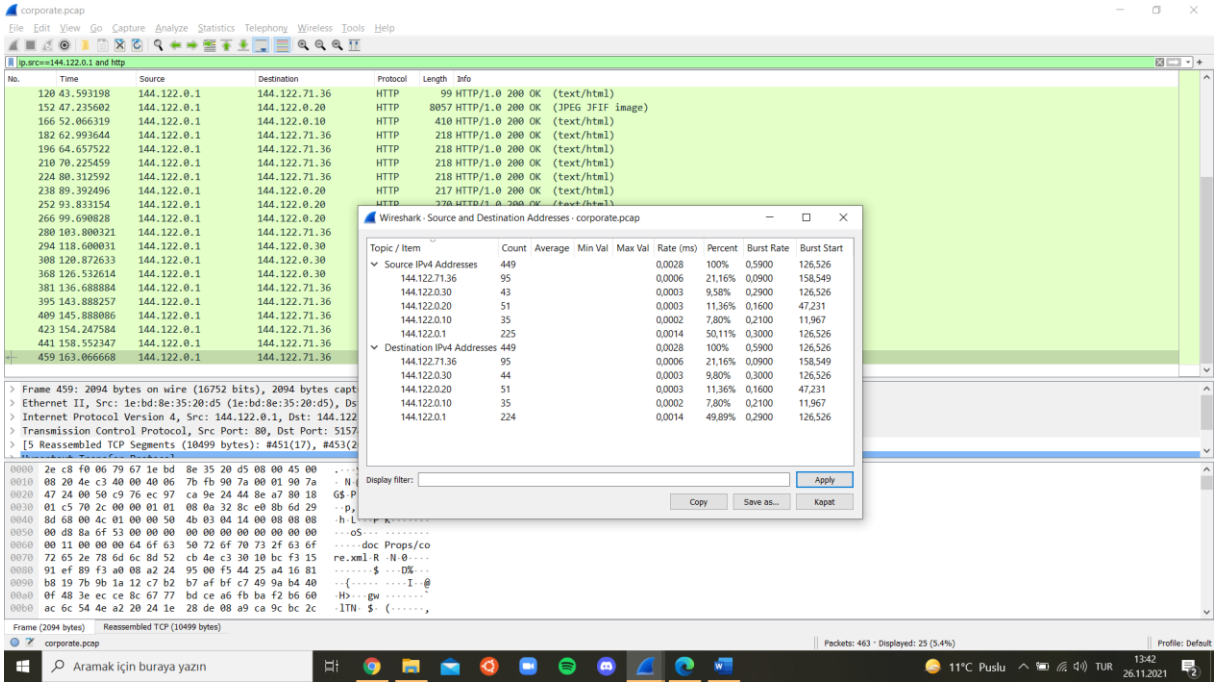

Figure 1: Users

2) Since HTTP responses are given by that IP address, I used two filters for this question ("ip.src==144.122.0.1 and http") to check the response messages. When I checked the response message contents, (text/html) formats are for login responses. According to the "Welcome to the server …" messages, below is the list of successful logins:
   - alice, donna, alice, bob, root, thomas
     Hence, there are 6 successful logins to the system.

3) I filtered the HTTP protocols since the login process requires HTTP requests. Then, I examined some GET requests and their responses. For example, passwords "password", "password1", "qwerty" etc. for the root user, the response contains the message "Invalid password for user: root". So, the GET request format for the login process is:

GET login/"username"/"password"

For user bob, there are 2 requests. One of them is with password “password”, the other one with “password1”. For the first request, it is denied because of the wrong credentials. For the second one, the response message contains “Welcome to the file server, bob”. So bob’s password is “password1”.

4) To detect malicious activity, I thought checking the application layer would be more suitable. So, I used the HTTP filter again. As shown in the figure below, from the IP address 144.122.71.36, the password of the root user has been tried and rejected 4 times with the most used passwords such as “password”, “qwerty”, “1234”. This shows that the user who tries these passwords has no idea about the real password, hence s/he is not the root user and is trying to leak to the system. From the same IP, static/”user” URL was tried to be reached 3 times to obtain some files such as readme.md but the response is 404 since there is no endpoint like this. This might be considered as another malicious attempt.

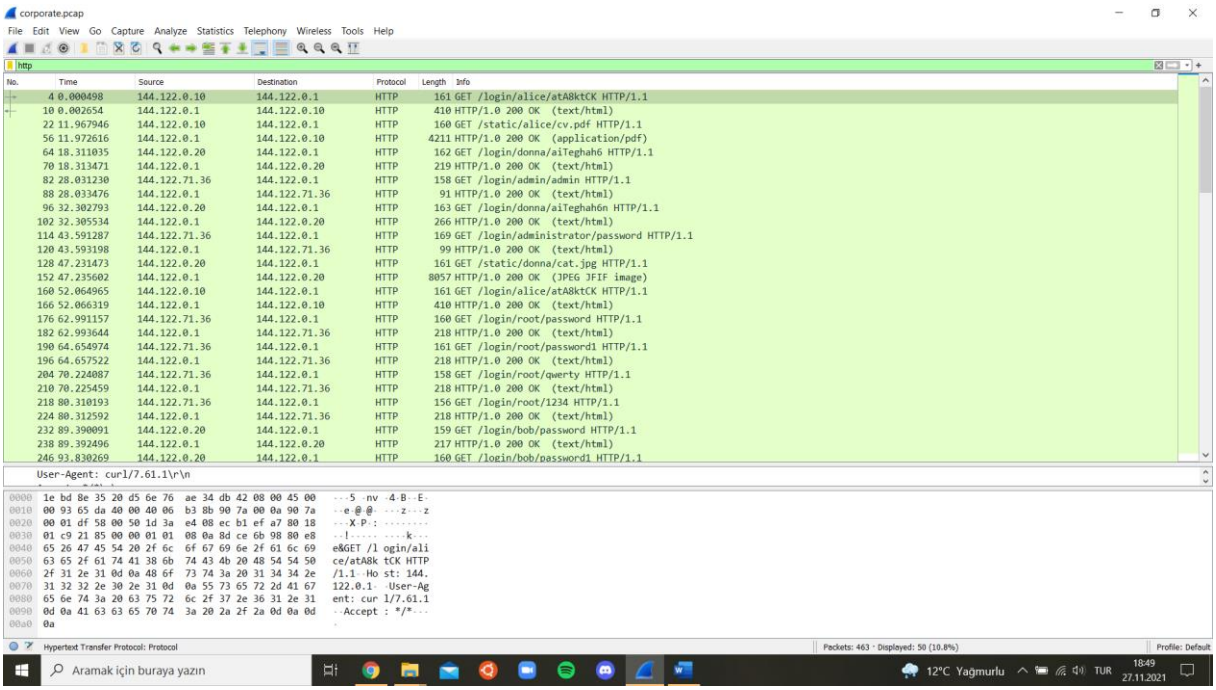

Figure 2: Malicious Activity

5) To check downloaded files, we again need to use http filter since it requires a GET request from the http server. By this filter, I went through the requests and receipt.xlsx file is reached by Alice with GET static/alice/receipt.xlsx. I also checked the response from the server and it contains related information about that file such as content length and file data. Hence, the answer is alice.

6) While investigating the passwords of users for question 3, weak passwords such as “password1” and “toor” (inverse of the root) attracted my attention. Those passwords might show vulnerability for those users.

   As a system vulnerability, a user tries 4 different passwords for the root user and s/he does not get banned or something. They might put a limit to the number of trials so that they cannot try until they find the correct password.

PART 2

1) There is one query request from my computer to the DNS server to get the http://ceng.metu.edu.tr's address and a corresponding response below that request.

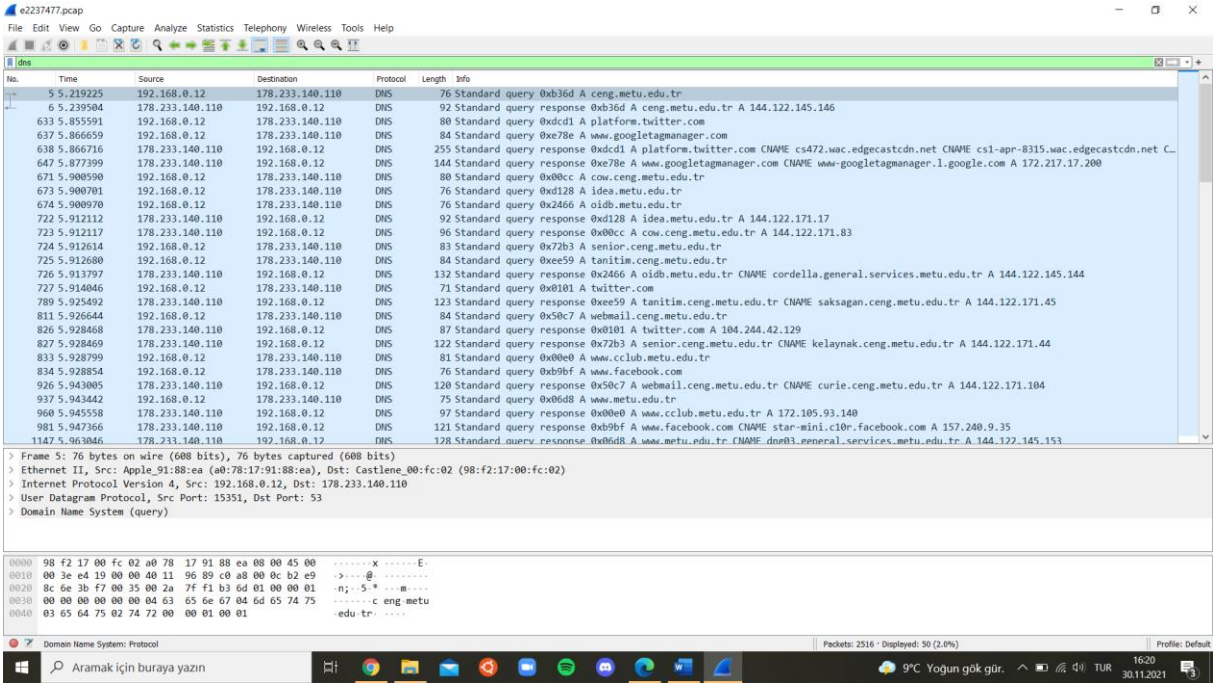

Figure 3: DNS packets

2) For the requests, there is one IP address mentioned in the next question. So, one server was queried for the DNS requests.

3) For requests, it is 192.168.206.110 and for responses, it is 192.168.0.12.

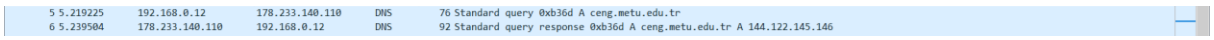

Figure 4: DNS IP addresses

4) Yes, it is cached. Because as I observe, the destination IP address of the DNS server is visible when I check my terminal.

5) a. TCP protocol is used in these requests. (Queries above the HTTP request)

b. They are not the HTTP type request and response since HTTP sends data over TCP. Hence, a TCP connection must be established between client and server before HTTP. That is the purpose of these queries.

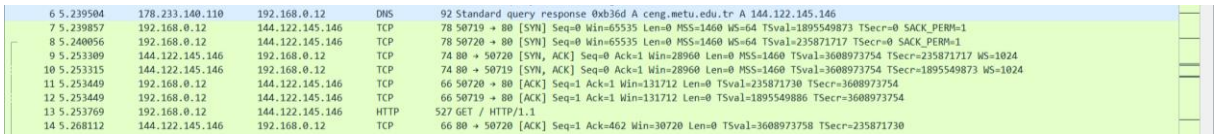

Figure 5: Protocol of the first request/response pair

6) I checked the contents of many requests to the http://ceng.metu.edu.tr's server. There are cookies in most of the other HTTP requests, but there is no cookie in the first HTTP request as shown in the figure:

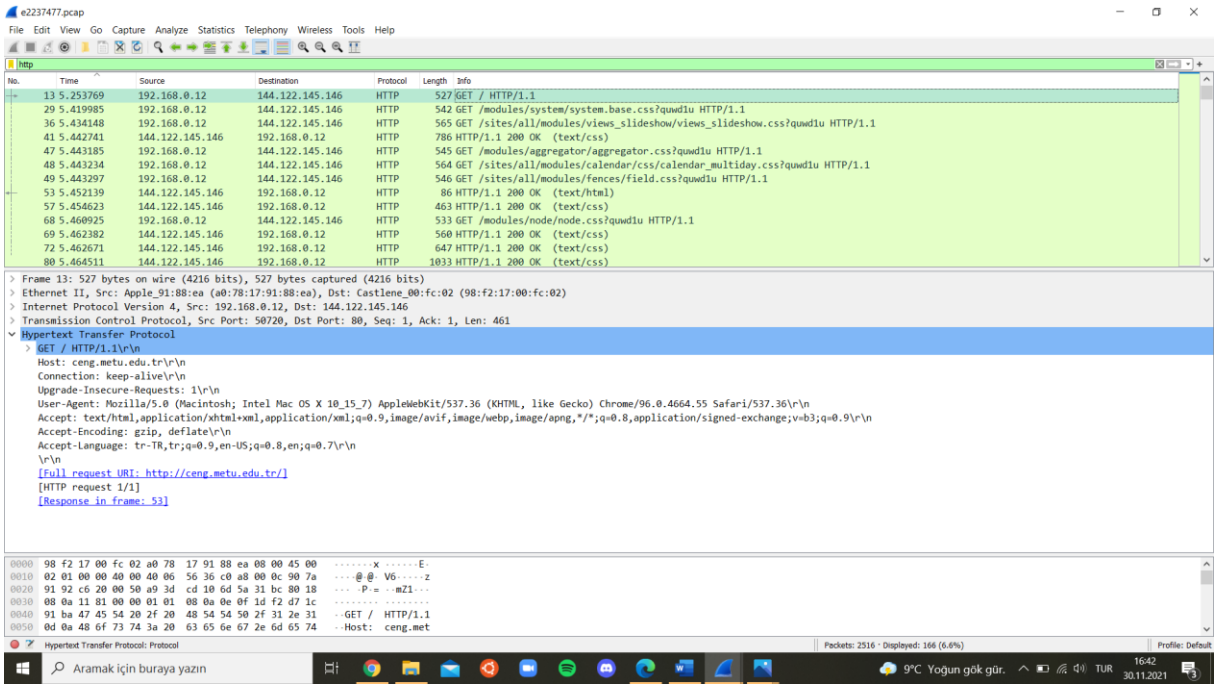

Figure 6: Cookies

7) a. Above figure also shows the user-agent string and it is:

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.55 Safari/537.36

b. As well as the browser I used(Chrome), the string includes other browsers although I have not even installed some of those. According to [1], there are some historical and security issues such as manipulating browser-sniffing web pages. Many web-browsers (including Microsoft's Internet-Explorer, Opera, and others) contain "Mozilla/5.0" at the beginning of their user agent string and I am already using Mozilla for this part of the homework. In addition, The "Gecko" refers to the browser's layout engine and the "AppleWebKit" is the actual rendering engine used by Chrome.

REFERENCES

1) https://www.linux.org/threads/user-agent-strings.11263/