Assignment:

After we try turn on capture of Wireshark, we try access the secret folder shared to us. As shown by the Wireshark, the first section of packets is displayed as following:

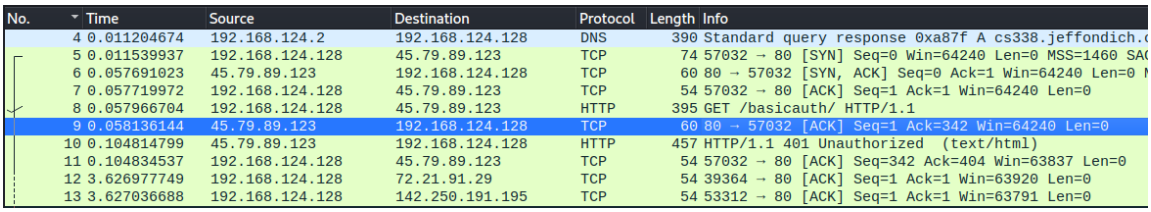
No.	Time	Source	Destination	Protocol
1	0.000000000	192.168.124.128	192.168.124.2	DNS
2	0.000043700	192.168.124.128	192.168.124.2	DNS
3	0.011204471	192.168.124.2	192.168.124.128	DNS
4	0.011204674	192.168.124.2	192.168.124.128	DNS

DNS protocol is the Domain Name System which is used to identify various resources reachable through the internet connected. We see two packets of DNS protocol are sent from my computer which has IP address of 192.168.124.128 to the secret folder which has IP address of 192.168.124.2 in order. The server then sends back two packets of DNS protocol to my computer in response. In the fourth frame, the server provides my computer with a new IP address which offers the service desired. A part of the message contained in this frame is displayed:

```
▸ Frame 4: 390 bytes on wire (3120 bits), 390 bytes captured (3120 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_fd:64:a3 (00:50:56:fd:64:a3), Dst: VMware_a2:cf:b9 (00:0c:29:a2:cf:b9)
▸ Internet Protocol Version 4, Src: 192.168.124.2, Dst: 192.168.124.128
▸ User Datagram Protocol, Src Port: 53, Dst Port: 41076
▾ Domain Name System (response)
    Transaction ID: 0xa87f
  ▸ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 4
    Additional RRs: 8
  ▸ Queries
  ▾ Answers
    ▸ cs338.jeffondich.com: type A, class IN, addr 45.79.89.123
  ▸ Authoritative nameservers
  ▸ Additional records
    [Request In: 1]
    [Time: 0.011204674 seconds]
```

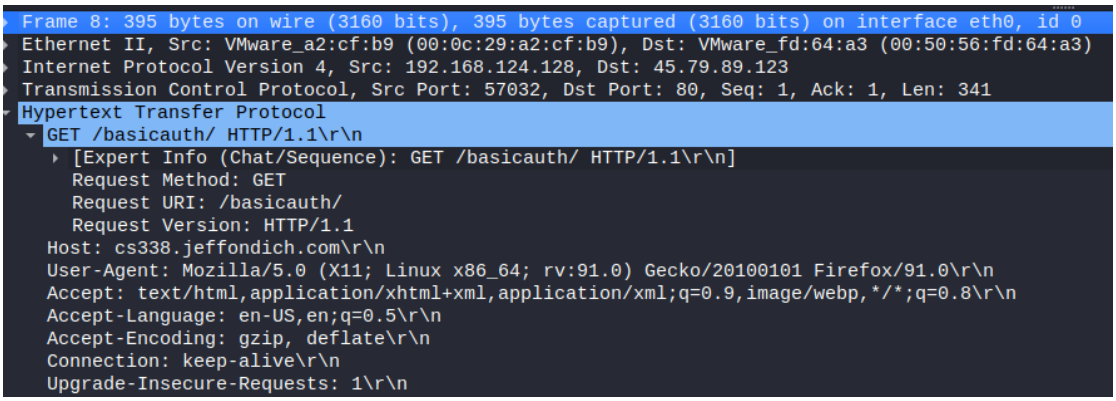
In the Answers section, IP address 45.79.89.123 is provided. My computer will then ask for the html page from this new server location. I understand this part with help from https://en.wikipedia.org/wiki/Domain_Name_System.

The next process is displayed:



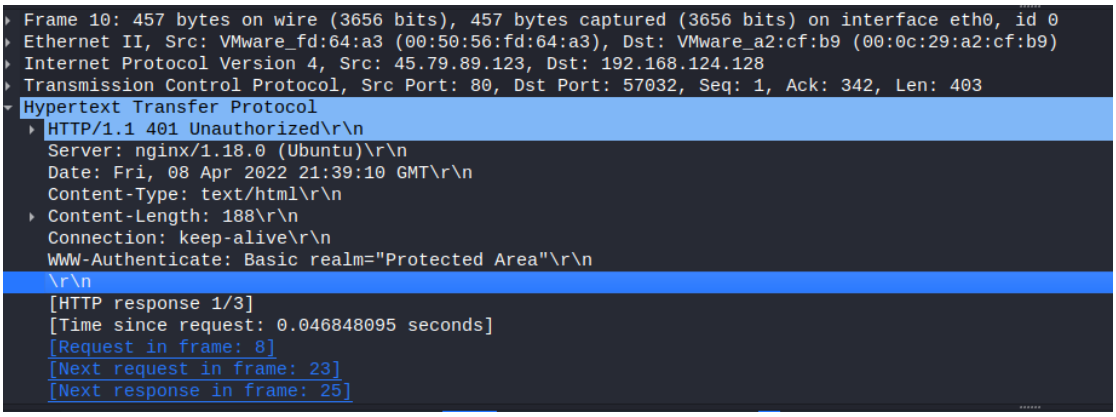
No.	Time	Source	Destination	Protocol	Length	Info
4	0.011204674	192.168.124.2	192.168.124.128	DNS	390	Standard query response 0xa87f A cs338.jeffondich.c
5	0.011539937	192.168.124.128	45.79.89.123	TCP	74	57032 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SAC
6	0.057691023	45.79.89.123	192.168.124.128	TCP	60	80 → 57032 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 M
7	0.057719972	192.168.124.128	45.79.89.123	TCP	54	57032 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
8	0.057966704	192.168.124.128	45.79.89.123	HTTP	395	GET /basicauth/ HTTP/1.1
9	0.058136144	45.79.89.123	192.168.124.128	TCP	60	80 → 57032 [ACK] Seq=1 Ack=342 Win=64240 Len=0
10	0.104814799	45.79.89.123	192.168.124.128	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
11	0.104834537	192.168.124.128	45.79.89.123	TCP	54	57032 → 80 [ACK] Seq=342 Ack=404 Win=63837 Len=0
12	3.626977749	192.168.124.128	72.21.91.29	TCP	54	39364 → 80 [ACK] Seq=1 Ack=1 Win=63920 Len=0
13	3.627036688	192.168.124.128	142.250.191.195	TCP	54	53312 → 80 [ACK] Seq=1 Ack=1 Win=63791 Len=0

Frames 5 to 7 contain the process of TCP 3-way Handshake between my computer of IP address 192.168.124.128 and the server of IP address 45.79.89.123. My computer says “hello, let’s talk!” with request for synchronization in frame 5 and the server responds by “okay!” with synchronization and acknowledgement in frame 6. Finally, my computer sends another acknowledgement in frame 7. After the process of Handshake, my computer can request for service in frame 8 by sending a HTTP protocol to the server. The message of such request is displayed:

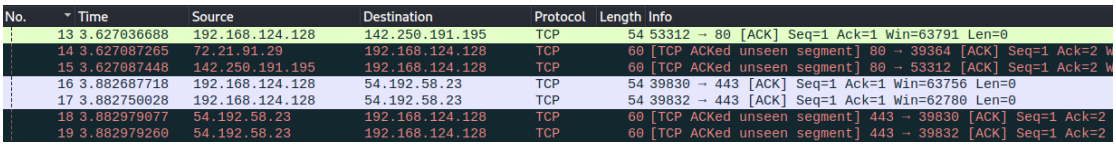


```
Frame 8: 395 bytes on wire (3160 bits), 395 bytes captured (3160 bits) on interface eth0, id 0
Ethernet II, Src: VMware_a2:cf:b9 (00:0c:29:a2:cf:b9), Dst: VMware_fd:64:a3 (00:50:56:fd:64:a3)
Internet Protocol Version 4, Src: 192.168.124.128, Dst: 45.79.89.123
Transmission Control Protocol, Src Port: 57032, Dst Port: 80, Seq: 1, Ack: 1, Len: 341
Hypertext Transfer Protocol
  GET /basicauth/ HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /basicauth/ HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /basicauth/
    Request Version: HTTP/1.1
  Host: cs338.jeffondich.com\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
```

In this packet, method GET is employed to ask for an HTML file from the server to be displayed for the secret folder. The server then responds in frame 9 with TCP protocol for acknowledgment that it will take care of the request. However, the server finds that my computer is not authorized for the access and authentication of identity is necessary. It responds to the request in frame 10 with a HTTP protocol where client is reminded of need for authentication as displayed:

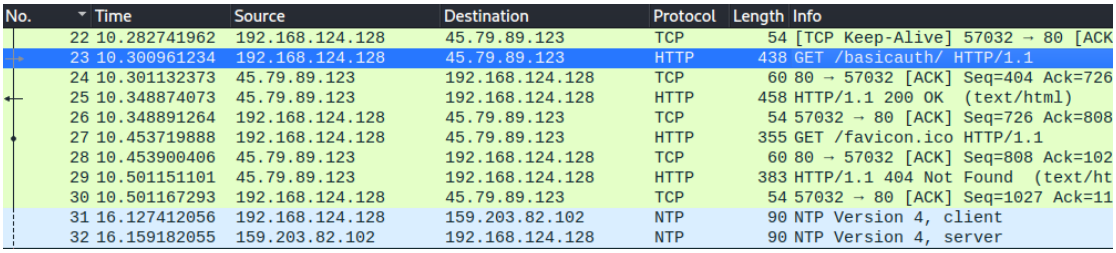


My computer replies to the denial of access with acknowledgement in frame 11 saying that “fine, I’ll get authorization first”. The computer then sends TCP protocols to servers with IP addresses of 72.21.91.29 and 142.250.191.195 in order to do process of authorization. The two servers correspond to the requirements of typing in username and password. The packets indicating process of authorization will be displayed below:



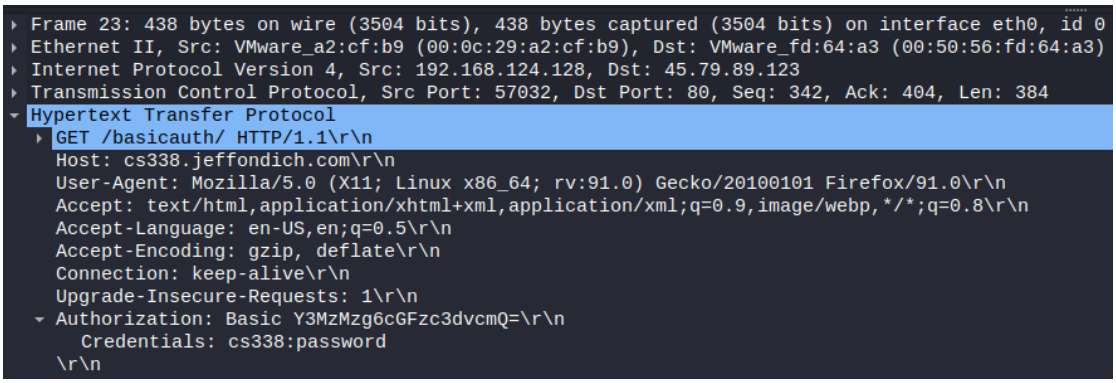
In frames 14 and 15 the two servers are sending packets of information to my computer with the protocol type indicating that ACKs for “invisible” packets are displayed in the capture. Encrypted information of username and password for authorization is then transferred to the client which is the browser I am using. Users like I would then type in username and password we know into the blanks for authorization. Packets containing this process are transferred in frames 16 and 17 to IP address of 54.192.58.23 with purpose of encryption. Encrypted information will then be sent back to my computer in frames 18 and 19. The browser will then use information acquired from frames 14, 15, 18, and 19 to verify authentication of client’s identity. After the verification, my computer will have

authorization to access the secret folder without being blocked. My computer then sends another request for HTML file as displayed below:



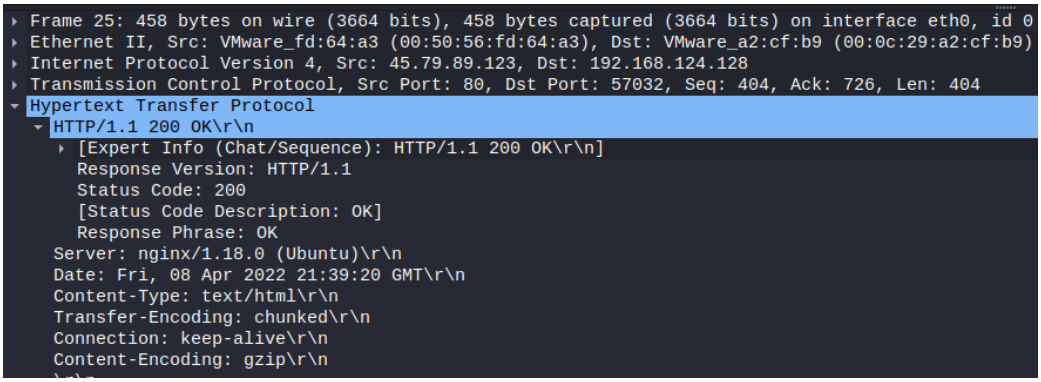
No.	Time	Source	Destination	Protocol	Length	Info
22	10.282741962	192.168.124.128	45.79.89.123	TCP	54	[TCP Keep-Alive] 57032 → 80 [ACK
23	10.300961234	192.168.124.128	45.79.89.123	HTTP	438	GET /basicauth/ HTTP/1.1
24	10.301132373	45.79.89.123	192.168.124.128	TCP	60	80 → 57032 [ACK] Seq=404 Ack=726
25	10.348874073	45.79.89.123	192.168.124.128	HTTP	458	HTTP/1.1 200 OK (text/html)
26	10.348891264	192.168.124.128	45.79.89.123	TCP	54	57032 → 80 [ACK] Seq=726 Ack=808
27	10.453719888	192.168.124.128	45.79.89.123	HTTP	355	GET /favicon.ico HTTP/1.1
28	10.453900406	45.79.89.123	192.168.124.128	TCP	60	80 → 57032 [ACK] Seq=808 Ack=102
29	10.501151101	45.79.89.123	192.168.124.128	HTTP	383	HTTP/1.1 404 Not Found (text/ht
30	10.501167293	192.168.124.128	45.79.89.123	TCP	54	57032 → 80 [ACK] Seq=1027 Ack=11
31	16.127412056	192.168.124.128	159.203.82.102	NTP	90	NTP Version 4, client
32	16.159182055	159.203.82.102	192.168.124.128	NTP	90	NTP Version 4, server

The client sends another request to the server of IP address 45.79.89.123 with HTTP protocol. The message contained in this packet is basically same with the last request in frame 8. However, a new characteristic of authorization is included in this packet to tell the server that the client is allowed to access the secret folder. The specific content of the message is displayed below:



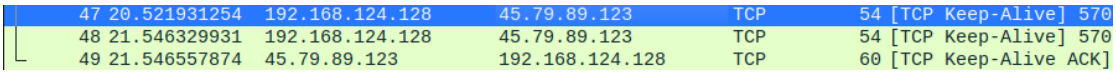
```
▸ Frame 23: 438 bytes on wire (3504 bits), 438 bytes captured (3504 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_a2:cf:b9 (00:0c:29:a2:cf:b9), Dst: VMware_fd:64:a3 (00:50:56:fd:64:a3)
▸ Internet Protocol Version 4, Src: 192.168.124.128, Dst: 45.79.89.123
▸ Transmission Control Protocol, Src Port: 57032, Dst Port: 80, Seq: 342, Ack: 404, Len: 384
▾ Hypertext Transfer Protocol
  ▸ GET /basicauth/ HTTP/1.1\r\n
    Host: cs338.jeffondich.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
  ▾ Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
      Credentials: cs338:password
    \r\n
```

The server will then examine the credentials and tells the client “Okay, I will take care of your request” in frame 24 with acknowledgment. It verifies client’s access to the HTML file and then responds in frame 25 with HTTP protocol as shown below:



```
▸ Frame 25: 458 bytes on wire (3664 bits), 458 bytes captured (3664 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_fd:64:a3 (00:50:56:fd:64:a3), Dst: VMware_a2:cf:b9 (00:0c:29:a2:cf:b9)
▸ Internet Protocol Version 4, Src: 45.79.89.123, Dst: 192.168.124.128
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 57032, Seq: 404, Ack: 726, Len: 404
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 200 OK\r\n
    ▸ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Server: nginx/1.18.0 (Ubuntu)\r\n
    Date: Fri, 08 Apr 2022 21:39:20 GMT\r\n
    Content-Type: text/html\r\n
    Transfer-Encoding: chunked\r\n
    Connection: keep-alive\r\n
    Content-Encoding: gzip\r\n
```

The server sends back a status code 200 to confirm that the request is handled successfully. HTML file displaying the secret folder will be sent to my computer to be displayed. My computer then replies to an acknowledgement in frame 26 confirming receiving information needed. The client sends another request to the server asking for favicon.ico in frame 27. The sever sends back an acknowledgement in frame 28. However, the server can't find it and must respond with a HTTP protocol containing status code of 404 in frame 29. The client then sends an acknowledgement of response before to the server in frame 30. At this stage, the process of accessing the secret folder is completed and the server will be communicating with my computer continuously to maintain the HTML page as following:



These packets of information constitute the complicated communication between clients and servers with encryption being employed as well. It was exciting to unveil the secrets lying behind these protocols.