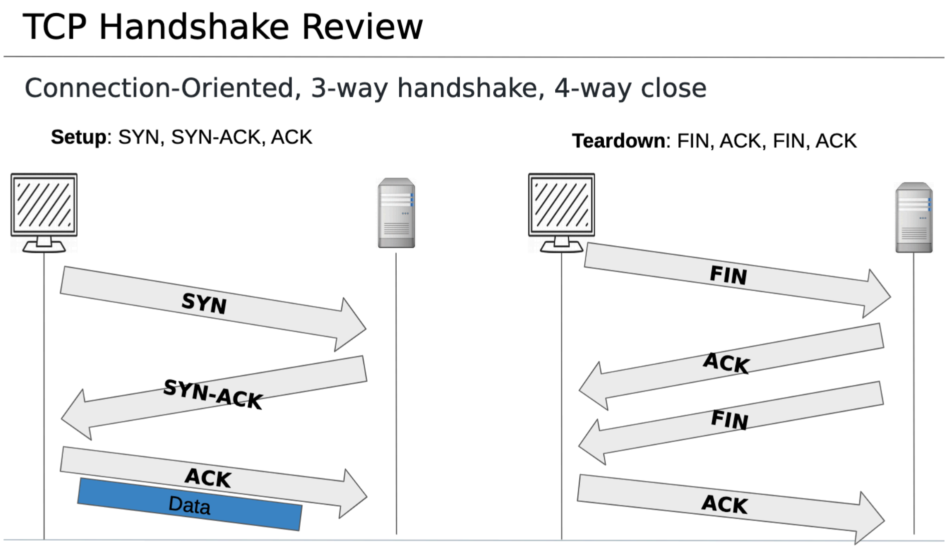
Homework Assignment 6: PCAPs for Days

Dissecting Mystery PCAPs

1-mystery.pcapng

The application layer represents a layer in the OSI model and is located in the TCP/IP protocol suite. In this pcap example, hypertext transfer protocol (HTTP) is used and the standardization as well as transmission control protocol (TCP). Within the application layer, there are processes such as "applications" and "services" that provide access to the network. There are also protocols which establish a communication interface and end-user services.



TCP Handshake Review

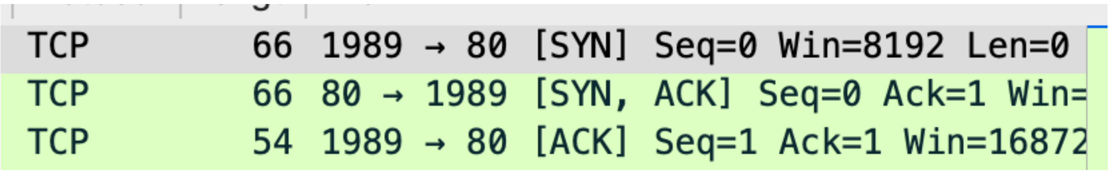
Connection-Oriented, 3-way handshake, 4-way close

Setup: SYN, SYN-ACK, ACK

Teardown: FIN, ACK, FIN, ACK

Figure 1. The steps taken in the communication of the HTTP/TCP protocol in mystery.pcapng

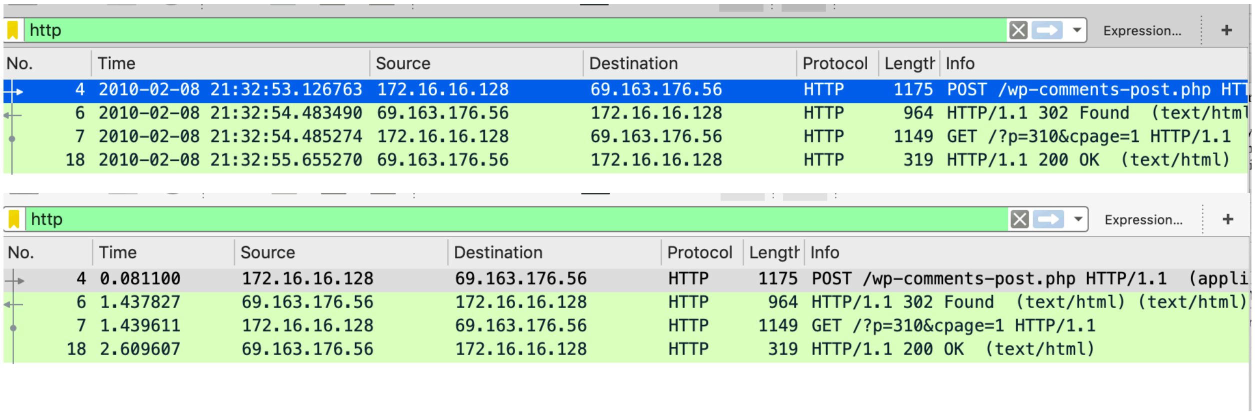
Figure 1 represents the type of communication taking place in the pcap. For instance, the IP source 172.16.16.128 sends a SYN request to 69.163.176.56. The destination IP returns with a SYN-ACK response, which is acknowledged by the source (ACK).



```
TCP        66  1989 → 80 [SYN] Seq=0 Win=8192 Len=0
TCP        66  80 → 1989 [SYN, ACK] Seq=0 Ack=1 Win=
TCP        54  1989 → 80 [ACK] Seq=1 Ack=1 Win=16872
```

Figure 2.

Furthermore, 1989 is the source port, whereas port 80 is the destination. The host in this pcap is www.chrissanders.org\r\n.

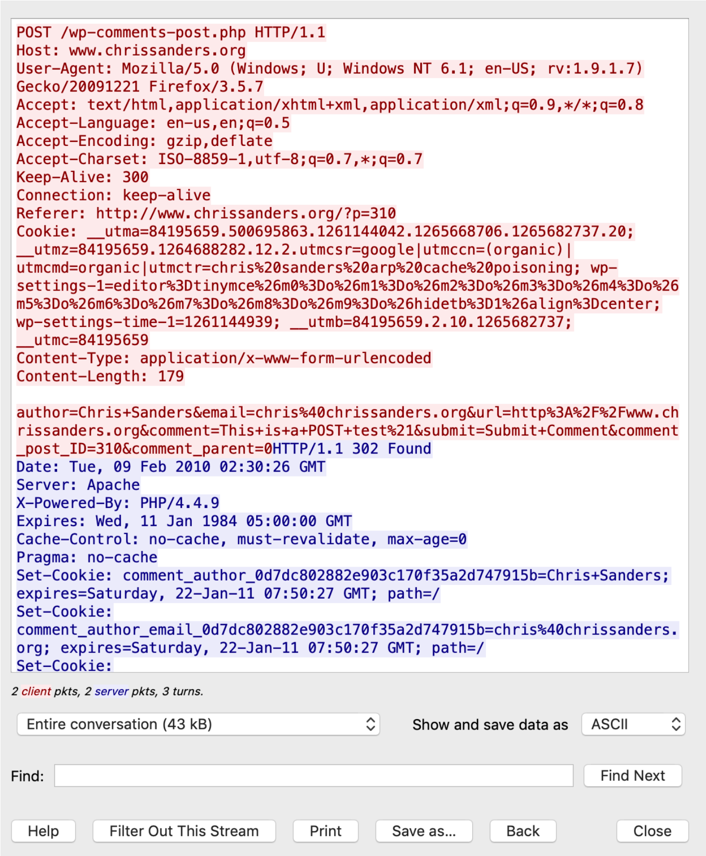


No.	Time	Source	Destination	Protocol	Length	Info
4	2010-02-08 21:32:53.126763	172.16.16.128	69.163.176.56	HTTP	1175	POST /wp-comments-post.php HTT
6	2010-02-08 21:32:54.483490	69.163.176.56	172.16.16.128	HTTP	964	HTTP/1.1 302 Found (text/html
7	2010-02-08 21:32:54.485274	172.16.16.128	69.163.176.56	HTTP	1149	GET /?p=310&cpage=1 HTTP/1.1
18	2010-02-08 21:32:55.655270	69.163.176.56	172.16.16.128	HTTP	319	HTTP/1.1 200 OK (text/html)

No.	Time	Source	Destination	Protocol	Length	Info
4	0.081100	172.16.16.128	69.163.176.56	HTTP	1175	POST /wp-comments-post.php HTTP/1.1 (appli
6	1.437827	69.163.176.56	172.16.16.128	HTTP	964	HTTP/1.1 302 Found (text/html) (text/html)
7	1.439611	172.16.16.128	69.163.176.56	HTTP	1149	GET /?p=310&cpage=1 HTTP/1.1
18	2.609607	69.163.176.56	172.16.16.128	HTTP	319	HTTP/1.1 200 OK (text/html)

Figure 3. http filter for the 1-mystery (1).pcapng

When applying the http protocol as a filter, information such as the time, source, IP destination and type of request is provided. Information such as the date and time is also shown. Figure 3 shows the source 172.16.16.128 sending a POST and GET request to the destination IP 69.163.176.56. The destination IP represents the host www.chrissanders.org. Further analysis with an http stream (figure 4) shows a conservation between 2 client packets and 2 server packets. 43 kB of data is transmitted.



```
POST /wp-comments-post.php HTTP/1.1
Host: www.chrissanders.org
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.7)
Gecko/20091221 Firefox/3.5.7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.chrissanders.org/?p=310
Cookie: __utma=84195659.500695863.1261144042.1265668706.1265682737.20;
__utmz=84195659.1264688282.12.2.utmcsr=google|utmccn=(organic)|
utmcmd=organic|utmctr=chris%20sanders%20arp%20cache%20poisoning; wp-
settings-1=editor%3Dtinymce%26m0%3Do%26m1%3Do%26m2%3Do%26m3%3Do%26m4%3Do%26
m5%3Do%26m6%3Do%26m7%3Do%26m8%3Do%26m9%3Do%26hidetb%3D1%26align%3Dcenter;
wp-settings-time-1=1261144939; __utmb=84195659.2.10.1265682737;
__utmc=84195659
Content-Type: application/x-www-form-urlencoded
Content-Length: 179

author=Chris+Sanders&email=chris%40chrissanders.org&url=http%3A%2F%2Fwww.ch
rissanders.org&comment=This+is+a+POST+test%21&submit=Submit+Comment&comment
_post_ID=310&comment_parent=0HTTP/1.1 302 Found
Date: Tue, 09 Feb 2010 02:30:26 GMT
Server: Apache
X-Powered-By: PHP/4.4.9
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Pragma: no-cache
Set-Cookie: comment_author_0d7dc802882e903c170f35a2d747915b=Chris+Sanders;
expires=Saturday, 22-Jan-11 07:50:27 GMT; path=/
Set-Cookie:
comment_author_email_0d7dc802882e903c170f35a2d747915b=chris%40chrissanders.
org; expires=Saturday, 22-Jan-11 07:50:27 GMT; path=/
Set-Cookie:
```

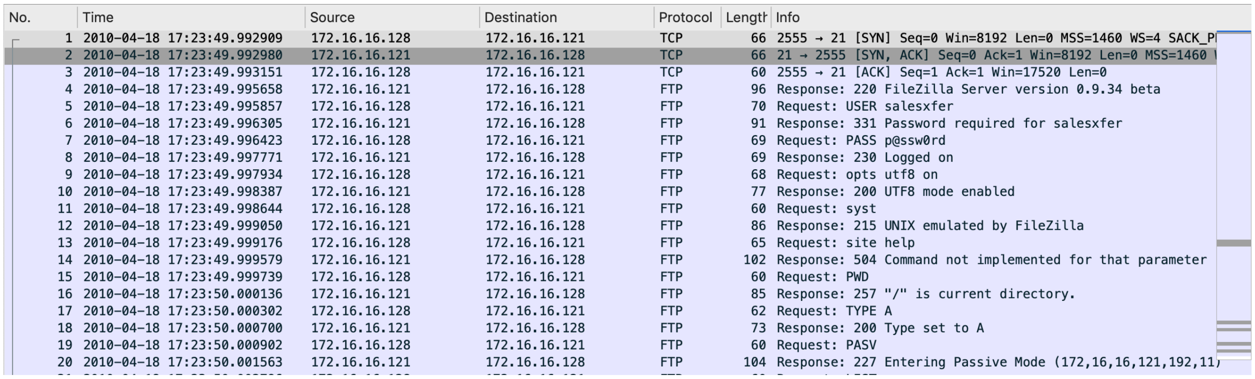
2 client pkts, 2 server pkts, 3 turns.

Entire conversation (43 kB)

Show and save data as: ASCII

Figure 4. the content type is text/html

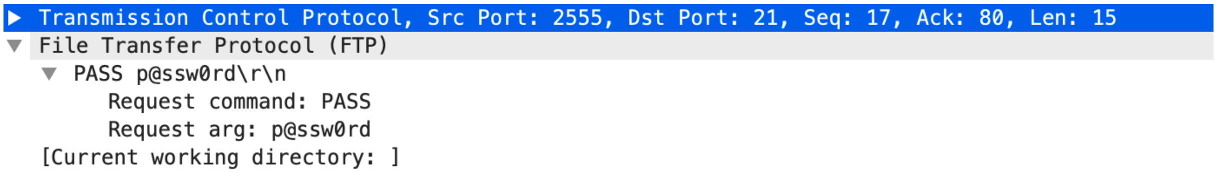
ftp_transfer.pcapng



```
No.   Time                        Source          Destination     Protocol  Length  Info
    1 2010-04-18 17:23:49.992909  172.16.16.128   172.16.16.121   TCP         66  2555 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_P
    2 2010-04-18 17:23:49.992980  172.16.16.121   172.16.16.128   TCP         66  21 → 2555 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
    3 2010-04-18 17:23:49.993151  172.16.16.128   172.16.16.121   TCP         60  2555 → 21 [ACK] Seq=1 Ack=1 Win=17520 Len=0
    4 2010-04-18 17:23:49.995658  172.16.16.121   172.16.16.128   FTP         96  Response: 220 FileZilla Server version 0.9.34 beta
    5 2010-04-18 17:23:49.995857  172.16.16.128   172.16.16.121   FTP         70  Request: USER salesxfer
    6 2010-04-18 17:23:49.996305  172.16.16.121   172.16.16.128   FTP         91  Response: 331 Password required for salesxfer
    7 2010-04-18 17:23:49.996423  172.16.16.128   172.16.16.121   FTP         69  Request: PASS p@ssw0rd
    8 2010-04-18 17:23:49.997771  172.16.16.121   172.16.16.128   FTP         69  Response: 230 Logged on
    9 2010-04-18 17:23:49.997934  172.16.16.128   172.16.16.121   FTP         68  Request: opts utf8 on
   10 2010-04-18 17:23:49.998387  172.16.16.121   172.16.16.128   FTP         77  Response: 200 UTF8 mode enabled
   11 2010-04-18 17:23:49.998644  172.16.16.128   172.16.16.121   FTP         60  Request: syst
   12 2010-04-18 17:23:49.999050  172.16.16.121   172.16.16.128   FTP         86  Response: 215 UNIX emulated by FileZilla
   13 2010-04-18 17:23:49.999176  172.16.16.128   172.16.16.121   FTP         65  Request: site help
   14 2010-04-18 17:23:49.999579  172.16.16.121   172.16.16.128   FTP        102  Response: 504 Command not implemented for that parameter
   15 2010-04-18 17:23:49.999739  172.16.16.128   172.16.16.121   FTP         60  Request: PWD
   16 2010-04-18 17:23:50.000136  172.16.16.121   172.16.16.128   FTP         85  Response: 257 "/" is current directory.
   17 2010-04-18 17:23:50.000302  172.16.16.128   172.16.16.121   FTP         62  Request: TYPE A
   18 2010-04-18 17:23:50.000700  172.16.16.121   172.16.16.128   FTP         73  Response: 200 Type set to A
   19 2010-04-18 17:23:50.000902  172.16.16.128   172.16.16.121   FTP         60  Request: PASV
   20 2010-04-18 17:23:50.001563  172.16.16.121   172.16.16.128   FTP        104  Response: 227 Entering Passive Mode (172,16,16,121,192,11)
```

Figure 5. ftp_transfer.pcapng

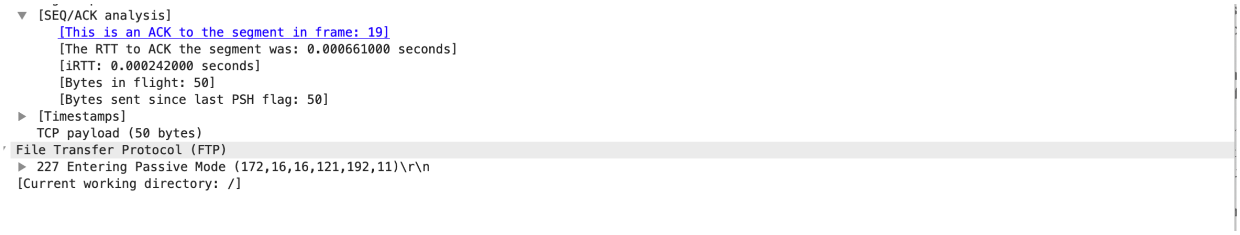
In the *ftp_transfer.pcapng* the source IP address is 172.16.16.128 whereas the destination IP is 172.16.16.121. The user Salesxfer is using UTF-8 encoding mode with Unix and one can suggest, that the objective is to convert the hex value back into character or vice versa. In UTF-8, files and strings which contain only 7-bit ASCII characters will be encoded. However, strings that contain up to 16-bit characters with bytes such as ‘\0’ or ‘\’ (figure 5) usually indicate a special meaning in the filename or other “C library function arguments”.



```
▶ Transmission Control Protocol, Src Port: 2555, Dst Port: 21, Seq: 17, Ack: 80, Len: 15
▼ File Transfer Protocol (FTP)
  ▼ PASS p@ssw0rd\r\n
      Request command: PASS
      Request arg: p@ssw0rd
  [Current working directory: ]
```

Figure 6.

In this example, a TCP handshake is established – communication between the computer and network. This initiates the file transfer protocol (FTP) used for transferring files to or exchanging files with a host computer. As shown in figure 6 the password “p@ssw0rd” is set to authenticate the request to Salesxfer. Also shown is the source port:2555 and destination port: 21. However, FTP will typically run on port 22 and establish two connections in parallel: control and data connection. It differs from the TCP protocol as it represents the communication between two computers and runs on top of TCP; similar to HTTP. TCP is a set of communication rules that allow network devices to communicate. Furthermore, the SYN-ACK highlighted in figure 5 represents a SYN message from the local device and ACK of the previous packet.



```
  ▼ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 19]
      [The RTT to ACK the segment was: 0.000661000 seconds]
      [iRTT: 0.000242000 seconds]
      [Bytes in flight: 50]
      [Bytes sent since last PSH flag: 50]
  ▶ [Timestamps]
    TCP payload (50 bytes)
 File Transfer Protocol (FTP)
  ▶ 227 Entering Passive Mode (172,16,16,121,192,11)\r\n
 [Current working directory: /]
```

Figure 7. Passive mode FTP

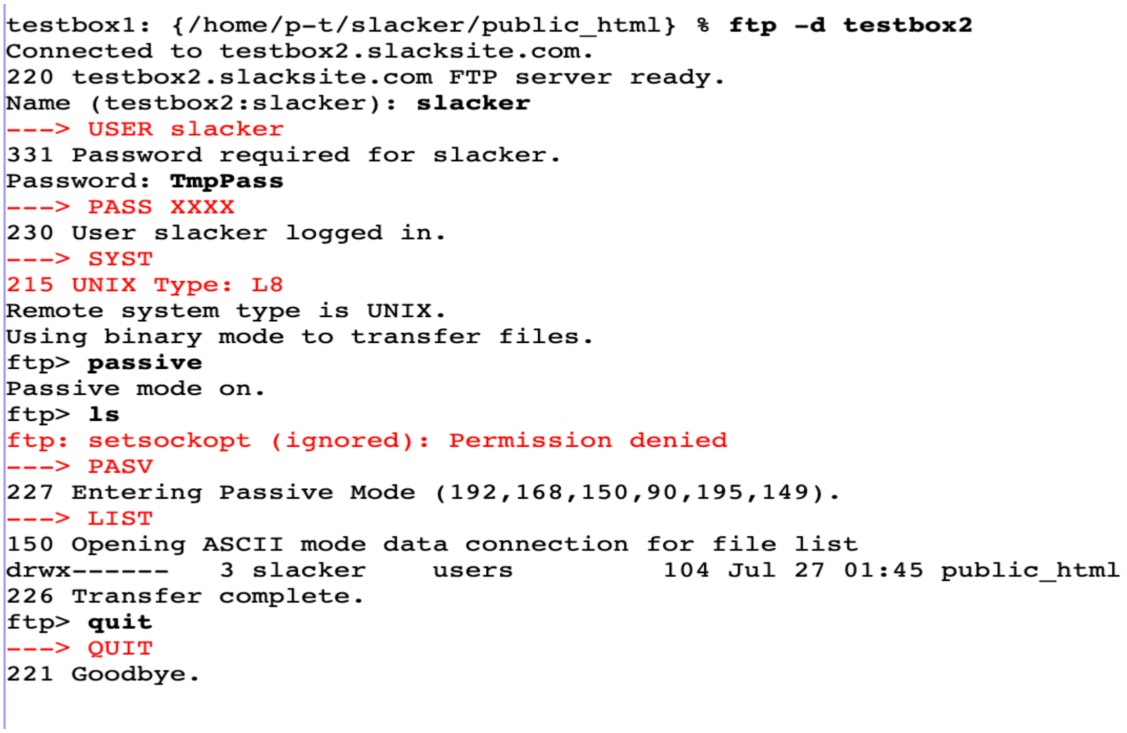
Further analysis shows the client initiates both connections through passive mode FTP. This is used to bypass firewalls, by filtering the incoming data port connection to the client from the server. This also solves the problem of the server initiating a connection to the client, which is beneficial to client but detrimental to the FTP server admin. For instance, allowing remote connections to high numbered ports on the server has the potential to be problematic. FTP daemons act as a solution as they allow the administrator to choose which ports the FTP server will use.

In passive mode FTP, the client will open two random ports locally. The first port contacts the server on port 21 (figure 6) here, the client will use the PASV command rather than the PORT command; thereby allowing the server to connect back to the data port. As a result, the server will open a random high port and send an ACK back to the client data port. In other words, the client will make both connections to the server, however one of them will be on a random high port. Figure 8 illustrates passive mode FTP.

```
Passive FTP :
command : client >1023 -> server 21
data    : client >1024 -> server >1023
```

figure 8. Passive mode FTP connection between client and server

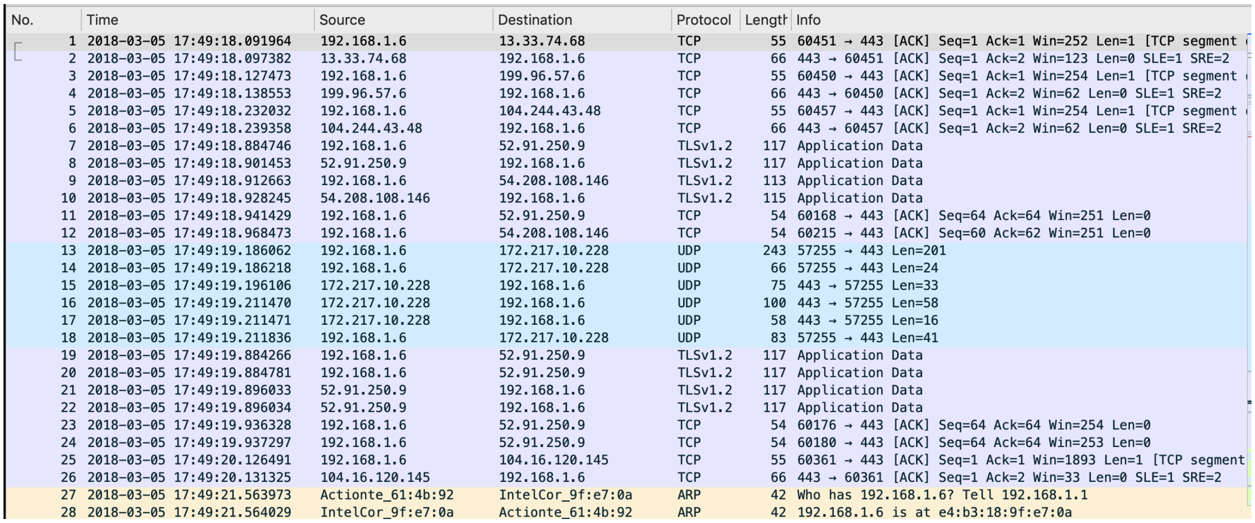
Furthermore, figure 9 shows an example of what the users Salesxfer spreadsheet would look like. However, the information is different with respect to this example. In the *ftp_transfer.pcapng* the user Salesxfer is using UTF-8 mode with Unix, rather than binary mode.



```
testbox1: {/home/p-t/slacker/public_html} % ftp -d testbox2
Connected to testbox2.slacksite.com.
220 testbox2.slacksite.com FTP server ready.
Name (testbox2:slacker): slacker
---> USER slacker
331 Password required for slacker.
Password: TmpPass
---> PASS XXXX
230 User slacker logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering Passive Mode (192,168,150,90,195,149).
---> LIST
150 Opening ASCII mode data connection for file list
drwx------   3 slacker    users         104 Jul 27 01:45 public_html
226 Transfer complete.
ftp> quit
---> QUIT
221 Goodbye.
```

Figure 9.

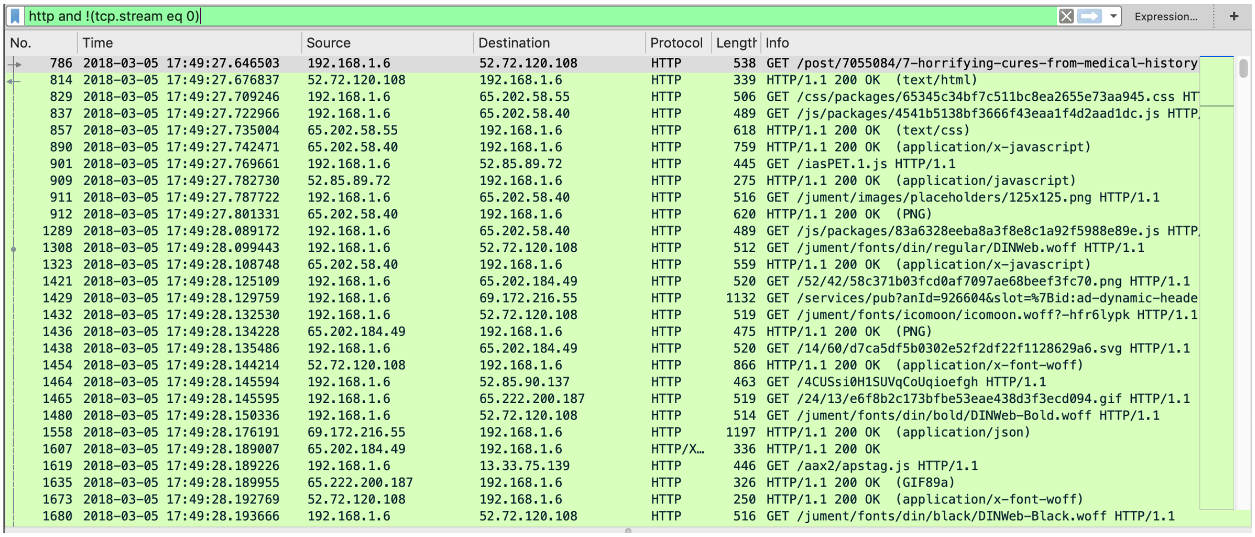
http_page.pcapng



No.	Time	Source	Destination	Protocol	Length	Info
1	2018-03-05 17:49:18.091964	192.168.1.6	13.33.74.68	TCP	55	60451 → 443 [ACK] Seq=1 Ack=1 Win=252 Len=1 [TCP segment
2	2018-03-05 17:49:18.097382	13.33.74.68	192.168.1.6	TCP	66	443 → 60451 [ACK] Seq=1 Ack=2 Win=123 Len=0 SLE=1 SRE=2
3	2018-03-05 17:49:18.127473	192.168.1.6	199.96.57.6	TCP	55	60450 → 443 [ACK] Seq=1 Ack=1 Win=254 Len=1 [TCP segment
4	2018-03-05 17:49:18.138553	199.96.57.6	192.168.1.6	TCP	66	443 → 60450 [ACK] Seq=1 Ack=2 Win=62 Len=0 SLE=1 SRE=2
5	2018-03-05 17:49:18.232032	192.168.1.6	104.244.43.48	TCP	55	60457 → 443 [ACK] Seq=1 Ack=1 Win=254 Len=1 [TCP segment
6	2018-03-05 17:49:18.239358	104.244.43.48	192.168.1.6	TCP	66	443 → 60457 [ACK] Seq=1 Ack=2 Win=62 Len=0 SLE=1 SRE=2
7	2018-03-05 17:49:18.884746	192.168.1.6	52.91.250.9	TLSv1.2	117	Application Data
8	2018-03-05 17:49:18.901453	52.91.250.9	192.168.1.6	TLSv1.2	117	Application Data
9	2018-03-05 17:49:18.912663	192.168.1.6	54.208.108.146	TLSv1.2	113	Application Data
10	2018-03-05 17:49:18.928245	54.208.108.146	192.168.1.6	TLSv1.2	115	Application Data
11	2018-03-05 17:49:18.941429	192.168.1.6	52.91.250.9	TCP	54	60168 → 443 [ACK] Seq=64 Ack=64 Win=251 Len=0
12	2018-03-05 17:49:18.968473	192.168.1.6	54.208.108.146	TCP	54	60215 → 443 [ACK] Seq=60 Ack=62 Win=251 Len=0
13	2018-03-05 17:49:19.186062	192.168.1.6	172.217.10.228	UDP	243	57255 → 443 Len=201
14	2018-03-05 17:49:19.186218	192.168.1.6	172.217.10.228	UDP	66	57255 → 443 Len=24
15	2018-03-05 17:49:19.196106	172.217.10.228	192.168.1.6	UDP	75	443 → 57255 Len=33
16	2018-03-05 17:49:19.211470	172.217.10.228	192.168.1.6	UDP	100	443 → 57255 Len=58
17	2018-03-05 17:49:19.211471	172.217.10.228	192.168.1.6	UDP	58	443 → 57255 Len=16
18	2018-03-05 17:49:19.211836	192.168.1.6	172.217.10.228	UDP	83	57255 → 443 Len=41
19	2018-03-05 17:49:19.884266	192.168.1.6	52.91.250.9	TLSv1.2	117	Application Data
20	2018-03-05 17:49:19.884781	192.168.1.6	52.91.250.9	TLSv1.2	117	Application Data
21	2018-03-05 17:49:19.896033	52.91.250.9	192.168.1.6	TLSv1.2	117	Application Data
22	2018-03-05 17:49:19.896034	52.91.250.9	192.168.1.6	TLSv1.2	117	Application Data
23	2018-03-05 17:49:19.936328	192.168.1.6	52.91.250.9	TCP	54	60176 → 443 [ACK] Seq=64 Ack=64 Win=254 Len=0
24	2018-03-05 17:49:19.937297	192.168.1.6	52.91.250.9	TCP	54	60180 → 443 [ACK] Seq=64 Ack=64 Win=253 Len=0
25	2018-03-05 17:49:20.126491	192.168.1.6	104.16.120.145	TCP	55	60361 → 443 [ACK] Seq=1 Ack=1 Win=1893 Len=1 [TCP segment
26	2018-03-05 17:49:20.131325	104.16.120.145	192.168.1.6	TCP	66	443 → 60361 [ACK] Seq=1 Ack=2 Win=33 Len=0 SLE=1 SRE=2
27	2018-03-05 17:49:21.563973	Actionte_61:4b:92	IntelCor_9f:e7:0a	ARP	42	Who has 192.168.1.6? Tell 192.168.1.1
28	2018-03-05 17:49:21.564029	IntelCor_9f:e7:0a	Actionte_61:4b:92	ARP	42	192.168.1.6 is at e4:b3:18:9f:e7:0a

Figure 10. http_page.pcapng

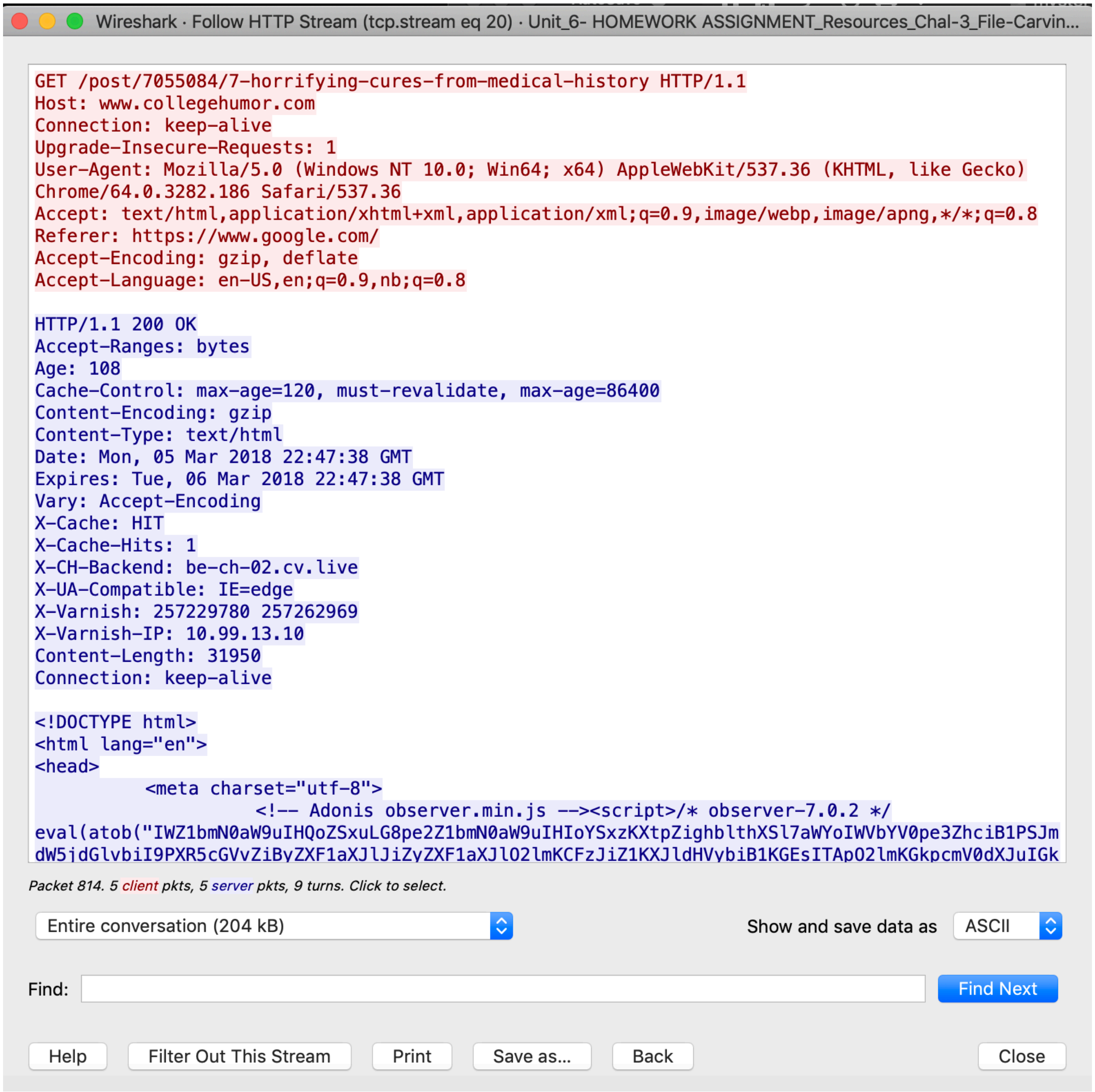
In the http_page.pcapng a TCP handshake initiates the connection between the computer and the network. The source IP address is 192.168.1.6 and the destination IP address is 13.33.74.68. However as shown in the image below, there are multiple destination IP addresses the host is sending get requests to. In figure 11, we can see the GET request has succeeded as indicated by the HTTP/1.1 200 OK response. In this case the server is responding using the HTTP protocol.



http and !(tcp.stream eq 0)

No.	Time	Source	Destination	Protocol	Length	Info
786	2018-03-05 17:49:27.646503	192.168.1.6	52.72.120.108	HTTP	538	GET /post/7055084/7-horrifying-cures-from-medical-history
814	2018-03-05 17:49:27.676837	52.72.120.108	192.168.1.6	HTTP	339	HTTP/1.1 200 OK (text/html)
829	2018-03-05 17:49:27.709246	192.168.1.6	65.202.58.55	HTTP	506	GET /css/packages/65345c34bf7c511bc8ea2655e73aa945.css HT
837	2018-03-05 17:49:27.722966	192.168.1.6	65.202.58.40	HTTP	489	GET /js/packages/4541b5138bf3666f43eaa1f4d2aad1dc.js HTTP
857	2018-03-05 17:49:27.735004	65.202.58.55	192.168.1.6	HTTP	618	HTTP/1.1 200 OK (text/css)
890	2018-03-05 17:49:27.742471	65.202.58.40	192.168.1.6	HTTP	759	HTTP/1.1 200 OK (application/x-javascript)
901	2018-03-05 17:49:27.769661	192.168.1.6	52.85.89.72	HTTP	445	GET /iasPET.1.js HTTP/1.1
909	2018-03-05 17:49:27.782730	52.85.89.72	192.168.1.6	HTTP	275	HTTP/1.1 200 OK (application/javascript)
911	2018-03-05 17:49:27.787722	192.168.1.6	65.202.58.40	HTTP	516	GET /jument/images/placeholders/125x125.png HTTP/1.1
912	2018-03-05 17:49:27.801331	65.202.58.40	192.168.1.6	HTTP	620	HTTP/1.1 200 OK (PNG)
1289	2018-03-05 17:49:28.089172	192.168.1.6	65.202.58.40	HTTP	489	GET /js/packages/83a6328eeba8a3f8e8c1a92f5988e89e.js HTTP
1308	2018-03-05 17:49:28.099443	192.168.1.6	52.72.120.108	HTTP	512	GET /jument/fonts/din/regular/DINWeb.woff HTTP/1.1
1323	2018-03-05 17:49:28.108748	65.202.58.40	192.168.1.6	HTTP	559	HTTP/1.1 200 OK (application/x-javascript)
1421	2018-03-05 17:49:28.125109	192.168.1.6	65.202.184.49	HTTP	520	GET /52/42/58c371b03fcd0af7097ae68beef3fc70.png HTTP/1.1
1429	2018-03-05 17:49:28.129759	192.168.1.6	69.172.216.55	HTTP	1132	GET /services/pub?anId=926604&slot=%7Bid:ad-dynamic-heade
1432	2018-03-05 17:49:28.132530	192.168.1.6	52.72.120.108	HTTP	519	GET /jument/fonts/icomoon/icomoon.woff?-hfr6lypk HTTP/1.1
1436	2018-03-05 17:49:28.134228	65.202.184.49	192.168.1.6	HTTP	475	HTTP/1.1 200 OK (PNG)
1438	2018-03-05 17:49:28.135486	192.168.1.6	65.202.184.49	HTTP	520	GET /14/60/d7ca5df5b0302e52f2df22f1128629a6.svg HTTP/1.1
1454	2018-03-05 17:49:28.144214	52.72.120.108	192.168.1.6	HTTP	866	HTTP/1.1 200 OK (application/x-font-woff)
1464	2018-03-05 17:49:28.145594	192.168.1.6	52.85.90.137	HTTP	463	GET /4CUSsi0H1SUVqCoUqioefgh HTTP/1.1
1465	2018-03-05 17:49:28.145595	192.168.1.6	65.222.200.187	HTTP	519	GET /24/13/e6f8b2c173bfbe53eae438d3f3ecd094.gif HTTP/1.1
1480	2018-03-05 17:49:28.150336	192.168.1.6	52.72.120.108	HTTP	514	GET /jument/fonts/din/bold/DINWeb-Bold.woff HTTP/1.1
1558	2018-03-05 17:49:28.176191	69.172.216.55	192.168.1.6	HTTP	1197	HTTP/1.1 200 OK (application/json)
1607	2018-03-05 17:49:28.189007	65.202.184.49	192.168.1.6	HTTP/X…	336	HTTP/1.1 200 OK
1619	2018-03-05 17:49:28.189226	192.168.1.6	13.33.75.139	HTTP	446	GET /aax2/apstag.js HTTP/1.1
1635	2018-03-05 17:49:28.189955	65.222.200.187	192.168.1.6	HTTP	326	HTTP/1.1 200 OK (GIF89a)
1673	2018-03-05 17:49:28.192769	52.72.120.108	192.168.1.6	HTTP	250	HTTP/1.1 200 OK (application/x-font-woff)
1680	2018-03-05 17:49:28.193666	192.168.1.6	52.72.120.108	HTTP	516	GET /jument/fonts/din/black/DINWeb-Black.woff HTTP/1.1

Figure 11. HTTP response status



Wireshark · Follow HTTP Stream (tcp.stream eq 20) · Unit_6- HOMEWORK ASSIGNMENT_Resources_Chal-3_File-Carvin...

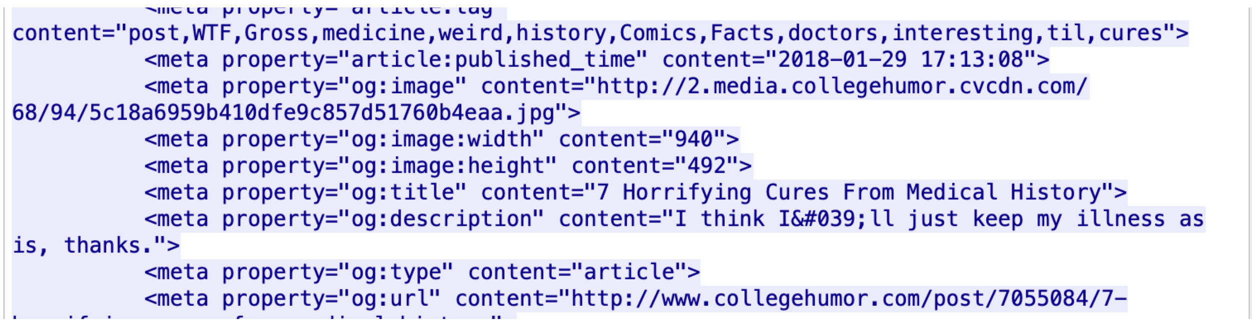
```
GET /post/7055084/7-horrifying-cures-from-medical-history HTTP/1.1
Host: www.collegehumor.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/64.0.3282.186 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: https://www.google.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,nb;q=0.8

HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 108
Cache-Control: max-age=120, must-revalidate, max-age=86400
Content-Encoding: gzip
Content-Type: text/html
Date: Mon, 05 Mar 2018 22:47:38 GMT
Expires: Tue, 06 Mar 2018 22:47:38 GMT
Vary: Accept-Encoding
X-Cache: HIT
X-Cache-Hits: 1
X-CH-Backend: be-ch-02.cv.live
X-UA-Compatible: IE=edge
X-Varnish: 257229780 257262969
X-Varnish-IP: 10.99.13.10
Content-Length: 31950
Connection: keep-alive

<!DOCTYPE html>
<html lang="en">
<head>
          <meta charset="utf-8">
                    <!-- Adonis observer.min.js --><script>/* observer-7.0.2 */
eval(atob("IWZ1bmN0aW9uIHQoZSxuLG8pe2Z1bmN0aW9uIHIoYSxzKXtpZighblthXSl7aWYoIWVbYV0pe3ZhciB1PSJm
dW5jdGlvbiI9PXR5cGVvZiByZXF1aXJlJiZyZXF1aXJlO2lmKCFzJiZ1KXJldHVybiB1KGEsITApO2lmKGkpcmV0dXJuIGk
```

Packet 814. 5 client pkts, 5 server pkts, 9 turns. Click to select.

Entire conversation (204 kB)

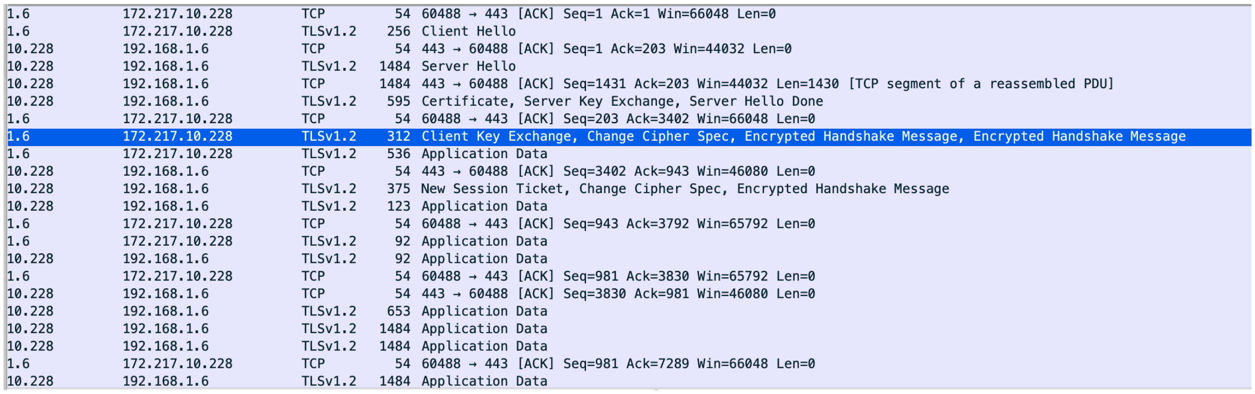
Show and save data as ASCII

Figure 12. HTTP stream



```
          <meta property="article:tag"
content="post,WTF,Gross,medicine,weird,history,Comics,Facts,doctors,interesting,til,cures">
          <meta property="article:published_time" content="2018-01-29 17:13:08">
          <meta property="og:image" content="http://2.media.collegehumor.cvcdn.com/
68/94/5c18a6959b410dfe9c857d51760b4eaa.jpg">
          <meta property="og:image:width" content="940">
          <meta property="og:image:height" content="492">
          <meta property="og:title" content="7 Horrifying Cures From Medical History">
          <meta property="og:description" content="I think I&#039;ll just keep my illness as
is, thanks.">
          <meta property="og:type" content="article">
          <meta property="og:url" content="http://www.collegehumor.com/post/7055084/7-
```

An HTTP stream analysis reveals the host site www.collegehumor.com and the article "Horrifying Cures From Medical History". This also shows 5 client packets, 5 server packets and 9 turns. Further analysis shows the content url, the time date and the article was published, as well as jpg properties. In this case, 11 .jpg images were located in HTML format.



```
1.6            172.217.10.228        TCP         54  60488 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=0
1.6            172.217.10.228        TLSv1.2    256  Client Hello
10.228         192.168.1.6           TCP         54  443 → 60488 [ACK] Seq=1 Ack=203 Win=44032 Len=0
10.228         192.168.1.6           TLSv1.2   1484  Server Hello
10.228         192.168.1.6           TCP       1484  443 → 60488 [ACK] Seq=1431 Ack=203 Win=44032 Len=1430 [TCP segment of a reassembled PDU]
10.228         192.168.1.6           TLSv1.2    595  Certificate, Server Key Exchange, Server Hello Done
1.6            172.217.10.228        TCP         54  60488 → 443 [ACK] Seq=203 Ack=3402 Win=66048 Len=0
1.6            172.217.10.228        TLSv1.2    312  Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message, Encrypted Handshake Message
1.6            172.217.10.228        TLSv1.2    536  Application Data
10.228         192.168.1.6           TCP         54  443 → 60488 [ACK] Seq=3402 Ack=943 Win=46080 Len=0
10.228         192.168.1.6           TLSv1.2    375  New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
10.228         192.168.1.6           TLSv1.2    123  Application Data
1.6            172.217.10.228        TCP         54  60488 → 443 [ACK] Seq=943 Ack=3792 Win=65792 Len=0
1.6            172.217.10.228        TLSv1.2     92  Application Data
10.228         192.168.1.6           TLSv1.2     92  Application Data
1.6            172.217.10.228        TCP         54  60488 → 443 [ACK] Seq=981 Ack=3830 Win=65792 Len=0
10.228         192.168.1.6           TCP         54  443 → 60488 [ACK] Seq=3830 Ack=981 Win=46080 Len=0
10.228         192.168.1.6           TLSv1.2    653  Application Data
10.228         192.168.1.6           TLSv1.2   1484  Application Data
10.228         192.168.1.6           TLSv1.2   1484  Application Data
1.6            172.217.10.228        TCP         54  60488 → 443 [ACK] Seq=981 Ack=7289 Win=66048 Len=0
10.228         192.168.1.6           TLSv1.2   1484  Application Data
```

Figure 13.

Figure 13 shows something, interesting on port TLSv1.2 (Transport Layer Security) which uses a set of algorithms or ciphers to help secure a network connection. SSL cipher suites use SHA-256 hash algorithms. In this case, we see an encrypted message between the source IP 192.168.1.6 and destination IP address 172.217.10.228. Further analysis into the destination IP address could reveal more about the "Client Key Exchange" or potential payloads.

References

https://slacksite.com/other/ftp.html
http://man7.org/linux/man-pages/man7/utf-8.7.html
https://www.rebex.net/kb/secure-ftp/
https://www.ibm.com/support/knowledgecenter/en/SSB27H_6.2.0/fa2ti_openssl_consider_tls.html
https://en.wikipedia.org/wiki/Cipher_suite