Chrisin Jacob

Syed Farabi

Wend Tin Basile Sam

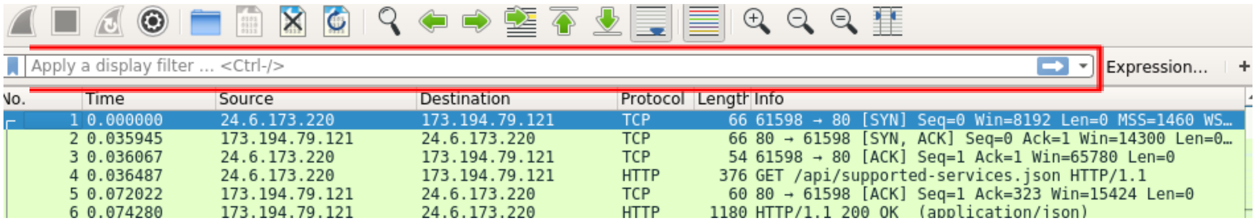
Lab 3 - Wireshark

November 9, 2023

CSCI 400

Wireshark Introduction

Task 3.3: Find A Specific Packet

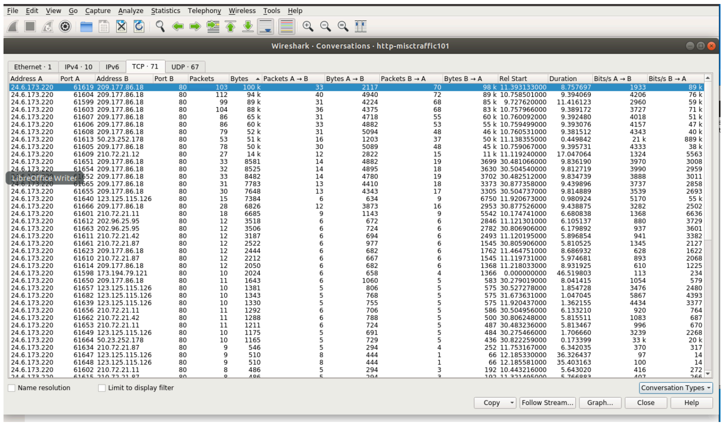


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	24.6.173.220	173.194.79.121	TCP	66	61598 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS…
2	0.035945	173.194.79.121	24.6.173.220	TCP	66	80 → 61598 [SYN, ACK] Seq=0 Ack=1 Win=14300 Len=0…
3	0.036067	24.6.173.220	173.194.79.121	TCP	54	61598 → 80 [ACK] Seq=1 Ack=1 Win=65780 Len=0
4	0.036487	24.6.173.220	173.194.79.121	HTTP	376	GET /api/supported-services.json HTTP/1.1
5	0.072022	173.194.79.121	24.6.173.220	TCP	60	80 → 61598 [ACK] Seq=1 Ack=323 Win=15424 Len=0
6	0.074280	173.194.79.121	24.6.173.220	HTTP	1180	HTTP/1.1 200 OK (application/json)

To find a specific packet, we use the research (Apply a display filter) bar to apply a filter that will help us to find the specific file. For example, we can the following syntax "*telnet && (telnet.data contains "john" && telnet.data contains "password")*" within the search bar to locate a single packet which contains the John's provided password when he attempts to use Telnet to log in as the "john" user.

Packet Introspection

Task 3.1: Find the Most Active TCP Flow



Address A	Port A	Address B	Port B	Packets	Bytes ▲	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
24.6.173.220	61619	209.177.86.18	80	103	100 k	33	2117	70	98 k	11.393133000	8.757697	1933	89 k
24.6.173.220	61604	209.177.86.18	80	112	94 k	40	4940	72	89 k	10.758501000	9.394069	4206	76 k
24.6.173.220	61599	209.177.86.18	80	99	89 k	31	4224	68	85 k	9.727620000	11.416123	2960	59 k
24.6.173.220	61603	209.177.86.18	80	104	88 k	36	4375	68	83 k	10.757966000	9.389172	3727	71 k
24.6.173.220	61607	209.177.86.18	80	86	65 k	31	4718	55	60 k	10.760092000	9.392480	4018	51 k
24.6.173.220	61606	209.177.86.18	80	86	60 k	33	4882	53	55 k	10.759499000	9.393076	4157	47 k
24.6.173.220	61608	209.177.86.18	80	79	52 k	31	5094	48	46 k	10.760531000	9.381512	4343	40 k
24.6.173.220	61613	50.23.252.178	80	53	51 k	16	1203	37	50 k	11.138355000	0.449842	21 k	889 k
24.6.173.220	61605	209.177.86.18	80	78	50 k	30	5089	48	45 k	10.759067000	9.395731	4333	38 k
24.6.173.220	61609	210.72.21.12	80	27	14 k	12	2822	15	11 k	11.119240000	17.047064	1324	5563
24.6.173.220	61651	209.177.86.18	80	33	8581	14	4882	19	3699	30.481066000	9.836190	3970	3008
24.6.173.220	61654	209.177.86.18	80	32	8525	14	4895	18	3630	30.504540000	9.812719	3990	2959
24.6.173.220	61652	209.177.86.18	80	33	8482	14	4780	19	3702	30.482512000	9.834739	3888	3011
24.6.173.220	61665	209.177.86.18	80	31	7783	13	4410	18	3373	30.877358000	9.439896	3737	2858
24.6.173.220	61655	209.177.86.18	80	30	7648	13	4343	17	3305	30.504737000	9.814889	3539	2693
24.6.173.220	61640	123.125.115.126	80	15	7384	6	634	9	6750	11.920673000	0.980924	5170	55 k
24.6.173.220	61666	209.177.86.18	80	28	6826	12	3873	16	2953	30.877526000	9.438875	3282	2502
24.6.173.220	61601	210.72.21.11	80	18	6685	9	1143	9	5542	10.174741000	6.680838	1368	6636
24.6.173.220	61612	202.96.25.95	80	12	3518	6	672	6	2846	11.121301000	6.105137	880	3729
24.6.173.220	61663	202.96.25.95	80	12	3506	6	724	6	2782	30.806906000	6.179892	937	3601
24.6.173.220	61611	210.72.21.42	80	12	3187	6	694	6	2493	11.120195000	5.896854	941	3382
24.6.173.220	61661	210.72.21.87	80	12	2522	6	977	6	1545	30.805906000	5.810525	1345	2127
24.6.173.220	61623	209.177.86.18	80	12	2444	6	682	6	1762	11.464751000	8.686932	628	1622
24.6.173.220	61610	210.72.21.87	80	12	2212	6	667	6	1545	11.119731000	5.974681	893	2068
24.6.173.220	61614	209.177.86.18	80	12	2050	6	682	6	1368	11.218033000	8.931925	610	1225
24.6.173.220	61598	173.194.79.121	80	10	2024	6	658	4	1366	0.000000000	46.519803	113	234
24.6.173.220	61650	209.177.86.18	80	11	1643	6	1060	5	583	30.279019000	8.041415	1054	579
24.6.173.220	61657	123.125.115.126	80	10	1381	5	806	5	575	30.527278000	1.854728	3476	2480
24.6.173.220	61682	123.125.115.126	80	10	1343	5	768	5	575	31.673631000	1.047045	5867	4393
24.6.173.220	61639	123.125.115.126	80	10	1330	5	755	5	575	11.920437000	1.362155	4434	3377
24.6.173.220	61656	210.72.21.11	80	11	1292	6	706	5	586	30.504956000	6.133210	920	764
24.6.173.220	61662	210.72.21.42	80	11	1288	6	788	5	500	30.806248000	5.815511	1083	687
24.6.173.220	61653	210.72.21.11	80	11	1211	6	724	5	487	30.483236000	5.813467	996	670
24.6.173.220	61649	123.125.115.126	80	10	1175	5	691	5	484	30.275466000	1.706660	3239	2268
24.6.173.220	61664	50.23.252.178	80	10	1165	5	729	5	436	30.822259000	0.173399	33 k	20 k
24.6.173.220	61634	210.72.21.87	80	9	546	5	294	4	252	11.753167000	6.342035	370	317
24.6.173.220	61647	123.125.115.126	80	9	510	8	444	1	66	12.185330000	36.326437	97	14
24.6.173.220	61648	123.125.115.126	80	9	510	8	444	1	66	12.185581000	35.403163	100	14
24.6.173.220	61602	210.72.21.11	80	8	486	5	294	3	192	10.443216000	5.643020	416	272
24.6.173.220	61615	210.72.21.87	80	8	486	5	294	3	192	11.321495000	5.766883	407	266

□ Name resolution □ Limit to display filter

Once we open our pcaps file (by using the command wireshark pcaps/http-misctraffic101.pcapng), then we click on the tab statistics, and then conversations. It opens the conversations window where we can find all the details about the traffic. Since we are looking for the most active tcp flow, we click on the TCP tab and filter data by the size of its bytes. When we apply the filter to analyze the packet traffic between A and B, we are able to observe all the conversations that occurred between the two. We can also filter data using directly the filter syntax on the research bar on wireshark's main page of the pcap file: ip.addr==24.6.173.220 && tcp.port==61639 && ip.addr==123.125.115.126 && tcp.port==80

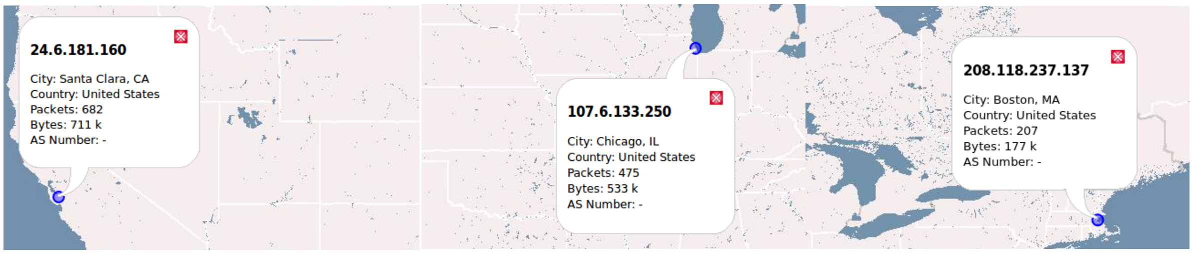
3.1 – 1:

Based on the bytes count, <24.6.173.220> and <209.177.86.18> are the addresses that participated in the most active IPv4 conversation.

3.1 – 2:

Number of packets in the Conversation: 103 Packets

Task 3.2: Geolocating IP Addresses



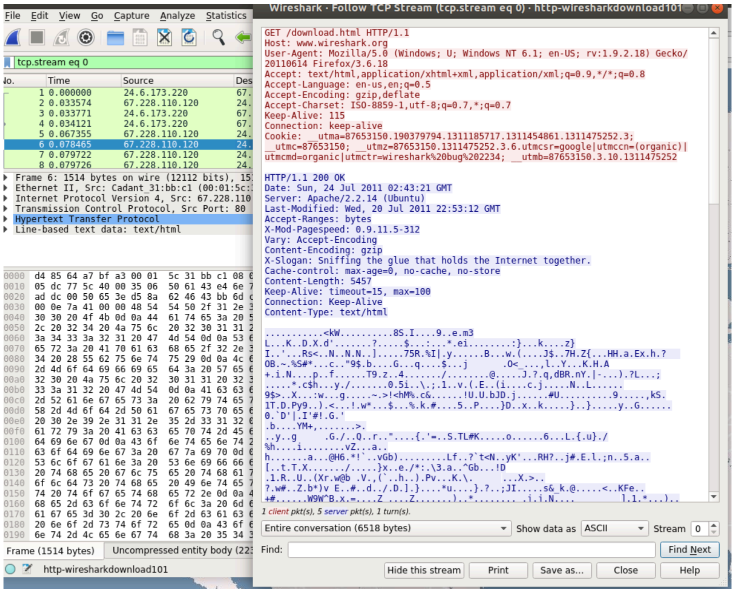
Thanks to data from MaxMind2, we were able to locate three geographic points on the map thanks to their IP addresses, which were displayed on the map. The three localities concerned are Santa Clara (CA), Chicago (IL), and Boston (MA).

3.2 – 3:

Packet Count: 682

Byte: 711k

Task 3.3: Reassemble text from TCP stream



tcp.stream eq 0

```
No.    Time       Source           Des
     1 0.000000   24.6.173.220     67.
     2 0.033574   67.228.110.120   24.
     3 0.033771   24.6.173.220     67.
     4 0.034121   24.6.173.220     67.
     5 0.067355   67.228.110.120   24.
     6 0.078465   67.228.110.120   24.
     7 0.079722   67.228.110.120   24.
     8 0.079726   67.228.110.120   24.
```

```
▸ Frame 6: 1514 bytes on wire (12112 bits), 15
▸ Ethernet II, Src: Cadant_31:bb:c1 (00:01:5c:
▸ Internet Protocol Version 4, Src: 67.228.110
▸ Transmission Control Protocol, Src Port: 80
▸ Hypertext Transfer Protocol
▸ Line-based text data: text/html
```

```
0000  d4 85 64 a7 bf a3 00 01  5c 31 bb c1 08 0
0010  05 dc 77 5c 40 00 35 06  50 61 43 e4 6e 7
0020  ad dc 00 50 65 3e d5 8a  62 46 43 bb 6d c
0030  00 0e 7a 41 00 00 48 54  54 50 2f 31 2e 3
0040  30 30 20 4f 4b 0d 0a 44  61 74 65 3a 20 5
0050  2c 20 32 34 20 4a 75 6c  20 32 30 31 31 2
0060  3a 34 33 3a 32 31 20 47  4d 54 0d 0a 53 6
0070  65 72 3a 20 41 70 61 63  68 65 2f 32 2e 3
0080  34 20 28 55 62 75 6e 74  75 29 0d 0a 4c 6
0090  2d 4d 6f 64 69 66 69 65  64 3a 20 57 65 6
00a0  32 30 20 4a 75 6c 20 32  30 31 31 20 32 3
00b0  33 3a 31 32 20 47 4d 54  0d 0a 41 63 63 6
00c0  2d 52 61 6e 67 65 73 3a  20 62 79 74 65 7
00d0  58 2d 4d 6f 64 2d 50 61  67 65 73 70 65 6
00e0  20 30 2e 39 2e 31 31 2e  35 2d 33 31 32 0
00f0  61 72 79 3a 20 41 63 63  65 70 74 2d 45 6
0100  64 69 6e 67 0d 0a 43 6f  6e 74 65 6e 74 2
0110  63 6f 64 69 6e 67 3a 20  67 7a 69 70 0d 0
0120  53 6c 6f 67 61 6e 3a 20  53 6e 69 66 66 6
0130  20 74 68 65 20 67 6c 75  65 20 74 68 61 7
0140  6f 6c 64 73 20 74 68 65  20 49 6e 74 65 7
0150  74 20 74 6f 67 65 74 68  65 72 2e 0d 0a 4
0160  68 65 2d 63 6f 6e 74 72  6f 6c 3a 20 6d 6
0170  61 67 65 3d 30 2c 20 6e  6f 2d 63 61 63 6
0180  20 6e 6f 2d 73 74 6f 72  65 0d 0a 43 6f 6
0190  6e 74 2d 4c 65 6e 67 74  68 3a 20 35 34 3
```

Frame (1514 bytes) | Uncompressed entity body (223

http-wiresharkdownload101

```
GET /download.html HTTP/1.1
Host: www.wireshark.org
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.18) Gecko/
20110614 Firefox/3.6.18
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: 115
Connection: keep-alive
Cookie: __utma=87653150.190379794.1311185717.1311454861.1311475252.3;
__utmc=87653150; __utmz=87653150.1311475252.3.6.utmcsr=google|utmccn=(organic)|
utmcmd=organic|utmctr=wireshark%20bug%202234; __utmb=87653150.3.10.1311475252

HTTP/1.1 200 OK
Date: Sun, 24 Jul 2011 02:43:21 GMT
Server: Apache/2.2.14 (Ubuntu)
Last-Modified: Wed, 20 Jul 2011 22:53:12 GMT
Accept-Ranges: bytes
X-Mod-Pagespeed: 0.9.11.5-312
Vary: Accept-Encoding
Content-Encoding: gzip
X-Slogan: Sniffing the glue that holds the Internet together.
Cache-control: max-age=0, no-cache, no-store
Content-Length: 5457
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

...........<kW..........8S.I....9..e.m3
L...K..D.X.d'.......?.....$...:...*.ei........:}...k....z}
I..'...Rs<..N..N.N..]......75R.%I|.y......B...w.(....J$..7H.Z{...HH.a.Ex.h.?
OB.~.%S#*...c.."9$.b....G...q....$...j       .O<_...,l..Y...K.H.A
+.i.N....p..f......T9.z..4......./........@.....J.?.q,dBR.nY.|-...).?L...;
.....*.c$h...y./.......0.5i..\.;.1..v.(.E..(i....c.j.....N..L......
9$>..X...:w...g.....~.>!<hM%.c&......!U.U.bJD.j......#U..........9.....,kS.
1T.D.Py9..).<...!.w*...$...%.k.#....5..P....}D..x..k.....}..}.....y..G......
0.`D'|.I'#!.G.'
.b....YM+,.......>.
..y..g     .G./..Q..r.."....{.'=..S.TL#K.....o......6...L.{.u}./
%h....i........vZ...a..
h.......a...@H6.*!`..vGb).........Lf..?`t<N..yK'...RH?..j#.E.l.;n..5.a..
[..t.T.X......./.....}x..e./*:.\3.a..^Gb...!D
.1.R..U..(Xr.w@b .V.,(`..h..).Pv...K.\.       ...X.>..
?.w#..Z.b*)v E..#..d../.D.].}....*u....}.?..;JI.....s&_k.@.....<..KFe..
+#......W9W^B.x.=....Z ....Z.......)..*........ .i.i.N....          ].1.*...)..
```

1 client pkt(s), 5 server pkt(s), 1 turn(s).

Entire conversation (6518 bytes) ▾ Show data as ASCII ▾ Stream 0

Find:

At this step, we have to reassemble a text from the TCP stream. To do it, once we open the packet, we right-click on frame 4, scroll down to follow, and select tcp stream. It opens the conversation stream that we have in the screenshot above. From this conversation stream, we can then detect the hidden message that is starting with X-Slogan. Such as we can apply the filter *<frame contains "X-Slogan">* to find the hidden message.

3.3 – 4:

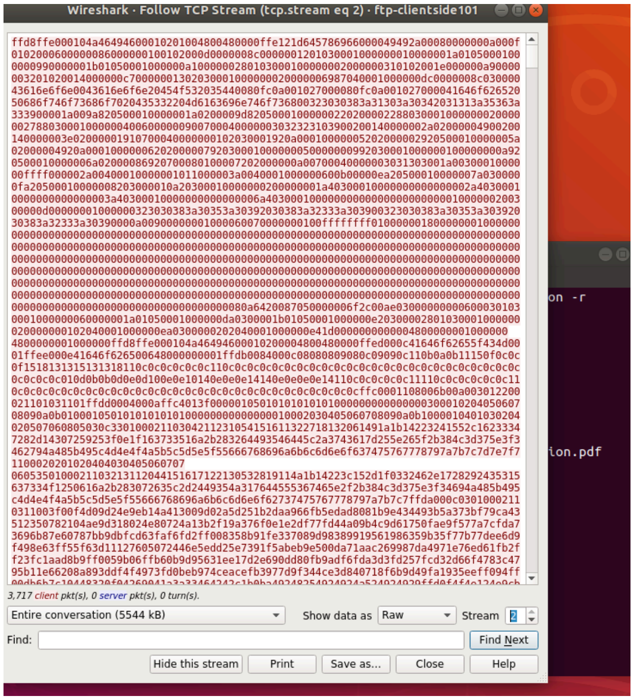
Frame 4 Hidden Message: < *Sniffing the glue that holds the Internet together* >

3.3 – 5:

Other Hidden Message: < *Sniff free or die.* >

Frame Number: < 28 >

Task 3.4: Extract binary file from FTP session


Wireshark · Follow TCP Stream (tcp.stream eq 2) · ftp-clientside101
on -r
ion.pdf

```
ffd8ffe000104a46494600010201004800480000ffe121d6457869660000049492a00080000000a000f
0102000600000086000000100102000d0000008c0000001201030001000000010000001a0105000100
0000990000001b01050001000000a10000002801030001000000020000003101020001e000000a90000
00320102001400000c7000000130203000100000002000000698704000100000dc0000008c030000
43616e6f6e0043616e6f6e20454f53203544080fc0a001027000080fc0a00102700004164f626520
50686f746f73686f7020435332204d6163696e746f7368003230303a83a31303a30342031313a35363a
333900001a009a8205000100000001a0200009d82050001000000220200002288030001000000020000
002788030001000000400600000090070004000000030323231039002001400000002a02000004900200
140000003e020000019107000400000001020300019 20a0001000000052020000029205000100000005a
020000004920a0001000000620200000792030001000000050000000992030001000000100000000a92
0500010000006a020000869207000801000072020000000a00700040000003031303001a00300010000
00ffff000002a0040001000000101100003a00400010000000600b00000ea20500010000007a030000
0fa2050001000000820300010a20300010000000200000001a40300010000000000000002a4030001
0000000000003a40300010000000000000006a403000100000000000000000000000001000000200 3
00000d000000010000003230303 83a30353a30392030383a32333a303900323030383a30353a303920
30383a32333a30390000a009000000001000060070000000100ffffffff0100000018000000001000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000080a6420087050000006f2c00ae030000000000600030103
000100000006000001a01050001000000da0300001b01050001000000e20300002801030001000000
0200000001020400010000000ea0300000202040001000000e41d000000000000048000000001000000
4800000001000000ffd8ffe000104a46494600010200004800480000ffed000c41646f62655f434d00
01ffee000e41646f626500648000000001ffdb0084000c08080809080c09090c110b0a0b11150f0c0c
0f15181313151313181110c0c0c0c0c0c110c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c
0c0c0c0c010d0b0b0d0e0d100e0e10140e0e0e14140e0e0e0e14110c0c0c0c0c11110c0c0c0c0c0c11
0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0cffc0001108006b00a003012200
021101031101ffdd0004000affc4013f0000010501010101010100000000000000000300010204050607
08090a0b010001050101010101010000000000000000010002030405060708090a0b1000010401030204
020507060805030c33010002110304211231054151611322718132061491a1b14223241552c1623334
7282d14307259253f0e1f163733516a2b283264493546445c2a3743617d255e265f2b384c3d375e3f3
462794a485b495c4d4e4f4a5b5c5d5e5f55666768696a6b6c6d6e6f637475767778797a7b7c7d7e7f7
110002020102040403040506070 7
06053501000211032131120441516171221305328191 14a1b14223c152d1f0332462e1728292435315
637334f1250616a2b283072635c2d2449354a3176445553674 65e2f2b384c3d375e3f34694a485b495
c4d4e4f4a5b5c5d5e5f55666768696a6b6c6d6e6f62737475767778797a7b7c7ffda000c0301000211
0311003f00f4d09d24e9eb14a413009d02a5d251b2daa966fb5edad8081b9e434493b5a373bf79ca43
51235078210 4ae9d318024e80724a13b2f19a376f0e1e2df77fd44a09b4c9d61750fae9f577a7cfda7
3696b87e60787bb9dbfcd63faf6fd2ff008358b91fe337089d98389919561986359b35f77b77dee6d9
f498e63ff55f63d11127605072446e5edd25e7391f5abeb9e500da71aac269987da4971e76ed61fb2f
f23fc1aad8b9ff0059b06ffb60b9d95631ee17d2e690dd80fb9adf6fda3d3fd257fcd32d66f4783c47
95b11e66208a893ddf4f4973fd0beb974ceacefb3977d9f344ce3d840718f6b9d49fa1935eeff094ff
00db6b7c10448320f04269041a3a33464242c1b0ba492482549249249a524924929ffd0f4f4e124e9cb
```

3,717 client pkt(s), 0 server pkt(s), 0 turn(s).

Entire conversation (5544 kB) Show data as Raw Stream 2

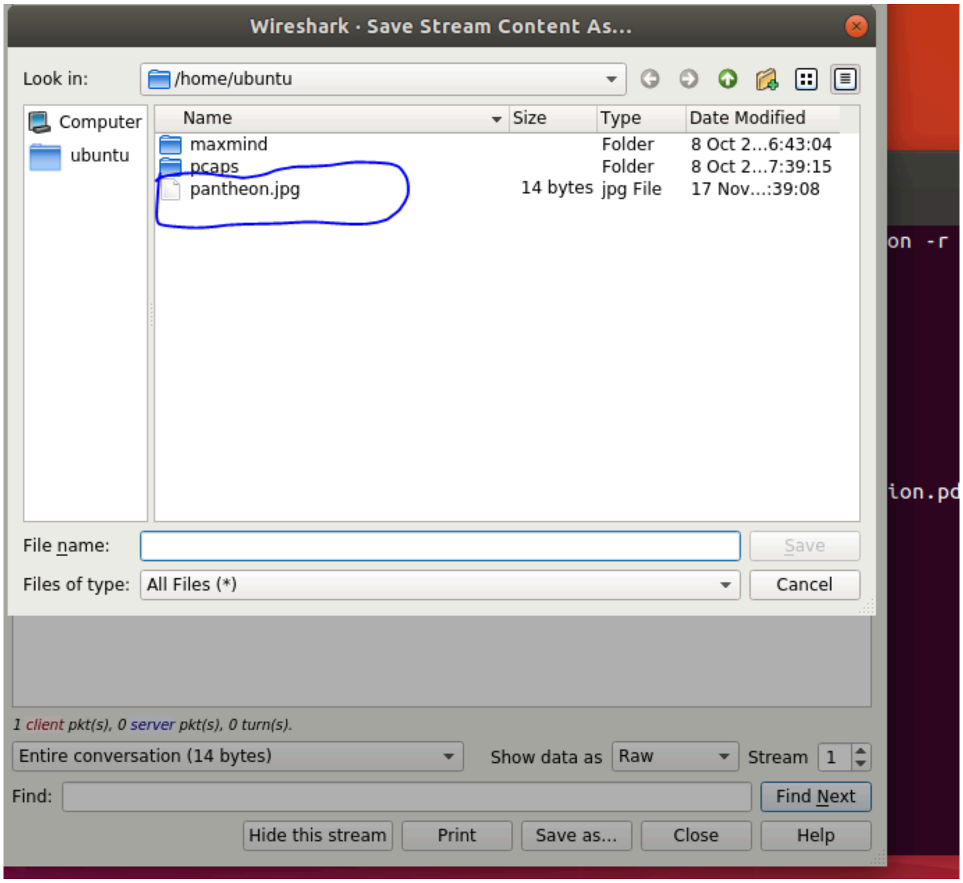
Find: Find Next

Hide this stream | Print | Save as... | Close | Help

To find a binary file from the FTP session, we select the specific packet followed by a right-click, scroll down to follow, and click on tcp stream. Once the conversation stream is opened, we go down to the bottom of the page and change the <show data as> into a raw version. It allows users to see the data inside into a binary format as shown in the screenshot above.

3.4 – 6:

Name of file: < pantheon.jpg>



Wireshark · Save Stream Content As...

Look in: /home/ubuntu

Name	Size	Type	Date Modified
maxmind		Folder	8 Oct 2...6:43:04
pcaps		Folder	8 Oct 2...7:39:15
pantheon.jpg	14 bytes	jpg File	17 Nov...:39:08

File name:

Files of type: All Files (*)

1 client pkt(s), 0 server pkt(s), 0 turn(s).

Entire conversation (14 bytes)

Show data as Raw

Stream 1

Find: