Professional Masters in Information and Cyber Security	
Course	CSE 805 Digital Forensic
Teaching Assistant	Sahebul Karim
Email	sahebul.du@gmail.com
Lab	05 Network Forensic

Lab 05: Network Forensic

- Investigating Network Traffic Using Wireshark

Wireshark

Wireshark is a powerful network protocol analyzer used for analyzing and troubleshooting networks. It allows users to capture and interactively browse the traffic running on a computer network in real-time. This tool is widely used by network administrators, security professionals, developers, and researchers to inspect network packets, understand protocols, and diagnose network issues.

Here's a breakdown of what Wireshark does:

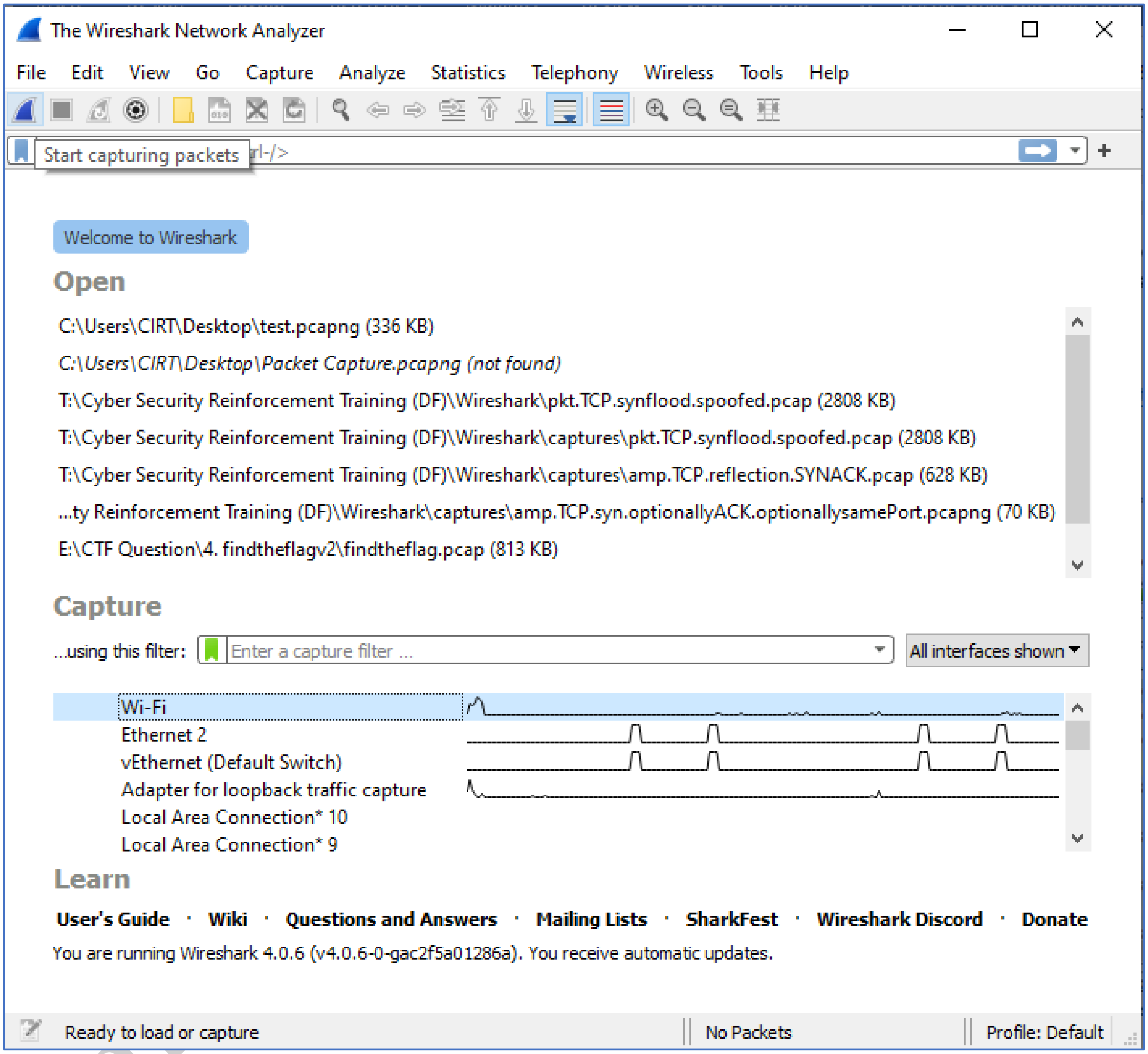
1. **Packet Capture:** Wireshark captures data packets traveling through a network interface, showing detailed information about each packet's contents, source, destination, protocols used, and more.
2. **Protocol Analysis:** It dissects and interprets these captured packets, allowing users to analyze different network protocols (HTTP, TCP, UDP, DNS, etc.), inspecting their headers, payloads, and interactions.
3. **Troubleshooting:** Network administrators often use Wireshark to diagnose network problems, identify irregularities, and pinpoint the source of issues like network congestion, latency, or misconfigurations.
4. **Security Analysis:** Security professionals use Wireshark to detect suspicious or malicious network traffic, analyze potential attacks, and understand the behavior of malware or cyber threats.

Wireshark provides a user-friendly interface that allows users to filter and search packets, visualize data flows, and export captured information for further analysis or reporting. Its versatility and comprehensive features make it an essential tool in network analysis and troubleshooting.

Download

https://www.wireshark.org/download.html

Installation



C:\Users\CIRT\Desktop\test.pcapng (336 KB)

C:\Users\CIRT\Desktop\Packet Capture.pcapng (not found)

T:\Cyber Security Reinforcement Training (DF)\Wireshark\pkt.TCP.synflood.spoofed.pcap (2808 KB)

T:\Cyber Security Reinforcement Training (DF)\Wireshark\captures\pkt.TCP.synflood.spoofed.pcap (2808 KB)

T:\Cyber Security Reinforcement Training (DF)\Wireshark\captures\amp.TCP.reflection.SYNACK.pcap (628 KB)

...ty Reinforcement Training (DF)\Wireshark\captures\amp.TCP.syn.optionallyACK.optionallysamePort.pcapng (70 KB)

E:\CTF Question\4. findtheflagv2\findtheflag.pcap (813 KB)

Wi-Fi
Ethernet 2
vEthernet (Default Switch)
Adapter for loopback traffic capture
Local Area Connection* 10
Local Area Connection* 9

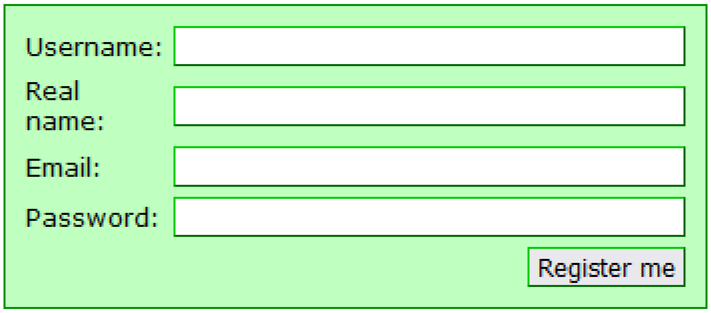
You are running Wireshark 4.0.6 (v4.0.6-0-gac2f5a01286a). You receive automatic updates.

- Wireshark captures all traffic on a network interface.
- Wireshark can capture HTTPS traffic, but it **cannot capture the actual payload of the encrypted HTTPS packets** unless you have the encryption keys. This is because HTTPS encrypts the data exchanged between the client and the server using SSL/TLS encryption, making it unreadable to anyone intercepting the traffic without the proper keys.

HTTPS provides a secure communication channel over an insecure network (like the internet) by encrypting the data exchanged between the client and the server. This encryption ensures that information such as login credentials, personal data, and sensitive information remains confidential during transmission.

1. Investigating HTTP traffic

1. Start capturing the interface traffic using Wireshark.
2. Go to the below link
   http://testasp.vulnweb.com/Register.asp?RetURL=%2FDefault%2Easp%3F
3. Fill up the form
   **use a dummy password



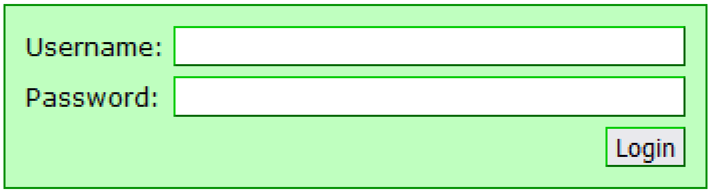
Username:

Real name:

Email:

Password:

4. Click Register me
5. Put user name and password
6. Click Login



Username:

Password:

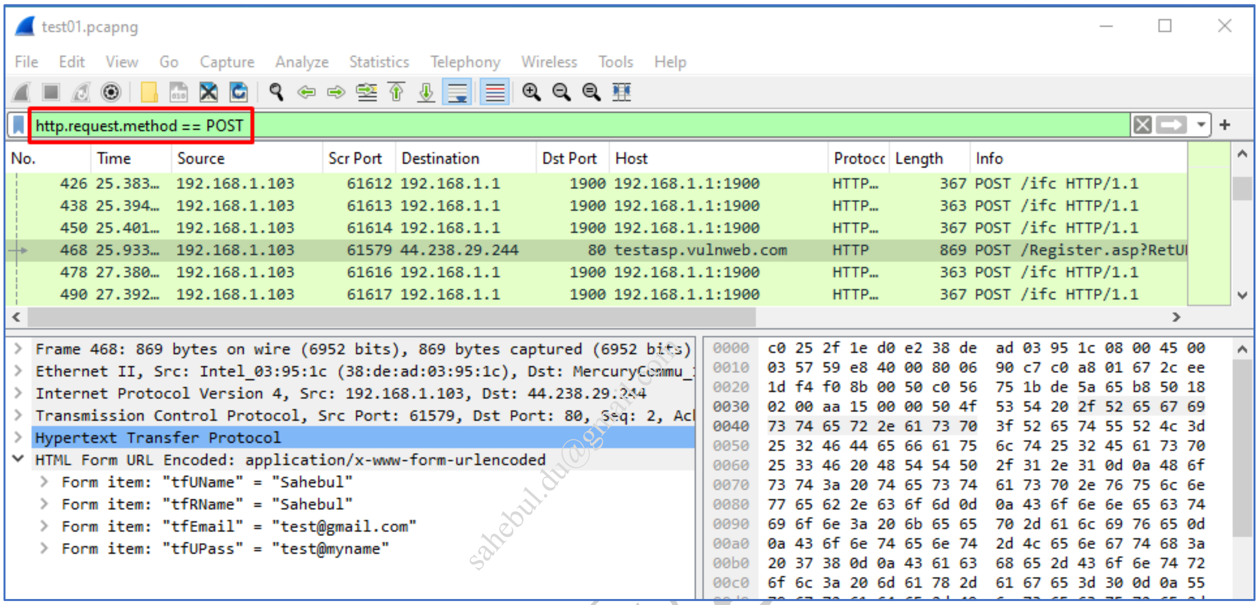
7. Stop capture and save as test01.pcapng.

Question 01:

a. Investigate the traffic to see if there are any plain text passwords stored in it?
b. Find the website IP?

Answer 01

- Generally, user credentials are stored in the POST requests. so, examining the packet containing the POST request help find the user credentials.
- Search for post request or use **http.request.method == POST**



```
http.request.method == POST
```

```
No.   Time       Source         Scr Port Destination    Dst Port Host                 Protocc Length Info
  426 25.383…   192.168.1.103    61612 192.168.1.1       1900 192.168.1.1:1900     HTTP…      367 POST /ifc HTTP/1.1
  438 25.394…   192.168.1.103    61613 192.168.1.1       1900 192.168.1.1:1900     HTTP…      363 POST /ifc HTTP/1.1
  450 25.401…   192.168.1.103    61614 192.168.1.1       1900 192.168.1.1:1900     HTTP…      367 POST /ifc HTTP/1.1
  468 25.933…   192.168.1.103    61579 44.238.29.244       80 testasp.vulnweb.com  HTTP       869 POST /Register.asp?RetU
  478 27.380…   192.168.1.103    61616 192.168.1.1       1900 192.168.1.1:1900     HTTP…      363 POST /ifc HTTP/1.1
  490 27.392…   192.168.1.103    61617 192.168.1.1       1900 192.168.1.1:1900     HTTP…      367 POST /ifc HTTP/1.1
```

```
> Frame 468: 869 bytes on wire (6952 bits), 869 bytes captured (6952 bits)
> Ethernet II, Src: Intel_03:95:1c (38:de:ad:03:95:1c), Dst: MercuryCommu_
> Internet Protocol Version 4, Src: 192.168.1.103, Dst: 44.238.29.244
> Transmission Control Protocol, Src Port: 61579, Dst Port: 80, Seq: 2, Ack
> Hypertext Transfer Protocol
v HTML Form URL Encoded: application/x-www-form-urlencoded
   > Form item: "tfUName" = "Sahebul"
   > Form item: "tfRName" = "Sahebul"
   > Form item: "tfEmail" = "test@gmail.com"
   > Form item: "tfUPass" = "test@myname"
```

```
0000  c0 25 2f 1e d0 e2 38 de  ad 03 95 1c 08 00 45 00
0010  03 57 59 e8 40 00 80 06  90 c7 c0 a8 01 67 2c ee
0020  1d f4 f0 8b 00 50 c0 56  75 1b de 5a 65 b8 50 18
0030  02 00 aa 15 00 00 50 4f  53 54 20 2f 52 65 67 69
0040  73 74 65 72 2e 61 73 70  3f 52 65 74 55 52 4c 3d
0050  25 32 46 44 65 66 61 75  6c 74 25 32 45 61 73 70
0060  25 33 46 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f
0070  73 74 3a 20 74 65 73 74  61 73 70 2e 76 75 6c 6e
0080  77 65 62 2e 63 6f 6d 0d  0a 43 6f 6e 6e 65 63 74
0090  69 6f 6e 3a 20 6b 65 65  70 2d 61 6c 69 76 65 0d
00a0  0a 43 6f 6e 74 65 6e 74  2d 4c 65 6e 67 74 68 3a
00b0  20 37 38 0d 0a 43 61 63  68 65 2d 43 6f 6e 74 72
00c0  6f 6c 3a 20 6d 61 78 2d  61 67 65 3d 30 0d 0a 55
```

Website IP: 44.238.29.244

2. Investigating HTTP traffic

Open the below log file in Wireshark: **Investigating http traffic 01.log**

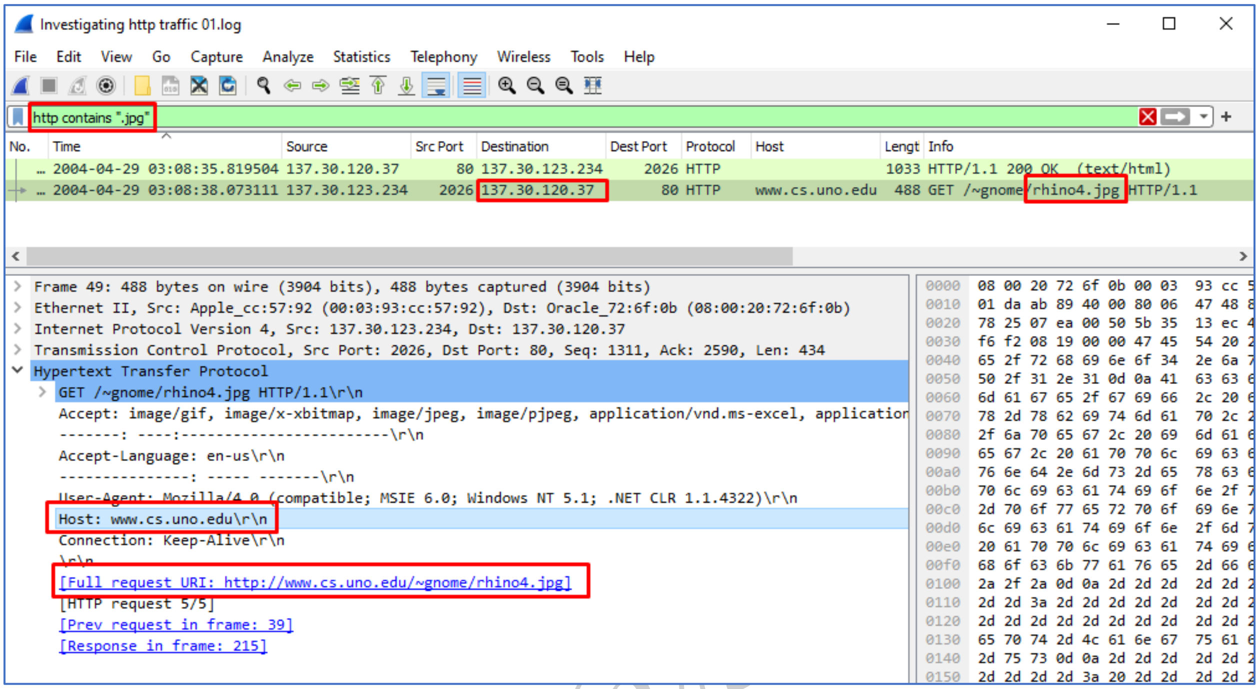
Question 02:

Any activities of uploading or downloading images using HTTP?

a. **Website name?**
b. **Website IP address?**
c. **Who is the owner of this IP (organization name)?**
d. **Dump the Image file and Generate MD5 hash of the image file?**

Answer 02

http contains ".jpg"



Investigating http traffic 01.log
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http contains ".jpg"

```
No. Time                         Source          Src Port Destination    Dest Port Protocol Host            Lengt Info
 … 2004-04-29 03:08:35.819504 137.30.120.37         80 137.30.123.234      2026 HTTP                           1033 HTTP/1.1 200 OK  (text/html)
 … 2004-04-29 03:08:38.073111 137.30.123.234      2026 137.30.120.37         80 HTTP     www.cs.uno.edu   488 GET /~gnome/rhino4.jpg HTTP/1.1
```

```
> Frame 49: 488 bytes on wire (3904 bits), 488 bytes captured (3904 bits)
> Ethernet II, Src: Apple_cc:57:92 (00:03:93:cc:57:92), Dst: Oracle_72:6f:0b (08:00:20:72:6f:0b)
> Internet Protocol Version 4, Src: 137.30.123.234, Dst: 137.30.120.37
> Transmission Control Protocol, Src Port: 2026, Dst Port: 80, Seq: 1311, Ack: 2590, Len: 434
v Hypertext Transfer Protocol
  > GET /~gnome/rhino4.jpg HTTP/1.1\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application
    -------: ----:-----------------------\r\n
    Accept-Language: en-us\r\n
    ---------------: ----- -------\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)\r\n
    Host: www.cs.uno.edu\r\n
    Connection: Keep-Alive\r\n
    \r\n
    [Full request URI: http://www.cs.uno.edu/~gnome/rhino4.jpg]
    [HTTP request 5/5]
    [Prev request in frame: 39]
    [Response in frame: 215]
```

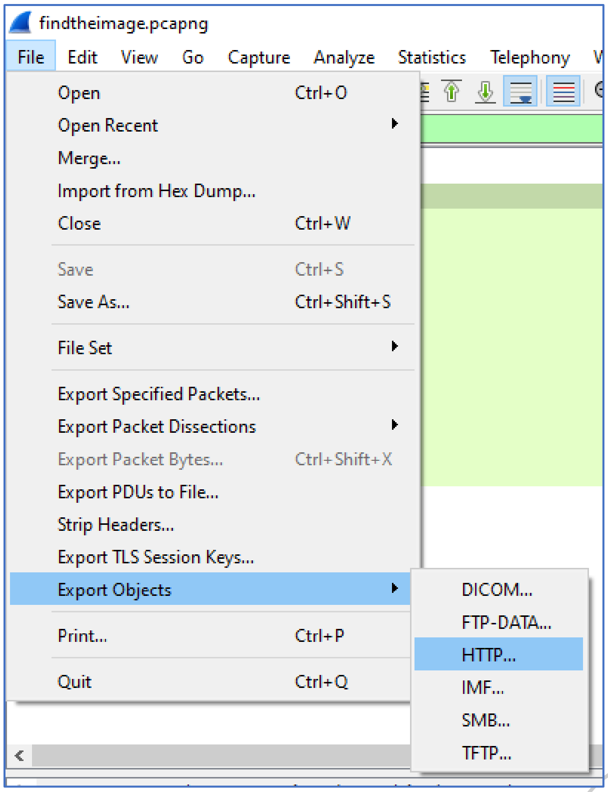
```
0000  08 00 20 72 6f 0b 00 03  93 cc 5
0010  01 da ab 89 40 00 80 06  47 48 8
0020  78 25 07 ea 00 50 5b 35  13 ec 4
0030  f6 f2 08 19 00 00 47 45  54 20 2
0040  65 2f 72 68 69 6e 6f 34  2e 6a 7
0050  50 2f 31 2e 31 0d 0a 41  63 63 6
0060  6d 61 67 65 2f 67 69 66  2c 20 6
0070  78 2d 78 62 69 74 6d 61  70 2c 2
0080  2f 6a 70 65 67 2c 20 69  6d 61 6
0090  65 67 2c 20 61 70 70 6c  69 63 6
00a0  76 6e 64 2e 6d 73 2d 65  78 63 6
00b0  70 6c 69 63 61 74 69 6f  6e 2f 7
00c0  2d 70 6f 77 65 72 70 6f  69 6e 7
00d0  6c 69 63 61 74 69 6f 6e  2f 6d 7
00e0  20 61 70 70 6c 69 63 61  74 69 6
00f0  68 6f 63 6b 77 61 76 65  2d 66 6
0100  2a 2f 2a 0d 0a 2d 2d 2d  2d 2d 2
0110  2d 2d 3a 2d 2d 2d 2d 2d  2d 2d 2
0120  2d 2d 2d 2d 2d 2d 2d 2d  2d 2d 2
0130  65 70 74 2d 4c 61 6e 67  75 61 6
0140  2d 75 73 0d 0a 2d 2d 2d  2d 2d 2
0150  2d 2d 2d 2d 3a 20 2d 2d  2d 2d 2
```

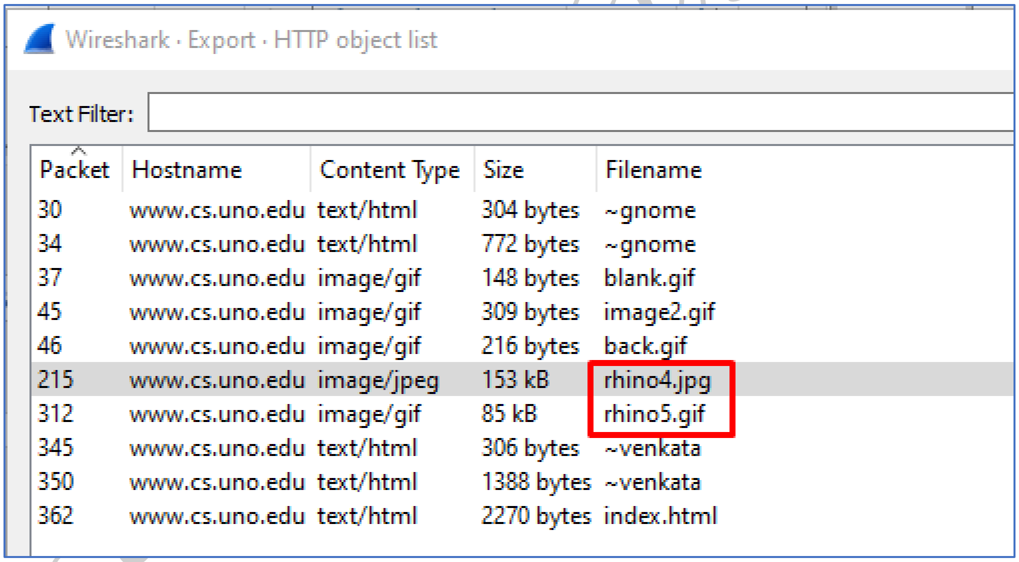
a. Website name?
   www.cs.uno.edu

b. Website IP address?
   137.30.120.37

c. Who is the owner of this IP (organization name)?
   Use whois to find out the IP details:
   Organization: University of New Orleans (UNO)

d. Dump the Image file and Generate MD5 hash of the image file?
   Use any online / offline hash calculator:
   Rhino4.jpg: AA64102AFFF71B93ED61FB100AF8D52A
   Rhino5.gif: 1E90B7F70B2ECB605898524A88269029




Wireshark · Export · HTTP object list

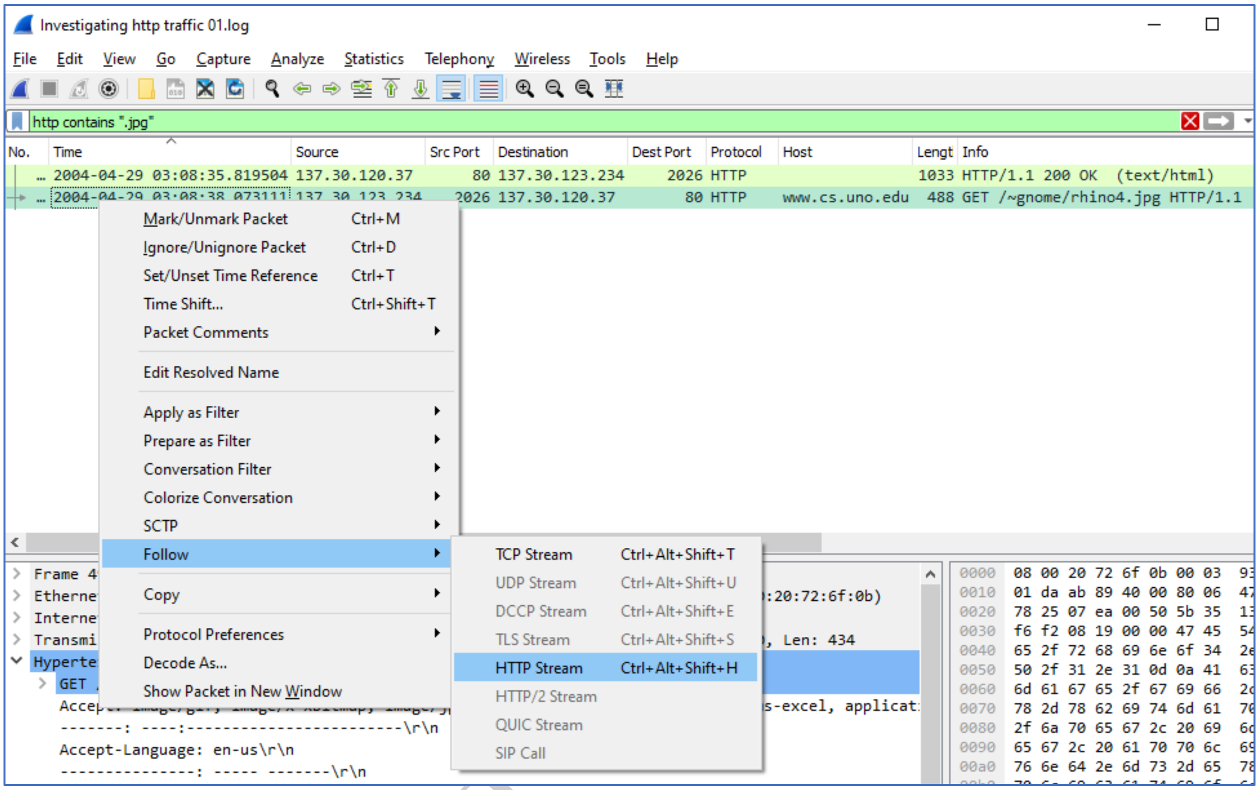
Text Filter:

Packet	Hostname	Content Type	Size	Filename
30	www.cs.uno.edu	text/html	304 bytes	~gnome
34	www.cs.uno.edu	text/html	772 bytes	~gnome
37	www.cs.uno.edu	image/gif	148 bytes	blank.gif
45	www.cs.uno.edu	image/gif	309 bytes	image2.gif
46	www.cs.uno.edu	image/gif	216 bytes	back.gif
215	www.cs.uno.edu	image/jpeg	153 kB	rhino4.jpg
312	www.cs.uno.edu	image/gif	85 kB	rhino5.gif
345	www.cs.uno.edu	text/html	306 bytes	~venkata
350	www.cs.uno.edu	text/html	1388 bytes	~venkata
362	www.cs.uno.edu	text/html	2270 bytes	index.html

More analysis of the http traffic

http contains ".jpg"

Follow HTTP Stream:


Investigating http traffic 01.log
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
DCCP Stream Ctrl+Alt+Shift+E
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
SIP Call

http contains ".jpg"

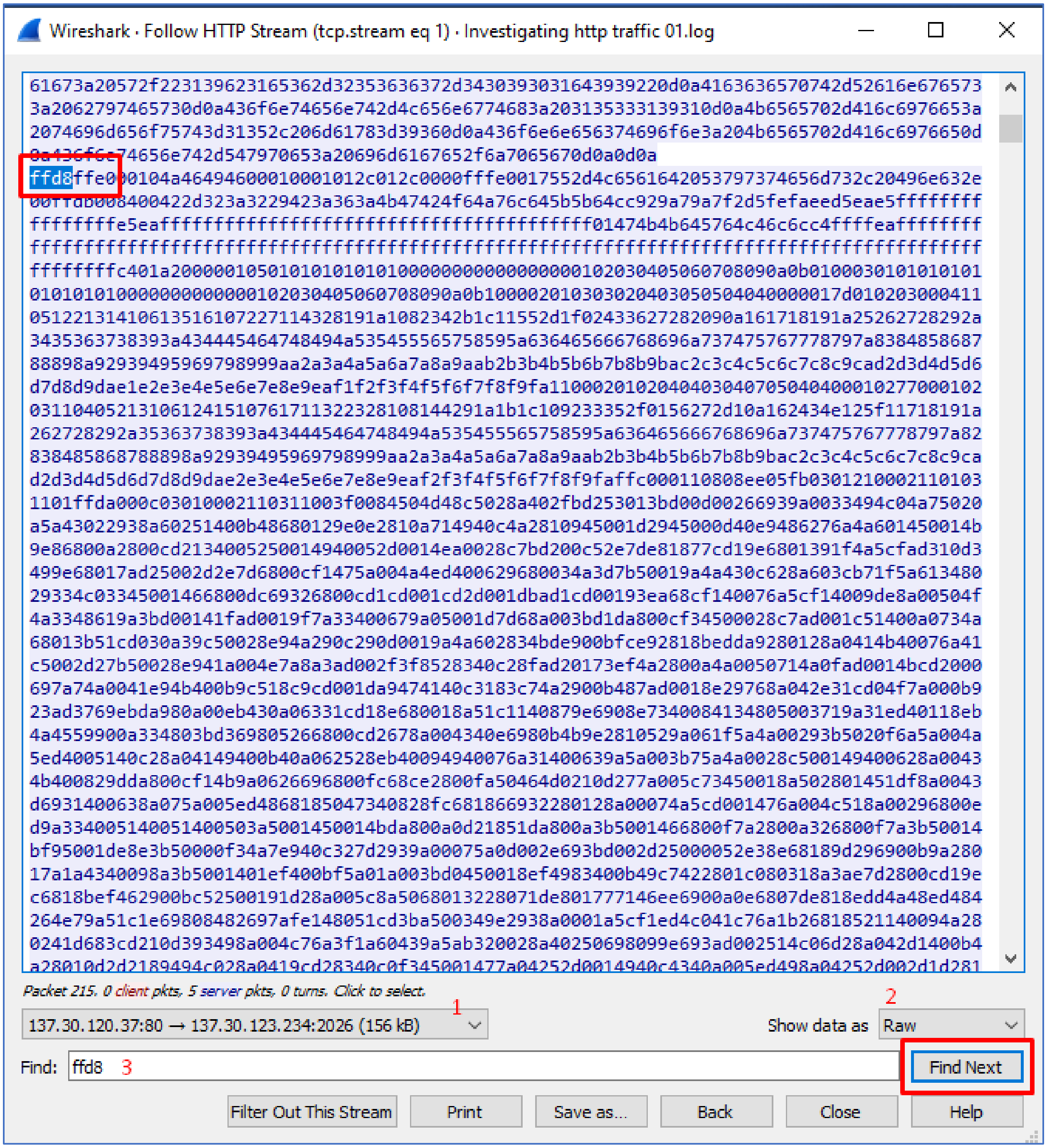
```
No.  Time                        Source          Src Port Destination     Dest Port Protocol Host            Lengt Info
…    2004-04-29 03:08:35.819504  137.30.120.37        80  137.30.123.234     2026 HTTP                      1033 HTTP/1.1 200 OK  (text/html)
…    2004-04-29 03:08:38.073111  137.30.123.234     2026  137.30.120.37        80 HTTP     www.cs.uno.edu    488 GET /~gnome/rhino4.jpg HTTP/1.1
```

```
> Frame 4
> Etherne                                   :20:72:6f:0b)
> Interne
> Transmi                                   , Len: 434
v Hyperte
  > GET
    Accep                                   s-excel, applicat
    -------: ----:-----------------------\r\n
    Accept-Language: en-us\r\n
    ---------------: ----- -------\r\n
```

```
0000  08 00 20 72 6f 0b 00 03  93
0010  01 da ab 89 40 00 80 06  47
0020  78 25 07 ea 00 50 5b 35  13
0030  f6 f2 08 19 00 00 47 45  54
0040  65 2f 72 68 69 6e 6f 34  2e
0050  50 2f 31 2e 31 0d 0a 41  63
0060  6d 61 67 65 2f 67 69 66  2c
0070  78 2d 78 62 69 74 6d 61  70
0080  2f 6a 70 65 67 2c 20 69  6d
0090  65 67 2c 20 61 70 70 6c  69
00a0  76 6e 64 2e 6d 73 2d 65  78
```

Search for the jpg image magic number:

The first two bytes in a JPEG file are always FFD8 and the last two bytes are always FFD9. **The first two bytes are referred to as file header or magic number** and the last two bytes are referred to as file footer.



Wireshark · Follow HTTP Stream (tcp.stream eq 1) · Investigating http traffic 01.log

```
61673a20572f223139623165362d32353636372d343039303164393922 0d0a4163636570742d52616e676573
3a2062797465730d0a436f6e74656e742d4c656e6774683a203135333139310d0a4b6565702d416c6976653a
2074696d656f75743d31352c206d61783d39360d0a436f6e6e656374696f6e3a204b6565702d416c6976650d
0a436f6e74656e742d547970653a20696d6167652f6a7065670d0a0d0a
ffd8ffe000104a46494600010001012c012c0000fffe0017552d4c6561642053797374656d732c20496e632e
00ffdb008400422d323a3229423a363a4b47424f64a76c645b5b64cc929a79a7f2d5fefaeed5eae5ffffffff
fffffffe5eafffffffffffffffffffffffffffffffffffffff01474b4b645764c46c6cc4ffffeaffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffc401a20000010501010101010100000000000000000102030405060708090a0b0100030101010101
01010101000000000000000102030405060708090a0b100002010303020403050504040000017d01020300041 1
0512213141061351610722711432819 1a1082342b1c11552d1f02433627282090a161718191a25262728292a
3435363738393a434445464748494a535455565758595a636465666768696a737475767778797a838485868 7
88898a92939495969798999aa2a3a4a5a6a7a8a9aab2b3b4b5b6b7b8b9bac2c3c4c5c6c7c8c9cad2d3d4d5d6
d7d8d9dae1e2e3e4e5e6e7e8e9eaf1f2f3f4f5f6f7f8f9fa110002010204040304070504040001027700010 2
031104052131061241510761711322328108144291a1b1c109233352f0156272d10a162434e125f11718191a
262728292a35363738393a434445464748494a535455565758595a636465666768696a737475767778797a82
838485868788898a92939495969798999aa2a3a4a5a6a7a8a9aab2b3b4b5b6b7b8b9bac2c3c4c5c6c7c8c9ca
d2d3d4d5d6d7d8d9dae2e3e4e5e6e7e8e9eaf2f3f4f5f6f7f8f9faffc000110808ee05fb030121000211010 3
1101ffda000c03010002110311003f0084504d48c5028a402fbd253013bd00d00266939a0033494c04a75020
a5a43022938a60251400b48680129e0e2810a714940c4a2810945001d2945000d40e9486276a4a601450014b
9e86800a2800cd213400525001494005 2d0014ea0028c7bd200c52e7de81877cd19e6801391f4a5cfad310d3
499e68017ad25002d2e7d6800cf1475a004a4ed400629680034a3d7b50019a4a430c628a603cb71f5a613480
29334c03345001466800dc69326800cd1cd001cd2d001dbad1cd00193ea68cf140076a5cf14009de8a00504f
4a3348619a3bd00141fad0019f7a33400679a05001d7d68a003bd1da800cf34500028c7ad001c51400a0734a
68013b51cd030a39c50028e94a290c290d0019a4a602834bde900bfce92818bedda9280128a0414b40076a41
c5002d27b50028e941a004e7a8a3ad002f3f8528340c28fad20173ef4a2800a4a0050714a0fad0014bcd2000
697a74a0041e94b400b9c518c9cd001da9474140c3183c74a2900b487ad0018e29768a042e31cd04f7a000b9
23ad3769ebda980a00eb430a06331cd18e680018a51c1140879e6908e734008413480500371 9a31ed40118eb
4a4559900a334803bd369805266800cd2678a004340e6980b4b9e2810529a061f5a4a00293b5020f6a5a004a
5ed4005140c28a04149400b40a062528eb40094940076a31400639a5a003b75a4a0028c500149400628a0043
4b400829dda800cf14b9a0626696800fc68ce2800fa50464d0210d277a005c73450018a502801451df8a0043
d6931400638a075a005ed48681850473408 28fc68186693228 0128a00074a5cd001476a004c518a00296800e
d9a3340051400 51400503a5001450014bda800a0d21851da800a3b5001466800f7a2800a326800f7a3b50014
bf95001de8e3b50000f34a7e940c327d2939a00075a0d002e693bd002d25000052e38e68189d296900b9a280
17a1a4340098a3b5001401ef400bf5a01a003bd0450018ef4983400b49c7422801c080318a3ae7d2800cd19e
c6818bef462900bc52500191d28a005c8a5068013228071de801777146ee6900a0e6807de818edd4a48ed484
264e79a51c1e69808482697afe148051cd3ba500349e2938a0001a5cf1ed4c041c76a1b2681852114009 4a28
0241d683cd210d393498a004c76a3f1a60439a5ab320028a40250698099e693ad002514c06d28a042d1400b4
a28010d2d2189494c028a0419cd28340c0f345001477a04252d0014940c4340a005ed498a04252d002d1d281
```

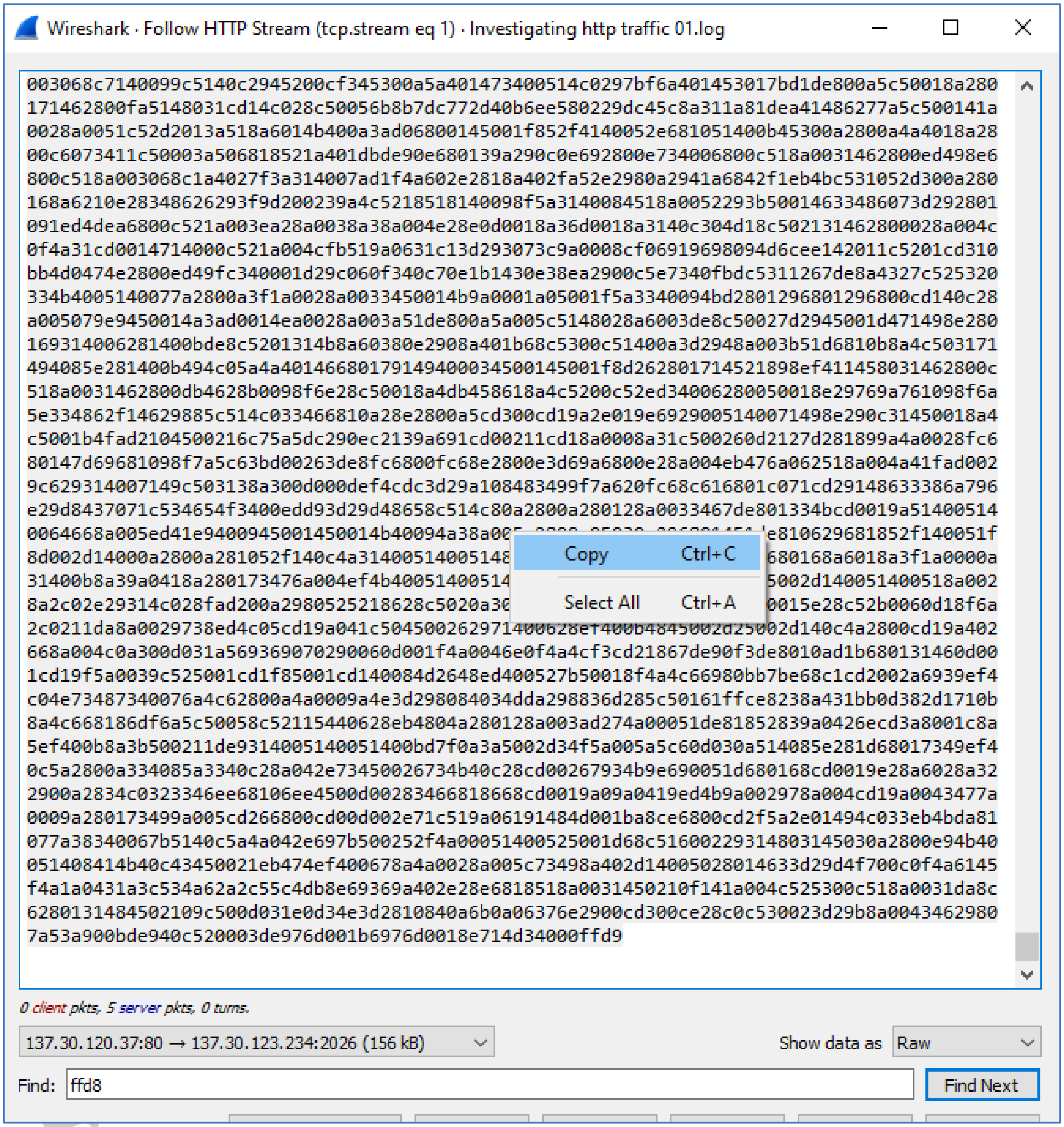
Packet 215. 0 client pkts, 5 server pkts, 0 turns. Click to select.

137.30.120.37:80 → 137.30.123.234:2026 (156 kB)

Show data as: Raw

Find: ffd8

Select and copy:



Wireshark · Follow HTTP Stream (tcp.stream eq 1) · Investigating http traffic 01.log

```
003068c7140099c5140c2945200cf345300a5a401473400514c0297bf6a401453017bd1de800a5c50018a280
171462800fa5148031cd14c028c50056b8b7dc772d40b6ee580229dc45c8a311a81dea41486277a5c500141a
0028a0051c52d2013a518a6014b400a3ad06800145001f852f4140052e681051400b45300a2800a4a4018a28
00c6073411c50003a506818521a401dbde90e680139a290c0e692800e734006800c518a0031462800ed498e6
800c518a003068c1a4027f3a314007ad1f4a602e2818a402fa52e2980a2941a6842f1eb4bc531052d300a280
168a6210e28348626293f9d200239a4c5218518140098f5a3140084518a0052293b50014633486073d292801
091ed4dea6800c521a003ea28a0038a38a004e28e0d0018a36d0018a3140c304d18c5021314628000 28a004c
0f4a31cd0014714000c521a004cfb519a0631c13d293073c9a0008cf06919698094d6cee142011c5201cd310
bb4d0474e2800ed49fc340001d29c060f340c70e1b1430e38ea2900c5e7340fbdc5311267de8a4327c525320
334b4005140077a2800a3f1a0028a0033450014b9a0001a05001f5a3340094bd2801296801296800cd140c28
a005079e9450014a3ad0014ea0028a003a51de800a5a005c5148028a6003de8c50027d2945001d471498e280
169314006281400bde8c5201314b8a60380e2908a401b68c5300c51400a3d2948a003b51d6810b8a4c503171
494085e281400b494c05a4a40146680179149400034500145001f8d262801714521898ef411458031462800c
518a0031462800db4628b0098f6e28c50018a4db458618a4c5200c52ed34006280050018e29769a761098f6a
5e334862f14629885c514c033466810a28e2800a5cd300cd19a2e019e692900514007 1498e290c31450018a4
c5001b4fad2104500216c75a5dc290ec2139a691cd00211cd18a0008a31c500260d2127d281899a4a0028fc6
80147d69681098f7a5c63bd00263de8fc6800fc68e2800e3d69a6800e28a004eb476a062518a004a41fad002
9c629314007149c503138a300d000def4cdc3d29a108483499f7a620fc68c616801c071cd29148633386a796
e29d8437071c534654f3400edd93d29d48658c514c80a2800a280128a0033467de801334bcd0019a51400514
0064668a005ed41e9400945001450014b40094a38a00                    e810629681852f140051f
8d002d14000a2800a281052f140c4a31400514005148                    680168a6018a3f1a0000a
31400b8a39a0418a280173476a004ef4b40051400514                    5002d14005140051 8a002
8a2c02e29314c028fad200a2980525218628c5020a30                    0015e28c52b0060d18f6a
2c0211da8a0029738ed4c05cd19a041c504500262971400628ef400b4845002d25002d140c4a2800cd19a402
668a004c0a300d031a56936907029006 0d001f4a0046e0f4a4cf3cd21867de90f3de8010ad1b680131460d00
1cd19f5a0039c525001cd1f85001cd140084d2648ed400527b50018f4a4c66980bb7be68c1cd2002a6939ef4
c04e73487340076a4c62800a4a0009a4e3d298084034dda298836d285c50161ffce8238a431bb0d382d1710b
8a4c668186df6a5c50058c52115440628eb4804a280128a003ad274a00051de81852839a0426ecd3a8001c8a
5ef400b8a3b500211de9314005140051400bd7f0a3a5002d34f5a005a5c60d030a514085e281d68017349ef4
0c5a2800a334085a3340c28a042e73450026734b40c28cd00267934b9e690051d680168cd0019e28a6028a32
2900a2834c0323346ee68106ee4500d00283466818668cd0019a09a0419ed4b9a002978a004cd19a0043477a
0009a280173499a005cd266800cd00d002e71c519a06191484d001ba8ce6800cd2f5a2e01494c033eb4bda81
077a38340067b5140c5a4a042e697b500252f4a00051400525001d68c516002293148031 45030a2800e94b40
051408414b40c43450021eb474ef400678a4a0028a005c73498a402d14005028014633d29d4f700c0f4a6145
f4a1a0431a3c534a62a2c55c4db8e69369a402e28e6818518a0031450210f141a004c525300c518a0031da8c
6280131484502109c500d031e0d34e3d2810840a6b0a06376e2900cd300ce28c0c530023d29b8a0043462980
7a53a900bde940c520003de976d001b6976d0018e714d34000ffd9
```

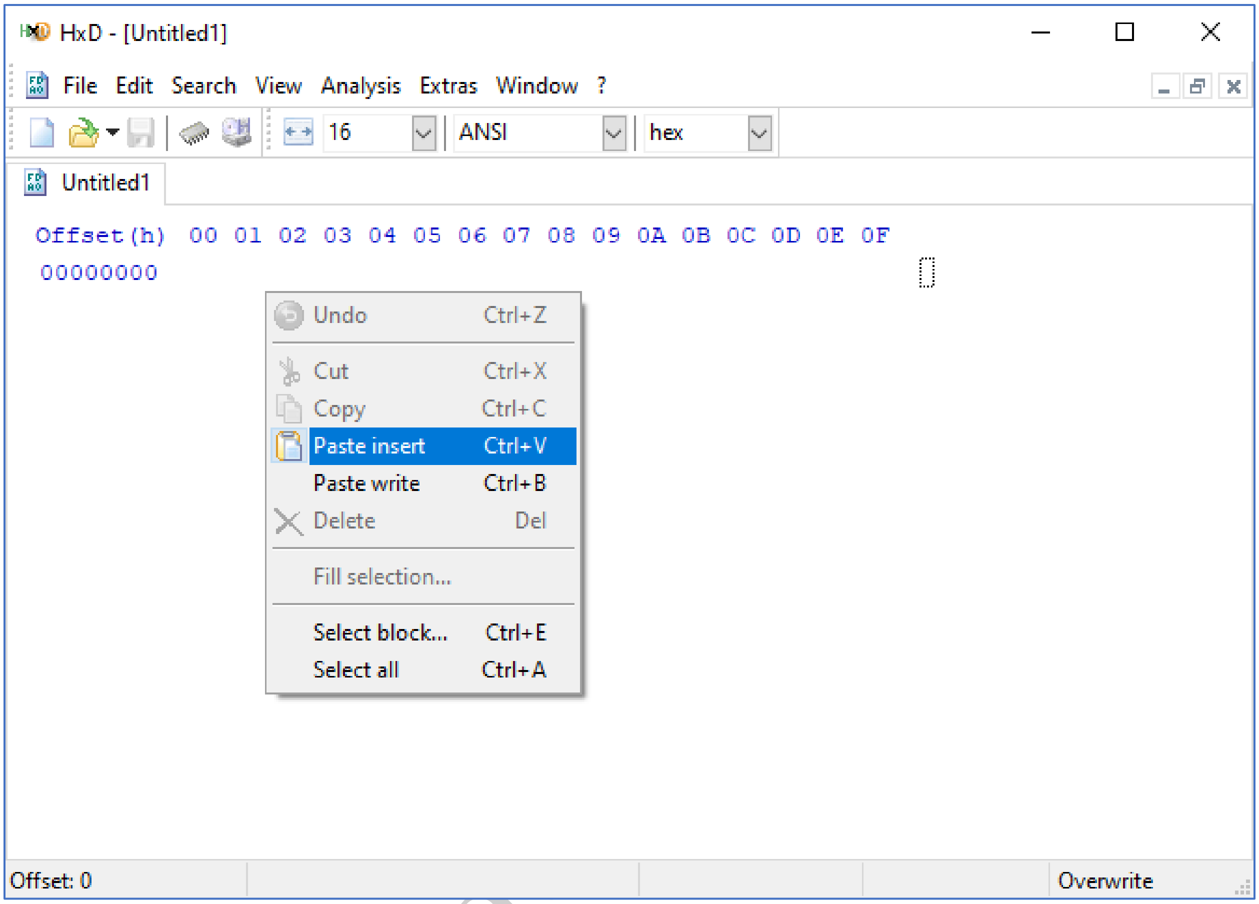
0 client pkts, 5 server pkts, 0 turns.

137.30.120.37:80 → 137.30.123.234:2026 (156 kB)

Show data as Raw

Find: ffd8

Use any Hex editor and pest it.



File > Save as > image.jpg

You will get the original image.