Wireshark packet sniffing for HTTP, SSL and TLS packets.

NTAL MINI PROJECT

BY:
AMEY BHADKAMKAR(4) ,AISHWARYA DANOJI(11)
AISHWARYA DHAGE(12) ,SAURABH FEGADE(14)

OVERVIEW

Project Scope

Network is under attack from so many directions and so security of network has become important.

One way of doing this is monitoring the packets sent through the network.In this project we are using wireshark to monitor the packets sent and received.

We will demonstrate sniffing of packets to obtain:

- username,password
- images
- demonstrate ssl handshaking procedure

Introduction

- Wireshark is a network packet analyzer.
- A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.
- It is a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course).
- In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, all that has changed.
- Wireshark is perhaps one of the best open source packet analyzers available today.

Purpose of using wireshark:

- Network security engineers use it to examine security problems
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals
- Network administrators use it to troubleshoot network problems

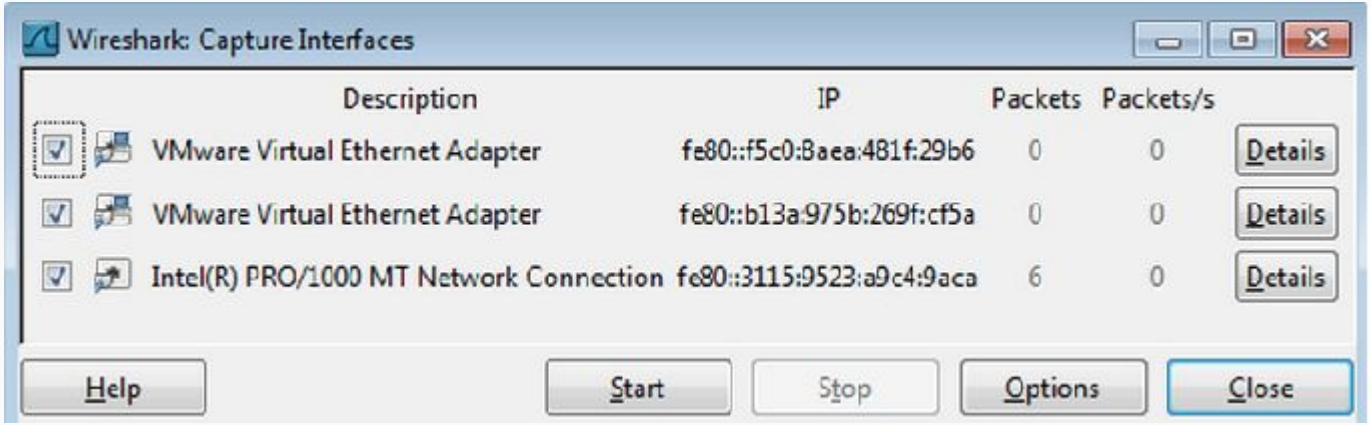
Features of Wireshark

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.

Proposed Work-
1)Sniffing packets(username and password)-

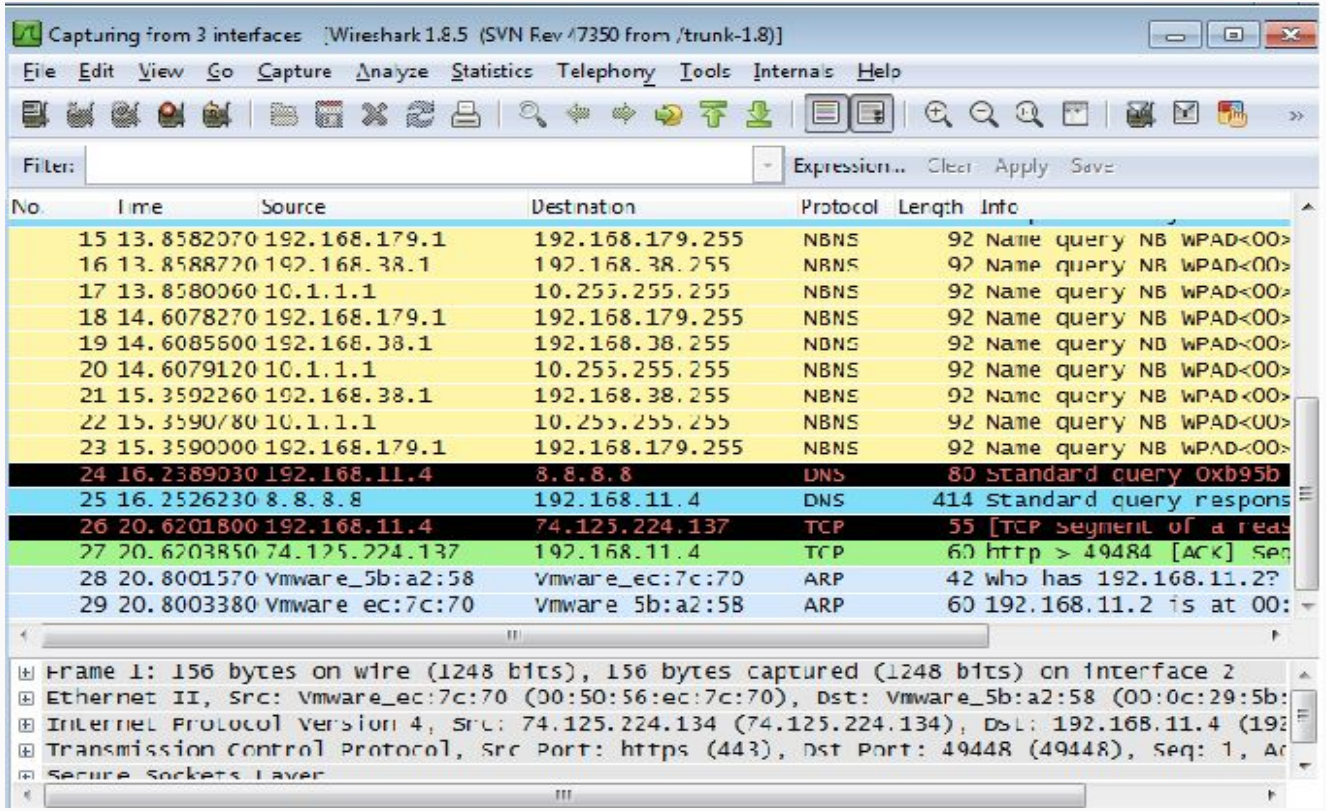
- Starting a Packet Capture-Click Start, Wireshark.

"Interface List".In the "Wireshark: Capture Interfaces" box, check all the interfaces, as shown below.Click the Start button.



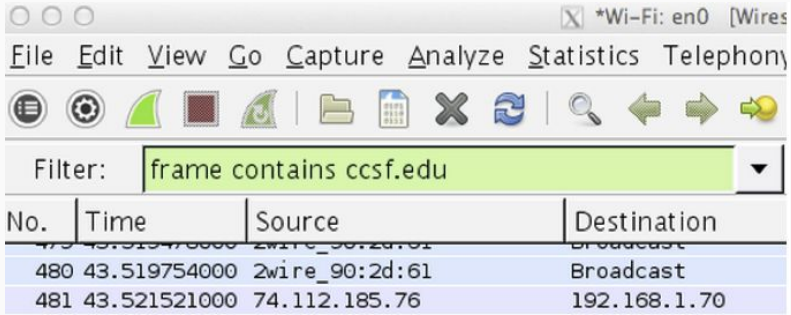
1)Sniffing packets(username and password)

You should see packets being captured



1)Sniffing packets(username and password)

- Sending a Test Password to gogoNET.
- Add username and password.
- Click the "Sign In" button.
- In the Wireshark window, box, click Capture, Stop.
- Filter packet of ccsf.edu



1)Sniffing packets(username and password)

Expand the "Follow TCP Stream" box so that you can see YOURNAME and the password
of topsecretpassword

Follow TCP Stream

Stream Content

```
POST /main/authorization/doSignIn?target=http%3A%2F%2Fwww.gogo6.com%2F HTTP/1.1
Host: www.gogo6.com
Connection: keep-alive
Content-Length: 68
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://www.gogo6.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/32.0.1700.102 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: http://www.gogo6.com/main/authorization/doSignIn?target=http%3A%2F%2Fwww.gogo6.com%2F
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Cookie: xg_cookie_check=1; xn_visitor=e7d57974-275e-4988-96a4-6d11e4772e1e; xgdi=1797716935;
2__utmz=^ning.1406623303837:3993126.1390855304.10.9.utmcsr=samsclass.info|utmccn=(referral)|
utmcmd=referral|utmcct=/ipv6/proj/gogo-Mac.htm; xg_pc=%257B%2522mobileRedirectsEnabled%2522%253Anull%
252C%2522lastVisit%2522%253A%25221391112266-12fjzd71j37c0%2522%257D; xgl=y; xg_sc=%7B%7D; xn_track=rp
%252C%25252F%252Crc%252C0%252Csi%252C1391122720%252Cse%252C1391123620;
__utma=3993126.1343598327.1390022246.1391112268.1391122720.10; __utmb=3993126.1.10.1391122720;
__utmc=3993126; __utmz=3993126.1390855304.8.8.utmcsr=samsclass.info|utmccn=(referral)|
utmcmd=referral|utmcct=/ipv6/proj/gogo-Mac.htm; ning_session=/QuNB4f8jkAmOL6A8IyOPI4pOf4NjTqVzlX1EqB
+YXYsEGOXljkZQuCa8kZTSoYOUiJ257FSyWI=; 2__utmv=^ning.1391123146078:;
2__utma=^ning.1454195146079:3993126.2067282596.1390022248.1391112268.1391122720.12;
2__utmb=^ning.1391124946079:3993126.4.10.1391122720; 2__utmc=^ning.1454195146079:3993126

xg_token=$emailAddress=YOURNAME%40ccsf.edu&password=topsecretpasswodHTTP/1.1 200 OK
Date: Thu, 30 Jan 2014 23:08:14 GMT
Server: Ning HTTP Server 2.0
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: ning_session=/QuNB4f8jkAmOL6A8IyOPI4pOf4NjTqVzlX1EqB
+YXYsEGOXljkZQiyOJgUJMFhkEI8d1KKXyx8=;Path=/;Domain=.gogo6.com;Expires=Fri, 31-Jan-14 00:08:14 GMT
X-XN-Trace-Token: 3da9db1f-fa50-41cb-b788-dca3f42dcf29
Date: Thu, 30 Jan 2014 23:08:14 GMT
```

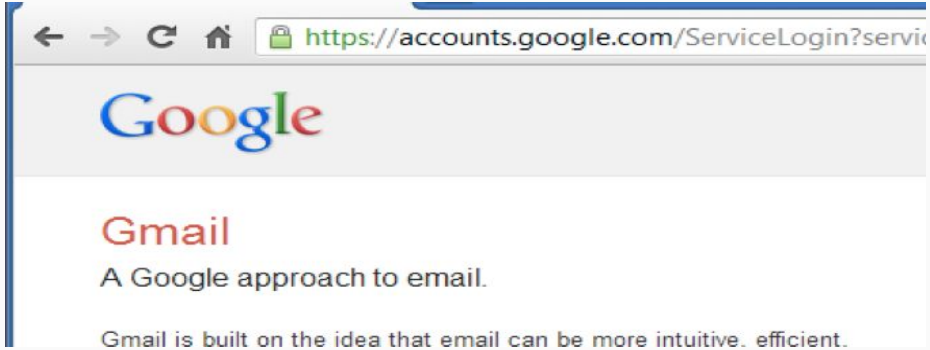
Entire conversation (9073 bytes)

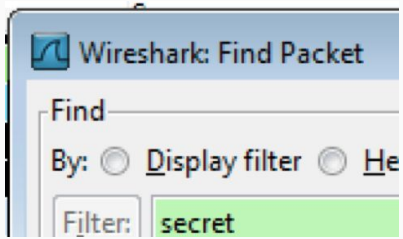
Find | Save As | Print | ASCII | EBCDIC | Hex Dump | C Arrays | Raw

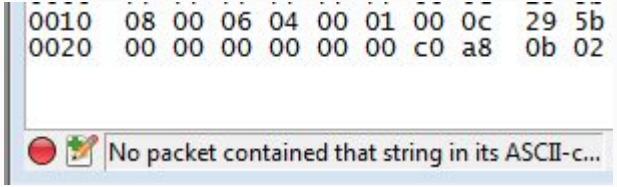
Help | Filter Out This Stream | Close

1)Sniffing packets(username and password)

- Using a Secure Password Transmission-
- Enter a Username and password
- Searching for the Password in Wireshark
- A message appears briefly in the status bar at the bottom of the Wireshark window, saying "No packet contained that string".
- The password cannot be found because Gmail encrypts it before transmitting it.







2) IMAGE SNIFFING

- Start Wireshark.
- Wireshark-> Interface list -> Capture Interfaces
- Open a web browser.
- Search for jpg or png images.
- Stop packet capturing and filter http.
- Search for jpeg or png file in the "Info".

Select the packets and from the details of the packets mentioned below you can do two things-

1)Open JPEG files directly" means that you can open them via File→Open. Wireshark will display the JPEG file as a single "packet". You can open and analyze MP3 files in the same way.

2)If you want to export a JPEG you've captured in an HTTP session you can use File→Export→Objects→HTTP.

IMAGE SNIFFING

5622	45.008991	216.58.199.130	192.168.1.106	HTTP	404 HTTP/1.1 200 OK (text/javascript)
5643	45.043550	104.31.76.172	192.168.1.106	HTTP	612 HTTP/1.1 200 OK (JPEG JFIF image)
5646	45.044308	192.168.1.106	104.31.76.172	HTTP	552 GET /promo/soundotcom200x125.jpg HTTP/1.1
5669	45.075873	103.243.220.231	192.168.1.106	HTTP	539 HTTP/1.1 302 Found
5687	45.104617	203.94.200.25	192.168.1.106	HTTP	1274 HTTP/1.1 200 OK (text/html)

```
▷ Frame 5643: 612 bytes on wire (4896 bits), 612 bytes captured (4896 bits) on interface 0
▷ Ethernet II, Src: BestItWo_1a:6d:74 (00:1e:a6:1a:6d:74), Dst: IntelCor_24:7b:ab (ac:72:89:24:7b:ab)
▷ Internet Protocol Version 4, Src: 104.31.76.172, Dst: 192.168.1.106
▷ Transmission Control Protocol, Src Port: 80, Dst Port: 51614, Seq: 138351, Ack: 1585, Len: 558
▷ [11 Reassembled TCP Segments (14558 bytes): #5611(1400), #5612(1400), #5614(1400), #5615(1400), #5617(1400), #5618(1400),
▷ Hypertext Transfer Protocol
◢ JPEG File Interchange Format
    Marker: Start of Image (0xffd8)
  ▷ Marker segment: Reserved for application segments - 1 (0xFFE1)
  ▷ Marker segment: Reserved for application segments - 12 (0xFFEC)
  ▷ Marker segment: Reserved for application segments - 1 (0xFFE1)
  ▷ Marker segment: Reserved for application segments - 14 (0xFFEE)
```

IMAGE SNIFFING

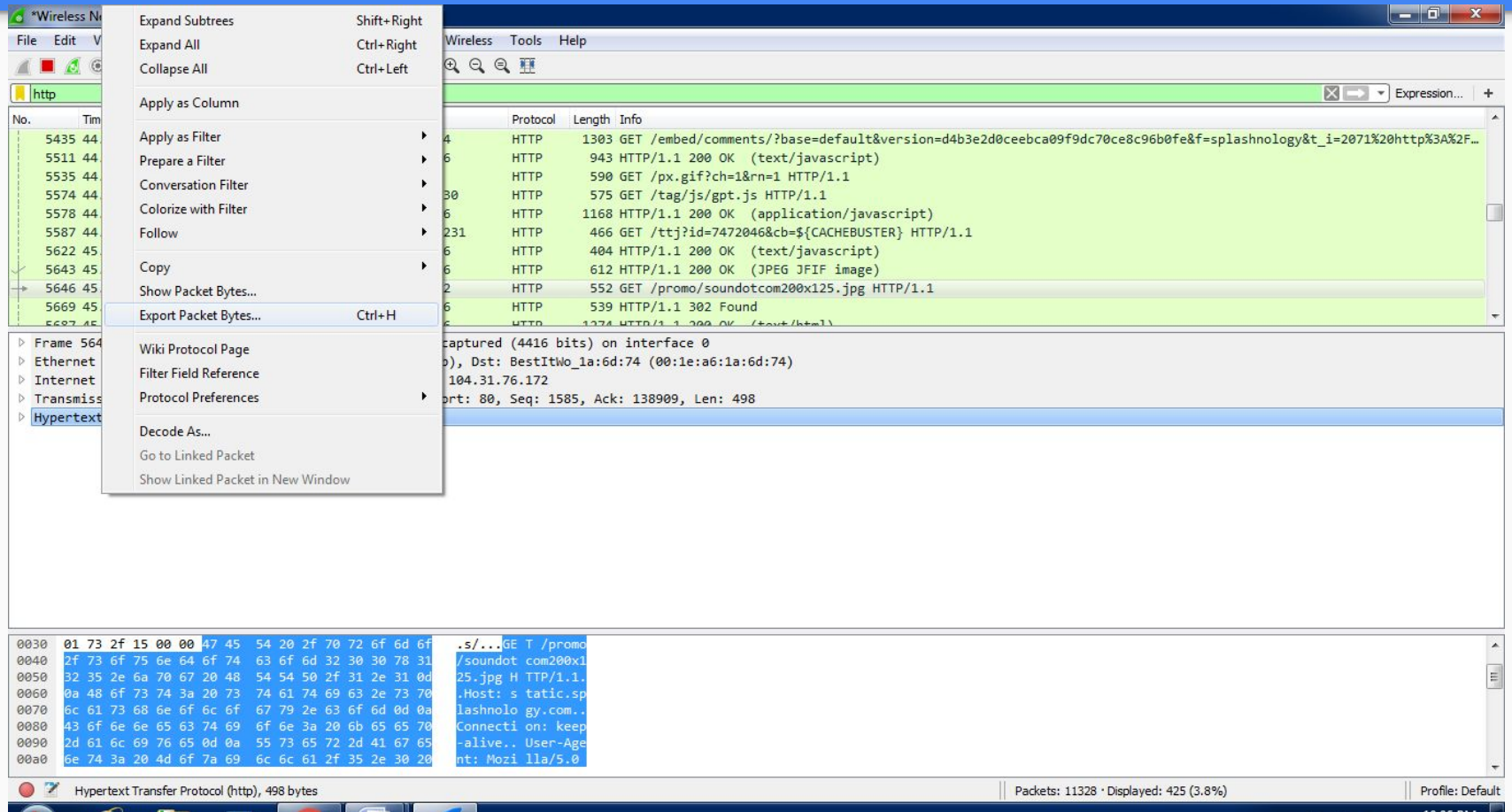


IMAGE SNIFFING

*Wireless Network Connection

Wireshark · Export Selected Packet Bytes

Desktop ▸ Search Desktop

Organize ▾ New folder

Favorites
Desktop
Downloads
Recent Places

Libraries
Documents
Music
Pictures
Videos

Homegroup

Libraries
System Folder

Homegroup
System Folder

MTNL
System Folder

Computer
System Folder

Network

File name: image1.jpg

Save as type: Raw data (*.bin *.dat *.raw)

Hide Folders

Save

Cancel

Expression... +

s/?base=default&version=d4b3e2d0ceebca09f9dc70ce8c96b0fe&f=splashnology&t_i=2071%20http%3A%2F…
text/javascript)
n=1 HTTP/1.1
HTTP/1.1
application/javascript)
6&cb=${CACHEBUSTER} HTTP/1.1
text/javascript)
JPEG JFIF image)
com200x125.jpg HTTP/1.1

b:ab)

, #5617(1400), #5618(1400), #5624(1400), #5625(1400), #5628(1400), #5629(1400), #5643(558)]

Marker segment: Define quantization table(s) (0xFFDB)
Start of Frame header: Start of Frame (non-differential, Huffman coding) - Baseline DCT (0xFFC0)
Marker segment: Define Huffman table(s) (0xFFC4)
Start of Segment header: Start of Scan (0xFFDA)

```
01b0  0a 0d 0a ff d8 ff e1 00  18 45 78 69 66 00 00 49   ........ .Exif..I
01c0  49 2a 00 08 00 00 00 00  00 00 00 00 00 00 00 ff   I*...... ........
01d0  ec 00 11 44 75 63 6b 79  00 01 00 04 00 00 00 4b   ...Ducky .......K
01e0  00 00 ff e1 03 2d 68 74  74 70 3a 2f 2f 6e 73 2e   .....-ht tp://ns.
01f0  61 64 6f 62 65 2e 63 6f  6d 2f 78 61 70 2f 31 2e   adobe.co m/xap/1.
0200  30 2f 00 3c 3f 78 70 61  63 6b 65 74 20 62 65 67   0/.<?xpa cket beg
0210  69 6e 3d 22 ef bb bf 22  20 69 64 3d 22 57 35 4d   in="..."  id="W5M
```

Frame (612 bytes) | Reassembled TCP (14558 bytes)

JPEG File Interchange Format (image-jfif), 14123 bytes

Packets: 11806 · Displayed: 425 (3.6%)

Profile: Default

10:08 PM
11/1/2016

3)SSL HANDSHAKING

Setting Up the Project

1. Create a non-Administrator User Account.Once this is done, you will need to restart your Windows 7 computer to continue. You will not need to log into your Unprivileged account just yet.
2. Next, we need to open a Command Prompt window. Click on the Start button, then type **cmd** into the *Search programs and files* box
3. Execute the following commands:
   SETX SSLKEYLOGFILE C:\keys\session-keys.log
   MKDIR \keys

« User Accounts ▸ Manage Accounts ▸ Create New Account

Search Con...

Name the account and choose an account type

This name will appear on the Welcome screen and on the Start menu.

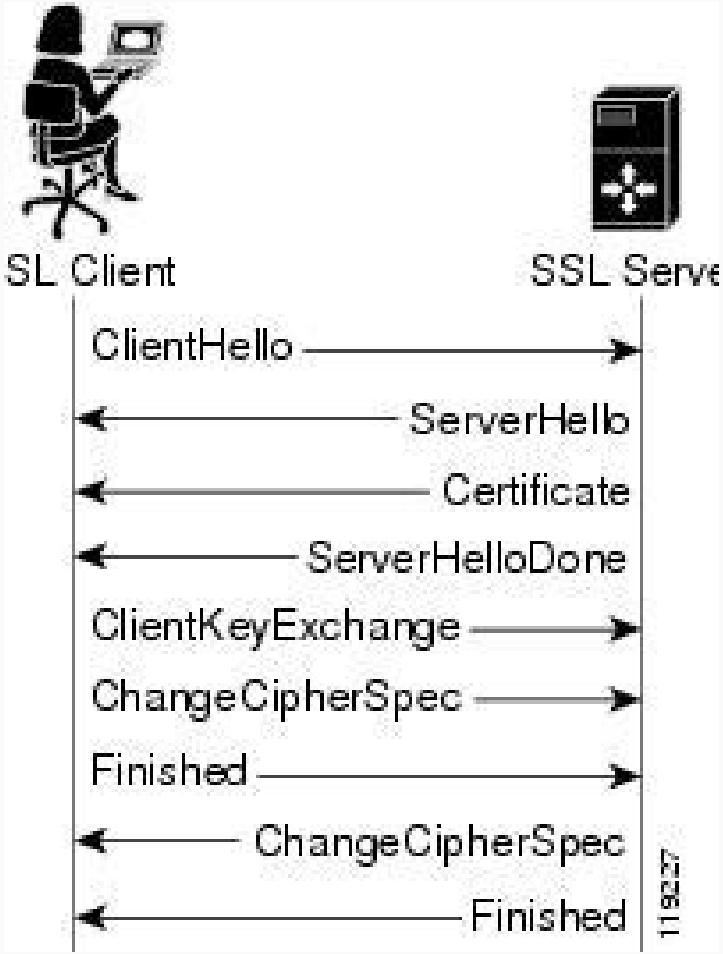
YOURNAME-BROKEN

- (●) Standard user
  Standard account users can use most software and change system settings that do not affect other users or the security of the computer.
- ( ) Administrator
  Administrators have complete access to the computer and can make any desired changes. Based on notification settings, administrators may be asked to provide their password or confirmation before making changes that affect other users.

We recommend that you protect every account with a strong password.

Why is a standard account recommended?

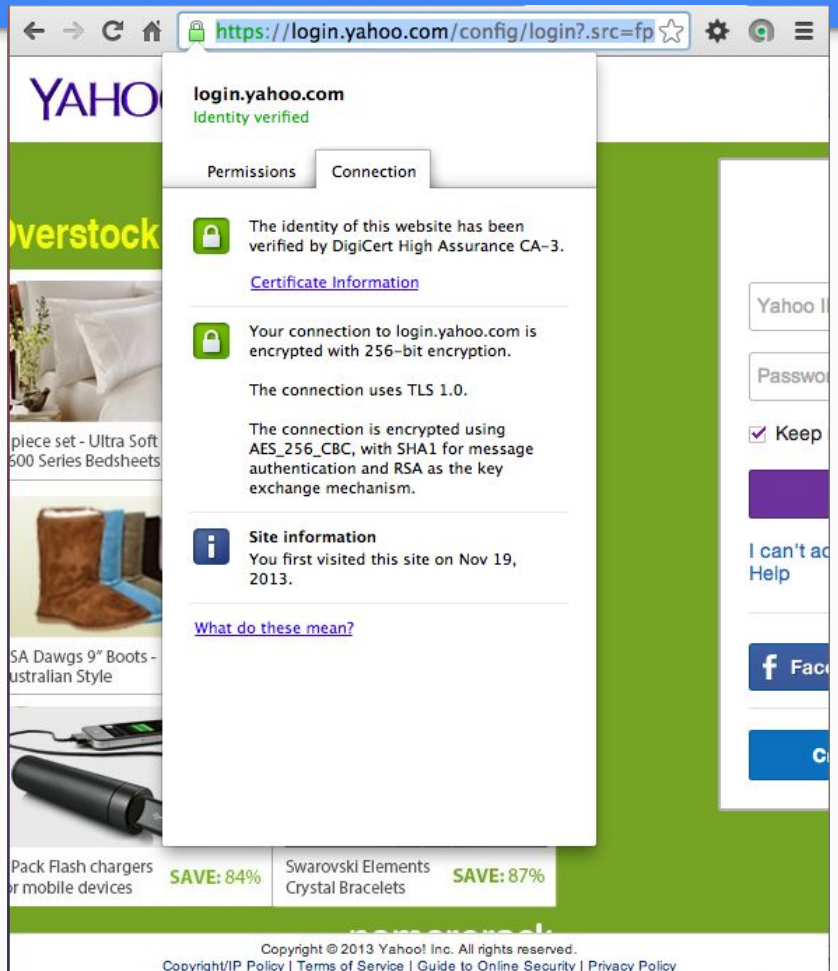
Create Account | Cancel


SL Client
SSL Serve
ClientHello
ServerHello
Certificate
ServerHelloDone
ClientKeyExchange
ChangeCipherSpec
Finished
ChangeCipherSpec
Finished
119227

1	ClientHello	Client initiates the handshake by sending the ClientHello me proposes the SSL parameters to use during the SSL sessio
2	ServerHello	Server responds with the ServerHello message that contain parameters that it selects for use during the SSL session.
3	Certificate	Server sends the client its public key certificate.
4	ServerHelloDone	Server concludes its part of the SSL negotiations.
5	ClientKeyExchange	Client sends session key information that it encrypts using t key.
6	ChangeCipherSpec	Client instructs the server to activate the negotiated SSL pa future messages that it sends.
7	Finished	Client instructs the server to verify that the SSL negotiation successful.
8	ChangeCipherSpec	Server instructs the client to activate the negotiated SSL pa future messages that it sends.
9	Finished	Server instructs the client to verify that the SSL negotiation

- Starting a Packet Capture-Click the Start button.
- Open a web browser and open yahoo or any site that uses TLS 1.0 OR TLS 1.2 protocol.
- Stop packet capturing and filter for ssl packets
- Look for "Client Hello " packet in the "info " column.Right click and select Follow TCP stream.
- The packets containing only TLSV1 handshake packets would be filtered and displayed.

10.	74.125.236.22	TLSv1	196	Client Hello
74.125.236.22	10	TLSv1	1514	Server Hello
74.125.236.22	10.	TLSv1	324	Certificate, Server Hello Done
10.	74.125.236.22	TLSv1	193	Client Key Exchange
10.	74.125.236.22	TLSv1	101	Change Cipher Spec, Encrypted Handshake Message
74.125.236.22	10.	TLSv1	101	Change Cipher Spec, Encrypted Handshake Message
10.	74.125.236.22	TLSv1	696	Application Data
74.125.236.22	10.	TLSv1	608	Application Data, Application Data
10.	74.125.236.22	TLSv1	594	Application Data
74.125.236.22	10.	TLSv1	1404	Application Data, Application Data



File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: ssl Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
119	4.526828	10.30.52.132	74.125.236.53	TLSv1	238	Client Hello
121	4.579867	74.125.236.53	10.30.52.132	TLSv1	1484	Server Hello
123	4.579887	74.125.236.53	10.30.52.132	TLSv1	574	Certificate, Server Key Exchange, Server Hello Done
125	4.586063	10.30.52.132	74.125.236.53	TLSv1	224	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
126	4.640613	74.125.236.53	10.30.52.132	TLSv1	292	Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message
127	4.640641	74.125.236.53	10.30.52.132	TLSv1	119	Application Data
129	4.641252	10.30.52.132	74.125.236.53	TLSv1	111	Application Data
130	4.641294	10.30.52.132	74.125.236.53	TLSv1	826	Application Data
133	4.853659	74.125.236.53	10.30.52.132	TLSv1	456	Application Data
134	4.853895	74.125.236.53	10.30.52.132	TLSv1	461	Application Data
140	4.917522	10.30.52.132	74.125.135.84	SSL	230	Client Hello
142	4.983053	74.125.135.84	10.30.52.132	TLSv1	1484	Server Hello
144	4.983082	74.125.135.84	10.30.52.132	TLSv1	558	Certificate, Server Key Exchange, Server Hello Done

```
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 167
▼ Handshake Protocol: Client Hello
  Handshake Type: Client Hello (1)
  Length: 163
  Version: TLS 1.0 (0x0301)
  ▶ Random
  Session ID Length: 0
  Cipher Suites Length: 72
  ▶ Cipher Suites (36 suites)
  Compression Methods Length: 1
  ▶ Compression Methods (1 method)
```

```
0000  00 1f 9d f2 bc c3 ac 16  2d 0c 69 5e 08 00 45 00   ........ -.i^..E.
0010  00 e0 2e 3a 40 00 40 06  96 89 0a 1e 34 84 4a 7d   ...:@.@. ....4.J}
0020  ec 35 e9 ef 01 bb d1 ae  8c b9 11 62 e9 3d 80 18   .5...... ...b.=..
0030  03 91 76 27 00 00 01 01  08 0a 00 91 a2 8c db 50   ..v'.... .......P
```

File: "/tmp/wireshark_eth2_20130... Packets: 276 Displayed: 69 Marked: 0 Dropped: 0 Profile: Default

RESULT:

- Thus,we have observed packets from a unsecured and secured password website and also observed how packets are captured of ssl ,tls and https.
- Wireshark allows live capturing of packets.
- Website with secure password transmission won't allow us to see the password thus securing the details of user.
- SSL allows transmission of information securely in encryped format.

REFERENCES

1)https://samsclass.info/120/proj/p3-wireshark.htm
2)https://samsclass.info/106/proj13/p1_WireShark_HTTP.htm
3)https://samsclass.info/120/proj/p6x-wireshark-ssl.html
4)https://en.wikipedia.org/wiki/Transport_Layer_Security

THANK YOU