

Experiment No. : 1

Date: 02/08/2024

Title : Packet Sniffing using Wireshark

Problem Definition : To capture packets of user-credentials during login process and check whether they are encrypted or not using packet sniffing tool, Wireshark. Perform this on website using HTTP and HTTPS protocol and inspect whether the websites the user-credentials or not.

Pre-requisite : Knowledge about various networking protocols such as HTTP, HTTPS, TCP etc and some knowledge about wireshark tool .

Theory :

Packet sniffing:

Packet sniffing is a form of wire-tap applied to computer networks instead of phone networks. It came into vogue with Ethernet, which is known as a "shared medium" network. This means that traffic on a segment passes by all hosts attached to that segment. Ethernet cards have a filter that prevents the host machine from seeing traffic addressed to other stations. Sniffing programs turn off the filter, and thus see everyones traffic.

Today's networks may already contain built-in sniffing modules. Most hubs support the RMON standard, which allow the intruder to sniff remotely using SNMP, which has weak authentication. Many corporations employ Network Associates "Distributed Sniffer Servers", which are set up with easy to guess passwords. Windows NT machines often have a "Network Monitoring Agent" installed, which again allows for remote sniffing.

Packets sniffing is difficult to detect, but it can be done. But the difficulty of the solution means that in practice, it is rarely done.

The popularity of packet sniffing stems from the fact that it sees *everything*. Typical items sniffed include:

SMTP, POP, IMAP traffic

Allows intruder to read the actual e-mail.

POP, IMAP, HTTP Basic, Telnet authentication

Reads passwords off the wire in clear-text.

SMB, NFS, FTP traffic Reads

files of the wire.

SQL database

Reads financial transactions and credit card numbers.

Not only can sniffing read information that helps break into a system, it is an intrusion by itself because it reads the very files the intruder is interested in.

Top 10 Packet Sniffing tools:

1: [Wireshark](#)

Wireshark (known as Ethereal until a trademark dispute in Summer 2006) is a fantastic open source network protocol analyzer for Unix and Windows. It allows you to examine data from a live network or from a capture file on disk. You can interactively browse the capture data, delving down into just

the level of packet detail you need. Wireshark has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session. It also supports hundreds of protocols and media types. A tcpdump-like console version named tethereal is included. One word of caution is that Ethereal has suffered from dozens of remotely exploitable security holes, so stay up-to-date and be wary of running it on untrusted or hostile networks (such as security conferences).

<http://media-2.cacotech.com/video/wireshark/introduction-to-wireshark/>

2: [Tcpdump](#)

Tcpdump is the IP sniffer we all used before Ethereal (Wireshark) came on the scene, and many of us continue to use it frequently. It may not have the bells and whistles (such as a pretty GUI or parsing logic for hundreds of application protocols) that Wireshark has, but it does the job well and with fewer security holes. It also requires fewer system resources. While it doesn't receive new features often, it is actively maintained to fix bugs and portability problems. It is great for tracking down network problems or monitoring activity. There is a separate Windows port named WinDump. TCPDump is the source of the Libpcap/WinPcap packet capture library, which is used by [Nmap](#) among many other tools.

3: [Cain and Abel](#)

UNIX users often smugly assert that the best free security tools support their platform first, and Windows ports are often an afterthought. They are usually right, but Cain & Abel is a glaring exception. This Windows-only password recovery tool handles an enormous variety of tasks. It can recover passwords by sniffing the network, cracking encrypted passwords using Dictionary, BruteForce and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, revealing password boxes, uncovering cached passwords and analyzing routing protocols. It is also [well documented](#).

4: [Kismet](#)

Kismet is a console (ncurses) based 802.11 layer2 wireless network detector, sniffer, and intrusion detection system. It identifies networks by passively sniffing (as opposed to more active tools such as [NetStumbler](#)), and can even decloak hidden (non-beaconing) networks if they are in use. It can automatically detect network IP blocks by sniffing TCP, UDP, ARP, and DHCP packets, log traffic in Wireshark/TCPDump compatible format, and even plot detected networks and estimated ranges on downloaded maps. As you might expect, this tool is commonly used for wardriving. Oh, and also warwalking, warflying, and warskating, ...

5: [Dsniff](#)

This popular and well-engineered suite by Dug Song includes many tools. dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspys passively monitor a network for interesting data

(passwords, e-mail, files, etc.). arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching). sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected ssh and https sessions by exploiting weak bindings in ad-hoc PKI. A separately maintained partial Windows port is available here. Overall, this is a great toolset. It handles pretty much all of your password sniffing needs.

6: [NetStumbler](#)

Netstumbler is the best known Windows tool for finding open wireless access points (“wardriving”). They also distribute a WinCE version for PDAs and such named [Ministumbler](#). The tool is currently free but Windows-only and no source code is provided. It uses a more active approach to finding WAPs than passive sniffers such as [Kismet](#) or [KisMAC](#).

7: [Ettercap](#)

Ettercap is a terminal-based network sniffer/interceptor/logger for ethernet LANs. It supports active and passive dissection of many protocols (even ciphered ones, like ssh and https). Data injection in an established connection and filtering on the fly is also possible, keeping the connection synchronized. Many sniffing modes were implemented to give you a powerful and complete sniffing suite. Plugins are supported. It has the ability to check whether you are in a switched LAN or not, and to use OS fingerprints (active or passive) to let you know the geometry of the LAN.

8: [Ngrep](#)

ngrep strives to provide most of GNU grep’s common features, applying them to the network layer. ngrep is a pcap-aware tool that will allow you to specify extended regular or hexadecimal expressions to match against data payloads of packets. It currently recognizes TCP, UDP and ICMP across Ethernet, PPP, SLIP, FDDI, Token Ring and null interfaces, and understands bpf filter logic in the same fashion as more common packet sniffing tools, such as tcpdump and snoop.

9: [Ntop](#)

Ntop shows network usage in a way similar to what top does for processes. In interactive mode, it displays the network status on the user’s terminal. In Web mode, it acts as a Web server, creating an HTML dump of the network status. It sports a NetFlow/sFlow emitter/collector, an HTTP-based client interface for creating ntop-centric monitoring applications, and RRD for persistently storing traffic statistics.

10: [EtherApe](#)

EtherApe is a graphical network monitor for Unix modeled after etherman. Featuring link layer, IP and TCP modes, EtherApe displays network activity graphically with a color coded protocols display. Hosts and links change in size with traffic. It supports Ethernet, FDDI, Token

Ring, ISDN, PPP and SLIP devices. It can filter traffic to be shown, and can read traffic from a file as well as live from the network.

Wireshark:

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.

You could think of a network packet analyzer as 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).

Some intended purposes

Here are some examples people use Wireshark for:

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

Beside these examples Wireshark can be helpful in many other situations too.

Features

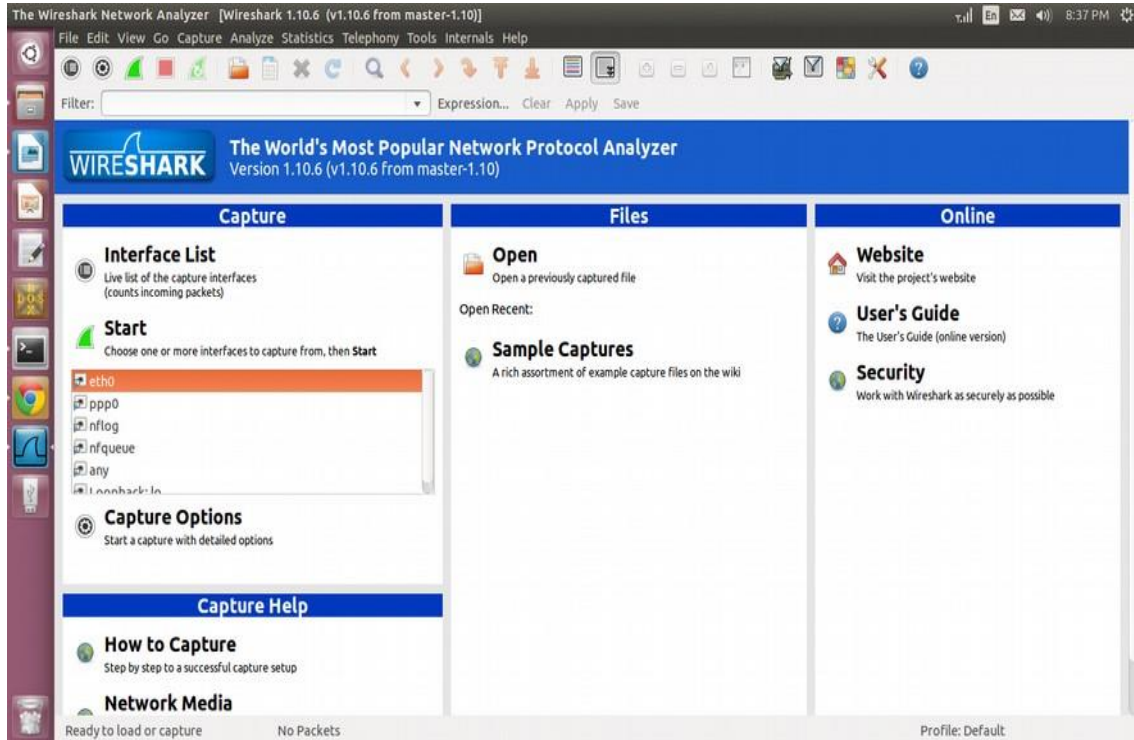
The following are some of the many features Wireshark provides:

- ☐ 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.
- ☐ *Search* for packets on many criteria.
- *Colorize* packet display based on filters.
- Create various *statistics*.
- **...and a lot more!**

Procedure :

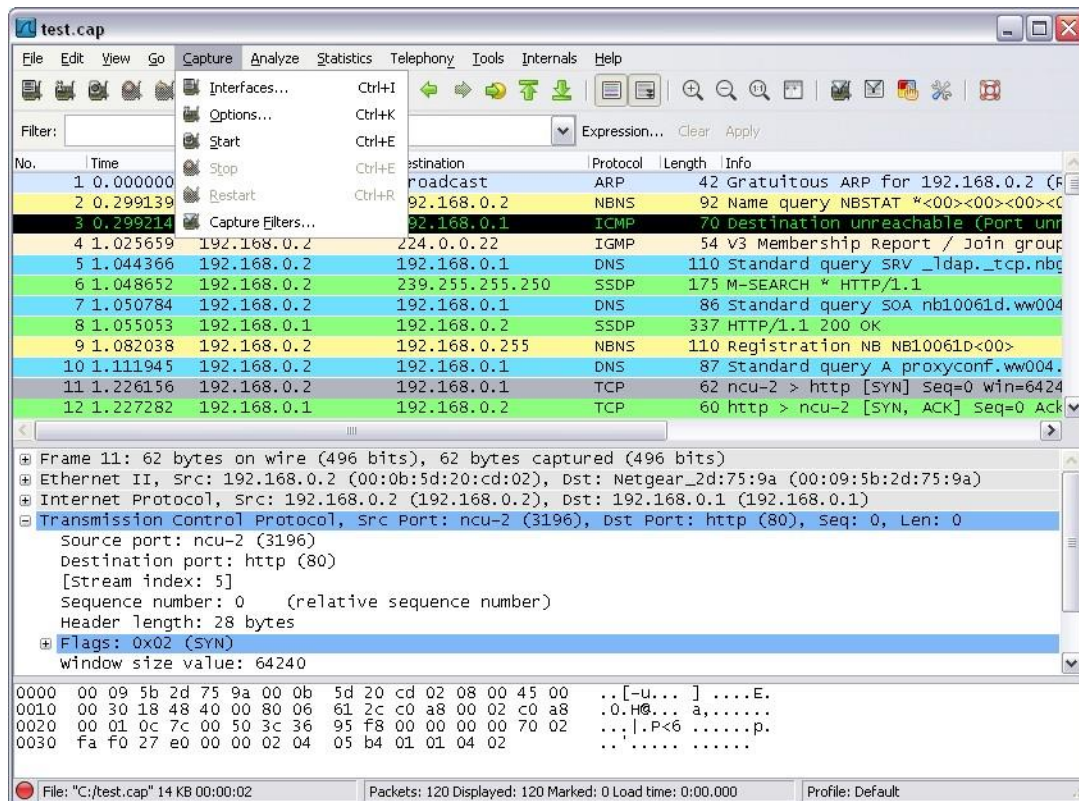
Start Wireshark.

On a Linux or Unix environment, select the Wireshark or Ethereal entry in the desktop environment's menu, or run "wireshark" (or "ethereal") from a root shell in a terminal emulator. Then Configure Wireshark



After starting Wireshark, do the following:

1. Select Capture | Interfaces

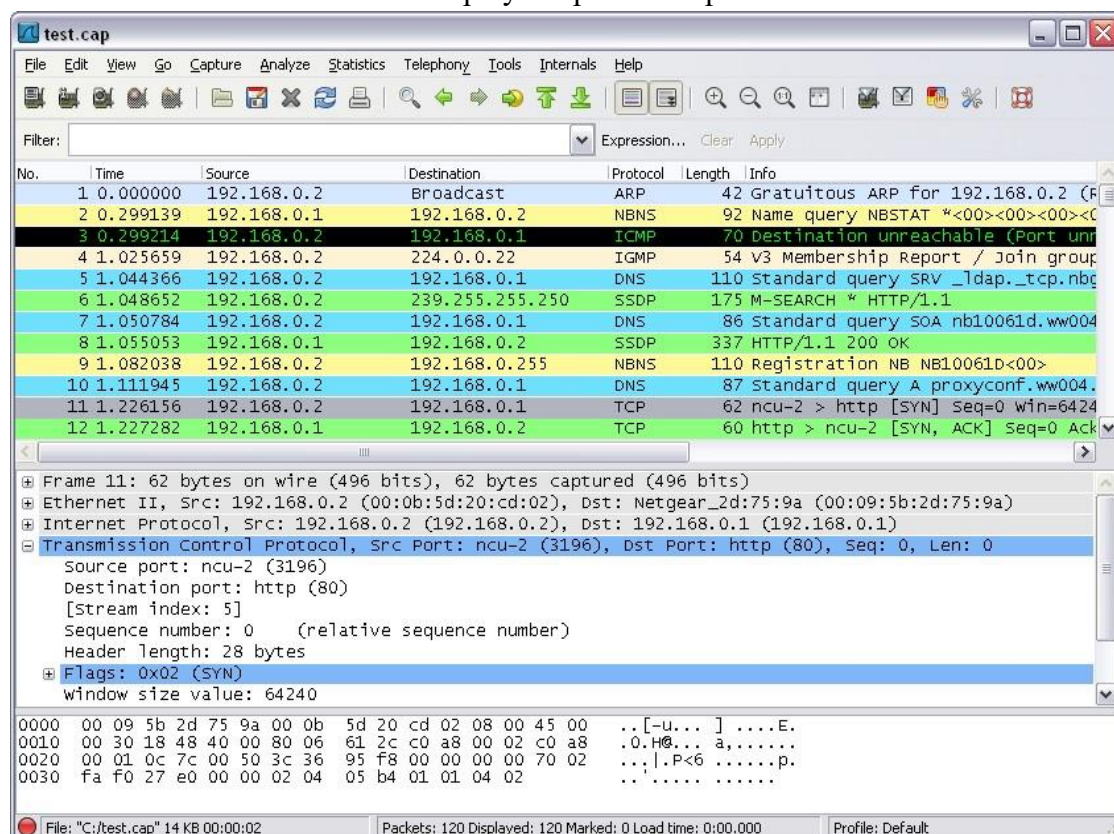


2. Select the interface on which packets need to be captured.
3. If capture options need to be configured, click the Options button for the chosen interface.

Note the following recommendations for traces that are to be analysed by Novell Technical Services:

- Capture packet in promiscuous mode: This option allows the adapter to capture all traffic not just traffic destined for this workstation. It should be enabled.
- Limit each packet to: Leave this option unset. Novell Support will always want to see full frames.
- Filters: Generally, Novell Support prefers an unfiltered trace. For documentation on filters, please refer to TID 10084702 - How to configure a capture filter for Ethereal (formerly NOVL90720).
- Capture file(s): This allows a file to be specified to be used for the packet capture. By default Wireshark will use temporary files and memory to capture traffic. Specify a file for reliability.
- Use multiple files, Ring buffer with: These options should be used when Wireshark needs to be left running capturing data for a long period of time. The number of files is configurable. When a file fills up, it will wrap to the next file. The file name should be specified if the ring buffer is to be used.
- Stop capture after xxx packet(s) captured: Novell Technical Support would most likely never use this option. Leave disabled.
- Stop capture after xxx kilobyte(s) captured: Novell Technical Support would most likely never use this option. Leave disabled.
- Stop capture after xxx second(s): Novell Technical Support would most likely never use this option. Leave disabled.

- Update list of packets in real time: Disable this option if the problem that's being investigated is occurring on the same workstation as where Wireshark is running.
 - Automatic scrolling in live capture: Wireshark will scroll the window so that the most current packet is displayed.
 - Hide capture info dialog: Disable this option so that you can view the count of packets being captured for each protocol.
 - Enable MAC name resolution: Wireshark contains a table to resolve MAC addresses to vendors. Leave enabled.
 - Enable network name resolution: Wireshark will issue DNS queries to resolve IP host names. Also will attempt to resolve network network names for other protocols. Leave disabled.
 - Enable transport name resolution: Wireshark will attempt to resolve transport names. Leave disabled.
4. Now click the Start button to start the capture.
 5. Recreate the problem. The capture dialog should show the number of packets increasing. If not, then stop the capture. Examine the interface list and pick the one that is not associated with the WANIP. It will probably be a long alpha-numeric string. If packets are still not being captured, try removing any filters that have been defined.
 6. Once the problem which is to be analyzed has been reproduced, click on Stop. It might take a few seconds for Wireshark to display the packets captured.



Results :

Packet sniffing/snooping was studied and understood. Wireshark tool for sniffing was used. It was studied how to capture specific packets from network traffic, get information about the packets like the protocol used, etc. It was also analysed to find out if websites have secure encryption for their

login page, i.e., username and password. It was discovered that oosai .com and fmworldmusic.com were two of the sites that do not use encryption for their password while login.

The screenshot displays a web browser window with the URL `fmworldmusic.com`. The page features a navigation bar with links for Home, Tamil Songs, Hindi Songs, Telugu Songs, Search, Create Playlist, Greetings, Community, Suggestion, and Contact. A search bar is present, and a user profile for 'anthony pillai' is visible. The main content area includes a 'Hello, welcome!' message, a 'Radio Station' dropdown, and a 'Now Playing Songs' section. A sidebar on the right shows a tweet and a list of users listening to a song.

Below the browser window, a Wireshark packet capture is shown. The filter is set to `tcp`. The packet list shows several TCP packets between `10.0.1.22` and `10.0.1.2`. The packet details pane shows the following data:

```

Frame 185: 693 bytes on wire (5544 bits), 693 bytes captured (5544 bits)
Ethernet II, Src: d4:3d:7e:38:c4:a9 (d4:3d:7e:38:c4:a9), Dst: SpeedDra_0e:01:0b (00:13:3b:0e:01:0b)
01b0 65 72 65 72 3a 20 68 74 74 70 3a 2f 2f 66 6d 77 erer: ht tp://fmw
01c0 6f 72 6c 64 6d 75 73 69 63 2e 63 6f 6d 2f 0d 0a orldmusi c.com/..
01d0 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 Content- Length:
01e0 39 34 0d 0a 43 6f 6f 6b 69 65 3a 20 50 48 50 53 94..Cook ie: PHPS
01f0 45 53 53 49 44 3d 37 6d 72 65 35 63 34 6b 37 6b ESSID=7m re5c4k7k
0200 6e 35 62 39 67 36 75 30 31 63 67 70 39 75 71 36 n5b96u0 lcp9uq6
0210 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 ..Connec tion: ke
0220 65 70 2d 61 6c 69 76 65 0d 0a 50 72 61 67 6d 61 ep-alive ..Pragma
0230 3a 20 6e 6f 2d 63 61 63 68 65 0d 0a 43 61 63 68 : no-cac he..Cach
0240 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6e 6f 2d 63 68 e-Contro l: no-ca
0250 63 68 65 0d 0a 0d 0a 75 73 65 72 6e 61 6d 65 3d che....u sername=
0260 6a 61 63 71 75 65 6c 69 6e 65 61 6e 74 68 6f 6e jacqueli neanthon
0270 79 31 39 32 32 25 34 30 79 61 68 6f 6f 2e 63 6f y1992x40 yahoo.co
0280 2e 69 6e 26 70 61 73 73 3d 61 62 63 31 32 33 26 .in&pass =abc123&
0290 61 63 74 69 76 69 74 79 3d 6c 6f 67 69 6e 26 4c activity =login&L
02a0 6f 67 69 6e 3d 4c 6f 67 69 6e 26 72 65 6d 65 6d ogin=Log in&remem
02b0 62 65 72 3d 30 ber=0
  
```

The status bar at the bottom of Wireshark indicates: File: `/tmp/wireshark_eth0_20150...`, Packets: 592 Displayed: 12 Marked: 0 Dropped: 0, Profile: Default.

The screenshot displays a web browser window with the Oosai.com website. The website has a green header with the logo and navigation links: HOME, SEARCH, REGISTER, MY OOSAI, and FEEDBACK. Below the header, there's a section for 'Hi Jacqueline | My Playlist | Signout' and a 'Free Software Downloads' link. The main content area features a 'Microsoft' banner with the text 'with wide angle, integrated Skype.' and a 'LAST UPDATES' section listing various Tamil songs and artists. A sidebar on the left contains links to 'OOSAI PANEL' (Tamil Movies, Tamil Devotional Albums, etc.) and 'SONG OF THE DAY'.

Below the browser window, a Wireshark 1.6.7 packet capture is shown. The filter is set to 'tcp'. The packet list shows several TCP connections between 10.0.1.22 and 10.0.1.2. The selected packet (Frame 226) is a POST request to 'http://www.oosai.com/member_login.cfm'. The packet details show the following data:

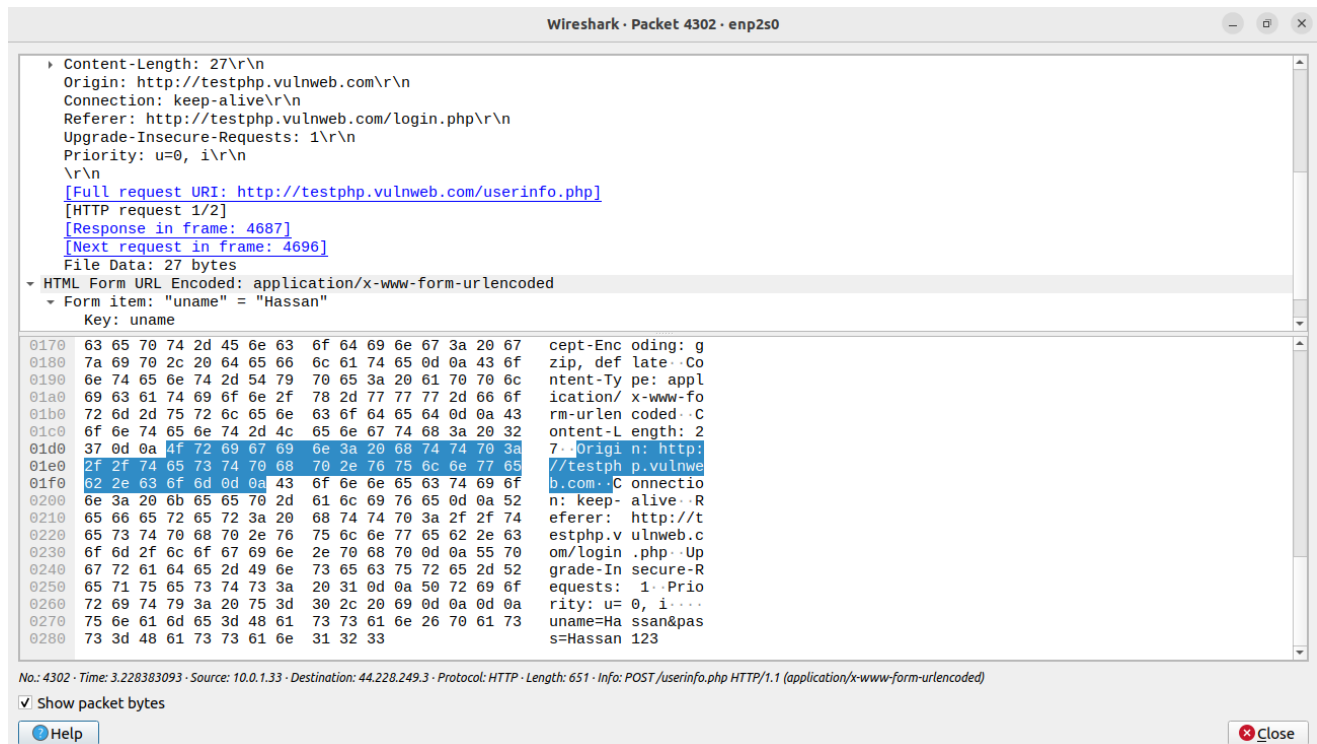
```

Frame 226: 603 bytes on wire (4824 bits), 603 bytes captured (4824 bits)
  Ethernet II, Src: d4:3d:7e:38:c4:a9 (d4:3d:7e:38:c4:a9), Dst: SpeedDra_0e:01:0b (00:13:3b:0e:01:0b)
  Internet Protocol Version 4, Src: 10.0.1.22 (10.0.1.22), Dst: 10.0.1.2 (10.0.1.2)
    Hypertext Transfer Protocol
      ..Referer: http://www.oosai.com/member_login.cfm
      ..Cookie: CFID=2416282; CFTOKEN=59270418
      ..Content-Type: application/x-www-form-urlencoded
      ..Content-Length: 66
      ..usrnm=jacqueline
      ..acq=1992%40yahoo.co
      ..in&pass=123456&submit=Login
  
```

References :

- <http://www.linuxjournal.com/content/packet-sniffing-basics>
- <http://www.internetgeeks.org/tech/hacking/top-10-data-packet-sniffing-analyzer-toolshackers/>
- <https://www.novell.com/support/kb/doc.php?id=3892415>
- https://www.wireshark.org/docs/wsug_html/
- <http://www.oosai.com/default.cfm?act=signout>
- <http://fmworldmusic.com/>

Lab practice (optional) :



Questions (Short, Long, MCQs) (optional) :

S1. What is Packet sniffing.

→ **Packet sniffing** is the process of intercepting and analyzing network traffic as it flows through a network. A packet sniffer captures data packets transmitted over a network, allowing network administrators or malicious actors to monitor, analyze, and inspect the contents of these packets.

S2. Differentiate promiscuous and non-promiscuous mode.

Mode	Promiscuous Mode	Non-Promiscuous Mode
Definition	The network interface captures all traffic, including packets not meant for it.	The network interface captures only the packets specifically addressed to it.
Use case	Used in packet sniffing, network monitoring, or intrusion detection.	Used in regular network communication.
Visibility	Can see all packets on the network.	Can only see packets meant for its own device.
Security Implications	Can be exploited for malicious purposes, such as unauthorized data capture.	Safer, as it restricts access to only intended traffic.