29_Lavena Babu

Don Bosco Institute of Technology, Mumbai 400070

Department of Information Technology

Experiment No. : 1

Date: 18/07/2022

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 databse
: 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.cacetech.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 asNetStumbler), 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 webspy 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

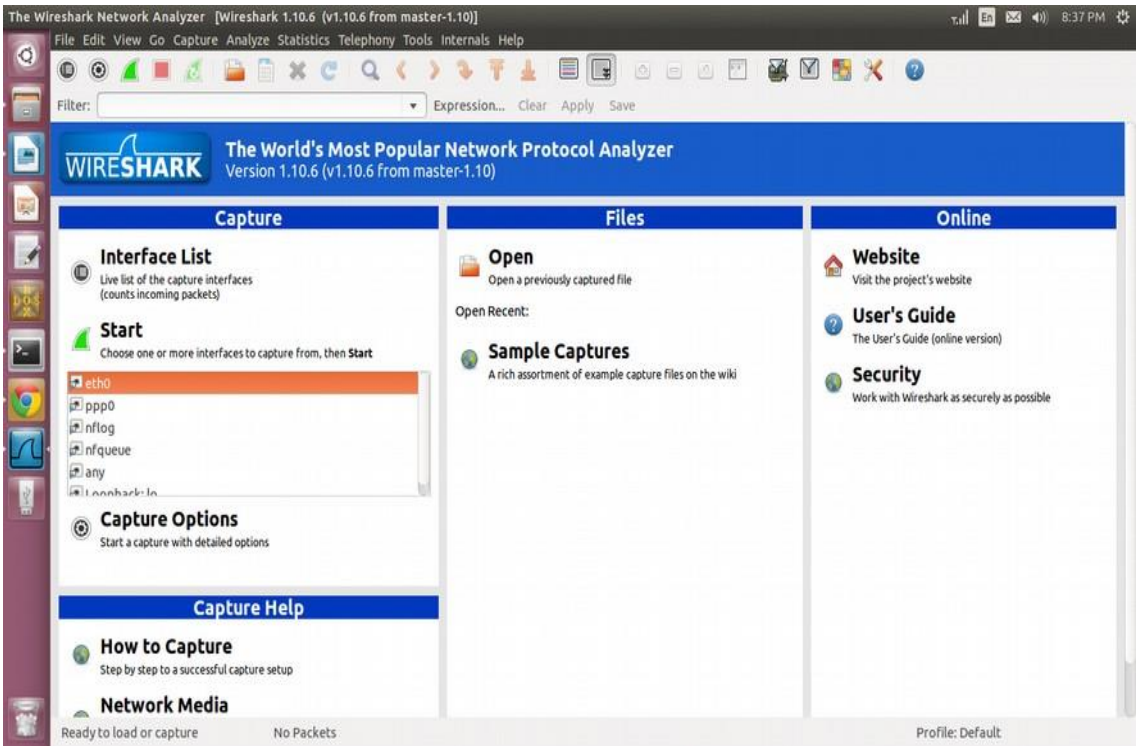
The following are some of the many features Wireshark provides:

- 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 :

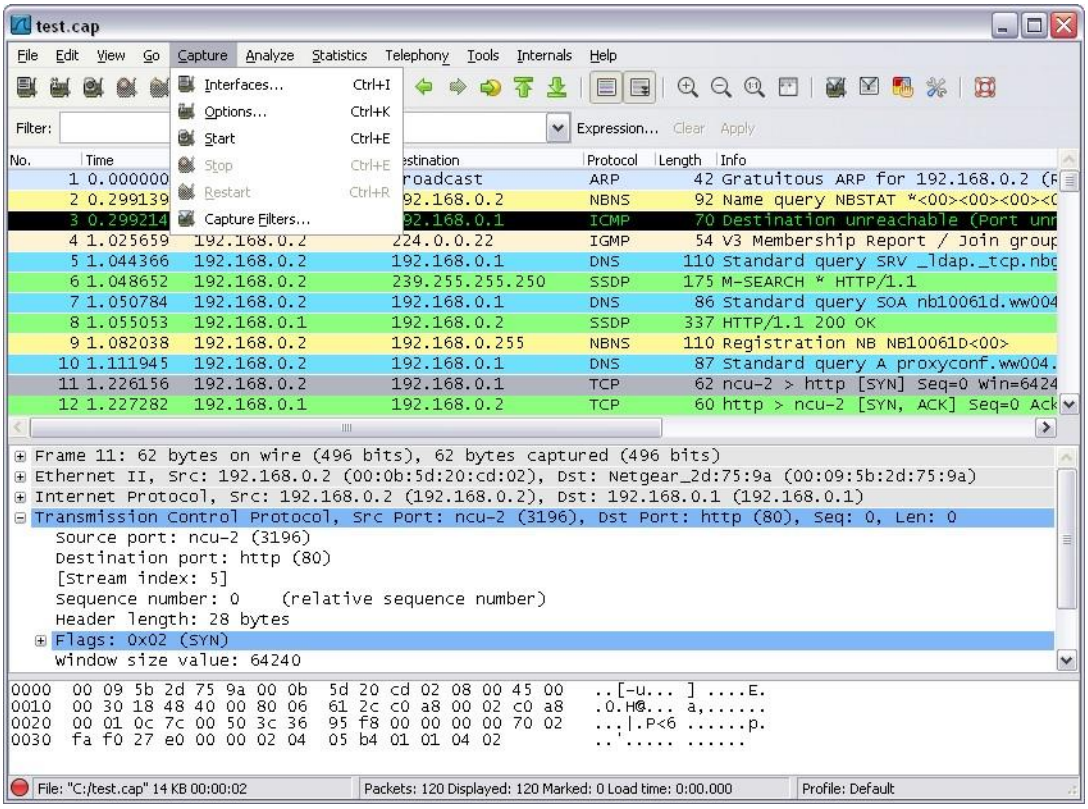
Start Wireshark.

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:

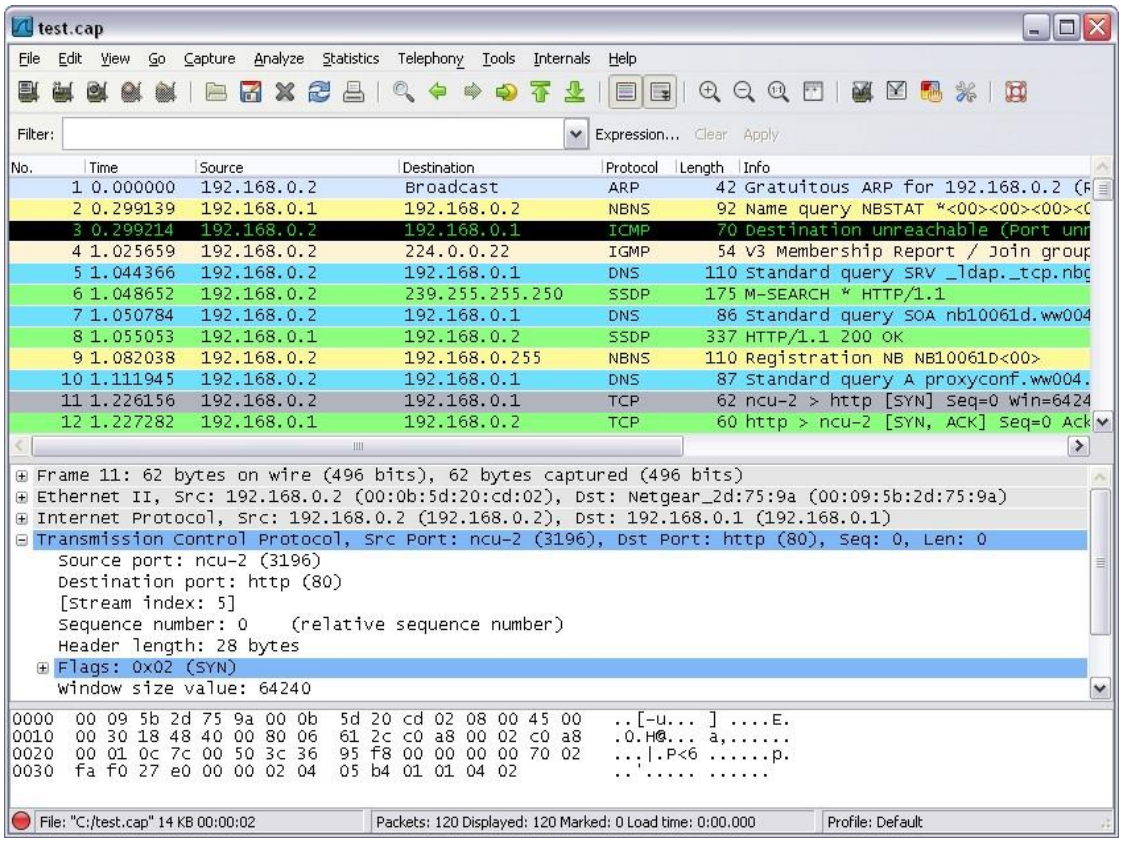
1.Select Capture | Interfaces



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 data for a long period of time. The number of files is configurable. When a file fills up, it 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.

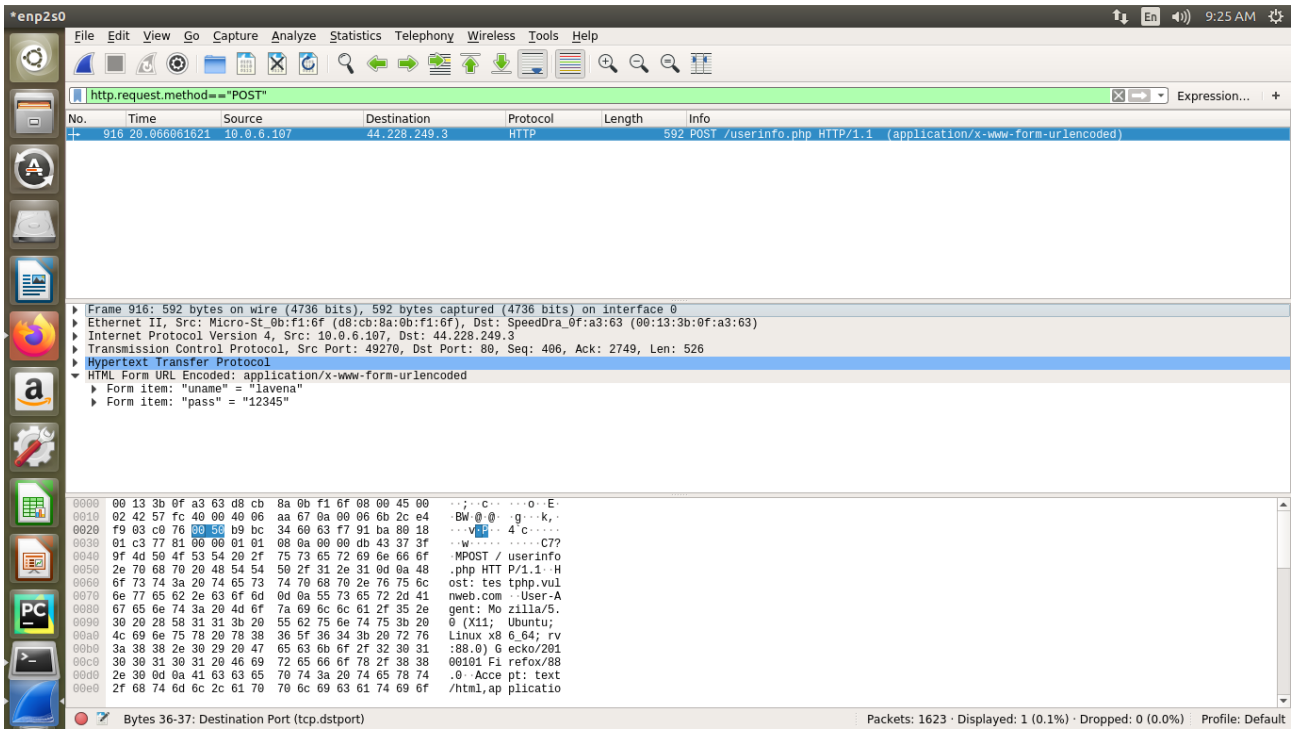
- Update list of packets in real time: Disable this option if the problem that's being investigated is occuring on the same workstation as where Wireshark is running.
- Automatic scrolling in live capture: Wireshark will scroll the window so that the most current packet is displayed.
- Hide capture info 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.



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/

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

S1. What is Packet sniffing.

When any data has to be transmitted over the computer network, it is broken down into smaller units at the sender's node called data packets and reassembled at receiver's node in original format. It is the smallest unit of communication over a computer network. It is also called a block, a segment, a datagram or a cell. The act of capturing data packet across the computer network is called packet sniffing.

S2. Differentiate promiscous and non-promiscous mode.

In promiscuous mode, the NIC allows all frames through, so even frames intended for other machines or network devices can be read. But, in non-promiscuous mode, when the NIC receives a frame, it drops it unless it is addressed to its specific media access control address or is a broadcast or multicast addressed frame.