

**LABORATORY REPORT**

**ON**

**Information & Network Security**

**CTBT CSE SVII L3**

**Submitted To**

**Dr. Rashi Chaudhary**

**Department of Cyber Security & Digital Forensics National Forensic Sciences University**

**Submitted By**

**DAYANANDA BINDHANI**

**(022200300004018)**

**National Forensic Sciences University**

**Delhi Campus, New Delhi – 110085, India**

TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
| Sr. No. | Practical Name | Pages |
| 1 | *Analysing network connections using 'netstat'. Using netstat to view information about incoming and outgoing network connections, routing table, etc. (include all the commands).* | 3-9 |
| 2 | *To monitor and analyse real-time TCP/IP connections on a Windows system using `CurrPorts`.* | 10-11 |
| 3 | *To monitor and analyse real-time TCP/IP connections on a Windows system using `TCPView`.* | 12-13 |
| 4 | *To create a disk image (.DD format) via FTK imager, do its analysis using Autopsy and generate the report.* | 14-17 |
| 5 | *Using Wireshark analyse and filter the TCP (SYN, ACK, Packet transmissions) details.* | 18-19 |
| 6 | *Using Wireshark analyse and filter the HTTP `get` method.* | 20-21 |
| 7 | *Using Wireshark analyse and filter the HTTP `post` method.* | 22-23 |
| 8 | *Using `nmap` to discover, scan, IP, ports, services, OS, versions.* | 24-26 |

**PRACTICAL – 1**

* **AIM:**

*Analysing network connections using `netstat`. Using netstat to view information about incoming and outgoing network connections, routing table, etc. (include all the commands).*

* **TOOLS/APPLICATIONS USED:**

netstat

* **THEORY:**

Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, this command displays active TCP connections.

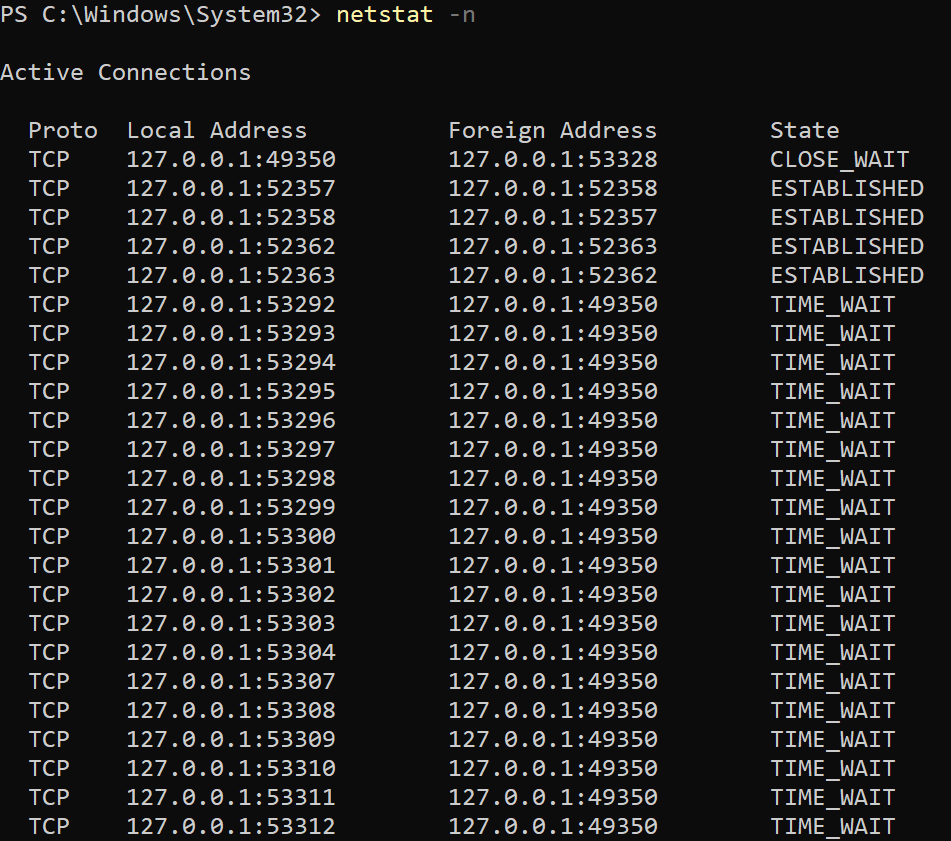
* **PROCEDURES:** 
  + Launch your terminal or command-line interface.
  + Execute netstat to view active and listening connections (incoming & outgoing).
  + Inspect the displayed list: local addresses, foreign addresses, connection states.
  + Display the routing table to check how network paths are configured.
  + Examine protocol statistics to identify unusual behaviour (e.g., many TCP in unusual states).
  + Look at interface statistics to detect packet drops, errors or high-volume traffic.
  + Correlate connection states (e.g., ESTABLISHED, TIME\_WAIT, SYN\_SENT) with expected network activity.
  + Note any unexpected entries: unknown remote addresses, unexpected listening ports, many connections in anomalous states.
  + Save/capture the output for your report or further analysis.
  + Conclude by interpreting what the results suggest about network health, routing correctness or suspicious activity.
* **OUTPUT:**

**COMMAND:** *netstat -a*

A screenshot of a computer

AI-generated content may be incorrect.

**COMMAND:** *netstat -n*



**COMMAND:** *netstat -b*

A screenshot of a computer program

AI-generated content may be incorrect.

**COMMAND:** *netstat -e*

A screen shot of a computer

AI-generated content may be incorrect.

**COMMAND:** *netstat -o*

A screenshot of a computer

AI-generated content may be incorrect.

**COMMAND:** *netstat -p tcp*

A screenshot of a computer screen

AI-generated content may be incorrect.

**COMMAND:** *netstat -s*

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**COMMAND:** *netstat -r*

A screenshot of a computer screen

AI-generated content may be incorrect.

**PRACTICAL – 2**

* **AIM:**

*To monitor and analyse real-time TCP/IP connections on a Windows system using `CurrPorts`.*

* **TOOLS/APPLICATIONS USED:**

CurrPorts

* **THEORY:**

CurrPorts is network monitoring software that displays the list of all currently opened TCP/IP and UDP ports on your local computer. For each port in the list, information about the process that opened the port is also displayed, including the process name, full path of the process, version information of the process (product name, file description, and so on), the time that the process was created, and the user that created it.

* **PROCEDURES:** 
  + Download CurrPorts and run it (it’s portable — no installation needed).
  + Open the program so you can see the list of currently open TCP & UDP ports, with info like process name, local/remote address and port, state, etc.
  + Use the “Options” or View menu to enable display of listening ports, established connections, and any states you care about (e.g., TIME\_WAIT, CLOSE\_WAIT).
  + Optionally adjust refresh interval or enable automatic refresh so you can watch changes in real-time.
  + Use filters (include/exclude) to focus on specific processes, ports, or remote IP ranges you want to monitor.
  + Review the list for unusual or unexpected entries: e.g., unknown process names, remote addresses you didn’t expect, many connections in odd states. Marked items may be flagged (e.g., pink) if the application is unidentified.
  + If needed, select one or more connections to close them (or the process that opened them) — useful for terminating unwanted/unknown connections.
  + Save or export the current list (to text, HTML, XML) for reporting or further investigation.
* **OUTPUTS:**

A screenshot of a computer

AI-generated content may be incorrect.

**PRACTICAL – 3**

* **AIM:**

*To monitor and analyse real-time TCP/IP connections on a Windows system using `TCPView`.*

* **TOOLS/APPLICATIONS USED:**

TCPView

* **THEORY:**

TCPView is a lightweight and straightforward utility that is part of the venerable Sysinternals Suite, now owned by Microsoft. It is a program that will show detailed listings of all TCP and UDP endpoints on the system, including the local and remote addresses and state of TCP connections. TCPView also reports the name of the process that owns the endpoint. It provides a more informative and conveniently presented subset of the Netstat program that ships with the OS. It’s download includes Tcpvcon, a command-line version with the same functionality.

* **PROCEDURES:** 
  + Download and launch TCPView.
  + Let it list all active TCP and UDP endpoints (local & remote addresses + owning process).
  + Observe the table: process name, PID, protocol, local/remote address/port, connection state.
  + Watch live updates: new connections shown in green, terminated in red, changed states highlighted.
  + Use filters/search to focus on specific ports, processes or remote addresses of interest.
  + Identify any unexpected or suspicious entries — e.g., unknown process making outbound connections, connections in odd states.
  + If needed, close a connection or terminate its owning process via right-click/context menu.
  + Save/export the view for reporting (to text/HTML) for later analysis.
  + After monitoring, interpret what you observed: normal vs unusual behavior, what the connections imply, and if any action is required.
* **OUTPUT:**

A screenshot of a computer program

AI-generated content may be incorrect.

**PRACTICAL – 4**

* **AIM:**

*To create a disk image (.DD format) via `FTK imager`, do its analysis using `Autopsy` and generate the report.*

* **TOOLS/APPLICATIONS USED:**
* FTK Imager
* Autopsy
* **THEORY:**

FTK Imager is a forensic-imaging and preview tool developed by AccessData (now under Exterro) that lets investigators create bit-for-bit copies (images) of storage media (hard drives, external drives, USBs) and capture volatile memory in some cases.

Autopsy is an open-source, graphical digital forensics platform built on top of The Sleuth Kit (TSK) that provides investigators a UI to analyse disk images, file systems, recover deleted files, perform timeline analysis, keyword search, web artefact extraction, etc.

* **PROCEDURES:** 
  + Prepare your forensic workstation and ensure you have write-blocking enabled on the source drive (so you don’t alter the original evidence).
  + Launch FTK Imager.
  + In FTK Imager: select **File → Create Disk Image**.
  + Choose the source evidence type (e.g., Physical Drive) and select the correct drive you want to image.
  + Choose the image destination: select destination folder, filename, and importantly select the “Raw (dd)” / “.DD” / “Raw image” format in the image type drop-down.
  + Enter case information or metadata (case number, examiner name, description) if required (helps documentation).
  + Optional but recommended: check the “Verify images after they are created” or similar verification/hash option. This generates hash values (MD5/SHA1) of the image to confirm integrity.
  + Click Start (or finish as prompted) and wait for the imaging process to complete. The tool will produce the raw image - .DD (or .001/.002 segments if fragmented) and the log/hash report.
  + Once completed, confirm the hash values match, and note the image filename, size, destination path, hash values and tool version for your documentation.
* Launch Autopsy.
* Create a new case: provide case name, base directory, etc.
* Add your data source: choose “Disk Image or VM File” and browse to your .DD image file you created earlier.
* Configure the ingest modules: select which modules (e.g., file type identification, keyword search, recent activity, hash lookup) to run.
* Run the ingest/analysis process: Autopsy will process the image, parse file systems, recover deleted files, carve unallocated space, build timelines, etc.
* Browse the results: review artefacts such as user files, system logs, browser history, email, USB device history, deleted files, timeline of events.
* Perform searches/filters as needed: keyword search, filter by file type, time ranges, hash matches, etc.
* Tag relevant items/artefacts: mark items of interest (evidence), add notes or bookmarks for reporting.
* Generate the report: use Autopsy’s report generation feature (export in HTML/PDF format) summarizing case metadata, evidence sources, findings, tagged artefacts, summary of results.
* **OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**PRACTICAL – 5**

* **AIM:**

*Using Wireshark analyse and filter the TCP (SYN, ACK, Packet transmissions) details.*

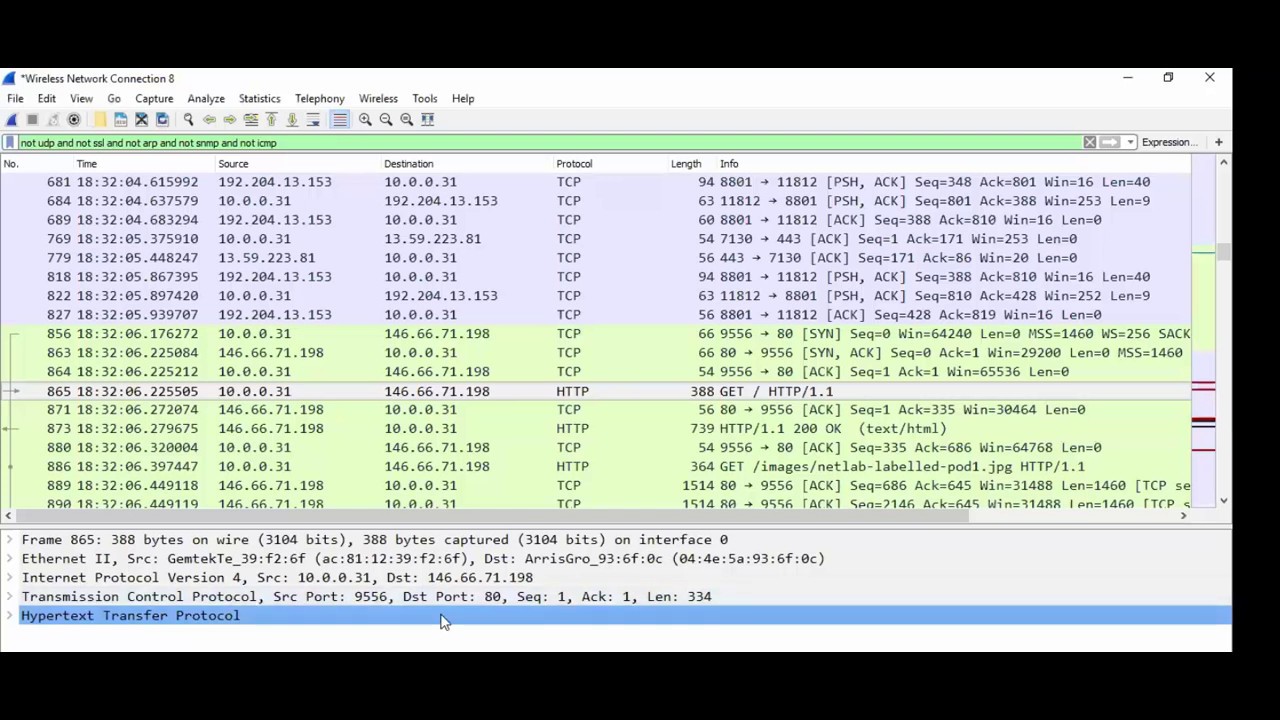
* **TOOLS/APPLICATIONS USED:**
* Wireshark
* **THEORY:**

Wireshark is a powerful, open-source network protocol analyzer that allows users to capture and interactively browse the traffic running on a computer network, providing deep inspection of hundreds of protocols.

Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.

* **PROCEDURES:** 
  + Open Wireshark and select the network interface through which traffic will be captured.
  + Start the packet capture on that interface.
  + Initiate a TCP connection (for example by opening a website or connecting to a service).
  + After the connection is established, stop the capture.
  + Apply a filter to display only TCP-related packets.
  + Locate and inspect the first three TCP packets of the conversation (SYN from client, SYN+ACK from server, ACK from client) to verify the handshake.
  + Examine subsequent packets in the TCP stream for sequence numbers, acknowledgment numbers, payload length and any anomalies (e.g., retransmissions).
  + Use the “Follow TCP Stream” feature (or equivalent) to view the entire conversation context.
  + Save the capture file and export relevant packet details, screenshots or summaries for your report.
  + In your report: note the interface used, start/stop times, key handshake packet details, any unusual findings in the TCP transmissions and your interpretation of what that indicates about the connection.
* **OUTPUT:**

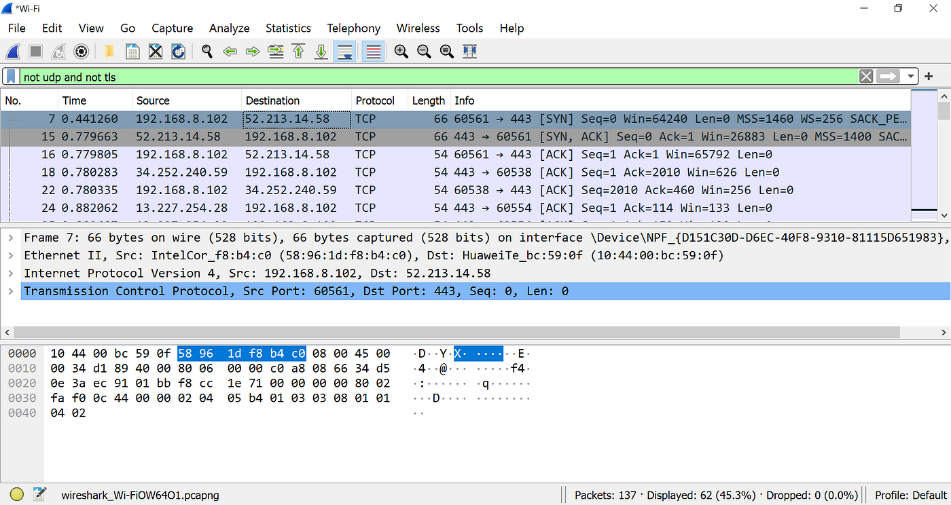
**TCP -> SYN**



**TCP -> SYN/ACK**



**TCP -> ACK**



**PRACTICAL – 6**

* **AIM:**

*Using Wireshark analyse and filter the HTTP `get` method.*

* **TOOLS/APPLICATIONS USED:**
* Wireshark
* **THEORY:**

Wireshark is a powerful, open-source network protocol analyzer that allows users to capture and interactively browse the traffic running on a computer network, providing deep inspection of hundreds of protocols.

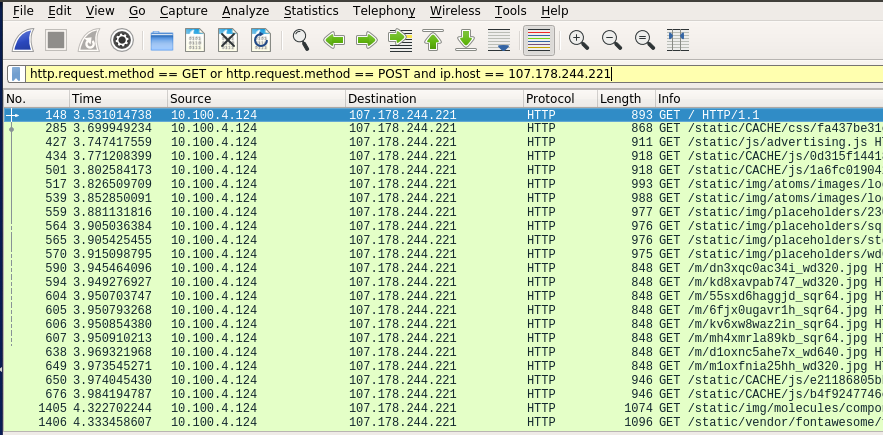
Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.

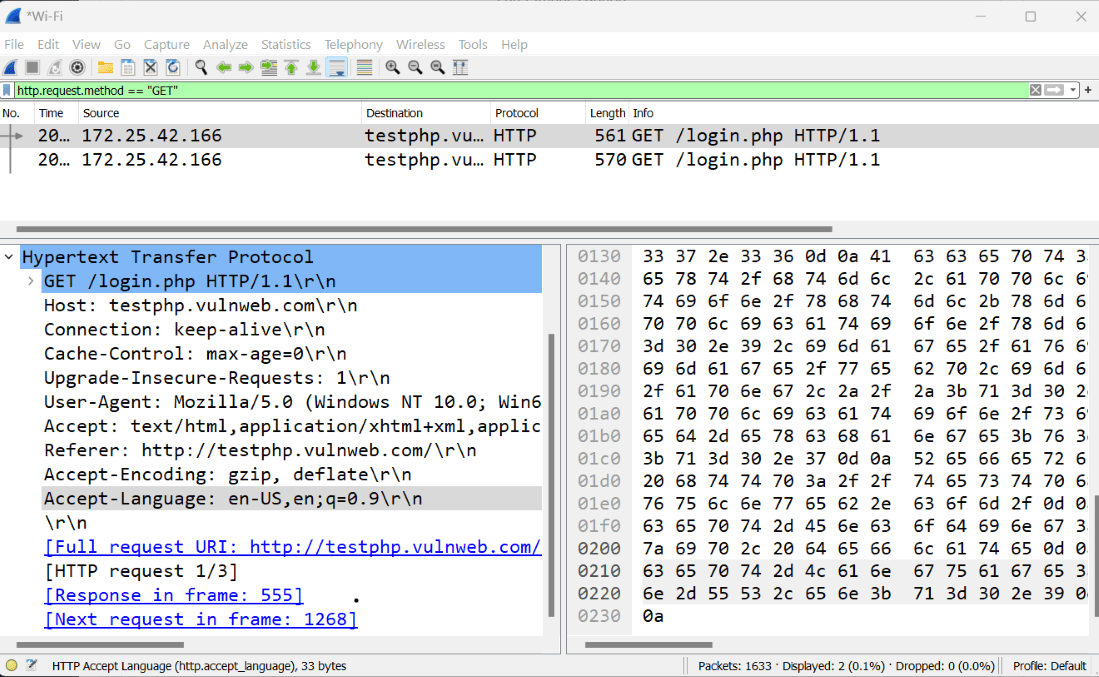
* **PROCEDURES:** 
  + Launch Wireshark and select the network interface to capture.
  + Start the packet capture and then trigger a simple HTTP GET (for example, open a webpage).
  + Stop the capture once the request and response are captured.
  + Apply a filter to only show HTTP traffic (so you focus on GET requests).
  + In the packet list, locate the GET request packet from the client to the server.
  + Expand its details: check the HTTP method, requested URL, headers, and payload length.
  + Locate the corresponding HTTP response packet from server to client: check status code, headers, data size.
  + Optionally follow the TCP stream to view the full request-response conversation.
* **OUTPUT:**

**HTTP METHOD `GET`**

A screenshot of a computer

AI-generated content may be incorrect.





**PRACTICAL – 7**

* **AIM:**

*Using Wireshark analyse and filter the HTTP `post` method.*

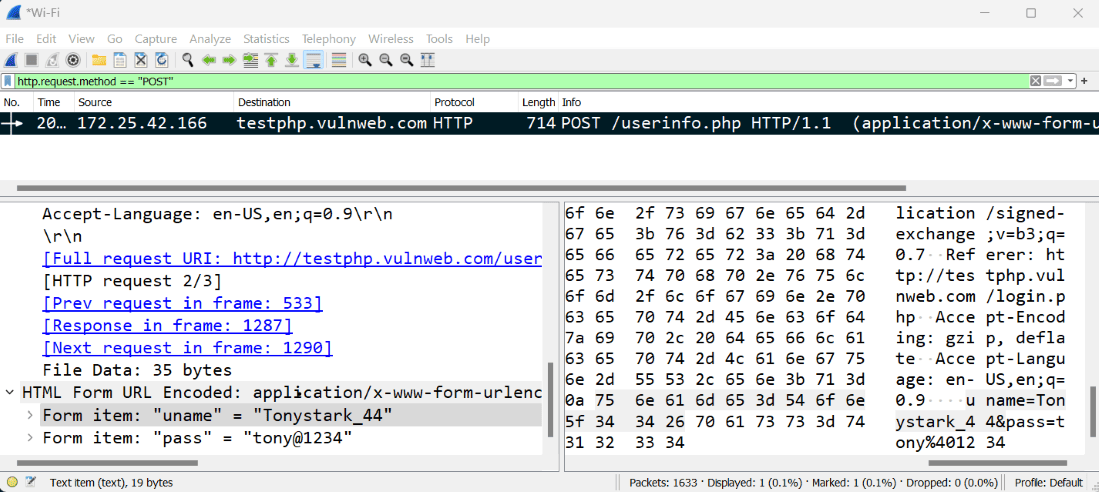
* **TOOLS/APPLICATIONS USED:**
* Wireshark
* **THEORY:**

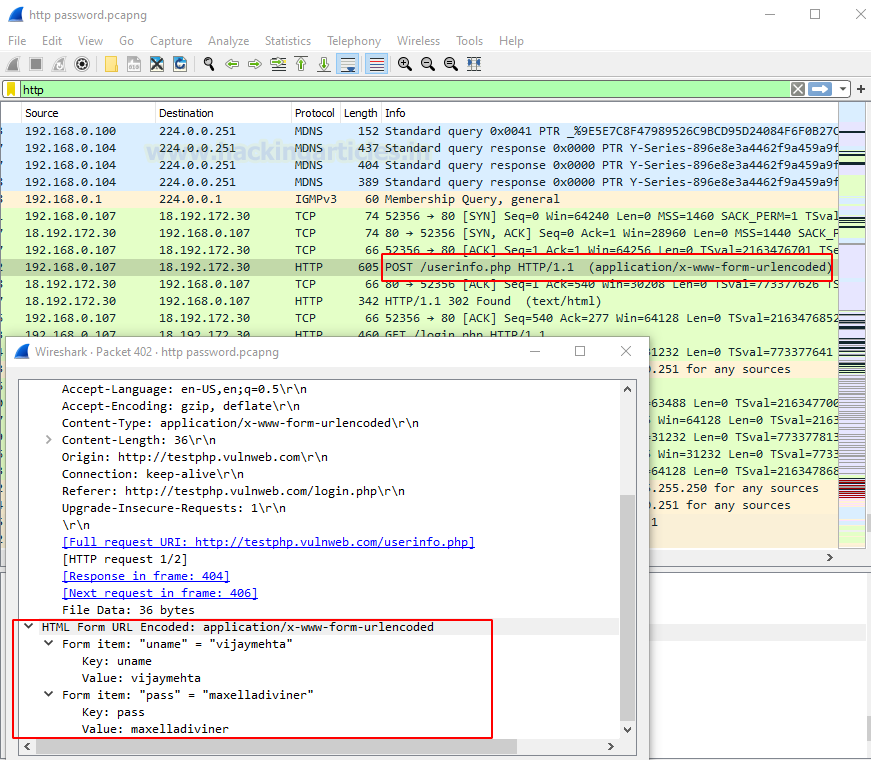
Wireshark is a powerful, open-source network protocol analyzer that allows users to capture and interactively browse the traffic running on a computer network, providing deep inspection of hundreds of protocols.

Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.

* **PROCEDURES:** 
  + Launch Wireshark and select the network interface to capture.
  + Start capturing packets.
  + Trigger a HTTP POST transaction (for example submit a form or an API request).
  + Stop the capture once the request and server response are present.
  + Apply a filter to show only HTTP traffic (so you isolate POST requests).
  + In the packet list, locate the POST request from client to server. Expand details: check HTTP method, URL, headers, payload length.
  + Locate the corresponding HTTP response from server to client: check status code, headers, data size.
* **OUTPUT:**

**HTTP METHOD `POST`**





**PRACTICAL – 8**

* **AIM:**

*Using `Nmap` to discover, scan, IP, ports, services, OS, versions etc.*

* **TOOLS/APPLICATIONS USED:**
* Nmap
* **THEORY:**

Nmap (“Network Mapper”) is an open-source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

* **PROCEDURES:** 
  + Determine the target host (IP address or hostname) you’ll scan.
  + Launch Nmap on your scanning machine with appropriate privileges.
  + Perform a basic host/port discovery to see which IPs are up and which ports respond.
  + Once you identify live host(s), perform a port scan to find open/closed/filtered TCP/UDP ports.
  + Enable service/version detection to identify services running on open ports and their versions.
  + Enable OS / device detection to attempt to identify the operating system and device type.
  + Review the scan results: IP address, open ports, service names & versions, OS details, device type.
* **OUTPUT:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen with white text

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A computer screen with white text

AI-generated content may be incorrect.**