Network Security and Forensics

Lab Session 3

Submitted To:-

Dr. Lokesh Chauhan Sir

Submitted By:-

Saloni Rangari

M.Tech. AIDS

I. Perform the depth protocol analysis by Wireshark.

1. Set up Wireshark to capture packets on a network interface.

1.1. **Open Wireshark:** Launch the application. You will see a list of available network interfaces.

1.2. **Select the Network Interface:** Choose the network interface you want to capture packets from. This could be your Ethernet or Wi-Fi connection. Click on the interface to highlight it.

1.3. **Start Capturing Packets:** Click the "Start capturing packets" button (the shark fin icon) or go to the menu and select Capture -> Start. You can also use the keyboard shortcut Ctrl + E to begin capturing.

a) Capture and analyze various types of traffic, such as HTTP, DNS, and FTP

i) Capturing Traffic

(1) Open Wireshark: Launch the Wireshark application.

(2) Select the Network Interface: Choose the appropriate network interface (e.g., Ethernet or Wi-Fi) from the list of available interfaces.

(3) Start Capturing Packets: Click on the "Start capturing packets" button (the shark fin icon) or go to Capture -> Start to begin capturing traffic.

ii) Generating Traffic

(1) Perform Activities: While capturing packets, generate the desired types of network traffic. For example:

(2) HTTP: Visit websites, download files, or make HTTP requests.

(3) DNS: Resolve domain names by visiting websites or using DNS lookup tools.

(4) FTP: Connect to FTP servers and transfer files.

(5) Stop the Capture: Once you have captured sufficient data, stop the capture by clicking the red square button or selecting Capture -> Stop.

iii) Analyzing HTTP Traffic

(1) Apply the HTTP Display Filter: In the display filter bar, type http and press Enter. This will filter the captured packets to show only HTTP traffic.

(2) Examine HTTP Requests and Responses: Expand the HTTP packets to view details such as:

(3) Request Methods: GET, POST, PUT, DELETE, etc.

(4) Request and Response Headers: Analyze headers like Host, User-Agent, Content-Type, etc.

(5) Request and Response Bodies: View the content being transferred, such as HTML pages or JSON data.

iv) Investigate HTTP Status Codes: Look for HTTP status codes in the responses to understand the server's behavior, such as:

(1) 200 OK: Successful request

(2) 404 Not Found: Resource not found

(3) 500 Internal Server Error: Server-side error

iv) Analyzing DNS Traffic

(1) Apply the DNS Display Filter: In the display filter bar, type dns and press Enter. This will filter the captured packets to show only DNS traffic.

(2) Examine DNS Queries and Responses: Expand the DNS packets to view details such as:

(3) Query Name: The domain name being resolved

(4) Query Type: The type of DNS record being queried (A, AAAA, MX, etc.)

(5) Response Codes: The status of the DNS response, such as NOERROR or NXDOMAIN

(6) Identify Authoritative DNS Servers: Look for the authoritative DNS servers in the DNS response packets.

v) Analyzing FTP Traffic

(1) Apply the FTP Display Filter: In the display filter bar, type ftp and press Enter. This will filter the captured packets to show only FTP traffic.

(2) Examine FTP Commands and Responses: Expand the FTP packets to view details such as:

(3) FTP Commands: USER, PASS, LIST, RETR, STOR, etc.

(4) FTP Responses: The server's responses to the FTP commands

(5) Identify FTP Authentication: Look for USER and PASS commands to see if FTP authentication is used.

(6) Analyze File Transfers: Observe the RETR and STOR commands to see which files are being downloaded or uploaded.

b) Examine packet headers, source and destination IP addresses, port numbers, and protocols.

i) **Packet Header Analysis**

(1) Expand the Packet Layers: Click on the arrows next to each layer to expand and view more details. Common layers include:

(2) Ethernet II: Displays the Ethernet frame details, including source and destination MAC addresses.

(3) Internet Protocol Version 4 (IPv4): Shows the IP header details.

(4) Transmission Control Protocol (TCP) or User Datagram Protocol (UDP): Displays transport layer details.

ii) **Identify Source and Destination IP Addresses**

(1) In the IPv4 section, look for:

(2) Source IP Address: Indicates the sender's IP address.

(3) Destination IP Address: Indicates the receiver's IP address.

iii) **Check Port Numbers**

(1) In the TCP or UDP section, identify:

(2) Source Port: The port number used by the sender.

(3) Destination Port: The port number used by the receiver.

(4) Common ports include:

(5) HTTP: Port 80

(6) HTTPS: Port 443

(7) FTP: Port 21

(8) DNS: Port 53

iv)**Determine the Protocol**

(1)The Protocol field in the IPv4 header indicates the transport layer protocol used, such as:

(2)TCP: Reliable, connection-oriented protocol.

(3)UDP: Connectionless protocol, often used for streaming or real-time applications.

(4)The protocol can also be seen in the Packet List Pane under the Protocol column.

(5)8. View Additional Information

(6)For more detailed analysis, you can also expand sections like:

(7)HTTP: To view request methods, headers, and content.

(8)DNS: To see query types and response codes.

Use the Follow Feature

To analyze the entire conversation between two endpoints, right-click on a TCP or UDP packet and select Follow > TCP Stream or UDP Stream. This will filter and display all packets in that particular stream.

Save and Export Data

If you need to keep a record of your findings, you can save the captured packets or export specific packet details by going to File -> Export Specified Packets.

c) Use display filters to focus on specific types of traffic

1. **Open Wireshark**: Launch the application and start capturing packets on your desired network interface.
2. **Locate the Display Filter Bar**: This is located just above the packet list pane.
3. **Enter Basic Filters**: Start with simple filters to focus on specific protocols:
    - **HTTP Traffic**: Type http and press Enter.
    - **DNS Traffic**: Type dns and press Enter.
    - **FTP Traffic**: Type ftp and press Enter.
4. **Filter by IP Address**: To focus on traffic from or to a specific IP address:
    - Source IP: ip.src == 192.168.1.10
    - Destination IP: ip.dst == 192.168.1.20
5. **Filter by Port Number**: To isolate traffic on specific ports:
    - HTTP: tcp.port == 80
    - HTTPS: tcp.port == 443
    - DNS: udp.port == 53
6. **Combine Filters**: Use logical operators to create complex filters:
    - AND: http && ip.src == 192.168.1.10
    - OR: http || dns
    - NOT: http && !ip.src == 192.168.1.10
7. **Clear Filters**: To reset the view, delete the filter expression and press Enter.

d) Experiment with different display filter expressions

1. Start Capturing Traffic: Follow the steps above to capture network traffic.
2. Try Different Filters: Enter various expressions in the display filter bar:
    - Filter by Protocol:
        - tcp to see all TCP packets.
        - udp for UDP packets.
    - Filter by Specific Conditions:
        - http.request.method == "GET" to see all GET requests.
        - dns.qry.name == "example.com" to filter DNS queries for a specific domain.
3. Use Comparison Operators: Experiment with operators:
    - ip.addr != 192.168.1.10 to exclude traffic from a specific IP.
    - tcp.port >= 1000 to see all TCP traffic on high-numbered ports.
4. Regular Expressions: Use regex for advanced filtering:
    - http.request.uri matches "\.png$" to find HTTP requests for PNG images.
5. Observe Changes: As you apply different filters, observe how the packet list updates. Take note of:
    - The number of packets displayed.
    - The types of packets that meet your criteria.

e) Explore and understand color-coded packet markings.

1. View the Packet List Pane: As you capture and filter packets, observe the colors of the entries in the packet list.
2. Default Color Coding: By default, Wireshark uses a variety of colors to indicate different protocols:
    - Light Purple: TCP packets
    - Light Green: HTTP packets
    - Dark Green: DNS packets
    - Red: Packets with errors or issues
3. Customize Color Rules: You can modify the color rules to suit your preferences:
    - Go to View -> Coloring Rules to see the current rules.
    - You can add, edit, or delete rules based on your needs.
4. Use Color Coding for Quick Analysis: When analyzing traffic, use the color-coding to quickly identify:
    - The types of protocols in use.
    - Any potential issues indicated by error colors.
5. Practice with Color-Coded Filters: Combine color-coding with display filters:
    - Use a filter like http and observe how the color coding helps you quickly identify relevant packets.

f) Analyze the handshake process

1. **Start capturing packets** on the desired network interface.
2. **Generate TCP traffic** by visiting a website or performing an action that initiates a TCP connection.
3. **Stop the capture** once you have enough data.
4. **Apply a display filter** to isolate the TCP handshake packets:

   tcp.flags.syn == 1 && tcp.flags.ack == 0

This filter will show only SYN packets, which initiate the handshake.

5. **Examine the SYN packets** to identify the source and destination ports, and the initial sequence numbers used by each host.
6. **Look for the corresponding SYN-ACK packets** from the destination host, which acknowledge the SYN and provide their own initial sequence number.
7. **Identify the final ACK packets** that complete the handshake, acknowledging the SYN-ACK.
8. **Observe the sequence and acknowledgment numbers** incrementing throughout the handshake process.

g) Investigate DNS query and response packets.

1. Apply the display filter dns to isolate DNS traffic.
2. Look for DNS query packets containing questions, such as A records for domain names.
3. Identify the corresponding DNS response packets, which will have the same transaction ID as the query.
4. Examine the response packets for the returned IP addresses or other DNS record types.
5. Check the DNS response codes to ensure a successful resolution (NOERROR).

h) Examine HTTP requests and responses, including headers and content.

1. Apply the display filter http to focus on HTTP traffic.
2. Identify HTTP request packets, which will contain the request method (GET, POST, etc.), requested URI, and headers.
3. Look for the corresponding HTTP response packets, which will have the same TCP conversation.
4. Examine the response packets for the returned status code (200 OK, 404 Not Found, etc.), headers, and content.
5. Observe the sequence and acknowledgment numbers to understand the flow of the HTTP conversation.

II. Packet Filtering and Display Filters:

a) Use Wireshark to capture network traffic in a specific scenario (e.g., browsing a website, downloading a file).

1. Open Wireshark and select the appropriate network interface for capturing traffic.
2. Start capturing packets by clicking the shark fin icon or using the shortcut Ctrl + E.
3. Perform the desired scenario, such as browsing a website or downloading a file, to generate relevant network traffic.
4. Stop the capture once you have collected sufficient data.

b) Apply display filters in Wireshark to isolate specific types of packets (e.g., HTTP, TCP).

1. Locate the display filter bar in Wireshark, usually at the top of the packet list pane.
2. Enter a display filter expression to isolate specific types of packets:
    - HTTP traffic: http
    - TCP traffic: tcp
    - DNS traffic: dns
    - FTP traffic: ftp
3. Press Enter to apply the filter and observe the updated packet list.

c) Experiment with different display filters and observe how they affect the captured traffic view.

1. Try various display filter expressions to see their impact on the captured traffic view:
    - Filter by IP address: ip.addr == 192.168.1.10
    - Filter by port number: tcp.port == 80
    - Combine filters with logical operators:
        - AND: http && ip.src == 192.168.1.10
        - OR: http || dns
        - NOT: http && !ip.src == 192.168.1.10
2. Observe how the packet list changes based on the applied filter:
    - The number of packets displayed
    - The types of packets that match the filter criteria
3. Clear the filter by deleting the expression and pressing Enter to reset the view.

d) How can display filters help in focusing on relevant packets during network analysis?

Display filters in Wireshark are crucial for focusing on relevant packets during network analysis. They help in the following ways:

1. Isolating specific types of traffic: By filtering packets based on protocols, IP addresses, or ports, you can quickly identify and analyze the traffic of interest.
2. Troubleshooting network issues: Display filters allow you to narrow down the packet view to specific scenarios, making it easier to pinpoint the source of problems.
3. Enhancing performance: Applying relevant filters reduces the amount of data displayed, improving Wireshark's performance, especially when dealing with large captures.
4. Facilitating targeted analysis: Focused filters enable you to concentrate on specific aspects of the network traffic, such as HTTP requests, DNS queries, or TCP handshakes, for in-depth analysis.
5. Saving time and effort: By quickly isolating relevant packets, display filters save time and effort compared to manually searching through the entire packet capture.