Understanding Network Traffic By Using WireShark (Feel free to use this work for your project, if you need to)

Creating a comprehensive manual for Wireshark would be a lengthy task, but I can provide you with a basic overview of how to use Wireshark and what you can learn from analyzing network traffic.

Wireshark Manual: Understanding Network Traffic

Introduction to Wireshark:

Wireshark is a powerful network protocol analyzer that allows you to capture and inspect the data traveling back and forth on your network. It supports hundreds of protocols and can display the data live or save it to analyze offline.

Capturing Traffic:

- **Selecting an Interface:** When you start Wireshark, choose the network interface you want to capture traffic from. This could be your Ethernet adapter or Wi-Fi network card.
- **Start Capturing:** Click on the interface, and Wireshark will start capturing packets. You can set filters to capture specific types of traffic (e.g., only HTTP, traffic to/from a specific IP).

Analyzing Traffic:

Wireshark provides detailed information about each captured packet. Here's what you can understand from the traffic:

- **Source and Destination Addresses:** Identify the source and destination IP addresses and ports involved in the communication.
- **Protocol Analysis:** Wireshark decodes packets and shows the protocol used (e.g., HTTP, DNS, TCP, UDP).
- **Packet Details:** Clicking on a packet reveals its raw data, allowing you to inspect the actual content of the packet.

Examples:

1. Analyzing HTTP Traffic:

- Filter by HTTP: **http**
- Wireshark will display all HTTP requests and responses.

- You can see URLs, request methods (GET, POST), and response status codes.

2. Detecting DNS Queries:

- Filter by DNS: **dns**
- Analyze DNS requests and responses.
- Identify domains being accessed and corresponding IP addresses.

3. Monitoring TCP Conversations:

- Filter by TCP: **tcp**
- Wireshark shows TCP connections.
- Analyze sequence numbers, ACK flags, and window sizes for each packet.

4. Identifying Suspicious Traffic:

- Look for unusual patterns, such as multiple connection attempts to a single IP address.
- Check for unexpected protocols or traffic from unknown sources.

5. Decrypting HTTPS Traffic (if you have access to encryption keys):

- Use the "Follow SSL Stream" feature to view decrypted HTTPS traffic if you possess the necessary encryption keys.

Metadata in Wireshark:

Wireshark captures metadata associated with each packet, providing valuable information for analysis:

- **Timestamps:** Wireshark records the time each packet is captured, allowing you to analyze the timing of network events.
- **Packet Length:** Displays the size of each packet, which can indicate the volume of data being transmitted.
- **Protocol Information:** Wireshark identifies the protocol type, version, and other specific details about each packet, aiding in protocol analysis.
- **Capture Interface Details:** Information about the network interface used for capturing, including IP and MAC addresses.

- **Expert Info:** Wireshark's built-in expert system highlights potential issues, such as malformed packets or suspicious behavior.

Learning from Network Traffic:

- **Identify Communication Patterns:**
  - Analyze patterns in communication, such as regular data exchanges between specific IP addresses and ports.
  - Detect abnormal spikes in traffic volume, which might indicate network congestion or malicious activities.
- **Diagnose Performance Issues:**
  - Identify slow response times by analyzing delays between request and response packets.
  - Monitor packet loss and retransmissions, which can indicate network congestion or unreliable connections.
- **Security Analysis:**
  - Detect unauthorized access attempts by examining failed login attempts or unusual connection patterns.
  - Identify potential security threats, such as port scans, DDoS attacks, or malware communications.
- **Troubleshoot Connectivity Problems:**
  - Investigate connection failures by examining packets with reset flags or ICMP error messages.
  - Identify DNS resolution issues by analyzing DNS request/response packets.

Wireshark Filtering Techniques:

Wireshark provides powerful filtering options to focus on specific packets of interest:

- **Display Filters:** These filters allow you to display specific types of packets. For example:
  - **ip.addr == 192.168.1.1**: Displays packets with the IP address 192.168.1.1.
  - **http**: Displays HTTP packets only.
- **Capture Filters:** Filters applied before capturing packets, allowing you to capture specific traffic only. For example:
  - **host 192.168.1.1**: Captures packets to and from IP address 192.168.1.1.
  - **port 80**: Captures packets on port 80 (HTTP).

- **Logical Operators:** Combine filters for more complex queries:
    - **ip.src == 192.168.1.1 && http**: Displays HTTP packets originating from IP address 192.168.1.1.
- **Save and Export Filtered Data:**
    - After applying filters, you can save the filtered packets to a new file for in-depth analysis or to share with others.

1. Unusual Traffic Patterns:

- **Unexplained Traffic Spikes:** Sudden and unexplained increases in network traffic might indicate unauthorized activities.
- **Consistent Off-Hours Traffic:** If there's unexpected network activity during non-business hours, it could be a sign of intrusion.

2. Analyze Protocols and Ports:

- **Unusual Protocols:** Look for unfamiliar or uncommon protocols in use. For instance, if you're seeing unusual protocols like SSH, FTP, or Telnet, investigate further.
- **Unusual Ports:** Traffic on unusual ports might indicate unauthorized services running on your network.

3. Inspect DNS Traffic:

- **DNS Requests:** Unusual DNS requests, especially to suspicious domains, can indicate malware trying to communicate with a command and control server.

4. Look for ARP Spoofing:

- **Duplicate IP Addresses:** Wireshark can help you identify ARP spoofing attacks by detecting duplicate IP addresses.

5. Analyze TLS/SSL Certificates:

- **Certificate Mismatch:** If you're monitoring encrypted traffic (like HTTPS), check for certificate mismatches or self-signed certificates, which could indicate a man-in-the-middle attack.

6. Analyze Packet Timing:

- **Packet Delays:** Check for unexpected delays in packet transmission. Delays might indicate interception attempts or network congestion caused by an intrusion.

7. Analyze Outbound Traffic:

- **Unusual Outbound Traffic:** Investigate outbound traffic, especially large file transfers or connections to unfamiliar IP addresses.

8. Check for Unauthorized Devices:

- **Unknown MAC Addresses:** Identify devices on your network by their MAC addresses. If you see unfamiliar MAC addresses, they might be unauthorized devices.

9. Use Intrusion Detection Systems (IDS):

- **Signature-Based Detection:** Utilize IDS tools that use signature-based detection to identify known patterns of attacks.
- **Behavioral Analysis:** IDS solutions can also detect abnormal network behavior, flagging potential threats.

10. Monitor Firewall Logs:

- **Firewall Logs:** Analyze your firewall logs in conjunction with Wireshark data. Look for denied or unusual connections.

11. Regularly Monitor Network Traffic:

- **Regular Audits:** Regularly monitor your network traffic, ideally in real-time, to quickly detect any suspicious activities.

1. Unusual Traffic Patterns:

Example: You notice a sudden spike in traffic on your network.

Wireshark Use: Wireshark can display the rate of incoming packets in real-time. If you observe an unexpected spike, select the corresponding packets, analyze their content, and investigate the source and destination IPs to identify the cause.

2. Analyze Protocols and Ports:

Example: You observe Telnet traffic on a port that's usually closed.

Wireshark Use: Use a display filter like **telnet** to specifically view Telnet traffic. Analyze the packets to see which devices are communicating and inspect the content to understand what's being transmitted.

3. Inspect DNS Traffic:

Example: You notice a device continuously querying a suspicious domain.

Wireshark Use: Apply a filter like **dns.qry.name == example.com** to focus on DNS queries related to the suspicious domain. Examine the DNS response packets to determine the IP addresses associated with the domain.

4. Look for ARP Spoofing:

Example: Your network experiences IP conflicts or devices suddenly lose network connectivity.

Wireshark Use: Analyze ARP packets by applying a filter like **arp**. Look for duplicate IP addresses associated with different MAC addresses, indicating potential ARP spoofing attacks.

5. Analyze TLS/SSL Certificates:

Example: Users receive SSL certificate warnings when accessing a website.

Wireshark Use: Filter encrypted traffic (e.g., **tls** or **ssl**). Inspect SSL/TLS handshake packets to check for certificate mismatches or anomalies in the certificate chain.

6. Analyze Packet Timing:

Example: Users report slow internet connectivity during specific hours.

Wireshark Use: Check the timestamps of packets sent during the reported times. Long delays between request and response packets could indicate network congestion or interception attempts.

7. Analyze Outbound Traffic:

Example: Suspicious files are found on an internal server.

Wireshark Use: Filter traffic originating from the server (e.g., **ip.src == server_IP**). Analyze outgoing packets for unusual destinations or large file transfers. Look for patterns, such as repetitive connections to external servers.

8. Check for Unauthorized Devices:

Example: Unknown devices are connected to your Wi-Fi network.

Wireshark Use: Monitor ARP packets (**arp** filter) to identify MAC addresses associated with IP addresses. Compare the identified devices with your authorized devices to spot unauthorized ones.

9. Use Intrusion Detection Systems (IDS):

Example: An IDS alerts you to a potential intrusion attempt.

Wireshark Use: Correlate IDS alerts with Wireshark data. Use Wireshark to analyze packets around the time of the alert, helping you understand the nature of the attempted intrusion.

10. Monitor Firewall Logs:

Example: A firewall log shows denied connection attempts.

Wireshark Use: Filter packets related to the denied connections (e.g., **ip.dst == denied_IP**). Analyze these packets to understand the nature of the connection attempts, such as the originating IP, port numbers, and protocols used.

11. Regularly Monitor Network Traffic:

Example: Periodic network audits reveal unexpected activities.

Wireshark Use: Set up scheduled captures and periodically review the captured data. By comparing different captures over time, you can identify trends and anomalies in your network traffic.
Remember, interpreting Wireshark data requires expertise. If you're not confident

in your analysis, consult a network security professional to ensure accurate interpretation and appropriate action.