Wireshark

I. What did you do?

a. Network and Protocol Identification and Packet Capturing:

- I used Wireshark within Coursera, launched the application and selected the appropriate network interface (e.g., Ethernet) to capture traffic.
- I started capturing packets by clicking on the interface. Wireshark immediately began displaying packets in real-time or after my activity in the browser or PowerShell.
- I identified networks and protocols and applied display (e.g., dns, http) and capture (e.g. port 1812 for capturing RADIUS traffic) filters to focus on specific protocols and/or types of traffic.
- I also opened stored capture files to analyze previously captured network traffic.
- I used Wireshark's built-in statistics tools (Statistics > Conversations) to get an overview of the traffic distribution among different protocols.
- I set capture filters (e.g., tcp port 80, host 192.168.1.1) to limit the capture to relevant packets, reducing noise and focusing on specific traffic. I allowed Wireshark to run for a set period, capturing data continuously, then stopped and closed it to switch to a new test.

b. How the Software Works:

- Wireshark captures packets by putting the network interface into promiscuous mode, allowing it to capture all packets on the network segment. It decodes the raw binary data of packets into human-readable form, detailing various protocol layers (e.g., Ethernet, IP, TCP/UDP, Application layer protocols).

- It provides powerful filtering options to include or exclude specific traffic types, IP addresses, ports, protocols, etc.
- It offers various tools for visualizing network traffic, including statistical summaries.

c. Output is Analyzed by:

- Using the “Follow TCP Stream” feature to reconstruct the entire conversation for a specific TCP connection.
- Analyzing the protocol hierarchy to understand the distribution of different protocols in the captured data.
- Inspecting the individual packets, drilling down into the packet details pane to examine headers and payloads of specific protocols.
- Checking Wireshark’s “Expert Information” for warnings, errors, and potential issues highlighted by the software.
- Using graphs to visualize traffic trends over time, identifying spikes and unusual patterns.
- For instance, for HTTP traffic, GET and POST requests can be examined to see details such as URLs, headers, and form data. For RADIUS traffic, authentication requests and responses can be analyzed. For Telnet and SSH, the plain text and encrypted data can be compared.

II. What are the results?

a. Networks and Protocols Identified:

- In the captured packets, I identified various protocols, including TCP, HTTPS, HTTP, DNS, RADIUS, Telnet, and SSH. The HTTP traffic included GET and POST requests, showing web page accesses and form submissions. DNS traffic showed domain name queries and responses. RADIUS packets displayed authentication requests and responses. Telnet packets revealed plain text communication, while SSH packets were encrypted.

- I identified the following protocols and their ports:
    - Non-encrypted:
        - RADIUS (port 1812)
        - HTTP (port 80)
        - DNS (port 53)
        - Telnet (port 23)
    - Encrypted:
        - SSH (port 22)
        - HTTPS (port 443, over TLS)

b. Do Any Packets Reflect a Cyber-Attack?

- I did not notice any packets that reflect a cyber-attack in the captured data. However, to identify potential cyber-attacks, I would look for unusual patterns such as repeated failed login attempts, unexpected data transfers, or known attack signatures. Anomalies in the size of packets, frequency, or timing could also indicate malicious activity.

c. How Can You Tell?

- Indicators of a cyber-attack might include:
    - Consistent attempts to connect to closed ports can indicate scanning.
    - Repeated failed authentication attempts (e.g. brute force attacks).
    - Large volumes of data being transferred unexpectedly.
    - Packets that do not conform to protocol standards might be crafted for malicious purposes.
    - Abnormal traffic patterns, like high traffic volume from unknown sources or devices, can be signs of attacks.

d. Analyze the Attack Surface:

- The attack surface includes all exposed points where an attacker can interact with the network. This includes:
  - Every device on the network presents an entry point for attackers. IoT devices, in particular, are often less secure.
  - Open Ports & Services running on open ports can be exploited if they have vulnerabilities (e.g. HTTP, DNS, RADIUS, Telnet).
  - Unencrypted Traffic such as HTTP traffic can be intercepted and analyzed, making it susceptible to man-in-the-middle attacks.
  - Authentication mechanisms such as HTTP basic authentication and RADIUS.

By capturing and analyzing traffic, I identified potential vulnerabilities such as unencrypted credentials in HTTP Basic Authentication and Telnet sessions. These areas require attention to mitigate risks, such as using encryption (e.g., HTTPS, SSH) and monitoring for unusual activity.

III. What did you learn?

a. Takeaways from Using Wireshark:

- Wireshark provides comprehensive visibility into network traffic, providing detailed information about the protocols and data being transmitted.
- It offers detailed insights into each packet, essential for deep analysis.
- Effective for identifying potential security issues, such as unusual traffic patterns or signs of scanning.
- The non-encrypted protocols and their ports are:
  - RADIUS (port 1812)
  - HTTP (port 80)
  - DNS (port 53)
  - Telnet (port 23, over TLS)
- The encrypted protocols and their ports are:
  - - SSH (port 22)
  - - HTTPS (port 443, over TLS).

- Additionally, the secret key is logged into the SSL log file to help with decrypting.

b. Learnings about Digital Networks, Packets, and Attack Surface:

- I learned that digital networks include various protocols that facilitate communication between devices. Each packet carries specific information that can reveal much about the network's operations and potential security issues. Understanding the attack surface is crucial for identifying and mitigating risks. In the future, I can use Wireshark to regularly monitor network traffic, identify potential security threats, and ensure compliance with security policies.

c. Future Use:

- Use Wireshark regularly for continuous network monitoring.
- Employ Wireshark for detailed analysis during incident responses to identify and mitigate attacks.
- Use Wireshark during security audits to ensure network integrity and compliance.

d. Value to the Organization:

- Wireshark can be invaluable for an organization by providing insights into network traffic, helping to detect and respond to security incidents. It can inform security planning by identifying vulnerabilities and ensuring that security measures are effective. Regular traffic analysis can aid in risk assessment, ensuring that the organization is aware of potential threats and can take proactive measures to protect its assets.