# **Network Security Monitoring and Analysis with Wireshark**

## **1. Introduction to Wireshark**

Wireshark is a powerful, open-source network protocol analyzer. It allows you to capture and interactively browse the traffic running on a computer network.

### **1.1 Installation**

* Download Wireshark from the official website ([www.wireshark.org](http://www.wireshark.org))
* Install on your chosen platform (Windows, macOS, or Linux)
* Ensure you have the necessary permissions to capture network traffic

## **2. Basic Capture and Analysis**

Capturing all network traffic, rather than just traffic to and from your device, allows for a more complete view of network activity. This is crucial for thorough network security monitoring and troubleshooting.

**Methods for Capturing All Network Traffic**

### **Virtual Machine Bridged Mode**

If your network devices are virtual machines, you can capture all traffic by placing Wireshark on the host machine.

Steps:

1. Set all VMs to bridged network mode.
2. Run Wireshark on the host machine.
3. Capture on the virtual bridge interface.
4. Start capturing.

Pros:

* Great for lab environments.
* Doesn't require additional physical hardware.

Cons:

* Only works for virtual environments.
* May miss some types of traffic depending on hypervisor configuration.

### **Wireless Monitoring Mode**

For wireless networks, you can use a wireless adapter that supports monitor mode.

Steps:

1. Ensure you have a wireless adapter that supports monitor mode.
2. Put the adapter into monitor mode (command varies by OS).
3. In Wireshark, select the monitor mode interface.
4. Start capturing.

Pros:

* Can capture all wireless traffic in range.
* Useful for analyzing wireless-specific issues.

Cons:

* Only captures wireless traffic.
* May require special hardware or drivers.

### **2.1 Starting a Capture**

1. Open Wireshark
2. Select the network interface you want to monitor
3. Click the blue shark fin icon to start capturing

### **2.2 Understanding the Interface**

* Packet List Pane: Shows a summary of each packet captured
* Packet Details Pane: Provides a detailed view of the selected packet
* Packet Bytes Pane: Displays the raw data of the packet

### **2.3 Basic Filtering**

* Use the display filter bar to focus on specific traffic
* Example filters:
  + http: Show only HTTP traffic
  + ip.addr == 192.168.1.100: Show traffic to/from a specific IP
  + tcp.port == 80: Show traffic on port 80

## **3. Security Analysis Techniques**

### **3.1 Identifying Suspicious Traffic**

1. Look for unusual port numbers or protocols
2. Check for large amounts of traffic to/from a single IP
3. Identify failed connection attempts

Example filter: tcp.flags.syn == 1 and tcp.flags.ack == 0

### **3.2 Detecting Port Scans**

1. Look for multiple SYN packets to different ports from the same source
2. Use the Conversations feature (Statistics > Conversations)

Example filter: tcp.flags.syn == 1 and tcp.flags.ack == 0 and ip.dst == YOUR\_IP

### **3.3 Analyzing HTTP Traffic**

1. Filter for HTTP traffic: http
2. Look for suspicious user agents or unusual requests
3. Check for sensitive data in clear text

Example filter: http.request.method == "POST"

### **3.4 Investigating DNS Traffic**

1. Filter for DNS: dns
2. Look for unusual domain names or high volumes of requests
3. Check for DNS tunnelling attempts

Example filter: dns.qry.name contains "suspicious"

## **4. Advanced Analysis Techniques**

### **4.1 Using the "Follow TCP Stream" Feature**

1. Right-click on a packet > Follow > TCP Stream
2. Analyze the entire conversation between two hosts

### **4.2 Detecting Malware Communication**

1. Look for periodic beaconing to unusual IP addresses
2. Check for encoded or encrypted data in payload

Example filter: ip.addr == SUSPICIOUS\_IP and data

### **4.3 Identifying Data Exfiltration**

1. Look for large outbound transfers, especially using protocols like FTP or HTTP POST
2. Check for unusual destinations or times of transfer

Example filter: ftp-data or http.request.method == "POST"

### **4.4 SSL/TLS Analysis**

1. Filter for SSL/TLS traffic: ssl
2. Look for weak cipher suites or outdated protocol versions
3. If you have the private keys, you can decrypt the traffic

Example filter: ssl.handshake.ciphersuite == 0x0004 (for weak RC4 cipher)

## **5. Creating Custom Filters and Coloring Rules**

### **5.1 Custom Filters**

1. View > Internals > Supported Protocols
2. Create complex filters using logical operators (and, or, not)

Example: (ip.src == 192.168.1.100 or ip.dst == 192.168.1.100) and tcp.port == 443

### **5.2 Coloring Rules**

1. View > Coloring Rules
2. Create new rules to highlight specific types of traffic

Example: Add a red background for failed SQL login attempts

## **6. Generating Reports**

### **6.1 Basic Statistics**

1. Use the Statistics menu for quick overviews
2. Endpoints, Conversations, and Protocol Hierarchy are particularly useful

### **6.2 IO Graphs**

1. Statistics > IO Graph
2. Visualize network usage over time

### **6.3 Expert Information**

1. Analyze > Expert Information
2. Quick way to identify potential issues in the capture

## 

## **7. Best Practices for Network Security Monitoring**

1. Regularly update Wireshark to the latest version
2. Use targeted captures to reduce noise
3. Combine Wireshark with other security tools for comprehensive monitoring
4. Be aware of privacy and legal considerations when capturing traffic
5. Use time display formats to correlate events across multiple logs
6. Save important captures for future reference or incident response

## **8. Hands-on Scenarios**

### **Scenario 1: Detecting a Port Scan**

1. Start a capture
2. Use Nmap from another machine to scan your target
3. Analyze the traffic in Wireshark
4. Look for patterns of SYN packets to multiple ports

### **Scenario 2: Identifying a Compromised Machine**

1. Look for unusual outbound connections
2. Check for periodic beaconing to suspicious IP addresses
3. Analyze the payload of suspicious packets

### **Scenario 3: Investigating a Potential Data Breach**

1. Look for large data transfers to external IP addresses
2. Analyze HTTP POST requests for sensitive data
3. Check for unusual activity during non-business hours

Network security monitoring is an ongoing process. Regular practice and staying updated with the latest threats and analysis techniques are key to effective monitoring.

This documentation provides a comprehensive guide to using Wireshark for network security monitoring and analysis. It covers basic usage, advanced techniques, and practical scenarios to help you develop your skills in network traffic analysis.