Lab 4: Network Traffic Analysis Using Scapy

Course: FSCT 8561 – Security Applications
Instructor: Dr. Maryam R. Aliabadi
Lab Duration: 3 Hours

Overview

This lab introduces network traffic analysis using Python and the Scapy library. Instead of capturing live network traffic, you will analyze a pre-captured PCAP file containing botnet activity, including a simulated Distributed Denial-of-Service (DDoS) attack. You will inspect packet headers and payloads, identify protocols, detect abnormal traffic patterns, and practice intrusion detection techniques. This lab emphasizes practical network forensics and anomaly detection.

Learning Objectives

- Understand packet-based network communication
- Capture live traffic using Scapy
- Analyze Ethernet, IP, and TCP/UDP packet headers
- Inspect payloads for potentially sensitive information
- Evaluate network security and server exposure
- Apply Python scripting for IDS-style traffic analysis

Pre-requisite

- Completed Labs 1–3
- Understanding of TCP/IP protocols

Required Reading & Tutorials

- Mastering Python for Networking and Security – Chapter 6
  - https://learning.oreilly.com/library/view/mastering-python-for/9781839217166/
  - Source code on: https://github.com/PacktPublishing/Mastering-Python-for-Networking-and-Security-Second-Edition
- Scapy Documentation: https://scapy.readthedocs.io/
- Scapy Tutorial: https://youtu.be/4ghZXpkqt8M
- tcpdump Reference: https://www.tcpdump.org/manpages/tcpdump.1.html

Lab Scenario

You are tasked with monitoring network traffic from a test server. Your goal is to capture packets using Scapy, inspect protocol details, identify unusual traffic patterns, and detect any unencrypted sensitive information. This lab demonstrates how network traffic can expose potential security risks.

Environment Setup

- Install Python and Scapy.
- Ensure access to the network interface (requires root/administrator privileges).
- Prepare a test server or virtual machine for authorized packet capture.

Part 1 – Packet Capture

In this part, you will capture live network packets from the test server using Scapy. This allows observation of the actual traffic flowing across the network.

1. **Start live packet capture using Scapy**
2. **Apply filters**
    a. TCP only: sniff(filter="tcp", count=50, prn=packet_callback)
    b. HTTP (port 80): sniff(filter="tcp port 80", count=50, prn=packet_callback)
    c. DNS (port 53): sniff(filter="udp port 53", count=50, prn=packet_callback)
3. **Capture at least 50 packets for inspection and later analysis**
4. **Expected Observations:** Packet summaries, IP addresses, ports, and protocols.

Part 2 – Packet Inspection

Inspect the captured packets to understand network headers, protocol types, and traffic patterns.

1. **Inspect individual packet details:** Examine Ethernet, IP, and TCP/UDP headers.
1. **Extract source and destination IPs, ports, and protocol types.**
2. **Identify TCP flags, packet length, and timing.**
2. **Expected Observations:** Correct identification of headers, protocol distribution, and detection of unusual patterns.

Part 3 – Sensitive Data Detection

Examine packet payloads to detect unencrypted sensitive information such as credentials or personal data.

1. **Inspect payloads for unencrypted credentials or sensitive information.**

2. **Check for metadata exposure:** HTTP headers, cookies, user-agent, or server version info.
3. **Expected Observations:** Identification of plain text sensitive data and unusual packet contents or patterns.

Part 4 – Traffic Analysis Script

develop a Python script to automate packet capture and analysis:

1. Captures packets from a live interface.
2. Logs source/destination IP addresses, ports, and protocols.
3. Counts packets per protocol type.
4. Outputs findings in a readable format.
5. Expected Observations: Accurate logging, correct protocol counts, and clear output for analysis.

Part 5 – Traffic Analysis & IDS Scripting (PCAP-Based)

In this section, you will analyze a pre-captured network traffic file (.pcap) that contains botnet activity, including a simulated Distributed Denial-of-Service (DDoS) attack. Instead of capturing live traffic, you will process recorded packets to detect abnormal traffic patterns using Python and Scapy.

Detect potential flooding attacks using the provided PCAP file: ***botnet-capture-20110812-rbot.pcap***

1. Load the PCAP file using Scapy
2. Inspect each packet:
    a. TCP/UDP protocols
    b. Source and destination IPs
3. Implement threshold-based anomaly detection:
    a. Track number of packets per unique source IP
    b. Use a 5-second sliding window based on timestamps
    c. Detection rule: **more than 20 packets in 5 seconds → alert**
4. Ensure alerts print **once per offending IP**.
5. Output a summary:
    a. Total TCP Packets: X
    b. Total UDP Packets: Y
    c. Suspicious IPs Detected: N

Expected Observations: One or more IPs exceeding threshold, Repeated packets from same source, Short inter-arrival times and Normal background traffic remains below detection thresholds.

Part 6 – Reflection Questions

1. Why is packet inspection important for network security?
2. What types of attacks rely on unencrypted network traffic?
3. What are the limitations of passive packet sniffing?
4. How does encryption help prevent information leakage?

Part 7 – Security Analysis

Write 300–400 words analyzing the security implications of your observations. Include:

- Risks observed from captured traffic
- Potential threats and vulnerabilities
- Suggestions for mitigating exposure
- Limitations of packet capture in real-world environments

Deliverables

- `Traffic_sniffer.py` – live capture & analysis
- `Anomaly_Detector.py` – IDS/PCAP analysis script
- A recording or screenshots demonstrate packet captures and analysis.
- Security analysis and reflection report
- Submit one PDF file only. This PDF is the Security analysis and Reflection Report and must include cloud links to all required technical artifacts (code base, recordings.).
- Filename format:

  Lab4-FirstName-LastName-StudentNumber.pdf

Good Luck!

Lab 4 Grading Rubric (Total: 100 points)

Component	**Points**	**Full Credit**		**Partial Credit**	**Minimal/No Credit**
Packet Capture Implementation	15	Captures at least 50 packets with correct filtering		Captures fewer packets or minor filter errors	Packet capture missing or incorrect
Packet Inspection Accuracy	15	Correctly identifies headers, IPs, ports, and protocols		Minor errors or incomplete inspection	Inspection missing or incorrect
Traffic Analysis Script	15	Logs traffic details, counts per protocol, readable output		Script partially works or minor output issues	Script missing or does not run
Sensitive Data Identification	15	Correctly identifies potential sensitive/unencrypted data		Partial identification or minor errors	Missing or incorrect
PCAP-Based Traffic Analysis script	25	Detects flooding behavior, prints alerts, summary output		Partially works, minor detection/output issues	Missing or does not run
Reflection & Security Discussion	15	300–400 words, clearly analyzes findings and security implications		Partial coverage or less detailed	Missing, generic, or incomplete