Network Intrusion Detection and Prevention System (NIDPS)
Lab Assignment - 3

System and Network Security (CS5.470)

Hard Deadline: 7 April 2025 (23:59 PM)
Total Marks: 100

Introduction to IDS

- An **intrusion** is an unauthorized attempt to access, manipulate, or harm a system.

- Examples of intrusions:
  - **Unauthorized login**: A hacker using leaked passwords from a data breach to access your online banking account.
  - **Malware injection**: Clicking on a fake email link (phishing) that downloads ransomware, locking all files until a ransom is paid.
  - **Spyware**: A rogue app on your smartphone tracking your keystrokes and stealing your personal information.
  - **Denial-of-Service (DoS) attack**: Attackers flooding an e-commerce website with fake traffic, making it inaccessible to real customers.

- An **IDS** monitors and analyzes network/system activities to identify potential security threats.

Signature-based IDS

- Detects attacks by comparing activities to a database of known attack patterns.
- Similar to an **antivirus**, which scans files for known malware signatures.
- Highly effective against **previously identified threats**.

- **Limitations:**
  - Cannot detect new or modified attacks (zero-day threats).
  - Requires constant updates to stay effective.

- **Example:**
  - A system detects a brute-force attack because it matches a known pattern of failed login attempts within a short time.

Anomaly-based IDS

- Monitors normal system behavior and flags deviations as potential threats.
- Can detect **new, unknown, or evolving attacks**.
- Uses machine learning, statistical models, or other approaches for behavior analysis.

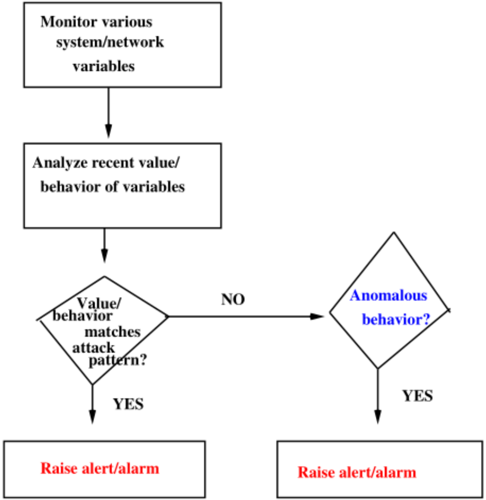
- **Limitations:**
  - Higher chance of **false positives** (flagging normal activities as threats).
  - Requires proper **training data** to define normal behavior accurately.

- **Example:**
  - In a company, a new unknown device with an unrecognized MAC address tries to access restricted files.

Intrusion Prevention vs Detection

- **Intrusion Prevention (Stopping Attacks Before They Happen)**:
  - Setting up strong password policies.
  - Firewalls blocking malicious traffic.
  - Multi-factor authentication.
  - Access controls to prevent unauthorized access.
- **Intrusion Detection (Catching Attacks When They Happen)**:
  - Like a alarm/notification that detects an intruder breaking into your system.
  - Antivirus detecting malware or IDS monitoring network activity for suspicious behavior.
  - Monitoring network traffic and system logs to detect threats.

Prevention versus Detection



Monitor various system/network variables
Analyze recent value/ behavior of variables
Value/ behavior matches attack pattern?
NO
Anomalous behavior?
YES
YES
Raise alert/alarm
Raise alert/alarm

Figure: Tasks performed by an IDS.

NIDPS

Assignment Tasks:

Develop a Network Intrusion Detection and Prevention System (NIDPS)

- Detect malicious activities using signature and anomaly-based methods.
- Log detected attacks.
- Dynamically block threats.
- Provide a CLI-based management interface.

Sample Package Installation (Python):

Install scapy, python-nmap, numpy sklearn

1. Network Traffic Monitoring [10]

Capturing Network Packets using Scapy

- Use Packet sniffer - Import the sniff function from Scapy.
- Define a callback function to print summary of the captured packet.
- Start sniffing each network packets.
- Print packet information

  Time: timestamp, **Src**: srcip : srcport, **Dst**: dstip : dstport, **Protocol**: protocol

2. Intrusion Detection Module [30]

1. **Port Scanning Detection → Anomaly-Based Detection**

1. **Multiple Port Scanning**
   - Identifies abnormal behavior by tracking connection attempts to multiple ports within a short time window
   - Legitimate users rarely connect to multiple random ports rapidly, so deviation from normal behavior is flagged as an attack.

For example:
Identify hosts that attempt to connect to more than 6 different ports within 15 seconds.

2. Intrusion Detection Module

2. Sequential Port Scanning

- Detects a pattern of sequential port accesses from the same IP address.
- Attacker (single IP) systematically scans multiple ports in a sequential order (e.g., 80, 81, 82, 83, ...)

Action [1,2] : Logging Attacker's IP and Targeted Ports

- The system should flag and log such suspicious IP addresses along with the ports they attempt to access.
- Identify repeated scanning attempts from the same attacker.

*** Grace Marks for Further Extension*

2. Intrusion Detection Module

2. OS Fingerprinting Detection → Signature-Based Detection

- Attackers determine an operating system by analyzing its response to specially crafted TCP packets.
- Uses predefined patterns of SYN, ACK, and FIN flag combinations to classify OS behavior.
- Each OS responds uniquely based on its TCP/IP stack implementation.
- Active tools (e.g., `nmap -O target_ip`) send multiple TCP probes.
- The system detects multiple unique TCP packets from the same source.
- IDS logs: `Suspicious OS fingerprinting attempt from target_ip`.

Example:
If an IP sends 5+ different SYN, ACK, and FIN flag combinations within 20 seconds, it is flagged.

3. Intrusion Prevention and Logging [20]

Block detected threats

- Use `iptables` firewall commands.
- Dynamically block malicious IPs upon detection.
- Prevent further malicious attempts from flagged attackers.
- Provide an option to manually unblock previously blocked IPs.

Example: Blocking with iptables

- `sudo iptables -A INPUT -s 192.168.1.5 -j DROP`
- This command blocks all traffic from the attacker's IP.
- To unblock: `sudo iptables -D INPUT -s 192.168.1.5 -j DROP`

4. Alert and Logging System [20]

Maintain a log file (`ids.log`) for detected intrusions.

- Displays a summary report of detected intrusions upon request.
- Log should contain:
  - **Date, Time**: When the attack occurred.
  - **Intrusion Type**: Port scan, OS fingerprinting, etc.
  - **Attacker IP**: Source of the attack.
  - **Targeted Ports/Flags**: Specific attack details.
  - **Time Span of Attack**: Duration of suspicious activity.

Log format:

Date Time — Intrusion Type — Attacker IP — Targeted Ports — Time Span Of Attack

Sample Entry:

21-03-25 14:30:12 — Port Scanning — 192.168.1.5 — 22, 80, 443, 8080 — 12s

5. Command-Line Interface (CLI) for IDS Management [10]

- Provide an interactive interface for managing the Intrusion Detection System (IDS).
- Enable users to monitor, control, and configure IDS operations efficiently.

Key Functionalities for Display Menu:

1. Start/Stop IDS: Enable or disable the intrusion detection system.
2. View Live Traffic: Display ongoing network activity in real-time.
3. View Intrusion Logs: Check recorded attack details from the log file.
4. Display Blocked IPs: Show a list of IPs currently blocked by the system.
5. Clear Block List: Remove all blocked IPs at once.
6. Unblock an IP: Allow a specific IP to regain access.
7. *Note: If you extend any functionality, add in menu.*
8. Exit: Quit the CLI interface.

Testing & Validation

- The IDS will run in the background, listening for malicious activity.
- The attack simulation scripts will send packets to different ports on the local machine
- The IDS will detect and log suspicious activity.
- The difference in seconds between two timestamps can be used to detect frequent connections (potential attack).
- Port similarity checks can be used to unblock connections
- Validate if a given IP address is in the correct format using dot count and character checks

Testing Module Overview

- Simulates a test of the Intrusion Detection System (IDS) by creating a list of packets and analyzing them.
- Analyze the packet to extract port address and IP details
- Identify the protocol used and the flags specified
- Specific flags with TCP module can be used to simulate flooding attack
- Simulate port scan attacks using tools like `nmap` and custom scripts
- For generating test attacks of signature based anomalies, `hping3` can be used

Details for testing script implementation

Network traffic can be simulated using IP and TCP classes from scapy

Use `localhost` as source and destination IPs

The packets simulate various scenarios

1. Normal traffic (e.g., ACK and PSH flags).
2. SYN flood attack (multiple SYN packets to a port).
3. Port scan attack (multiple SYN packets from the same source port to different destination ports).

** Further Extension (Optional)

For Example:

- Identify new or unknown scanning behaviors using statistical thresholds.
- Generate a detailed summary of detected intrusions.
- Save and export intrusion logs for future analysis.
- Implement a GUI-based interface for ease of use.
- Enable automatic rule updates for improved threat detection, etc.

Submission Instructions

Submit a zipped file named `<team_ID>_lab3.zip` containing:

- Source Code files
- **Documentation / README [10]**
  - Setup instructions on how to build and run each program.
  - Explanation of implementation steps.
  - Description of input and output formats.
- Log file **(**`ids.log`**)** : Contains records of detected intrusions.