Intrusion Detection System (IDS) Report

Hagay Cohen, Imri Shai

September 29, 2024

1 Conclusions from the R&D

The research we conducted on data leak attacks provided valuable insights into the types of attacks, methods of exfiltration, and detection mechanisms. We found that data leak attacks can be detected through packet inspection, user behavior analytics, and packet header analysis. These detection methods are implemented in our Intrusion Detection System (IDS) as discribed below.

2 System Architecture

Our IDS is designed to monitor network traffic and detect suspicious activities that may indicate data leaks. The system architecture is shown in Figure 1. The IDS consists of three main components: flow classifier, detection engine, and statistical analysis module.

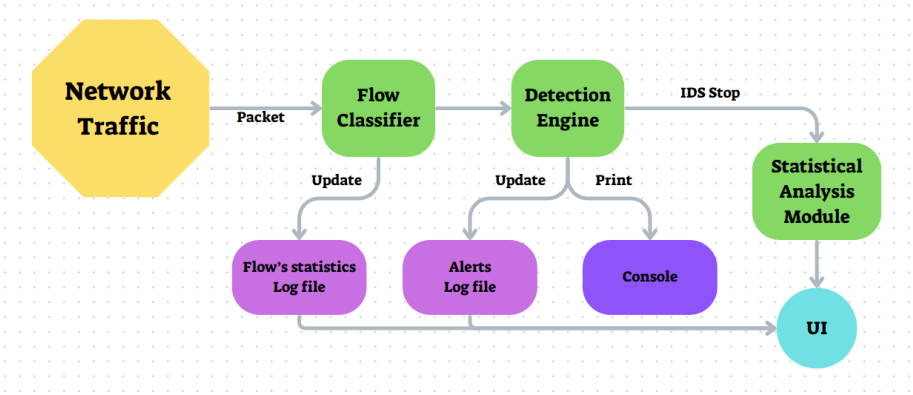


Figure 1: System Architecture

2.1 Flow Classifier

For each packet, the flow classifier extracts a 5-tuple key (source IP, destination IP, source port, destination port, protocol) and assigns it to a flow. The flow classifier then updates the flow's statistics, such as packet count and byte count, and forwards the flow to the detection engine.

2.2 Detection Engine

The detection engine analyzes the flows received from the flow classifier to identify potential data leak attacks. It uses a set of rules and traffic patterns to detect suspicious activities, such as tunneling, social engineering, and backdoors misuse. When a potential attack is detected, the detection engine generates an alert, prints it to the console, saves it to a log file.

2.3 Statistical Analysis Module

When the IDS is ordered to stop, by gracefully shutting down or an end of a simulation, the statistical analysis module saves the statistics of the flows to a file. And then, it activates the UI module to display the statistics in a user-friendly way. All the alerts, flow data, and statistics imgaes are saved to further analysis. UI example is shown in Appendix A.

3 Detections Explanations

Our IDS is capable of detecting various types of intrusions. The detection mechanisms include:

- **Data Exfiltration:** Detects data leak attacks by analyzing packet headers and inspecting packet contents, while considering protocol-specific characteristics.
- **Backdoors Misuse:** Detects backdoors misuse by monitoring network traffic and identifying usege of known backdoor ports or usege of internal protocols.
- **Tunneling:** Identifies tunneling attacks by analyzing packet headers and detecting anomalies in network traffic.
- **Suspicious User Behavior:** Analyzes user behavior and identifies patterns that deviate from normal activities, such as excessive data transfer or after-hours access.
- **Unauthorized IP Address:** Detects unauthorized IP addresses by monitoring network traffic and identifying connections from unknown or blacklisted IP addresses.

4 Docker Network Simulation

We used Docker to simulate a network environment for testing our IDS. The setup includes containers for an internal server, a database, user machines, an external server, and the IDS itself.

4.1 Network Setup

The network is defined using a Docker Compose file with the following components:

- **Internal Server:** Nginx web server.
- **Database:** MySQL database.
- **User Machines:** Simulate normal and malicious activities.
- **External Server:** Simulates external threats.
- **IDS:** Monitors and detects suspicious activities.

4.2 Traffic Generation

Scripts in user containers generate normal and malicious traffic, including web browsing, and data exfiltration.

4.3 IDS Monitoring and Detection

The IDS inspects packets, analyzes user behavior, and examines packet headers to detect data leaks. Alerts and logs are generated for suspicious activities.

4.4 Results and Analysis

The IDS's effectiveness is evaluated by reviewing alerts and analyzing traffic patterns to identify vulnerabilities.

5 Test Cases

In the "volumes" directory under the "ids_scripts" folder, there is a file named `Test.py` that contains the test cases. The test cases are designed to validate the functionality of the IDS by simulating various attack scenarios and verifying that the IDS correctly identifies and generates alerts for these attacks. The test cases include:

- DNS Tunneling
- ICMP Tunneling
- Unauthorized Port Access

- After Business Hours Traffic
- Large Data Transfer
- Suspicious Traffic
- Unauthorized File Transfer
- Unusual Traffic
- Unusual User Activity

The test cases are executed by running the `Test.py` script, which generates traffic in PCAP format. The PCAP files can be analyzed using Wireshark and used as input to the IDS for detection.

6 Future Extensions

In the future, the IDS can be extended to include more sophisticated detection mechanisms, such as:

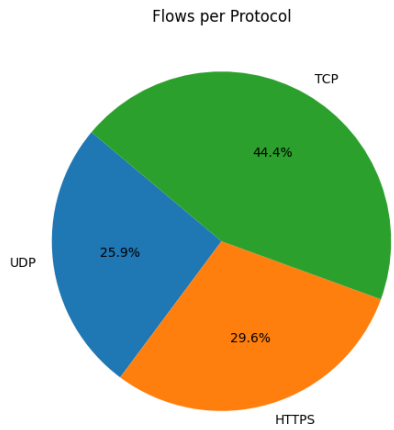
- **Machine Learning:** Implement machine learning algorithms to improve detection accuracy and reduce false positives.
- **Deep Packet Inspection:** Analyze packet contents to detect data leaks and identify malicious payloads.
- **Behavioral Analysis:** Monitor user behavior and identify patterns that deviate from normal activities.
- **Sandboxing:** Isolate suspicious files or applications in a sandbox environment to prevent data leaks.

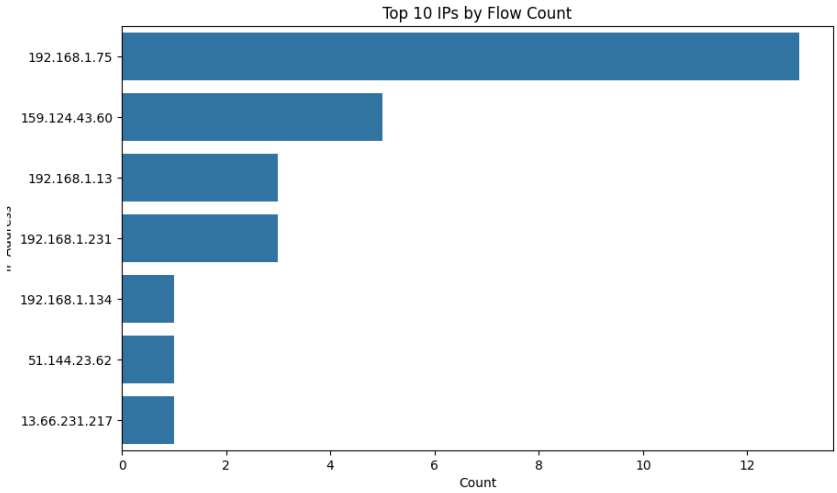
7 Conclusions and Notes

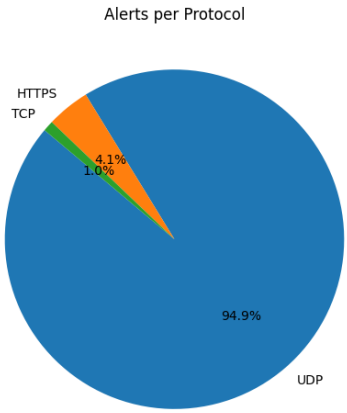
The IDS is a powerful tool for detecting data leak attacks, backdoors misuse, tunneling, and other security threats. After conducting research on data leak attacks and implementing the IDS, we have gained valuable insights into intrusion detection mechanisms and network security. The IDS can be further enhanced and diversified to address emerging threats and vulnerabilities in network environments.

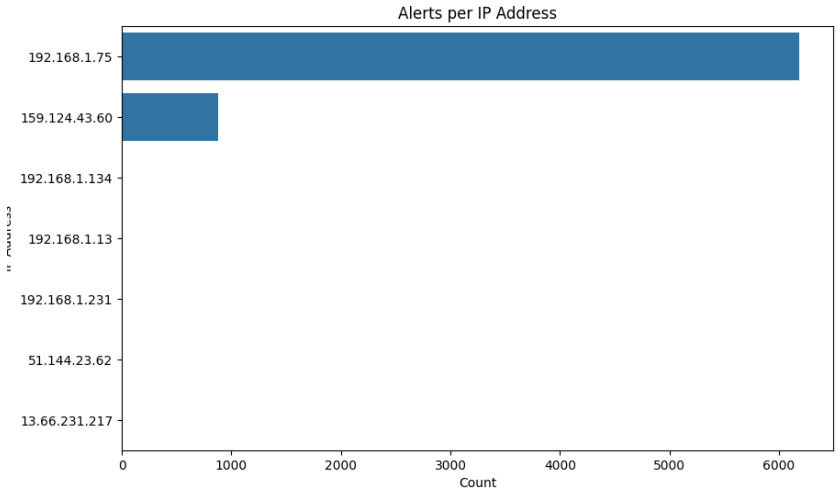
NOTE: The IDS is a proof-of-concept implementation. In a real-world scenario, the IDS would be deployed in a production environment with additional security measures and monitoring capabilities. Moreover, our IDS is designed to inspect unencrypted network traffic. Encrypted traffic, which is a common practice in modern networks, would require additional mechanisms, and some of the detection mechanisms may not be applicable.

A User Interface Example






Alerts per Protocol
HTTPS
TCP
4.1%
1.0%
94.9%
UDP


Alerts per IP Address
192.168.1.75
159.124.43.60
192.168.1.134
192.168.1.13
192.168.1.231
51.144.23.62
13.66.231.217
0
1000
2000
3000
4000
5000
6000
Count