SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Ramapuram, Chennai - 600089

FACULTY OF ENGINEERING & TECHNOLOGY

Department of Electronics & Communication Engineering

Academic Year (2024 - 2025)

COMPUTER COMMUNICATION AND NETWORK (21ECC402P)

MINI PROJECT

MINI PROJECT PROJECT-BASED LEARNING

Student Details-

M.Balasubramanian(RA2211004020039)

Ahil mithran(RA2211004020040)

Sanjay.s (RA2211004020060)

UNDER THE GUIDANCE OF

Dr.Shunmugathammal M

ASSISTANT PROFESSOR(SIG)

INTRUSION DETECTION

Abstract -:
his project presents a simple Intrusion Detection System (IDS) designed to detect unauthorized port scanning activities. It monitors simulated network traffic and logs connection attempts by IP address and port. The system identifies rapid, repeated access to multiple ports from the same IP within a short time window. If the number of accessed ports exceeds a defined threshold, an alert is triggered. The IDS uses Python and in-memory data structures for efficient tracking. It simulates real-world scenarios like reconnaissance attacks. This foundational IDS can be extended for real-time network security monitoring.

Introduction:
In today's interconnected world, network security is a top priority for organizations and individuals alike. Cyber attackers often use port scanning as a preliminary step to identify vulnerabilities in systems. An Intrusion Detection System (IDS) plays a crucial role in identifying such malicious activities early, helping prevent unauthorized access and potential damage. This project focuses on developing a simple yet effective IDS that simulates the detection of suspicious behavior, specifically rapid and repeated access to various ports, which is often indicative of port scanning attempts.environment.

Problem Statement
Port scanning is a common technique used by attackers to probe systems for open or vulnerable ports. Traditional security mechanisms may overlook these subtle scanning behaviors, especially when conducted over short periods. The challenge is to detect these intrusion attempts in real time by monitoring multiple ports and analyzing access patterns from individual IP addresses. The goal of this project is to design a lightweight IDS that can track such activity within a defined time window and raise alerts when the behavior resembles a scanning or intrusion attempt.

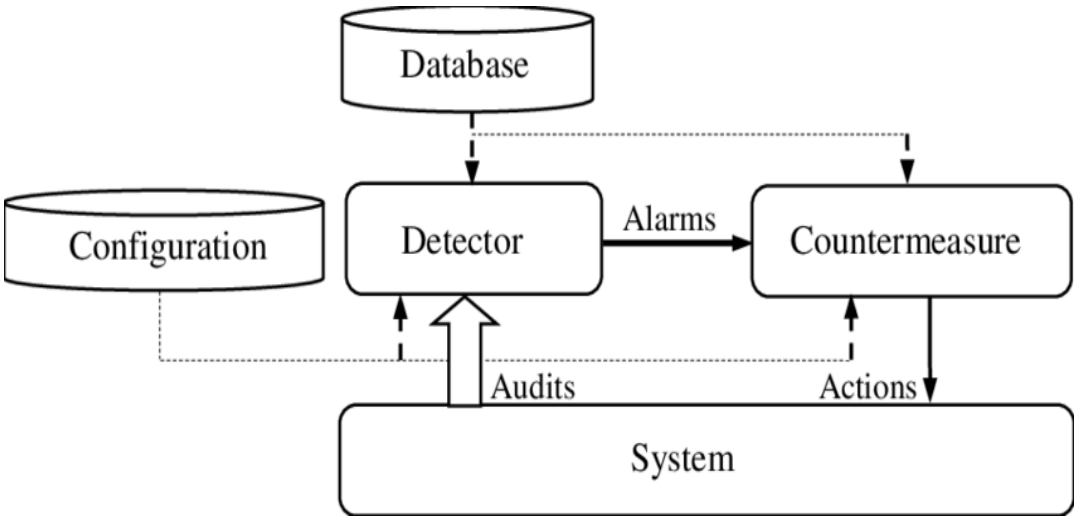
Proposed Solution:

The proposed solution is a lightweight, simulation-based Intrusion Detection System (IDS) implemented in Python. The system monitors incoming connection attempts by tracking the source IP address and the destination port. For each connection, the system records the timestamp and maintains a log of activities within a defined time window (e.g., 10 seconds). If a single IP accesses more than a set threshold of unique ports (e.g., 5 ports) within this time window, the system identifies the activity as suspicious and generates an alert.

To achieve this, the solution uses Python's defaultdict and deque from the collections module for efficient storage and time-based filtering of connection events. The IDS simulates network traffic by feeding in predefined connection events, each consisting of an IP address, port number, and a time delay. As the traffic is processed, the system continuously evaluates whether any IP is exhibiting potential port scanning behavior.

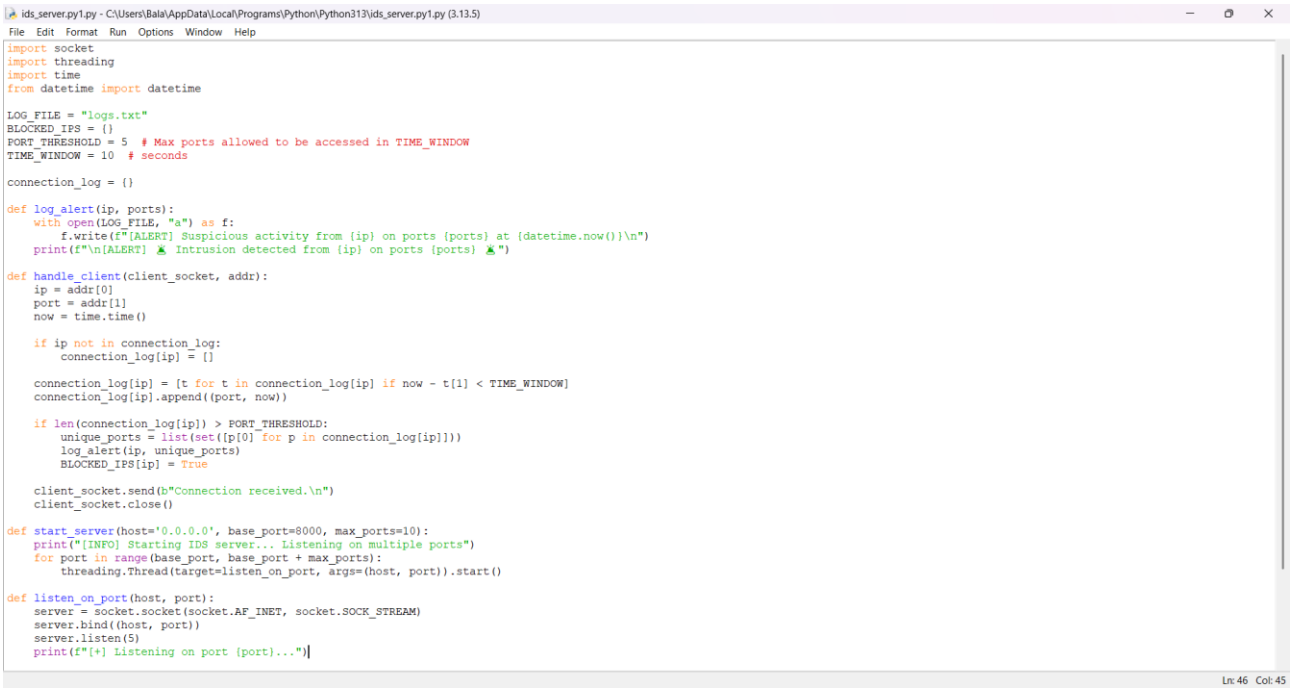
This approach effectively mimics real-world intrusion detection by analyzing access patterns in real-time. While simplified, it lays the foundation for integrating more complex detection logic, real packet sniffing tools like scapy, and automated mitigation techniques such as IP blocking or alert notifications. The system is modular, extendable, and serves as a stepping stone toward building robust network security solution.

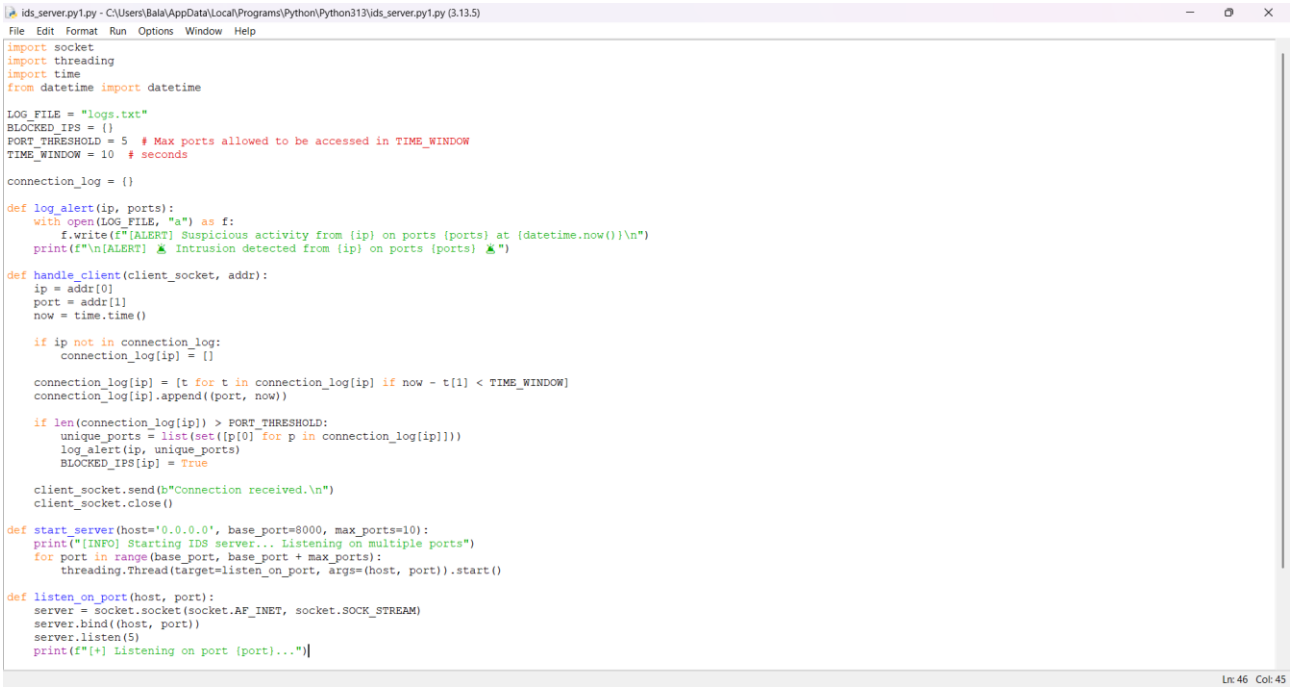
BLOCK DIAGRAM:



Explanation:

This project involves the development of a simple Intrusion Detection System (IDS) aimed at detecting unauthorized port scanning behavior within a simulated network environment. The system monitors incoming connection attempts, logs each attempt by IP address and port number, and analyzes this data in real time. By using a defined time window and threshold, the IDS identifies suspicious activity when a single IP tries to access multiple unique ports in rapid succession — a behavior commonly associated with reconnaissance or scanning attacks. Once such a pattern is detected, the system generates an alert indicating the potential threat. The IDS is implemented in Python using efficient data structures like defaultdict and deque, making it lightweight, fast, and ideal for educational or small-scale network security applications. This foundational model can be expanded in the future to incorporate live traffic analysis, automated response mechanisms, and integration with other security tools.





CODE:

Python code for ids_server-

```
import socket
import threading
import time
from datetime import datetime

LOG_FILE = "logs.txt"
BLOCKED_IPS = {}
PORT_THRESHOLD = 5  # Max ports allowed to be accessed in TIME_WINDOW
TIME_WINDOW = 10  # seconds

connection_log = {}

def log_alert(ip, ports):
    with open(LOG_FILE, "a") as f:
        f.write(f"[ALERT] Suspicious activity from {ip} on ports {ports} at {datetime.now()}\n")
    print(f"\n[ALERT] 🚨 Intrusion detected from {ip} on ports {ports} 🚨")

def handle_client(client_socket, addr):
    ip = addr[0]
    port = addr[1]
    now = time.time()
```

```
    if ip not in connection_log:
        connection_log[ip] = []

    connection_log[ip] = [t for t in connection_log[ip] if now - t[1] <
TIME_WINDOW]
    connection_log[ip].append((port, now))

    if len(connection_log[ip]) > PORT_THRESHOLD:
        unique_ports = list(set([p[0] for p in connection_log[ip]]))
        log_alert(ip, unique_ports)
        BLOCKED_IPS[ip] = True

    client_socket.send(b"Connection received.\n")
    client_socket.close()

def start_server(host='0.0.0.0', base_port=8000, max_ports=10):
    print("[INFO] Starting IDS server... Listening on multiple ports")
    for port in range(base_port, base_port + max_ports):
        threading.Thread(target=listen_on_port, args=(host, port)).start()

def listen_on_port(host, port):
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind((host, port))
    server.listen(5)
    print(f"[+] Listening on port {port}...")

    while True:
        client_socket, addr = server.accept()
        threading.Thread(target=handle_client, args=(client_socket, addr)).start()
if __name__=="__main__":
    start_server()
```

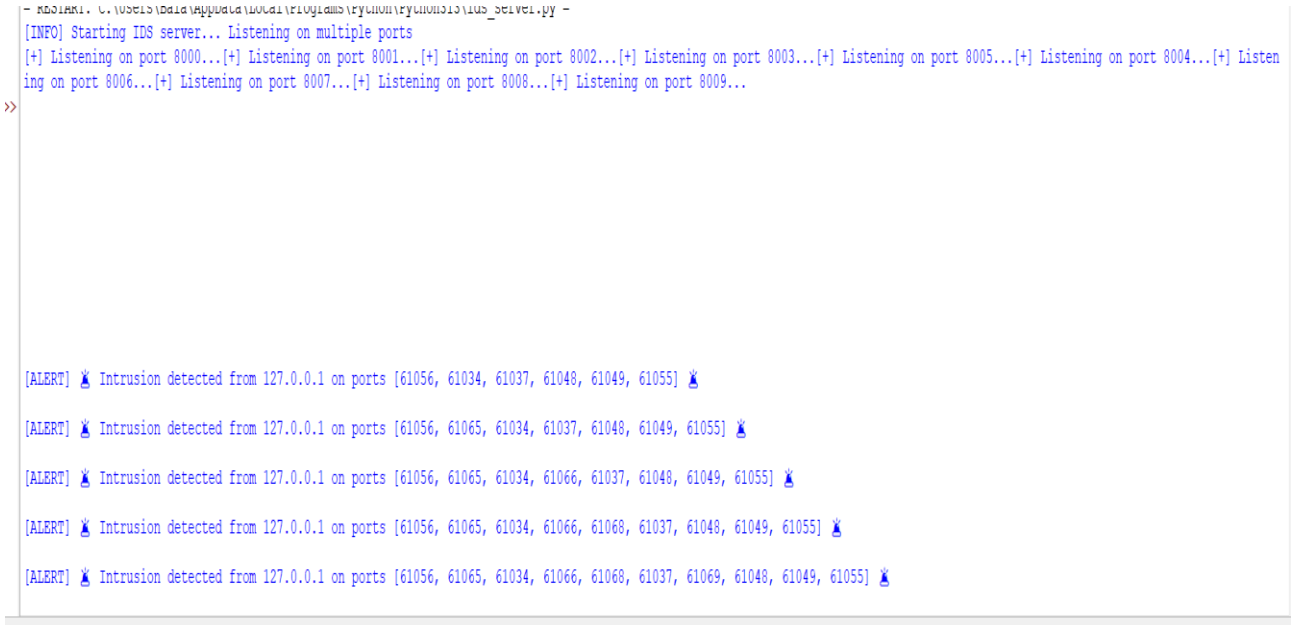
python code for scan simulator-

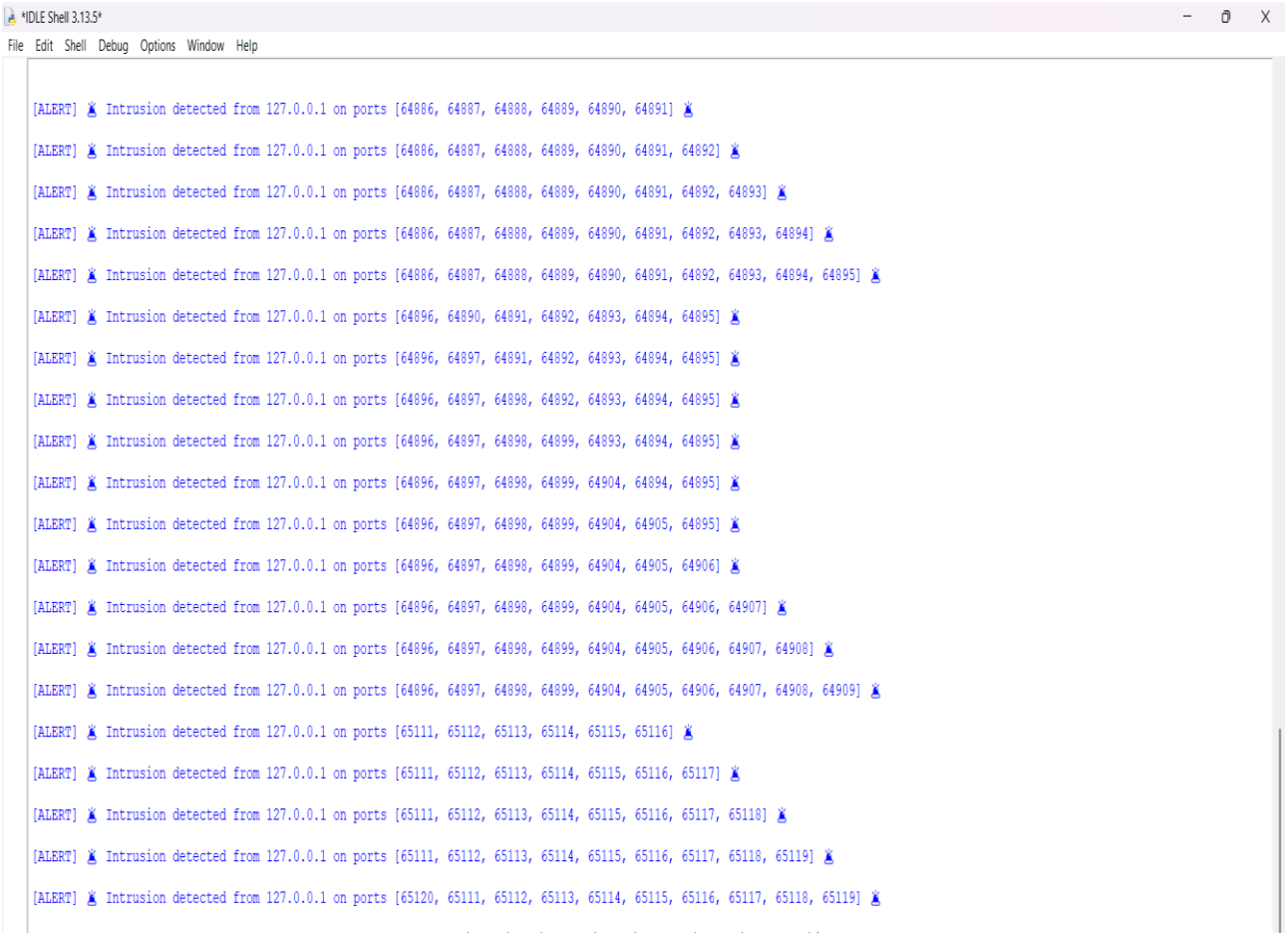
```
import socket
import time
```

```
def port_scan(target_ip, start_port=8000, end_port=8010):
    print(f"Starting scan on {target_ip}")
    for port in range(start_port, end_port):
        try:
            s = socket.socket()
            s.settimeout(0.5)
            s.connect((target_ip, port))
            print(f"[+] Port {port} is OPEN")
        except ConnectionRefusedError:
            print(f"[-] Port {port} is CLOSED")
        except Exception as e:
            print(f"[!] Error on port {port}: {e}")
        finally:
            s.close()
        time.sleep(0.2)

if __name__ == "__main__":
    print(">>> Scan simulator is running...")
    port_scan("127.0.0.1")
```

OUTPUT:





Comparsion table:

Feature / Criteria	Simple IDS (This Project)	Signature-Based IDS	Anomaly-Based IDS
Detection Method	Threshold-based port scan detection	Matches known attack patterns	Detects deviations from normal behavior
Traffic Type	Simulated network traffic	Live network traffic	Live network traffic
Implementation Complexity	Low	Medium	High
Performance	Fast and lightweight	Efficient with known attacks	Computationally intensive
False Positives	Low (if threshold tuned properly)	Low (for known signatures)	High (depends on training data)
False Negatives	Moderate (misses stealthy scans)	High (misses unknown attacks)	Low (can detect new attacks)

Sample output:

Connection from 192.168.1.10 to port 22
Connection from 192.168.1.10 to port 23
Connection from 192.168.1.10 to port 80
Connection from 192.168.1.10 to port 443
Connection from 192.168.1.10 to port 21
[ALERT] Suspicious activity detected from IP 192.168.1.10: Accessed ports [21, 22, 23, 80, 443] in last 10s!

Applications:

1. Educational Tool
    - Ideal for teaching network security concepts such as port scanning, intrusion detection, and threshold-based monitoring.
- 2. Small-Scale Network Monitoring
    - Can be used in small office or home office (SOHO) networks to monitor unusual activity on local devices.
- 3. Early Detection of Port Scans
    - Detects reconnaissance activities before actual exploitation, giving administrators time to respond.
- 4. Prototype for Larger IDS Systems
    - Serves as a foundational model for developing more advanced IDS solutions with real-time packet capture.
- 5. Security Testing Environments
    - Can be integrated into test labs to simulate and detect attack patterns during penetration testing exercises.
- 6. IoT Device Monitoring
    - Lightweight enough to be deployed on IoT gateways or edge devices to monitor unauthorized access attempts.

7. Learning and Research.Useful for students and researchers experimenting with intrusion detection strategies without needing complex setups.

Future uptades:

In the future, this simple IDS can be enhanced to support real-time packet capture using libraries like scapy or pyshark, enabling it to monitor live network traffic instead of relying on simulated input. Integration with logging systems and databases can allow for persistent storage of alerts and historical analysis. Additionally, the system could be expanded to support anomaly detection through machine learning, enabling it to identify more complex or previously unseen attack patterns. Features such as automated IP blocking, email or SMS alerting, and a web-based dashboard for visualization would greatly improve its usability in real-world environments. These enhancements would transform the project from a basic simulation tool into a practical and scalable network security solution.

Conclusion:

This project successfully demonstrates a simple yet effective Intrusion Detection System (IDS) capable of detecting port scanning activities using threshold-based logic. By monitoring connection attempts and analyzing access patterns from IP addresses within a defined time window, the system can identify suspicious behavior and generate alerts. While lightweight and easy to implement, the IDS provides a solid foundation for understanding basic intrusion detection principles and can be extended for more advanced network security applications.

REFERENCE

- *How to Build a Real-Time Intrusion Detection System with Python and Open-Source Libraries* – This tutorial walks through packet capture, traffic analysis, detection, and alerting using tools like scapy and Python FreeCodeCamp+1YouTube+1
  🔗 https://www.freecodecamp.org/news/build-a-real-time-intrusion-detection-system-with-python/
- *The Ultimate Port Scanning Guide* – Covers how IDS solutions detect port scanning and defensive strategies like firewalls and port knocking FreeCodeCampYouTube+15SecOps® Solution+15Twingate+15
  🔗 https://www.secopsolution.com/blog/the-ultimate-port-scanning-guide
- *Port Scanning Techniques* by Varonis – Discusses various scanning methods (e.g., FIN, Xmas, NULL) and how IDS systems recognize them SecOps® SolutionVaronis+1Vectra AI+1

🔗 https://www.varonis.com/blog/port-scanning-techniques

- *Intrusion Detection System Using Machine Learning Algorithms* – Explains how IDS can classify 'probing' (port scanning) via ML models Reddit+15Varonis+15Nmap+15GeeksforGeeks

  🔗 https://www.geeksforgeeks.org/machine-learning/intrusion-detection-system-using-machine-learning-algorithms/
- *Intrusion Detection System (Wikipedia)* – Offers an overview of IDS types, including signature and anomaly-based detection, and details on port scan identification GeeksforGeeksVaronis+7Wikipedia+7FreeCodeCamp+7

  🔗 https://en.wikipedia.org/wiki/Intrusion_detection_system
- **Snort** (Wikipedia) – A widely used open-source network-based IDS that detects port scans using signature rules Wikipedialifewire.com+2kassemfawaz.com+2Varonis+2

  🔗 https://en.wikipedia.org/wiki/Snort_(software)
- **Suricata** (Wikipedia) – A high-performance IDS/IPS capable of detecting port scans and other network attacks kassemfawaz.com+2Wikipedia+2lifewire.com+2Nmap+11Wikipedia+11GitHub+11

  🔗 https://en.wikipedia.org/wiki/Suricata_(software)