Title: Intrusion Prevention System Using Suricata and Python

Name: Waqas Ahmed Rind
CNIC: 4120222102853
Mobile #: 03323647267

Abstract/Summary:
This project demonstrates a basic Intrusion Prevention System (IPS) implemented on Kali Linux using Suricata for detection and a Python script for automated prevention. The system monitors SSH connections, detects repeated unauthorized attempts, and dynamically blocks the attacker using iptables. The project showcases practical implementation of network security tools and demonstrates real-time intrusion detection and prevention.

Motivation:
I am interested in the field of cybersecurity, and I wanted to explore the practical aspects of detecting and preventing attacks. I found it interesting to see how a system can automatically block brute-force attacks or other suspicious activity in real time. This project allows me to apply theoretical knowledge in a controlled environment and understand the working of an IPS.

Background and Overview:
The project focuses on network security and intrusion prevention. Suricata is a widely used IDS/IPS tool that monitors network traffic using rules to detect malicious behavior. The Python script complements Suricata by automating response actions using iptables to block malicious IP addresses. This project simulates attacks using Nmap to generate test traffic and demonstrates how an IPS can protect a system from repeated unauthorized access attempts.

Aim and Objective:
The aim of this project is to implement a functional Intrusion Prevention System that can detect and prevent SSH brute-force attacks. The objectives include:

- Installing and configuring Suricata on Kali Linux
- Writing custom rules to detect suspicious traffic
- Developing a Python script to respond to alerts by blocking malicious IP addresses
- Testing the system using simulated attacks and verifying the functionality

Methodology:

1. Set up the Kali Linux environment with Suricata and Python 3.
2. Configure network settings to ensure communication between the IPS machine and the attacker machine.
3. Write custom rules in Suricata to detect SSH brute-force attempts.
4. Implement a Python script to monitor Suricata logs and automatically block attacking IP addresses using iptables.
5. Simulate attacks using Nmap on a host machine to test detection and prevention.
6. Observe Suricata alerts, Python script outputs, firewall rules, and scan results to validate system functionality.

Timeline (Gantt Chart):

- Week 1: Project planning and environment setup
- Week 2: Installing tools and configuring Suricata
- Week 3: Writing custom rules and developing Python script
- Week 4: Testing the system and preparing demonstration

References:

1. Suricata Official Documentation – https://suricata.io
2. Kali Linux Documentation – https://www.kali.org/docs/
3. Nmap Network Scanning Guide – https://nmap.org/book/
4. Python 3 Official Documentation – https://www.python.org/doc/
5. Network Security Concepts – William Stallings, *Cryptography and Network Security*, 8th Edition