A Detailed Report on

Developing a tool that has the potential to detect Attacks like DoS/DDoS

By:
Gurrapu Chetan Raja

Abstract :

This project presents **ShieldGuard**, a web-based DoS and DDoS detection tool designed to monitor network activity and identify abnormal traffic behavior in real time. The system captures live network packets and evaluates traffic patterns within defined time windows to detect potential denial-of-service attacks. When an attack condition is identified, the tool determines the source IP and applies mitigation logic through firewall rules to reduce the impact of malicious traffic. An intuitive web dashboard provides real-time visualization of traffic behavior, attack status, packet statistics, and attacker information. The proof of concept is implemented in a controlled localhost environment using simulated attacks, while the design supports deployment on external servers and cloud environments, making ShieldGuard a flexible and effective solution for DoS/DDoS detection and mitigation.

Introduction :

With the increasing dependence on computer networks and internet-based applications, ensuring network security has become a critical challenge. Modern networks are constantly exposed to various cyber threats, including unauthorized access, data interception, and denial-of-service attacks. Among these, traffic flooding attacks such as Distributed Denial of Service (DDoS) and SYN flood attacks are particularly dangerous, as they can render services unavailable by overwhelming network resources.

Network Intrusion Detection Systems (NIDS) play a vital role in monitoring network traffic and identifying suspicious activities in real time. Traditional security mechanisms such as firewalls are often insufficient to detect complex or high-volume attacks, especially those that exploit standard protocol behavior. This creates the need for efficient traffic monitoring systems capable of analyzing packet-level information and detecting anomalies based on traffic patterns.

This project presents **ShieldGuard**, a real-time network traffic monitoring and attack detection tool developed using Python. The system captures live network packets and analyzes key parameters such as packet rate, source IP distribution, and TCP SYN flags to identify abnormal traffic behavior using threshold-based detection techniques. Upon detecting a potential attack, ShieldGuard classifies the network condition and identifies the suspected attacker IP address.

A lightweight web-based interface developed using Flask provides real-time visualization of network status and traffic statistics. Multithreading is employed to ensure continuous packet capture and analysis without affecting system performance. The proposed system offers a lightweight and cost-effective solution for basic DoS/DDoS detection and mitigation, while also serving as a foundation for future enhancements such as advanced analytics and machine learning–based anomaly detection.

Overview :

The proposed system **ShieldGuard** is a real-time network traffic monitoring and intrusion detection solution designed to identify abnormal traffic patterns and potential cyberattacks such as DoS/DDoS attacks. The **ShieldGuard** tool continuously captures live network packets from a selected network interface and analyzes them within fixed time windows to observe traffic behavior. Key parameters such as total packet count, packet frequency per source IP address, and TCP SYN flag occurrences are monitored to detect anomalies.

Based on predefined threshold values, the system classifies the network state as normal or under attack and identifies the most likely attacker IP address. To provide real-time visibility, the system includes a lightweight web-based dashboard that displays current network status, detected threats, and traffic statistics. Multithreading is utilized to ensure that packet capture, traffic analysis, and user interface operations run concurrently without performance degradation.

Technology Stack :

- **Python**
  Used as the core programming language due to its simplicity, extensive library support, and suitability for network traffic analysis and automation.
- **Scapy**
  A powerful packet manipulation and network analysis library used to capture live network packets and extract protocol-level information such as IP addresses and TCP flags.
- **Flask**
  A lightweight Python web framework used to develop the real-time dashboard and REST API for displaying network status, traffic behavior, and attack

information.

- **Threading Module**
  Enables concurrent execution of packet sniffing, detection logic, and web services to ensure real-time performance without blocking operations.
- **Collections (defaultdict)**
  Used for efficient counting and tracking of packets per source IP address during traffic analysis.
- **Operating System (Kali Linux)**
  Provides low-level network access and built-in security tools required for packet sniffing, traffic analysis, attack simulation, and firewall-based mitigation.
- **iptables**
  Linux firewall utility used to apply blocking rules for malicious source IP addresses during detected attack conditions.
- **hping3**
  A network traffic generation tool used to simulate DoS and DDoS attacks for testing and validating the detection and mitigation capabilities of ShieldGuard.
- **Chart.js**
  JavaScript visualization library used to display real-time traffic spikes and network behavior on the web dashboard.

What is DoS/DDoS attack?

- **DoS Attack:**
  A Denial of Service (DoS) attack is an attempt to make a system or network unavailable by flooding it with excessive traffic from a single source.
- **DDoS Attack:**
  A Distributed Denial of Service (DDoS) attack overwhelms a target by sending massive traffic from multiple compromised systems simultaneously.

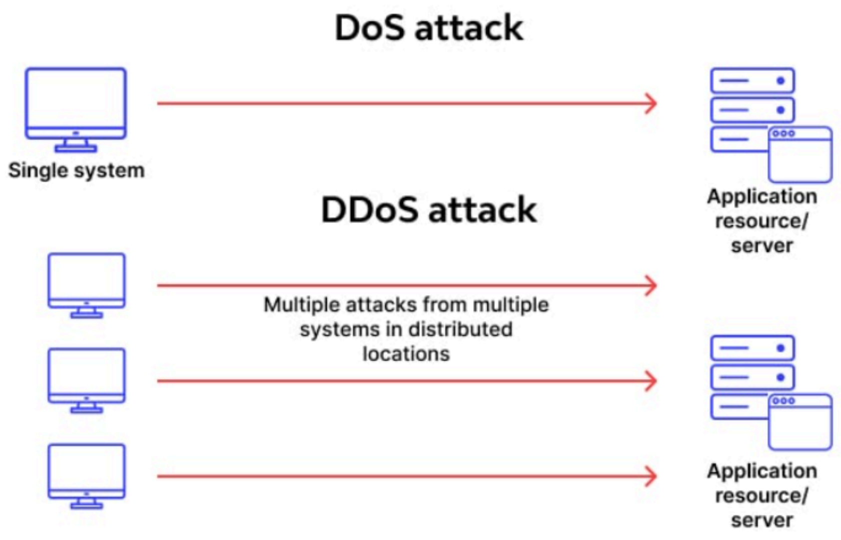

DoS attack
Single system
Application resource/ server
DDoS attack
Multiple attacks from multiple systems in distributed locations
Application resource/ server

Fig 1:DoS Attack and DDoS Attack

Types of DoS and DDoS Attacks :

- **DoS attacks** include SYN flood attacks, where excessive connection requests are sent to exhaust server resources, UDP flood attacks that overwhelm the target with large volumes of UDP packets, and ICMP (ping) flood attacks that consume network bandwidth.
- **DDoS attacks** include volumetric attacks that flood the network with massive traffic, protocol-based attacks that exploit weaknesses in network protocols such as TCP and UDP, and application-layer attacks that target specific services like web servers by sending a large number of HTTP requests.

OWASP Top 10 and DoS/DDoS :

The **OWASP Top 10** does not include Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks as standalone categories. This is because the OWASP Top 10 primarily focuses on vulnerabilities in web applications rather than network-level attacks. However, the impact of DoS and DDoS attacks, particularly on system availability, is indirectly addressed under availability-related risks, where attackers exploit application or protocol weaknesses to disrupt services and deny access to legitimate users.

Attack Vector :

Attack vectors refer to the methods used by attackers to launch DoS or DDoS attacks against a system. In the context of ShieldGuard, common attack vectors include TCP SYN floods, ICMP floods, UDP floods, and HTTP request floods. These attacks exploit network and transport layer protocols by sending excessive or malformed traffic to exhaust server resources and disrupt normal service availability.

Attack Arena :

The threat arena defines the environment in which attacks can occur. ShieldGuard considers multiple threat arenas, including local systems, on-premise servers, and cloud-based deployments. Attacks may originate from a single system in a local network or from multiple distributed sources across the internet. The tool is designed to operate across these environments by monitoring traffic behavior and adapting detection mechanisms based on the deployment context.

Attack Surface :

The attack surface represents all possible entry points where an attacker can target the system. For ShieldGuard, the attack surface includes network-facing services such as open ports, web servers, exposed APIs, and the network interface itself. Any service that accepts incoming network traffic can become a potential target for DoS or DDoS attacks if not properly monitored and protected.

Indicators of Compromise (IOC) :

- Abnormally high number of network packets within a short time window
- Excessive TCP SYN packets indicating potential SYN flood activity
- Repeated packet transmission from a single source IP address
- Sudden and continuous spike in overall network traffic
- Source IP generating traffic beyond predefined threshold limits

These indicators are used by the system ShieldGuard to identify and classify potential DoS or DDoS attacks.

Tactics, Techniques, and Procedures (TTPs) :

- **Tactic:** Service disruption by exhausting server resources
- **Technique:** TCP SYN flooding to exploit the TCP three-way handshake
- **Procedure:** Repeatedly sending a large number of SYN packets within a short time window from one or multiple source IP addresses without completing the connection

Indicators of Attack (IOA) :

- Continuous increase in packet rate over a short time window
- Rapid rise in TCP SYN packets before normal traffic completion
- Unusual traffic patterns that deviate from baseline behavior
- Multiple connection attempts without successful session establishment
- Repeated traffic bursts indicating active attack behavior

Working of ShieldGuard Tool:

- The ShieldGuard tool starts by initializing all required modules, including packet capture, traffic counters, detection logic, and the Flask-based web interface. Once the system is running, network packets are continuously captured from the configured network interface using the Scapy sniffing mechanism.
- Each incoming packet is inspected to extract IP and TCP information. The system increments packet counts per source IP and separately tracks TCP packets with the SYN flag. Packet capture operates within a fixed time window, after which the collected traffic data is evaluated.
- The detection logic calculates the total number of packets and compares the observed packet rate and SYN packet count against predefined threshold values. If the thresholds are exceeded, the system flags the traffic as abnormal and identifies the source IP address generating the highest number of packets as the potential attacker.
- The system updates the current network status, attacker IP information, and traffic history in memory. These results are made available in real time through Flask endpoints, allowing the web dashboard to display current traffic conditions and attack status. No automatic mitigation is applied in the current implementation; the system focuses on detection, monitoring, and visualization.

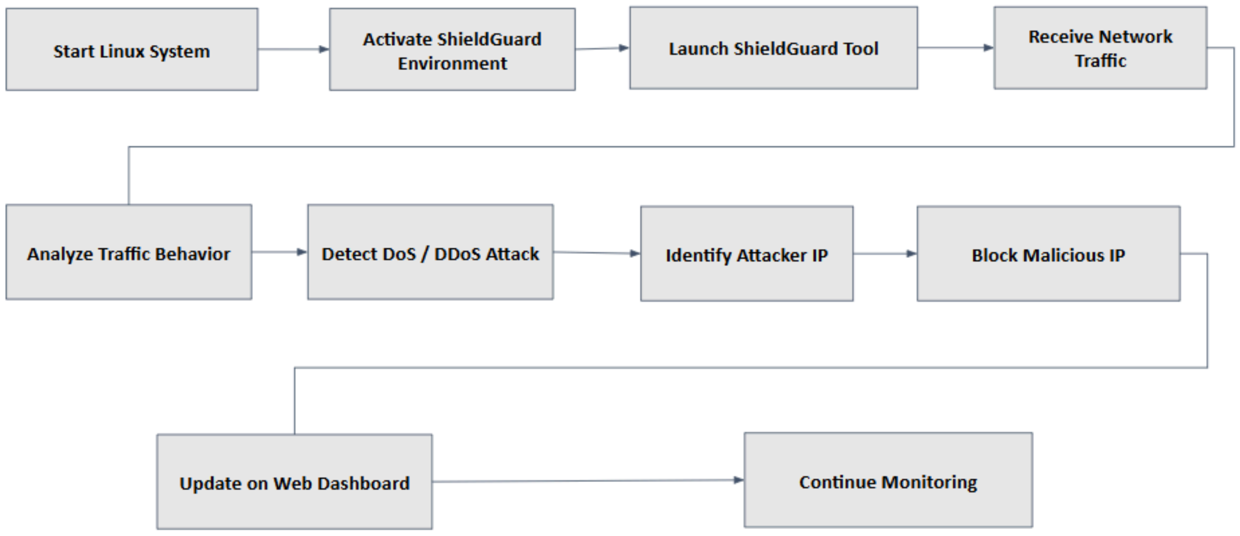


Fig 2: System Workflow Diagram of ShieldGuard DoS/DDoS Detection and Mitigation Tool

Project Execution Commands and Their Purpose :

1. System Update & Tool Installation

```
sudo apt update
```

#Updates package list.

```
sudo apt upgrade -y
```

#Upgrades installed packages.

```
sudo apt install python3 python3-pip python3-venv wireshark
tcpdump hping3 -y
```

#Installs:

- Python (development)
- Virtual environment support
- Packet capture tools
- Attack simulation tool (`hping3`)

2. Virtual Environment Setup

```
python3 -m venv shieldguard-env
```

#Creates an isolated Python environment for the project.

```
source shieldguard-env/bin/activate
```

#Activates the virtual environment.

```
pip install scapy flask
```

#Installs:

- **Scapy** → packet sniffing

- **Flask** → web dashboard

3. Project Folder & Code Creation

```
mkdir ShieldGuard
```

#Creates project directory.

```
cd ShieldGuard
```

#Enters project directory.

```
nano shieldguard.py
```

# Creates and opens the main Python file
#Full ShieldGuard code is pasted here

4. Run the ShieldGuard Tool

```
source shieldguard-env/bin/activate
```

#Ensures correct environment is active.

```
sudo python shieldguard.py
```

#Runs the ShieldGuard tool:

- Starts packet capture
- Starts detection engine
- Launches web dashboard
- Enables firewall access

5. Access Web Dashboard

```
http://127.0.0.1:5000
```

#Opens ShieldGuard web interface:

- Status
- IP details
- Packet count
- Traffic spike graph

6. Attack Simulation (PoC)

```
sudo hping3 -S --flood -p 80 127.0.0.1
```

#Simulates a **TCP SYN flood DoS attack**
# Generates high packet rate traffic
#Used to test detection logic

7. Stop Attack / Tool

```
Ctrl + C
```

#Stops:

- Attack simulation
- ShieldGuard monitoring
- Web dashboard

8. Firewall Verification

```
sudo iptables -L
```

#Verifies blocked IP rules (in real deployment).

9. General Utility Commands (Used During Setup)

```
ip addr
```

#Checks network interfaces.

```
ls
```

#Lists files/directories.

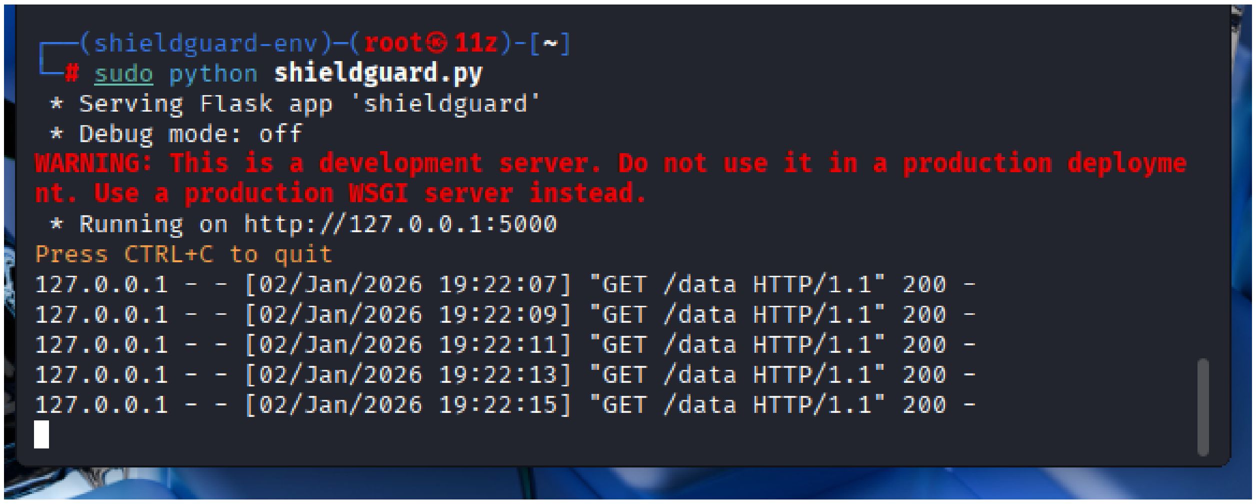
```
pwd
```

#Shows current directory.

```
deactivate
```

#Exits virtual environment.

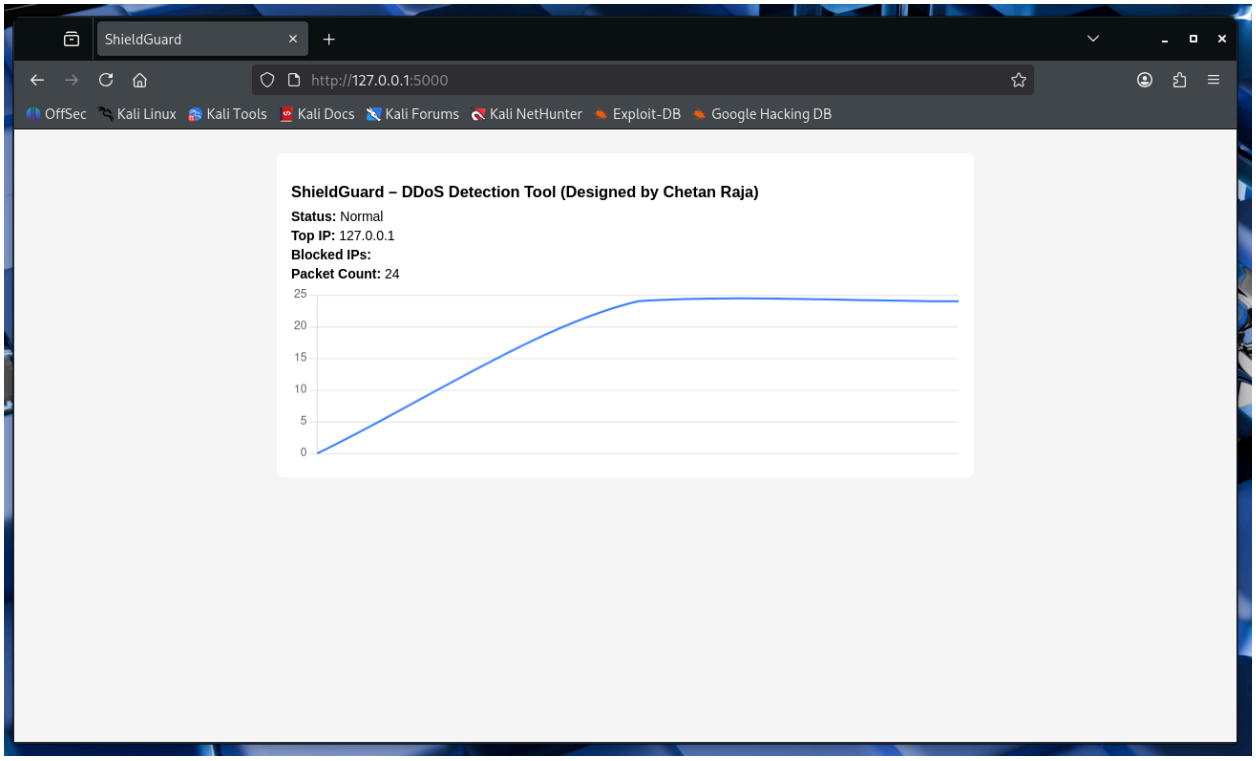
Proof of Concept (PoC) :



```
┌──(shieldguard-env)─(root㉿11z)-[~]
└─# sudo python shieldguard.py
 * Serving Flask app 'shieldguard'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployme
nt. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [02/Jan/2026 19:22:07] "GET /data HTTP/1.1" 200 -
127.0.0.1 - - [02/Jan/2026 19:22:09] "GET /data HTTP/1.1" 200 -
127.0.0.1 - - [02/Jan/2026 19:22:11] "GET /data HTTP/1.1" 200 -
127.0.0.1 - - [02/Jan/2026 19:22:13] "GET /data HTTP/1.1" 200 -
127.0.0.1 - - [02/Jan/2026 19:22:15] "GET /data HTTP/1.1" 200 -
```

PoC 1: ShieldGuard Tool Initialization

This screenshot shows the successful initialization of the ShieldGuard DDoS detection tool. The application starts the traffic monitoring engine and launches the web-based dashboard on the local server.

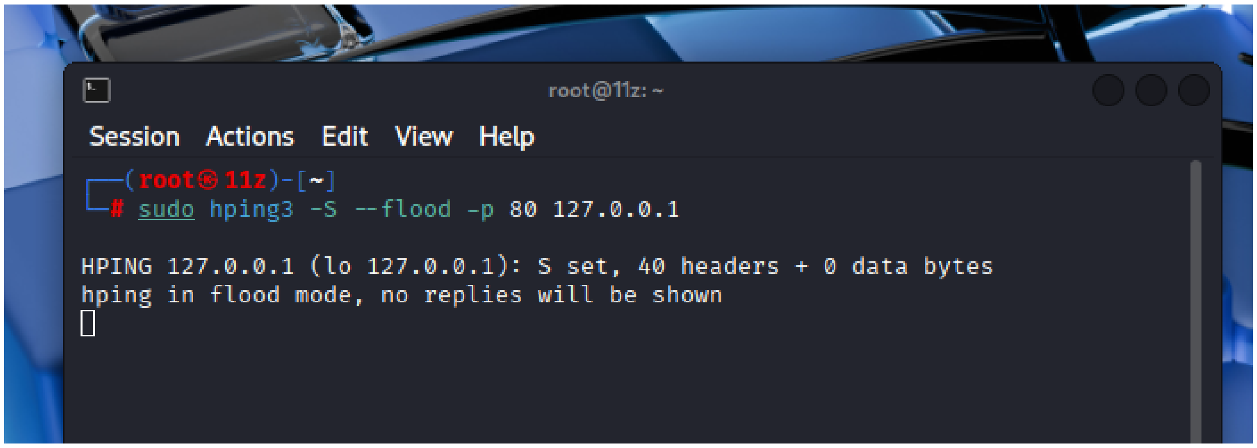


ShieldGuard – DDoS Detection Tool (Designed by Chetan Raja)

Status: Normal
Top IP: 127.0.0.1
Blocked IPs:
Packet Count: 24

PoC 2: Normal Traffic State

This screenshot represents the normal traffic condition. ShieldGuard monitors network activity and displays baseline traffic behavior with no attack detected.

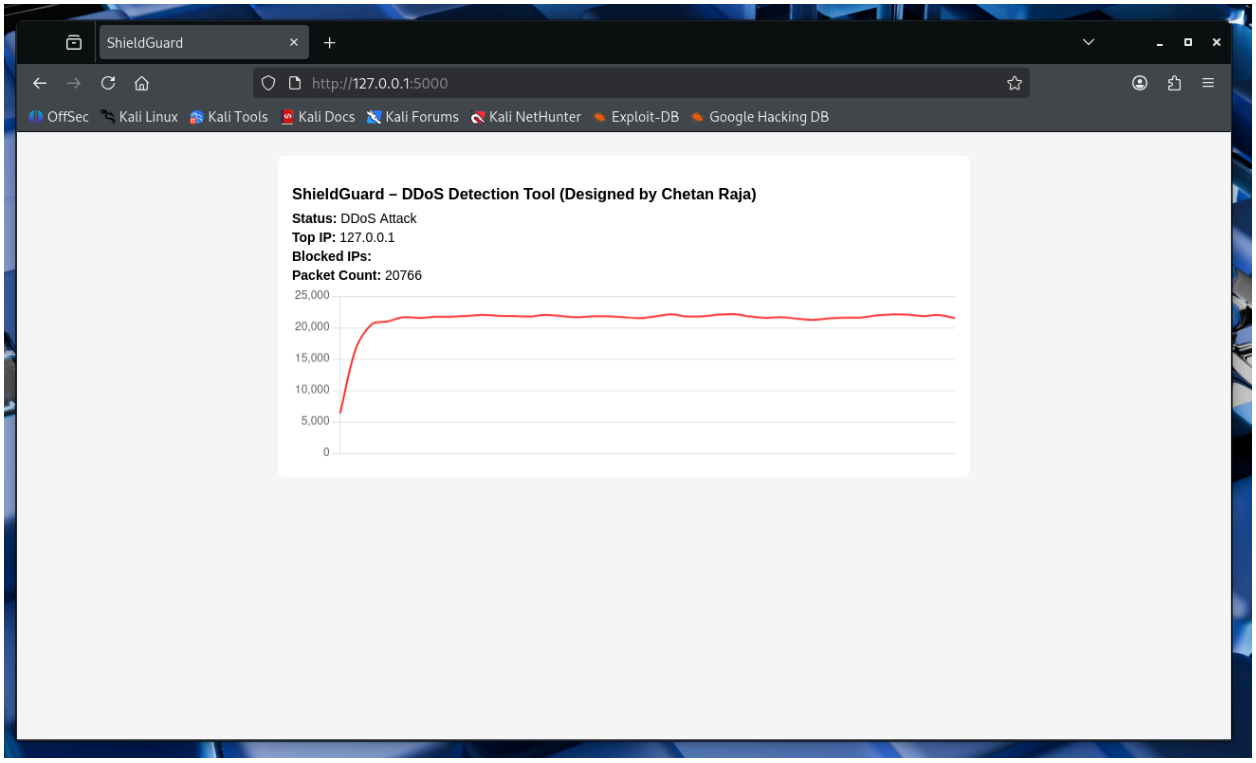


```
┌──(root㉿11z)-[~]
└─# sudo hping3 -S --flood -p 80 127.0.0.1

HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

PoC 3: Attack Simulation Using hping3

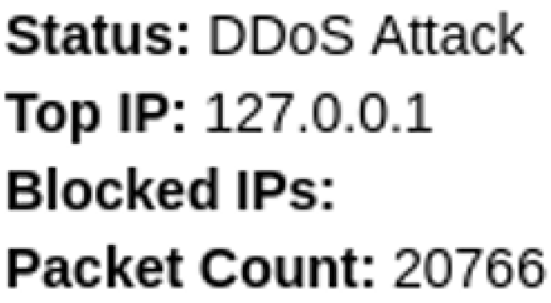
The screenshot has A TCP SYN flood attack is simulated using the hping3 tool. This generates high-rate packets to test the detection capability of ShieldGuard in a controlled environment.


ShieldGuard – DDoS Detection Tool (Designed by Chetan Raja)

Status: DDoS Attack
Top IP: 127.0.0.1
Blocked IPs:
Packet Count: 20766

PoC 4: DDoS Attack Detection

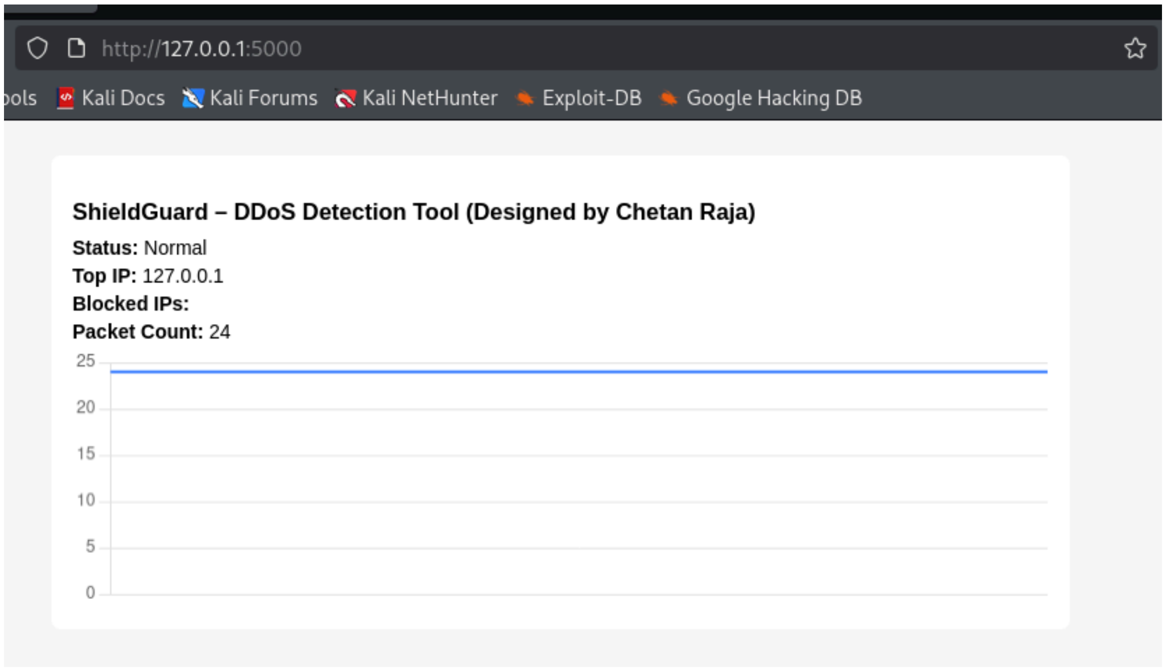
In This screenshot the ShieldGuard successfully detects abnormal traffic behavior and identifies the event as a DDoS attack. The sudden spike in packet volume is clearly visualized on the dashboard.


Status: DDoS Attack
Top IP: 127.0.0.1
Blocked IPs:
Packet Count: 20766

PoC 5: Attacker IP Identification

In this screenshot the system identifies the source IP responsible for the attack. In this localhost testing scenario, the attacker IP appears as 127.0.0.1.

Although the attack is detected, the localhost IP is not blocked to prevent self-denial of service during testing. In real-world deployments, ShieldGuard automatically blocks external attacker IPs using firewall rules.


ShieldGuard – DDoS Detection Tool (Designed by Chetan Raja)

Status: Normal
Top IP: 127.0.0.1
Blocked IPs:
Packet Count: 24

PoC 6: Recovery After Attack

After stopping the attack, ShieldGuard returns to a normal monitoring state, demonstrating the system's ability to recover and continue operation.

The Proof of Concept demonstrates real-time detection of a simulated DDoS attack, visualization of traffic spikes, attacker identification, and mitigation logic validated in a controlled localhost environment.

Impact of DoS and DDoS Attacks :

- Causes **service unavailability**, preventing legitimate users from accessing systems or applications.
- Leads to **system slowdowns or crashes** due to resource exhaustion.
- Results in **financial losses** caused by downtime and disrupted operations.
- Affects **business reputation and user trust** due to repeated service outages.
- Consumes excessive **network bandwidth**, impacting other connected services.
- May act as a **distraction for launching other cyberattacks** while defenses are overwhelmed.

Mitigation Strategies for DoS and DDoS Attacks :

- Implement **traffic monitoring and rate limiting** to control excessive requests.
- Use **firewall rules** to block malicious IP addresses and suspicious traffic patterns.
- Apply **threshold-based detection** to identify abnormal traffic behavior early.
- Enable **SYN flood protection** mechanisms to prevent TCP connection abuse.
- Deploy **Intrusion Detection and Prevention Systems (IDPS)** for continuous monitoring.
- Use **load balancing** to distribute traffic and reduce single-point overload.
- Regularly **update and harden systems** to reduce exploitable weaknesses.

Conclusion :

ShieldGuard enhances network security by providing real-time monitoring, detection, and mitigation of DoS and DDoS attacks with minimal system overhead. The system efficiently analyzes packet-level traffic to identify abnormal behavior and malicious sources within short time windows. Automated firewall-based blocking helps limit repeated attack attempts and maintain service availability. Its multithreaded and modular architecture ensures continuous, reliable operation and makes the system robust for deployment on both local machines and remote servers. Real-time visualization through a web-based dashboard improves situational awareness and simplifies network administration.

References :

1. **OWASP**
   OWASP Top 10 – Web Application Security Risks -
   https://owasp.org/www-project-top-ten/
2. **Scapy**
   Scapy Official Documentation - https://scapy.net/documentation/
3. **Flask**
   Flask Official Documentation - https://flask.palletsprojects.com/
4. **Internet Engineering Task ForceI(IETF)**
   RFC 793 – Transmission Control Protocol (TCP) -
   https://datatracker.ietf.org/doc/html/rfc793
5. **Cloudflare**
   Understanding DDoS Attacks -
   https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/
6. **Linux Foundation**
   iptables Documentation - https://netfilter.org/documentation/