Proof Of Concept: Homoglyph Shortener

Name: Purva Pawaskar, **Intern ID**: 259

NIPS Tool

What is an IPS?

An Intrusion Prevention System (IPS) is a network security tool that goes a step beyond detection. Like an IDS, it monitors network traffic for malicious activity and policy violations. However, an IPS is placed in line with traffic and can actively block or prevent detected intrusions in real-time, making it a proactive defence mechanism.

Why is IPS important?

While an IDS alerts you to a fire, an IPS acts as the automated fire sprinkler system. It provides an immediate, automated response to threats, which is crucial for stopping attacks like DoS, exploits, and brute-force attempts before they can cause significant damage.

Key Concepts in IPS:

- **Types of IPS:**
    - **Network-based (NIPS):** Deployed at strategic points in the network to monitor traffic to and from all devices. Our tool is a NIPS.
    - **Host-based (HIPS):** Installed on a single host or endpoint, monitoring its inbound and outbound traffic and system processes.
- **Detection & Prevention Techniques:**
    - **Signature-based:** Uses predefined patterns (signatures) of known attacks to detect and block them.
    - **Anomaly-based:** Establishes a baseline of normal network behaviour and blocks traffic that deviates significantly from it.
    - **Policy-based:** Administrators define security policies, and the IPS blocks any traffic that violates these policies.
- **Deployment Models:**

  - **Inline Mode:** The IPS is situated directly in the path of network traffic. All data must pass through it, allowing it to block malicious packets before they reach their target. This is the standard deployment for an IPS.
  - **Passive Mode:** The IPS monitors a copy of the traffic. While it can detect threats, its ability to block them is limited to sending TCP reset packets or updating firewall rules, rather than dropping the packets directly.

Main Tool: Enhanced IPS

Our tool is a comprehensive, GUI-based Intrusion Prevention System built to not only detect a variety of network threats but also to take immediate, automated action to block the source of the attack.

Libraries Used

- **tkinter:** The foundation for the graphical user interface (GUI), providing the main window, frames, buttons, themed widgets, and real-time display tables.
- **scapy:** The core engine for capturing, dissecting, and analysing network packets from a network interface in real-time.
- **threading & queue:** These allow the packet sniffing and analysis process to run in a non-blocking background thread, ensuring the GUI remains responsive while passing packets safely to the main thread.
- **subprocess & platform:** Used to execute system-level commands to interact with the host firewall (iptables on Linux, netsh advfirewall on Windows) to dynamically block malicious IP addresses.
- **re (Regex):** Provides regular expression capabilities for sophisticated pattern matching within packet payloads, crucial for detecting web attacks like SQL Injection and XSS.
- **collections.defaultdict:** A highly efficient data structure used to track threat metrics, such as ping counts, SYN packet rates, and scanned ports per IP address.
- **os:** Used to check for the necessary administrator privileges required to modify firewall rules.

Core Components & Logic

1. Threat Detection Thresholds (Rate-Based): The system tracks the rate of certain packet types over a specific time window to detect denial-of-service and scanning attacks.

- **ICMP Flood:** Flags and blocks an IP sending more than **10 ICMP pings** within **3 seconds**.
- **SYN Flood:** Flags and blocks an IP sending more than **30 TCP SYN packets** within **10 seconds**.
- **Port Scan:** Flags and blocks an IP that attempts to connect to more than **8 different ports** within **3 seconds**.
- **HTTP Violations:** Blocks an IP after it sends **2 or more** packets containing payloads that match web-attack signatures.

2. Pattern Matching (Signature-Based): The IPS inspects the raw payload of HTTP traffic for signatures of common web application attacks.

- **HTTP_SUSPICIOUS_REGEX:** A compiled regular expression that looks for pattern’s indicative of SQL Injection (union select, 1=1--) and Cross-Site Scripting (<script>, onerror=).
- **SQL_INJECTION_PATTERNS & XSS_PATTERNS:** Lists of specific byte strings (e.g., b'SELECT', b'alert(') that are highly indicative of an attack.

3. Automated Blocking Mechanism: This is the "Prevention" component. When a threat is confirmed:

- The block_ip function identifies the operating system.
- It constructs and executes the appropriate command (iptables for Linux or netsh for Windows) to add a new firewall rule that **DROPS** all incoming traffic from the attacker's IP address.
- The blocked IP is added to a central list, and the GUI is updated immediately.

Function Breakdown

- **IPS.__init__(self, root):** Initializes the main application, checks for admin privileges, defines themes, and creates the UI.

- **_create_widgets(self):** Builds all GUI elements, including the control buttons, statistics panel, and the panes for threat detection, blocked IPs, packet details, and logs.
- **start_protection(self) / stop_protection(self):** Controls the activation and deactivation of the packet sniffing thread.
- **packet_sniffer(self):** The function that runs in the background, using scapy.sniff() to capture all network traffic.
- **analyze_and_block(self, packet):** The core detection and prevention engine. For each packet, it checks against the rate-based thresholds and signature-based patterns. If a rule is violated, it calls the block_ip function.
- **block_ip(self, ip, reason):** The active response module. It executes system commands to add the attacker's IP to the firewall's blocklist and updates the tracking lists.
- **unblock_all_ips(self):** A manual override that removes all firewall rules created by the IPS, effectively unblocking all previously blocked IPs.
- **update_gui(self):** Periodically runs in the main thread to pull packets from the queue, pass them to the analysis engine, and update all visual components with new information.
- **show_threat_details(self, event):** Triggered when a user clicks on a threat in the table. It displays the full packet data and a summary of the threat in the details pane.

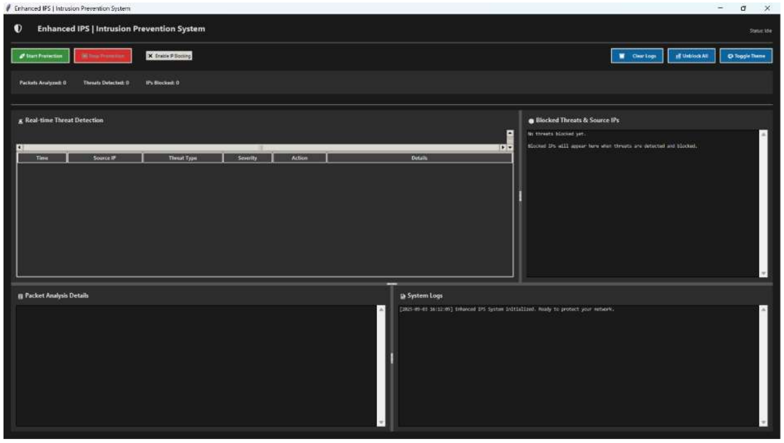
Working and Demonstration

The tool must be run with administrator/root privileges to allow it to capture network packets and modify firewall rules.

Step 1: Initial Interface When the tool starts, it presents a clean, comprehensive dashboard. The status is "Idle." Key areas are visible:

- **Controls:** "Start Protection," "Stop Protection," "Enable IP Blocking," and "Unblock All."
- **Stats Panel:** Shows live counts of packets analysed, threats detected, and IPs blocked.

- **Real-time Threat Detection:** A table where new threats will appear as they are detected.
- **Blocked Threats & Source IPs:** A summary view of all IPs that have been actively blocked by the firewall.
- **Packet Analysis & System Logs:** Detailed views for inspecting individual packets and viewing the IPS's operational history.


Enhanced IPS | Intrusion Prevention System
Enhanced IPS | Intrusion Prevention System
Status: Idle
Start Protection
Stop Protection
✕ Enable IP Blocking
Clear Logs
Unblock All
Toggle Theme
Real-time Threat Detection
Blocked Threats & Source IPs
Packet Analysis Details
System Logs

Packets Analyzed: 0 Threats Detected: 0 IPs Blocked: 0

Time	Source IP	Threat Type	Severity	Action	Details

```
No threats blocked yet.

Blocked IPs will appear here when threats are detected and blocked.
```

```
[2025-09-03 16:12:05] Enhanced IPS System initialized. Ready to protect your network.
```

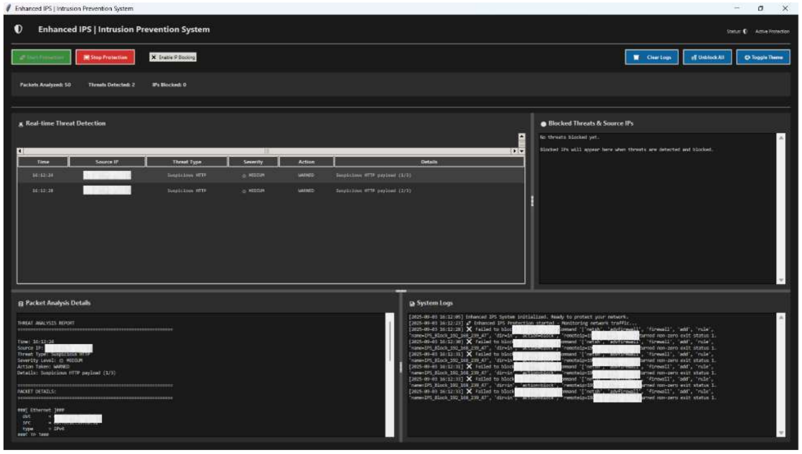
Step 2: Activating Protection Clicking **"Start Protection"** activates the IPS. The status changes to "Active Protection," and the tool begins capturing and analysing every packet on the network in real-time. The "Packets Analysed" counter starts to climb.

Step 3: Threat Detection and Automated Blocking When a malicious packet or pattern is detected—for example, a rapid series of TCP SYN packets from a single source—the IPS takes immediate action:

1. **Detection:** A new entry appears in the **"Real-time Threat Detection"** table, detailing the timestamp, source IP, threat type (e.g., "SYN Flood Attack"), severity, and the action taken ("BLOCKED").
2. **Prevention:** Simultaneously, the block_ip function is triggered. A firewall rule is created on the host machine.
3. **Logging:** The attacker's IP is added to the **"Blocked Threats & Source IPs"** list with details about why it was blocked. The "IPs Blocked" counter increments.

Step 4: Analysis and Mitigation An analyst can click on any threat in the detection table. This populates the **"Packet Analysis Details"** pane with the full, raw data of the malicious packet, allowing for in-depth forensic analysis.

The tool provides all the necessary information to understand the nature of the attack that was automatically prevented.



Enhanced IPS | Intrusion Prevention System
Enhanced IPS | Intrusion Prevention System
Status: ◐ Active Protection
Start Protection
Stop Protection
☒ Enable IP Blocking
Clear Logs
Unblock All
Toggle Theme

Packets Analyzed: 50   Threats Detected: 2   IPs Blocked: 0

Real-time Threat Detection

Time	Source IP	Threat Type	Severity	Action	Details
16:12:24	[redacted]	Suspicious HTTP	◎ MEDIUM	WARNED	Suspicious HTTP payload (1/3)
16:12:28	[redacted]	Suspicious HTTP	◎ MEDIUM	WARNED	Suspicious HTTP payload (2/3)

Blocked Threats & Source IPs

```
No threats blocked yet.

Blocked IPs will appear here when threats are detected and blocked.
```

Packet Analysis Details

```
THREAT ANALYSIS REPORT
==================================================

Time: 16:12:24
Source IP: [redacted]
Threat Type: Suspicious HTTP
Severity Level: ◎ MEDIUM
Action Taken: WARNED
Details: Suspicious HTTP payload (1/3)

==================================================
PACKET DETAILS:
==================================================

###[ Ethernet ]###
  dst       = [redacted]
  src       = [redacted]
  type      = IPv6
###[ IP ]###
```

System Logs

```
[2025-09-03 16:12:05] Enhanced IPS System initialized. Ready to protect your network.
[2025-09-03 16:12:23] 🚀 Enhanced IPS Protection started - Monitoring network traffic...
[2025-09-03 16:12:28] ✕ Failed to block[redacted]ommand '['netsh', 'advfirewall', 'firewall', 'add', 'rule',
'name=IPS_Block_192_168_239_47', 'dir=in', 'action=block', 'remoteip=1[redacted]urned non-zero exit status 1.
[2025-09-03 16:12:30] ✕ Failed to block[redacted]ommand '['netsh', 'advfirewall', 'firewall', 'add', 'rule',
'name=IPS_Block_192_168_239_47', 'dir=in', 'action=block', 'remoteip=1[redacted]urned non-zero exit status 1.
[2025-09-03 16:12:31] ✕ Failed to block[redacted]ommand '['netsh', 'advfirewall', 'firewall', 'add', 'rule',
'name=IPS_Block_192_168_239_47', 'dir=in', 'action=block', 'remoteip=1[redacted]urned non-zero exit status 1.
[2025-09-03 16:12:31] ✕ Failed to block[redacted]ommand '['netsh', 'advfirewall', 'firewall', 'add', 'rule',
'name=IPS_Block_192_168_239_47', 'dir=in', 'action=block', 'remoteip=1[redacted]urned non-zero exit status 1.
[2025-09-03 16:12:33] ✕ Failed to block[redacted]ommand '['netsh', 'advfirewall', 'firewall', 'add', 'rule',
'name=IPS_Block_192_168_239_47', 'dir=in', 'action=block', 'remoteip=19[redacted]urned non-zero exit status 1.
[2025-09-03 16:12:33] ✕ Failed to block[redacted]ommand '['netsh', 'advfirewall', 'firewall', 'add', 'rule',
'name=IPS_Block_192_168_239_47', 'dir=in', 'action=block', 'remoteip=19[redacted]urned non-zero exit status 1.
```