📘 User Guide: Honey-Auditor V3

Author: Sahil Naik **Version:** 3.0 **Date:** February 2026

1. Project Initialization

Download the repository from GitHub and prepare the environment for execution.

Instruction:

1. Open your terminal in Kali Linux.
2. Clone the repository using Git:

    git clone https://github.com/0xSahil21/Python-Honeypot-IPS.git

3. Navigate into the project directory:

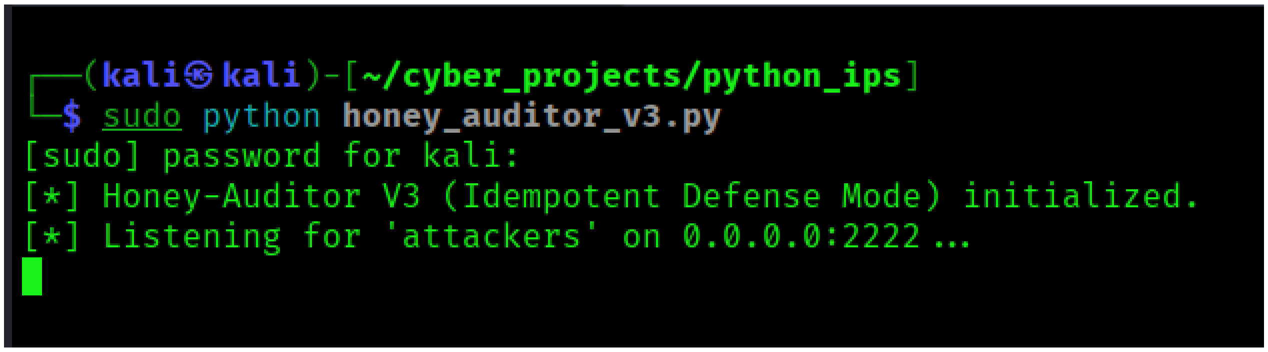
    cd Python-Honeypot-IPS

4. Verification:

    Run `ls` to confirm that `honey_auditor_v3.py` and other scripts are present.

5. Execute:

    Run sudo python honey_auditor_v3.py



```
┌──(kali㉿kali)-[~/cyber_projects/python_ips]
└─$ sudo python honey_auditor_v3.py
[sudo] password for kali:
[*] Honey-Auditor V3 (Idempotent Defense Mode) initialized.
[*] Listening for 'attackers' on 0.0.0.0:2222 ...
```

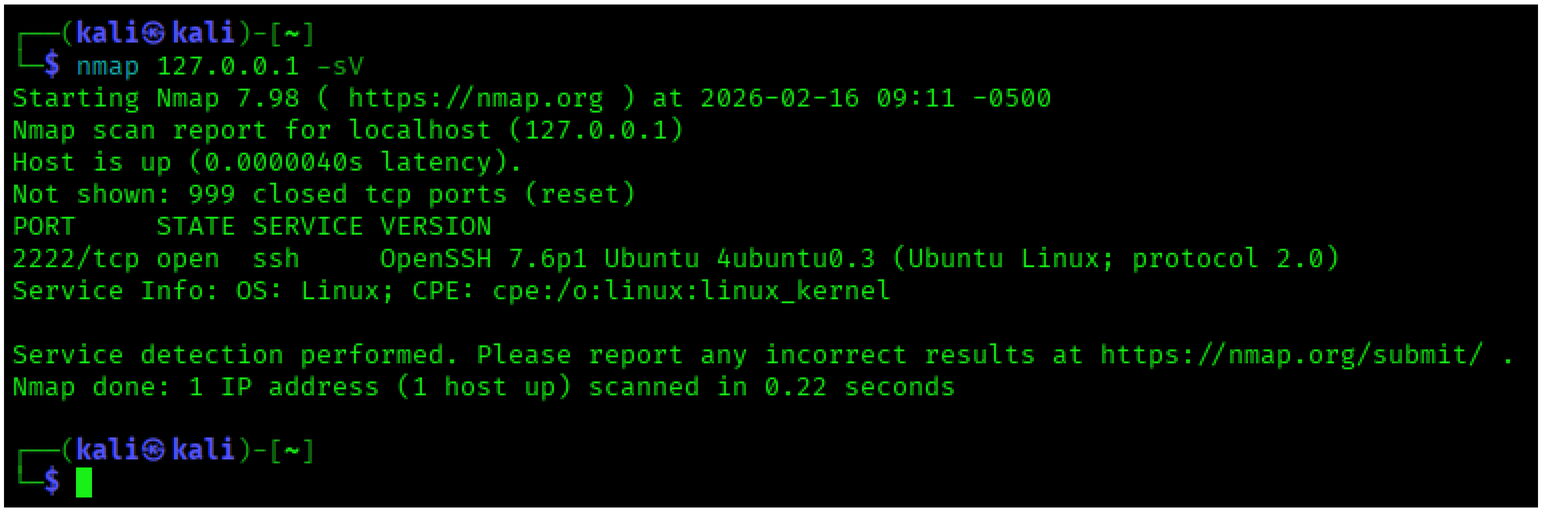
The source code :

```python
import socket
import sys
import os
import subprocess
import threading
from datetime import datetime

# Configuration
BIND_IP = "0.0.0.0"
BIND_PORT = 2222
LOG_FILE = "intrusions.log"

def log_event(ip, port):
    """Logs the intrusion to a file for GRC reporting."""
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    log_entry = f"{timestamp} - INTRUSION DETECTED - Source: {ip} - Port: {port}\n"

    with open(LOG_FILE, "a") as f:
        f.write(log_entry)

    print(f"\n[!] ALARM: Connection received from {ip}")

def block_ip(ip):
    """
    Interacts with Linux Kernel via iptables to ban the IP.
    Implements IDEMPOTENCY to prevent duplicate rules.
    """
    # 1. Check if rule already exists (Idempotency)
    check_cmd = ["iptables", "-C", "INPUT", "-s", ip, "-j", "DROP"]
    try:
        subprocess.check_call(check_cmd, stderr=subprocess.DEVNULL)

        print(f"[*] Idempotency Check: {ip} is already in the ban list. Skipping.")

        return

    except subprocess.CalledProcessError:

        # Rule does not exist, proceed to ban

        pass



    # 2. Add the DROP rule

    print(f"[!] AUTOMATION ENGAGED: Blocking {ip} in Firewall ...")

    ban_cmd = ["iptables", "-A", "INPUT", "-s", ip, "-j", "DROP"]

    subprocess.run(ban_cmd)

    print(f"[+] SUCCESS: {ip} has been neutralized.")



def handle_client(client_socket, ip):

    """Handles the connection and triggers defense."""

    # Send fake banner to trick scanners

    client_socket.send(b"SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3\n")



    # Log and Ban

    log_event(ip, BIND_PORT)

    block_ip(ip)



    client_socket.close()



def start_honeypot():

    print("[*] Honey-Auditor V3 (Idempotent Defense Mode) initialized.")

    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    server.bind((BIND_IP, BIND_PORT))

    server.listen(5)

    print(f"[*] Listening for 'attackers' on {BIND_IP}:{BIND_PORT} ...")



    while True:
        client, addr = server.accept()

        client_handler = threading.Thread(target=handle_client, args=(client, addr[0]))

        client_handler.start()


if __name__ == "__main__":

    if os.geteuid() != 0:

        sys.exit("[-] This script requires Root privileges to manage the firewall. Run with sudo.")


    try:

        start_honeypot()

    except KeyboardInterrupt:

        print("\n[*] Shutting down Honey-Auditor.")
```

2. Attack Simulation (Reconnaissance)

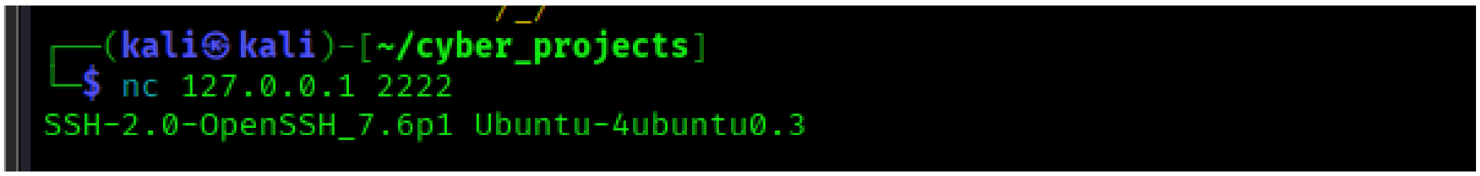
Simulate a malicious actor scanning or connecting to the trap.

- **Instruction:**
    1. Open a second terminal window (representing the Attacker).
    2. Run nmap and scan your local ip.



```
┌──(kali㉿kali)-[~]
└─$ nmap 127.0.0.1 -sV
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-16 09:11 -0500
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
2222/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

    3. Use Netcat (`nc`) to connect to the target IP: `nc 127.0.0.1 2222`.
    4. **Observation:** The honeypot accepts the connection and sends the fake SSH banner (`SSH-2.0-OpenSSH...`). This confirms the "Lure" is active.

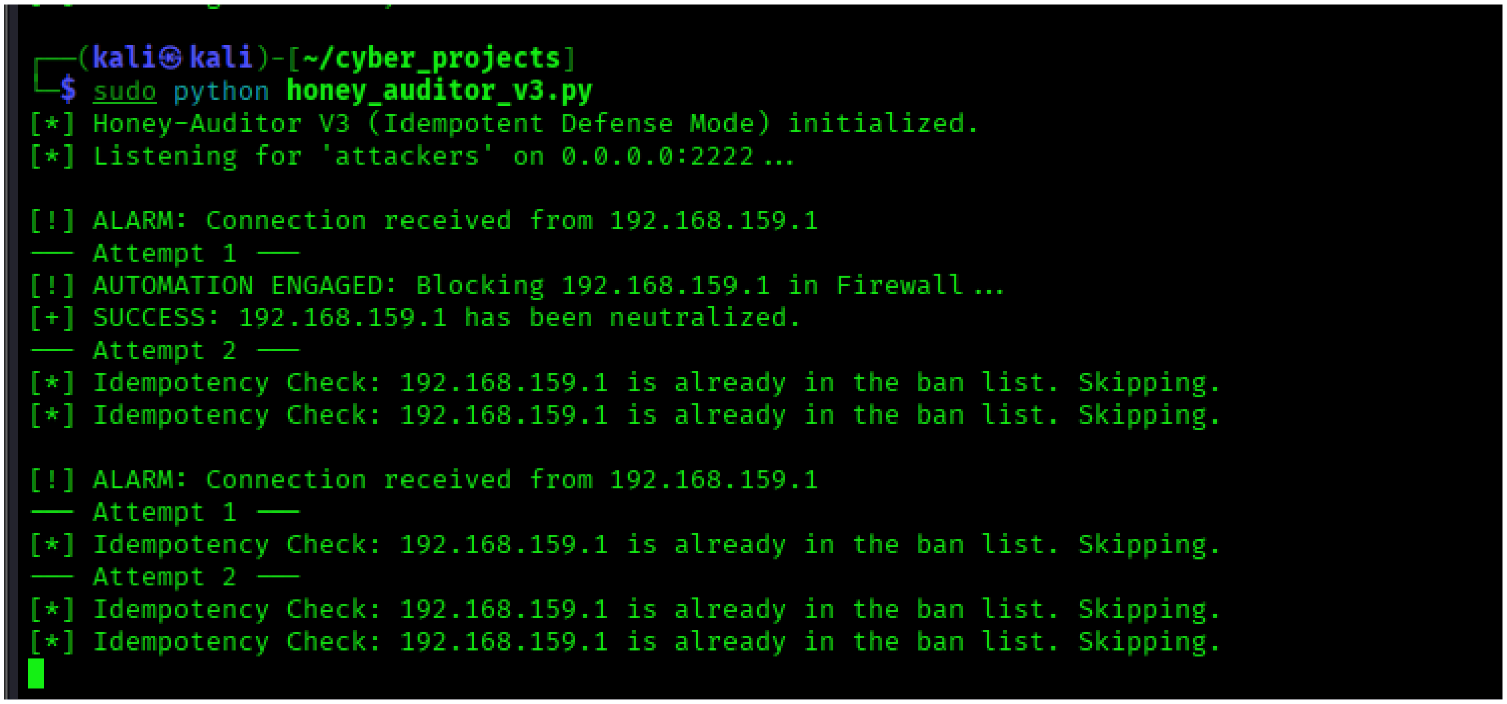


```
┌──(kali㉿kali)-[~/cyber_projects]
└─$ nc 127.0.0.1 2222
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
```

3. Active Defense in Action

Demonstrate the tool's core functionality : detecting an intrusion and automatically updating the firewall.

- **Instructions:**
    1. Switch back to your main terminal (Blue Team).
    2. As soon as the "Attacker" connects, observe the real-time output.



```
┌──(kali㉿kali)-[~/cyber_projects]
└─$ sudo python honey_auditor_v3.py
[*] Honey-Auditor V3 (Idempotent Defense Mode) initialized.
[*] Listening for 'attackers' on 0.0.0.0:2222 ...

[!] ALARM: Connection received from 192.168.159.1
─── Attempt 1 ───
[!] AUTOMATION ENGAGED: Blocking 192.168.159.1 in Firewall ...
[+] SUCCESS: 192.168.159.1 has been neutralized.
─── Attempt 2 ───
[*] Idempotency Check: 192.168.159.1 is already in the ban list. Skipping.
[*] Idempotency Check: 192.168.159.1 is already in the ban list. Skipping.

[!] ALARM: Connection received from 192.168.159.1
─── Attempt 1 ───
[*] Idempotency Check: 192.168.159.1 is already in the ban list. Skipping.
─── Attempt 2 ───
[*] Idempotency Check: 192.168.159.1 is already in the ban list. Skipping.
[*] Idempotency Check: 192.168.159.1 is already in the ban list. Skipping.
```

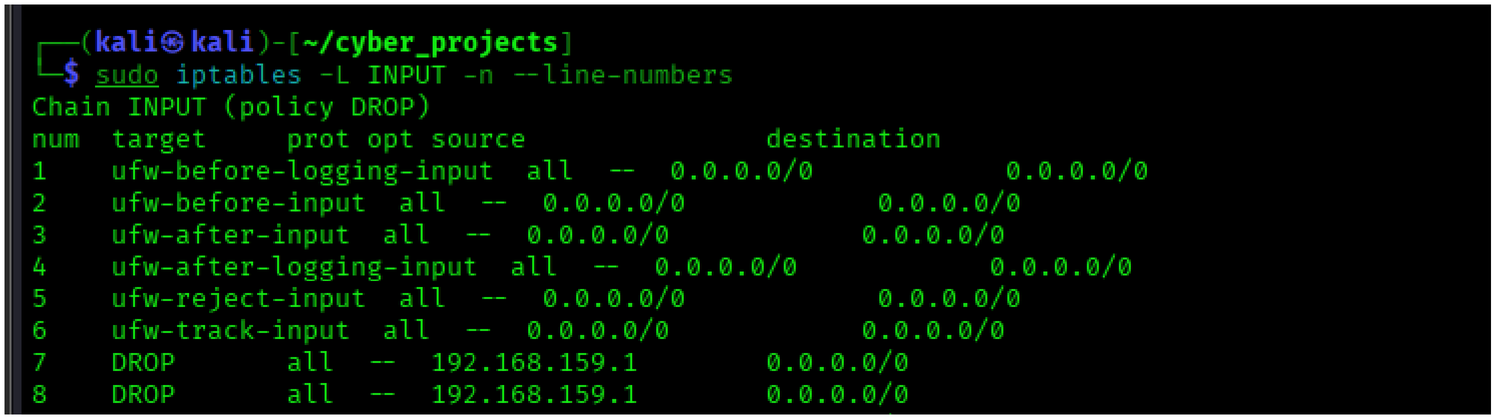
3. **Key Observations:**
    - **Detection:** The script logs the source IP immediately (`[!] ALARM`).
    - **Mitigation:** It automatically executes a kernel command to block the IP (`[+] SUCCESS`).
    - **Efficiency:** Notice the "Idempotency Check" messages. If the attacker tries again, the system intelligently skips adding a duplicate ban rule, saving system resources.

4. Verification (The "Jail" Check)

Prove that the Python script successfully manipulated the Linux Kernel to block the threat.

- **Instructions:**
    1. To confirm the ban is active in the Linux Kernel, run:

```
sudo iptables -L INPUT -n --line-numbers
```


```
┌──(kali㉿kali)-[~/cyber_projects]
└─$ sudo iptables -L INPUT -n --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ufw-before-logging-input  all  --  0.0.0.0/0            0.0.0.0/0
2    ufw-before-input  all  --  0.0.0.0/0            0.0.0.0/0
3    ufw-after-input  all  --  0.0.0.0/0            0.0.0.0/0
4    ufw-after-logging-input  all  --  0.0.0.0/0            0.0.0.0/0
5    ufw-reject-input  all  --  0.0.0.0/0            0.0.0.0/0
6    ufw-track-input  all  --  0.0.0.0/0            0.0.0.0/0
7    DROP       all  --  192.168.159.1        0.0.0.0/0
8    DROP       all  --  192.168.159.1        0.0.0.0/0
```

2. **What to look for:** You will see a specific `DROP` rule targeting the attacker's IP address (e.g., `192.168.159.1`). This confirms the "Active Defense" loop is complete.

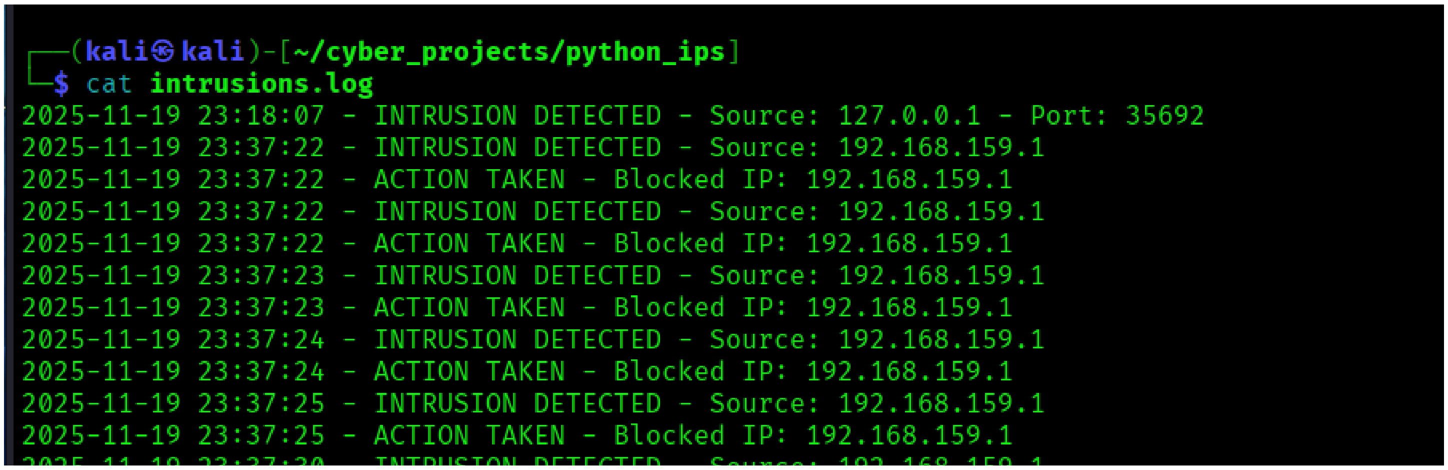
5. Forensic Reporting (Logs & PDF)

Review the evidence collected by the honeypot and generate a professional incident report.

- **Instructions:**
  1. **View Raw Logs:** To see the raw data captured by the honeypot, run:

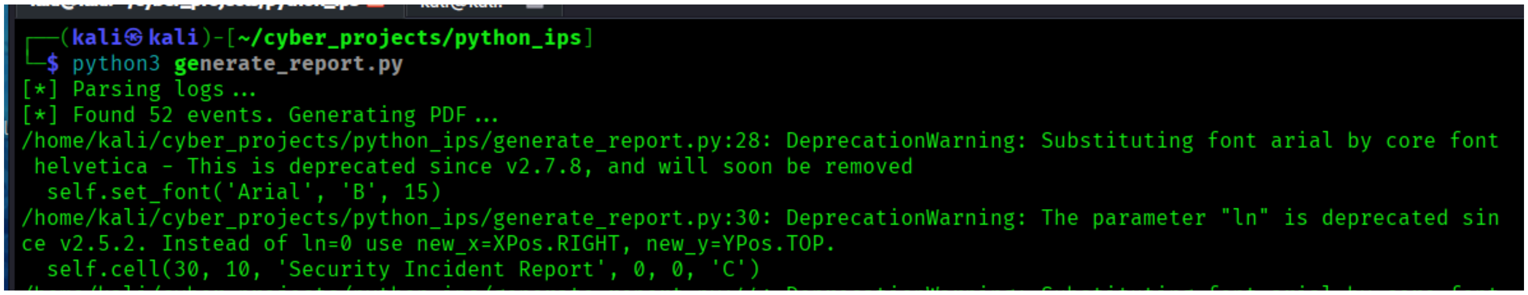
     cat intrusions.log

     Observation: You will see a timestamped entry confirming the "INTRUSION DETECTED" from the attacker's IP and "ACTION TAKEN".

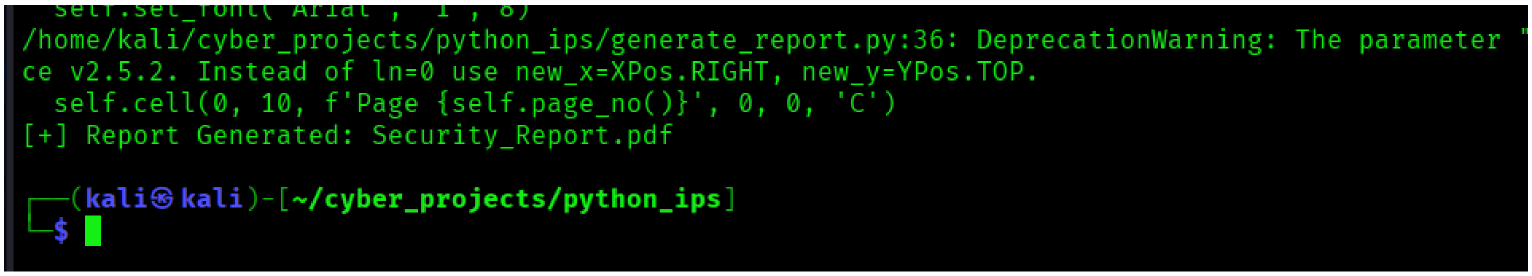

```
┌──(kali㉿kali)-[~/cyber_projects/python_ips]
└─$ cat intrusions.log
2025-11-19 23:18:07 - INTRUSION DETECTED - Source: 127.0.0.1 - Port: 35692
2025-11-19 23:37:22 - INTRUSION DETECTED - Source: 192.168.159.1
2025-11-19 23:37:22 - ACTION TAKEN - Blocked IP: 192.168.159.1
2025-11-19 23:37:22 - INTRUSION DETECTED - Source: 192.168.159.1
2025-11-19 23:37:22 - ACTION TAKEN - Blocked IP: 192.168.159.1
2025-11-19 23:37:23 - INTRUSION DETECTED - Source: 192.168.159.1
2025-11-19 23:37:23 - ACTION TAKEN - Blocked IP: 192.168.159.1
2025-11-19 23:37:24 - INTRUSION DETECTED - Source: 192.168.159.1
2025-11-19 23:37:24 - ACTION TAKEN - Blocked IP: 192.168.159.1
2025-11-19 23:37:25 - INTRUSION DETECTED - Source: 192.168.159.1
2025-11-19 23:37:25 - ACTION TAKEN - Blocked IP: 192.168.159.1
```

  2. **Generate PDF Report:** To convert these logs into a formal document for auditing:

     python3 generate_report.py

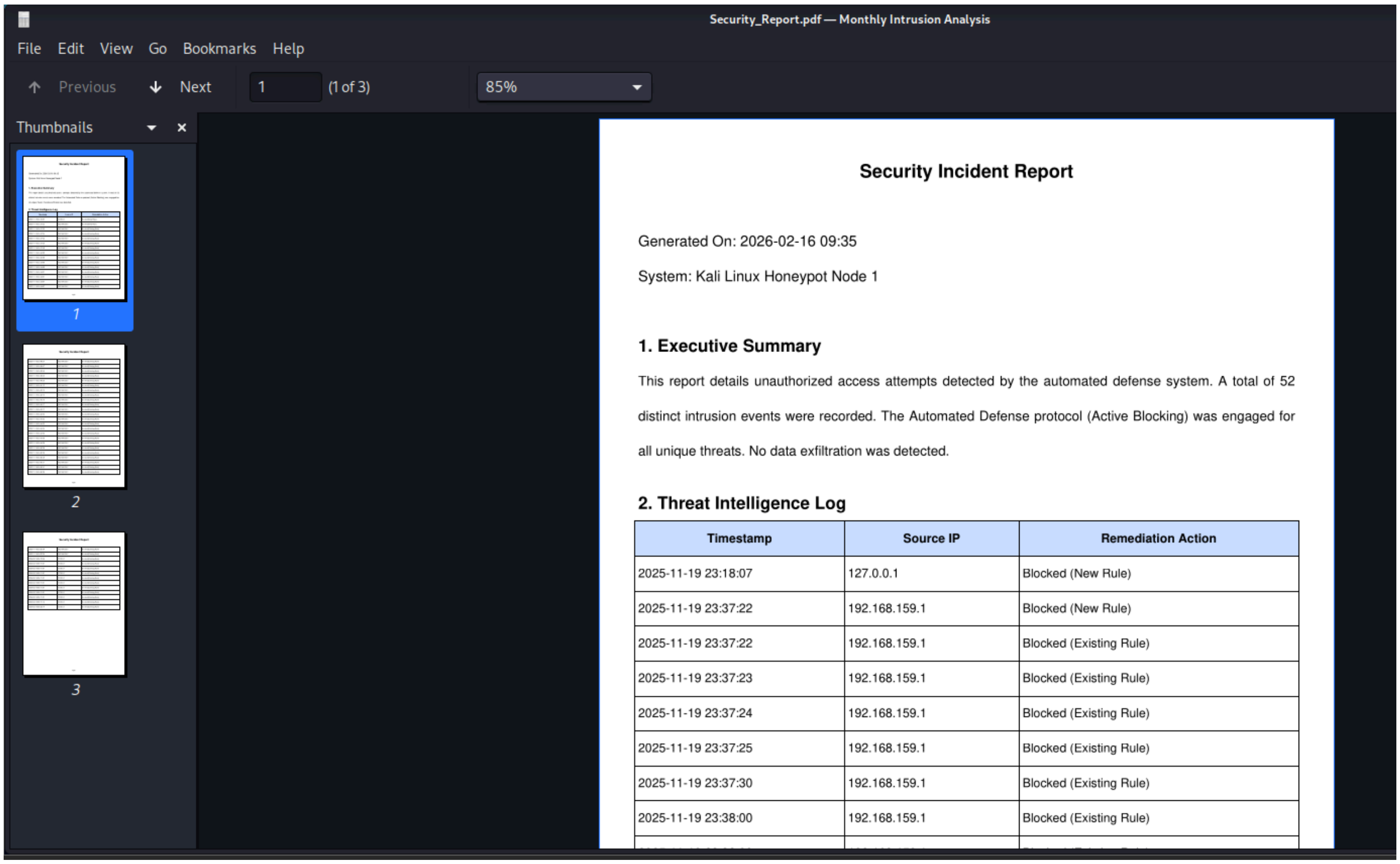


```
┌──(kali㉿kali)-[~/cyber_projects/python_ips]
└─$ python3 generate_report.py
[*] Parsing logs...
[*] Found 52 events. Generating PDF...
/home/kali/cyber_projects/python_ips/generate_report.py:28: DeprecationWarning: Substituting font arial by core font
 helvetica - This is deprecated since v2.7.8, and will soon be removed
  self.set_font('Arial', 'B', 15)
/home/kali/cyber_projects/python_ips/generate_report.py:30: DeprecationWarning: The parameter "ln" is deprecated sin
ce v2.5.2. Instead of ln=0 use new_x=XPos.RIGHT, new_y=YPos.TOP.
  self.cell(30, 10, 'Security Incident Report', 0, 0, 'C')
```



```
  self.set_font('Arial', 'I', 8)
/home/kali/cyber_projects/python_ips/generate_report.py:36: DeprecationWarning: The parameter "
ce v2.5.2. Instead of ln=0 use new_x=XPos.RIGHT, new_y=YPos.TOP.
  self.cell(0, 10, f'Page {self.page_no()}', 0, 0, 'C')
[+] Report Generated: Security_Report.pdf

┌──(kali㉿kali)-[~/cyber_projects/python_ips]
└─$ 
```

3. **Locate the Artifact:**
    - Run `ls` to confirm the file **Security_Report.pdf** has been created.
    - This PDF contains the executive summary and threat intelligence table derived from your logs.


Security_Report.pdf — Monthly Intrusion Analysis
File Edit View Go Bookmarks Help
Previous Next 1 (1 of 3) 85%
Thumbnails
1
2
3

Security Incident Report

Generated On: 2026-02-16 09:35

System: Kali Linux Honeypot Node 1

1. Executive Summary

This report details unauthorized access attempts detected by the automated defense system. A total of 52 distinct intrusion events were recorded. The Automated Defense protocol (Active Blocking) was engaged for all unique threats. No data exfiltration was detected.

2. Threat Intelligence Log

Timestamp	Source IP	Remediation Action
2025-11-19 23:18:07	127.0.0.1	Blocked (New Rule)
2025-11-19 23:37:22	192.168.159.1	Blocked (New Rule)
2025-11-19 23:37:22	192.168.159.1	Blocked (Existing Rule)
2025-11-19 23:37:23	192.168.159.1	Blocked (Existing Rule)
2025-11-19 23:37:24	192.168.159.1	Blocked (Existing Rule)
2025-11-19 23:37:25	192.168.159.1	Blocked (Existing Rule)
2025-11-19 23:37:30	192.168.159.1	Blocked (Existing Rule)
2025-11-19 23:38:00	192.168.159.1	Blocked (Existing Rule)