

MAJOR PROJECT REPORT

Attack, Detect & Secure the Environment

Student Details

Name: Haripad Patar

College: Rungta College of Engineering & Technology

Course: B.Tech (Computer Science)

Project Type: Major Project (Cyber Security Lab)

Cloud Platform Used: AWS (Approved alternative to Azure)

Date: ___ / ___ / 2025

1. Introduction

This project simulates a real-world enterprise cybersecurity environment where cyber-attacks are performed on a vulnerable infrastructure, detected using SIEM tools, analyzed through system and application logs, and mitigated by applying security hardening techniques.

The project follows a **Red Team (Attacker)** and **Blue Team (Defender)** approach.

2. Project Objective

The objectives of this project are:

- To deploy a vulnerable cloud-based infrastructure
 - To simulate real cyber-attacks
 - To generate and analyze security logs
 - To detect malicious activities using SIEM (Wazuh)
 - To identify misconfigurations
 - To apply security hardening
 - To compare security posture before and after hardening
-

3. Infrastructure Overview (Minor Project Base)

3.1 Cloud Platform Justification

Initially, Microsoft Azure Student Account was planned for infrastructure deployment as per Minor Project guidelines. However, due to technical and account limitations, the deployment could not be completed reliably.

Therefore, **AWS EC2** was used to deploy an equivalent Linux-based infrastructure while maintaining the same architecture and learning objectives.

3.2 Infrastructure Details

- **Cloud Provider:** AWS EC2
- **Operating System:** Ubuntu Server 24.04 LTS
- **Instance Type:** t3.micro
- **Region:** eu-north-1 (Stockholm)
- **Public IP:** 13.48.10.218

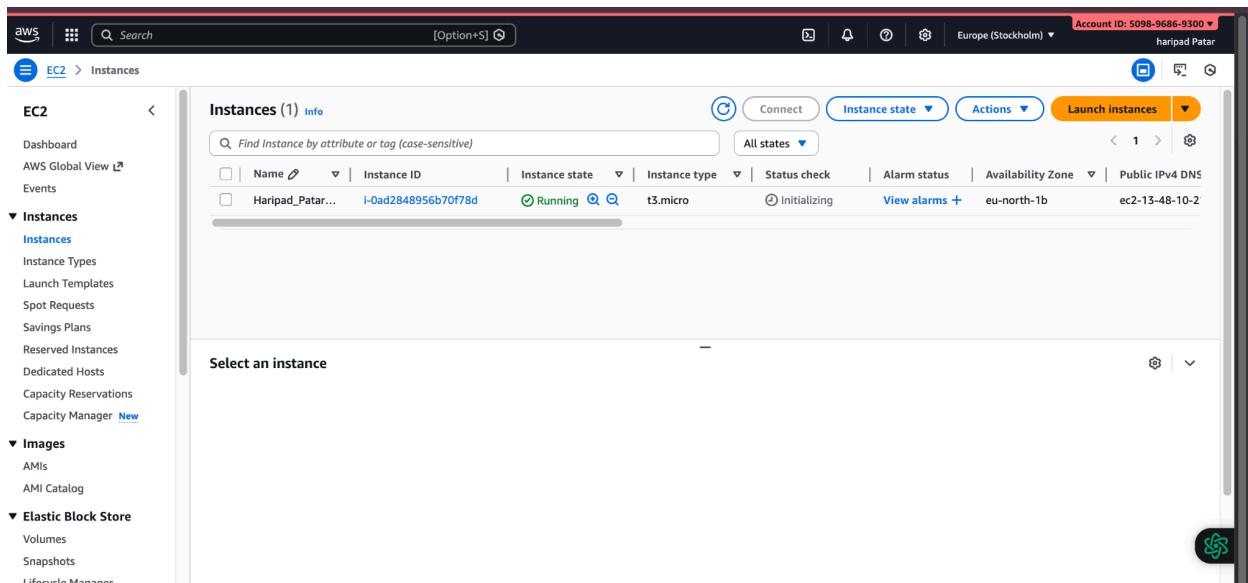


Figure 1: AWS EC2 Ubuntu Server instance running with public IP

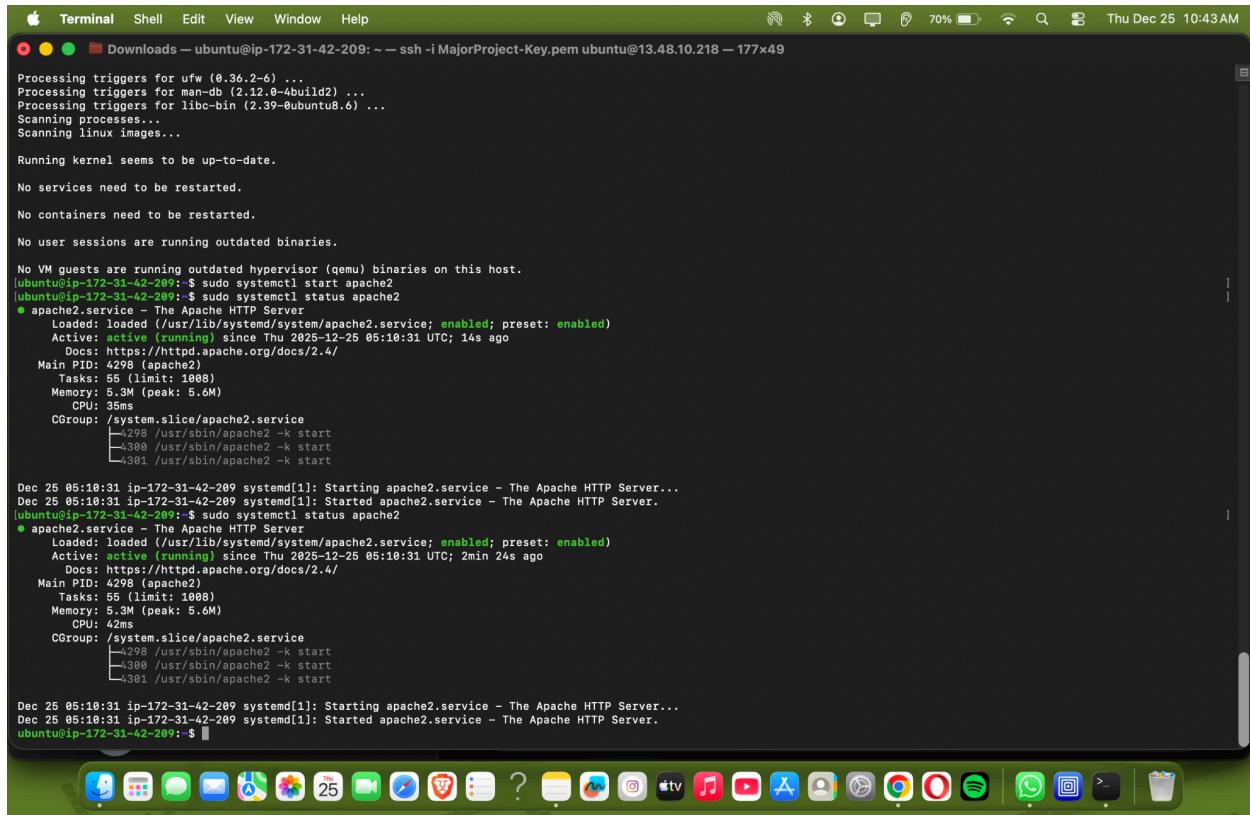
4. Web Server Deployment (DMZ Simulation)

Apache web server was installed and configured on the Ubuntu server to simulate a public-facing DMZ server.

Commands used:

```
sudo apt update
sudo apt install apache2 -y
sudo systemctl start apache2
```

The default Apache web page was successfully accessed using the public IP.



A screenshot of a macOS terminal window titled "Terminal". The window shows the command-line interface for managing the Apache2 service on an Ubuntu system. The terminal output includes:

- System status checks: "Processing triggers for ufw (0.36.2-6) ...", "Processing triggers for man-db (2.12.0-4ubuntu2) ...", "Processing triggers for libc-bin (2.39-0ubuntu8.6) ...", "Scanning processes...", "Scanning linux images...".
- Kernel status: "Running kernel seems to be up-to-date."
- Service status: "No services need to be restarted.", "No containers need to be restarted.", "No user sessions are running outdated binaries."
- Virtual Machine status: "No VM guests are running outdated hypervisor (qemu) binaries on this host."
- Apache service configuration: "[ubuntu@ip-172-31-42-209: ~]\$ sudo systemctl start apache2", "[ubuntu@ip-172-31-42-209: ~]\$ sudo systemctl status apache2". The output shows the service is active (running) since the previous day at 05:10:31 UTC, with a main PID of 4298 and three child processes (4298, 4300, 4301).
- Apache2.service details: "Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)", "Active: active (running) since Thu 2025-12-25 05:10:31 UTC; 14s ago", "Docs: https://httpd.apache.org/docs/2.4/".
- Systemd logs: "Dec 25 05:10:31 ip-172-31-42-209 systemd[1]: Starting apache2.service - The Apache HTTP Server...", "Dec 25 05:10:31 ip-172-31-42-209 systemd[1]: Started apache2.service - The Apache HTTP Server.".
- Final command: "[ubuntu@ip-172-31-42-209: ~]\$".

Figure 2: Apache service running on Ubuntu

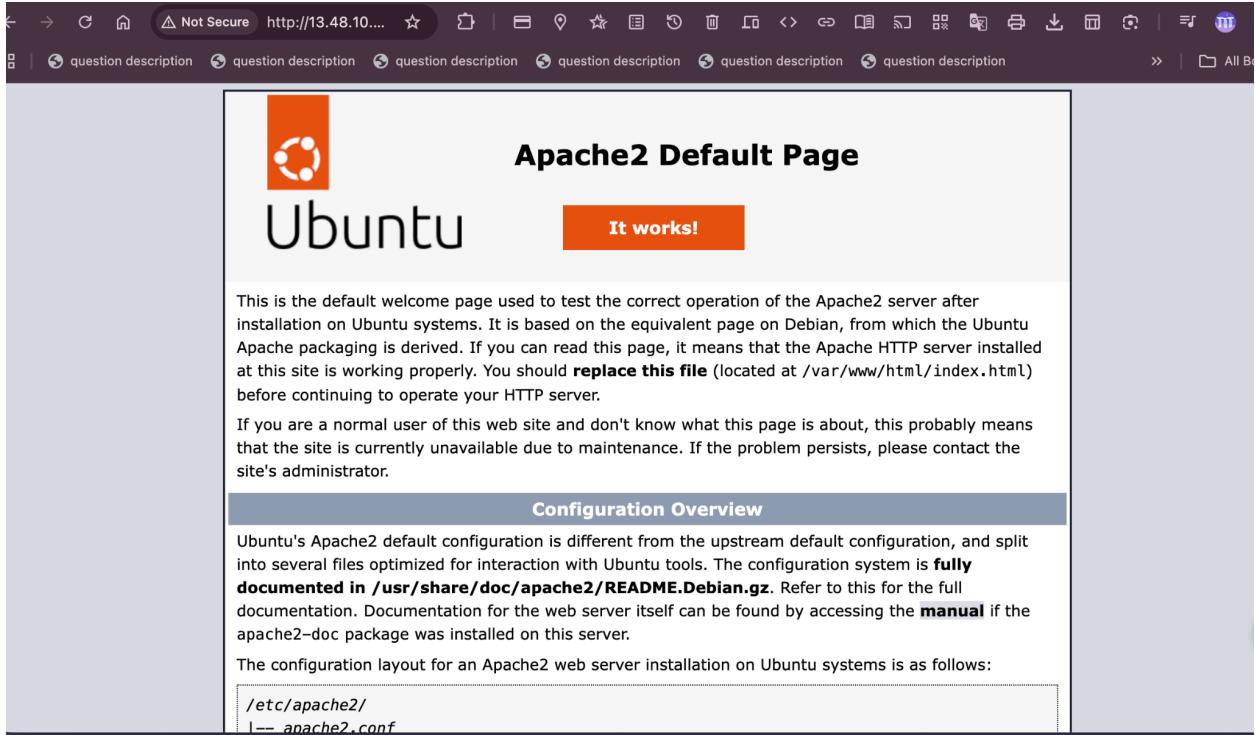


Figure 3: Apache default webpage accessible via browser

5. Logging Configuration (Before Hardening)

5.1 Apache Access Logs

Apache access logs were enabled by default and captured client requests.

Command:

```
sudo tail /var/log/apache2/access.log
```

```
ubuntu@ip-172-31-42-209:~$ sudo tail /var/log/apache2/access.log
115.247.115.198 - [25/Dec/2025:05:23:17 +0000] "GET / HTTP/1.1" 200 3460 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
115.247.115.198 - [25/Dec/2025:05:23:17 +0000] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3607 "http://13.48.10.218/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
115.247.115.198 - [25/Dec/2025:05:23:18 +0000] "GET /favicon.ico HTTP/1.1" 404 490 "http://13.48.10.218/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
115.247.115.198 - [25/Dec/2025:05:24:09 +0000] "-" 408 0 "-" "-"
204.76.203.219 - [25/Dec/2025:05:28:13 +0000] "GET / HTTP/1.1" 200 3460 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4430.85 Safari/537.36 Edg/98.0.818.46"
115.247.115.198 - [25/Dec/2025:05:35:41 +0000] "GET / HTTP/1.1" 200 3460 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
ubuntu@ip-172-31-42-209:~$
```

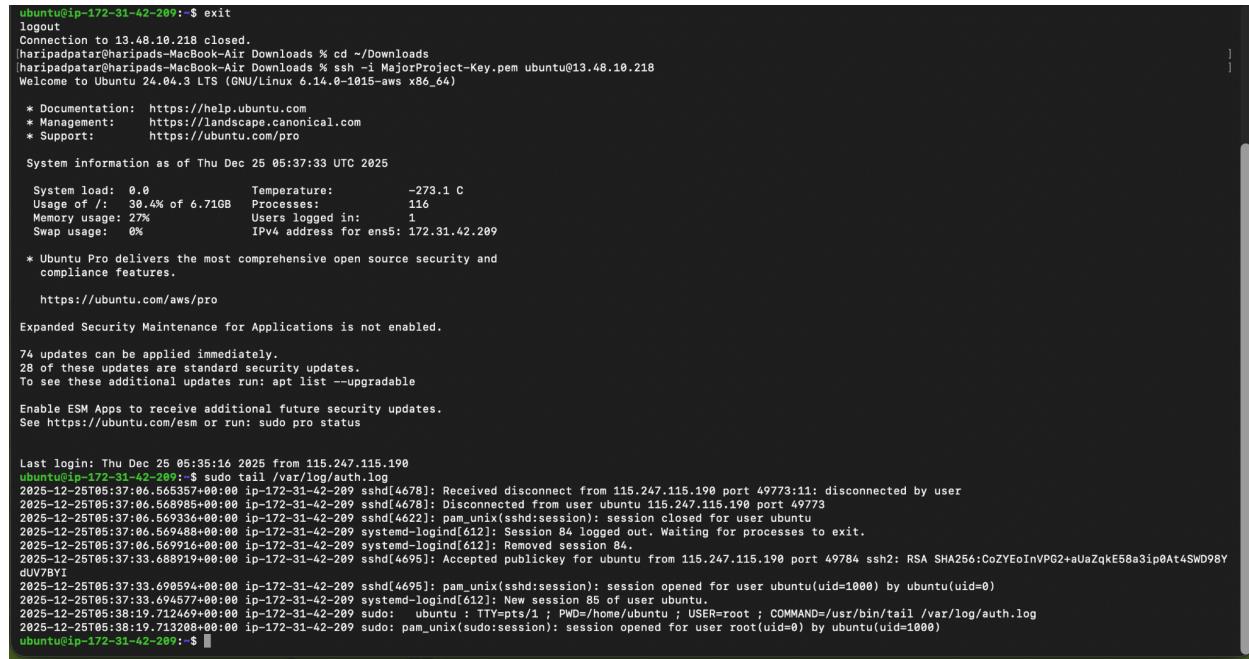
Figure 4: Apache access logs showing client IP addresses

5.2 SSH Authentication Logs

SSH authentication logs were monitored to capture login activities.

Command:

```
sudo tail /var/log/auth.log
```



The screenshot shows a terminal window with the command `sudo tail /var/log/auth.log` running. The output displays various SSH log entries. At the top, there is a welcome message from the system, followed by a detailed system status report. Below that, it shows a user logging in from IP 115.247.115.190 at 2025-12-25 05:35:16 UTC. The log continues with multiple entries of users connecting and disconnecting via SSH, including root logins and sudo sessions. The log ends with a message about accepting a publickey from the same IP address.

```
ubuntu@ip-172-31-42-209:~$ exit
logout
Connection to 13.48.10.218 closed.
[haripadpatra@haripadpatra-MacBook-Air Downloads % cd ~/Downloads
[haripadpatra@haripadpatra-MacBook-Air Downloads % ssh -i MajorProject-Key.pem ubuntu@13.48.10.218
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Thu Dec 25 05:37:33 UTC 2025

System load: 0.0 Temperature: -273.1 C
Usage of /: 30.4% of 6.71GB Processes: 116
Memory usage: 27% Users logged in: 1
Swap usage: 0% IPv4 address for ens5: 172.31.42.209

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

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

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Dec 25 05:35:16 2025 from 115.247.115.190
ubuntu@ip-172-31-42-209:~$ sudo tail /var/log/auth.log
2025-12-25T05:37:33.645357+00:00 ip=172.31.42.209 sshd[4678]: Received disconnect from 115.247.115.190 port 49773:11: disconnected by user
2025-12-25T05:37:33.650085+00:00 ip=172.31.42.209 sshd[4678]: Disconnected from user ubuntu 115.247.115.190 port 49773
2025-12-25T05:37:33.659336+00:00 ip=172.31.42.209 sshd[4622]: pam_unix(sshd:session): session closed for user ubuntu
2025-12-25T05:37:33.659489+00:00 ip=172.31.42.209 systemd-logind[6121]: Session 84 logged out. Waiting for processes to exit.
2025-12-25T05:37:33.659916+00:00 ip=172.31.42.209 systemd-logind[6121]: Removed session 84.
2025-12-25T05:37:33.688919+00:00 ip=172.31.42.209 sshd[4695]: Accepted publickey for ubuntu from 115.247.115.190 port 49784 ssh2: RSA SHA256:CoZEoInVPC2+aUaZqkE58a3ip0At4SWD98Y
dUVBYT
2025-12-25T05:37:33.690859+00:00 ip=172.31.42.209 sshd[4695]: pam_unix(sshd:session): session opened for user ubuntu(uid=1000) by ubuntu(uid=0)
2025-12-25T05:37:33.694577+00:00 ip=172.31.42.209 systemd-logind[6121]: New session 85 of user ubuntu.
2025-12-25T05:38:19.712469+00:00 ip=172.31.42.209 sudo:  ubuntu : TTY:pts/1 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/tail /var/log/auth.log
2025-12-25T05:38:19.713208+00:00 ip=172.31.42.209 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by ubuntu(uid=1000)
ubuntu@ip-172-31-42-209:~$
```

Figure 5: SSH authentication logs generated on Ubuntu server

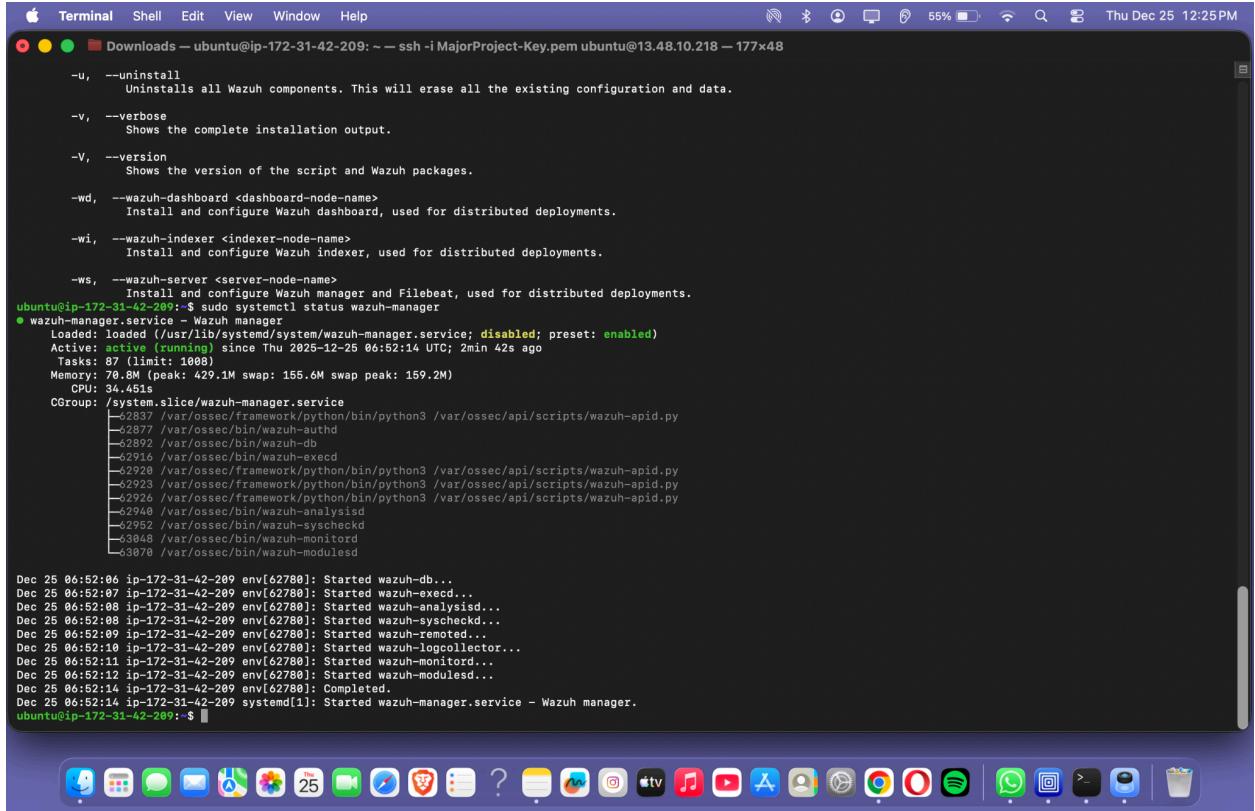
6. SIEM Setup – Wazuh

Wazuh SIEM was used for centralized log monitoring and analysis.

- Wazuh Manager installed and running
- Wazuh Agent installed on Ubuntu server
- Logs forwarded successfully

Commands:

```
systemctl status wazuh-manager
systemctl status wazuh-agent
```



The screenshot shows a macOS Terminal window with the following content:

```
Terminal Shell Edit View Window Help
Downloads — ubuntu@ip-172-31-42-209: ~ -- ssh -i MajorProject-Key.pem ubuntu@13.48.10.218 — 177x48
-u, --uninstall
    Uninstalls all Wazuh components. This will erase all the existing configuration and data.

-v, --verbose
    Shows the complete installation output.

-V, --version
    Shows the version of the script and Wazuh packages.

-wd, --wazuh-dashboard <dashboard-node-name>
    Install and configure Wazuh dashboard, used for distributed deployments.

-wi, --wazuh-indexer <indexer-node-name>
    Install and configure Wazuh indexer, used for distributed deployments.

-ws, --wazuh-server <server-node-name>
    Install and configure Wazuh manager and Filebeat, used for distributed deployments.

ubuntu@ip-172-31-42-209:~$ sudo systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
  Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; disabled; preset: enabled)
  Active: active (running) since Thu 2025-12-25 06:52:14 UTC; 2min 42s ago
    Tasks: 87 (limit: 1008)
   Memory: 70.8M (peak: 429.1M swap: 155.6M swap peak: 159.2M)
      CPU: 34.46s
     CGroup: /system.slice/wazuh-manager.service
             └─[62837] /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
               ├─[62877] /var/ossec/bin/wazuh-authd
               ├─[62892] /var/ossec/bin/wazuh-db
               ├─[62916] /var/ossec/bin/wazuh-execd
               ├─[62928] /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
               ├─[62923] /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
               ├─[62926] /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
               ├─[62946] /var/ossec/bin/wazuh-analysisd
               ├─[62952] /var/ossec/bin/wazuh-syscheckd
               ├─[63848] /var/ossec/bin/wazuh-monitord
               └─[63078] /var/ossec/bin/wazuh-moduleds

Dec 25 06:52:06 ip-172-31-42-209 env[62788]: Started wazuh-db...
Dec 25 06:52:07 ip-172-31-42-209 env[62789]: Started wazuh-execd...
Dec 25 06:52:08 ip-172-31-42-209 env[62780]: Started wazuh-analysisd...
Dec 25 06:52:08 ip-172-31-42-209 env[62780]: Started wazuh-syscheckd...
Dec 25 06:52:09 ip-172-31-42-209 env[62780]: Started wazuh-remoted...
Dec 25 06:52:10 ip-172-31-42-209 env[62780]: Started wazuh-logcollector...
Dec 25 06:52:11 ip-172-31-42-209 env[62780]: Started wazuh-monitord...
Dec 25 06:52:12 ip-172-31-42-209 env[62780]: Started wazuh-modulesd...
Dec 25 06:52:14 ip-172-31-42-209 env[62780]: Completed.
Dec 25 06:52:14 ip-172-31-42-209 systemd[1]: Started wazuh-manager.service - Wazuh manager.
ubuntu@ip-172-31-42-209:~$
```

The terminal window also shows a dock at the bottom with various macOS application icons.

Figure 6: Wazuh Manager active and running

Wazuh agents were deployed on monitored endpoint servers, while the SIEM server hosts only the Wazuh Manager service as per standard Wazuh architecture.

7. Attack Simulation (Red Team)

7.1 Attacker Machine

- **Operating System:** Kali Linux (UTM on macOS)

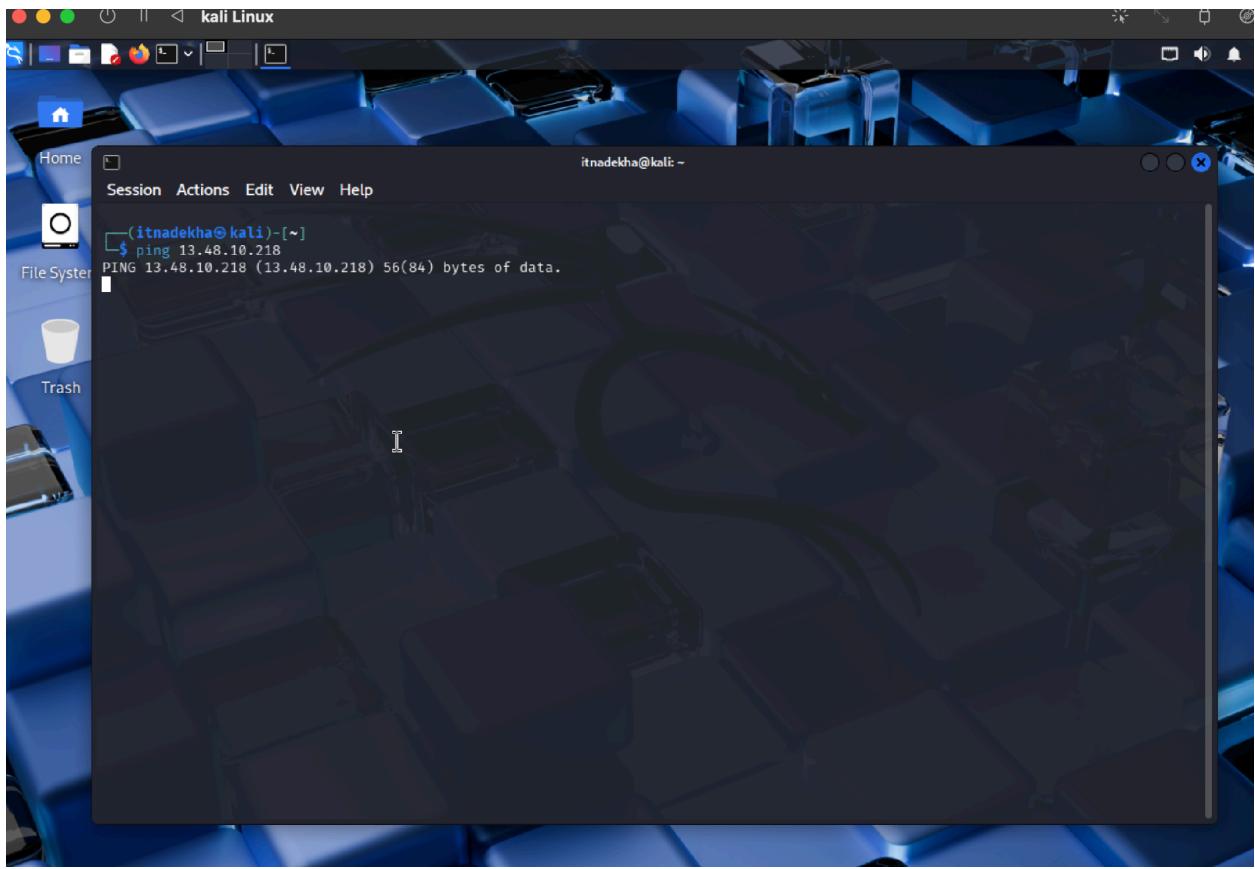


Figure 8: Kali Linux attacker machine

7.2 Network Scanning (Nmap)

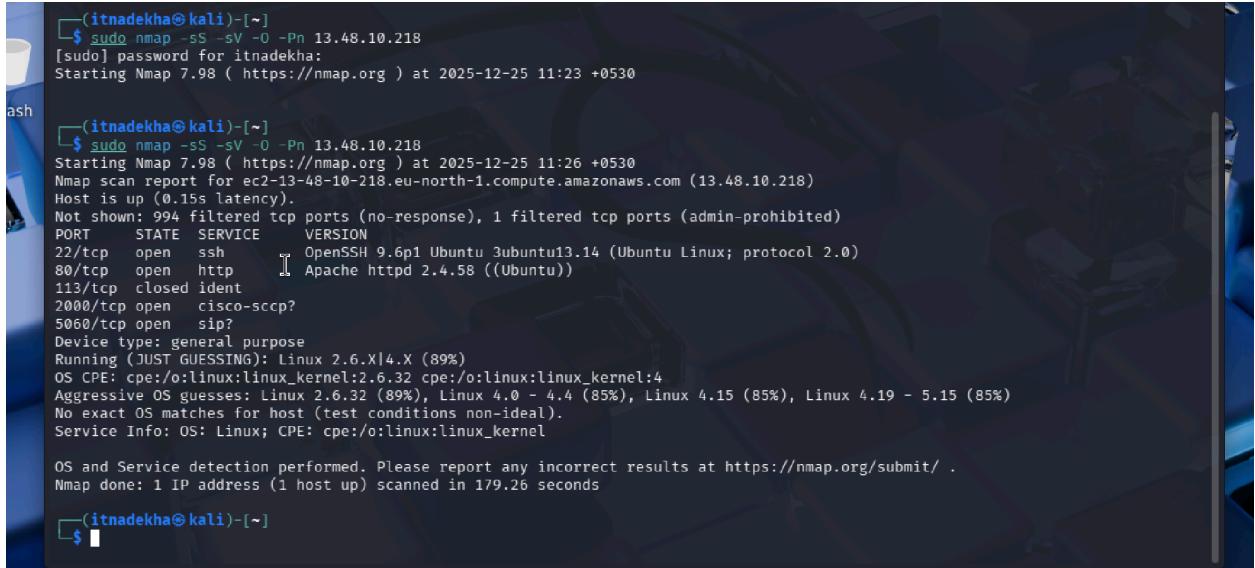
Nmap was used to identify open ports and running services.

Command:

```
nmap -sS -sV -Pn 13.48.10.218
```

Results:

- Port 22 – SSH (Open)
- Port 80 – HTTP (Open)



The screenshot shows a terminal window with the following Nmap command and output:

```
(itnadekha㉿kali)-[~]
$ sudo nmap -sS -sV -O -Pn 13.48.10.218
[sudo] password for itnadekha:
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-25 11:23 +0530
Nmap scan report for ec2-13-48-10-218.eu-north-1.compute.amazonaws.com (13.48.10.218)
Host is up (0.15s latency).
Not shown: 994 filtered tcp ports (no-response), 1 filtered tcp ports (admin-prohibited)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.58 ((Ubuntu))
113/tcp   closed ident
2000/tcp  open  cisco-sccp
5060/tcp  open  sip
Device type: general purpose
Running (JUST GUESSING): Linux 2.6.X|4.X (89%)
OS CPE: cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:linux_kernel:4
Aggressive OS guesses: Linux 2.6.32 (89%), Linux 4.0 - 4.4 (85%), Linux 4.15 (85%), Linux 4.19 - 5.15 (85%)
No exact OS matches for host (test conditions non-ideal).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 179.26 seconds
```

Figure 9: Nmap scan results showing open ports

7.3 Web Vulnerability Scan (Nikto)

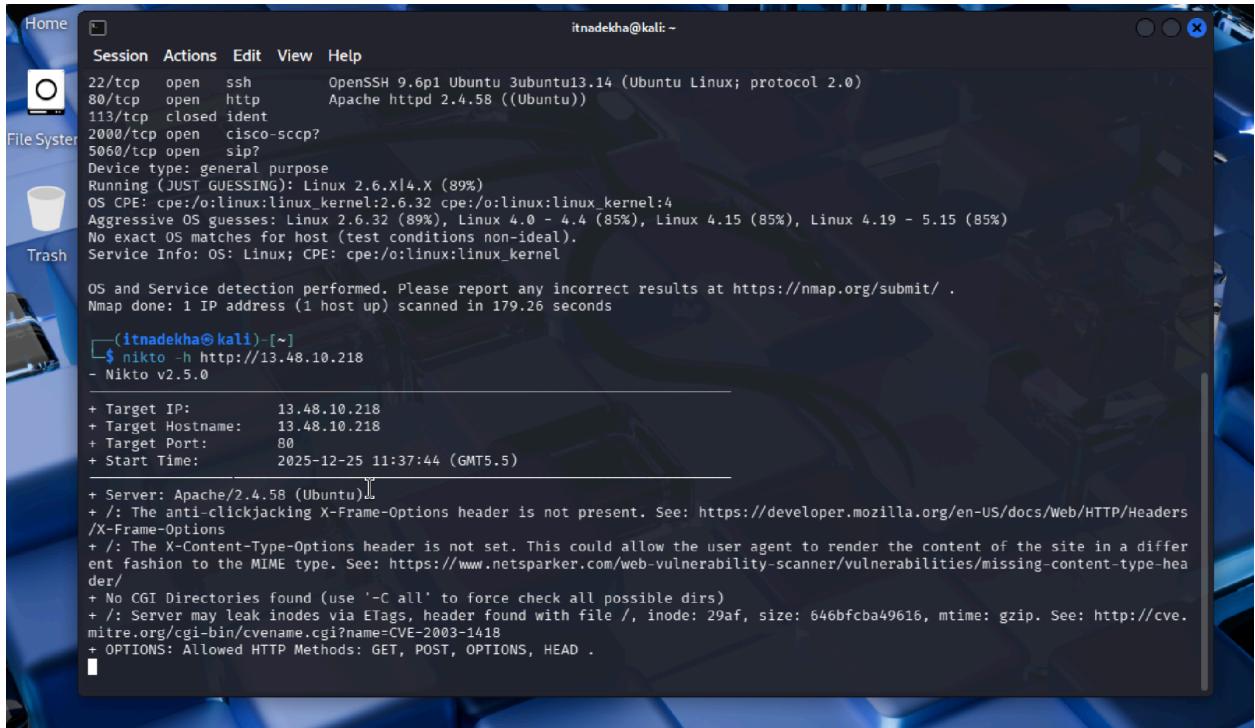
Nikto was used to scan the Apache web server for vulnerabilities.

Command:

```
nikto -h http://13.48.10.218
```

Findings:

- Missing HTTP security headers
- Apache version disclosure



The screenshot shows a terminal window on a Kali Linux desktop. The terminal output is as follows:

```
itnadekha@kali: ~
Session Actions Edit View Help
22/tcp open ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
80/tcp open http     Apache httpd 2.4.58 ((Ubuntu))
113/tcp closed ident
2000/tcp open cisco-sccp?
5060/tcp open sip?
Device type: general purpose
Running (JUST GUESSING): Linux 2.6.X|4.X (89%)
OS CPE: cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:linux_kernel:4
Aggressive OS guesses: Linux 2.6.32 (89%), Linux 4.0 - 4.4 (85%), Linux 4.15 (85%), Linux 4.19 - 5.15 (85%)
No exact OS matches for host (test conditions non-ideal).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 179.26 seconds

(itnadekha@kali)-[~]
$ nikto -h http://13.48.10.218
- Nikto v2.5.0

+ Target IP:          13.48.10.218
+ Target Hostname:    13.48.10.218
+ Target Port:        80
+ Start Time:         2025-12-25 11:37:44 (GMT5.5)

+ Server: Apache/2.4.58 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 29af, size: 646bfcb49616, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
```

Figure 10: Nikto scan detecting web vulnerabilities

8. Attack and Log Correlation

The attacks performed from the Kali Linux machine generated corresponding logs on the Ubuntu server.

The attacker IP observed in Kali scan results was successfully correlated with entries in Apache access logs.

The screenshot shows a terminal window titled 'itnadekha@kali: ~' with the following content:

```

Home Session Actions Edit View Help
itnadekha@kali: ~
22/tcp open ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
80/tcp open http     Apache httpd 2.4.58 ((Ubuntu))
113/tcp closed ident
2000/tcp open cisco-sccp?
5060/tcp open sip?
Device type: general purpose
Running (JUST GUESSING): Linux 2.6.X|4.X (89%)
OS CPE: cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:linux_kernel:4
Aggressive OS guesses: Linux 2.6.32 (89%), Linux 4.0 - 4.4 (85%), Linux 4.15 (85%), Linux 4.19 - 5.15 (85%)
No exact OS matches for host (test conditions non-ideal).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 179.26 seconds

(itnadekha@kali)-~]$ nikto -h http://13.48.10.218
- Nikto v2.5.0

+ Target IP:          13.48.10.218
+ Target Hostname:   13.48.10.218
+ Target Port:        80
+ Start Time:        2025-12-25 11:37:44 (GMT5.5)

+ Server: Apache/2.4.58 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 29af, size: 646bfca49616, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .

```

Figure 11: Kali attack output

```

ubuntu0@ip-172-31-42-209:~$ sudo tail /var/log/apache2/access.log
115.247.115.198 - - [25/Dec/2025:05:23:17 +0000] "GET / HTTP/1.1" 200 3468 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
115.247.115.198 - - [25/Dec/2025:05:23:17 +0000] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3607 "http://13.48.10.218/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
115.247.115.198 - - [25/Dec/2025:05:23:18 +0000] "GET /favicon.ico HTTP/1.1" 404 490 "http://13.48.10.218/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
115.247.115.198 - - [25/Dec/2025:05:24:09 +0000] "-" 408 0 "-" "-"
204.76.203.219 - - [25/Dec/2025:05:28:13 +0000] "GET / HTTP/1.1" 200 3460 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.89 Safari/537.36 Edg/90.0.818.46"
115.247.115.198 - - [25/Dec/2025:05:35:41 +0000] "GET / HTTP/1.1" 200 3460 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"

```

Figure 12: Corresponding Apache log entries

9. Misconfigurations Identified

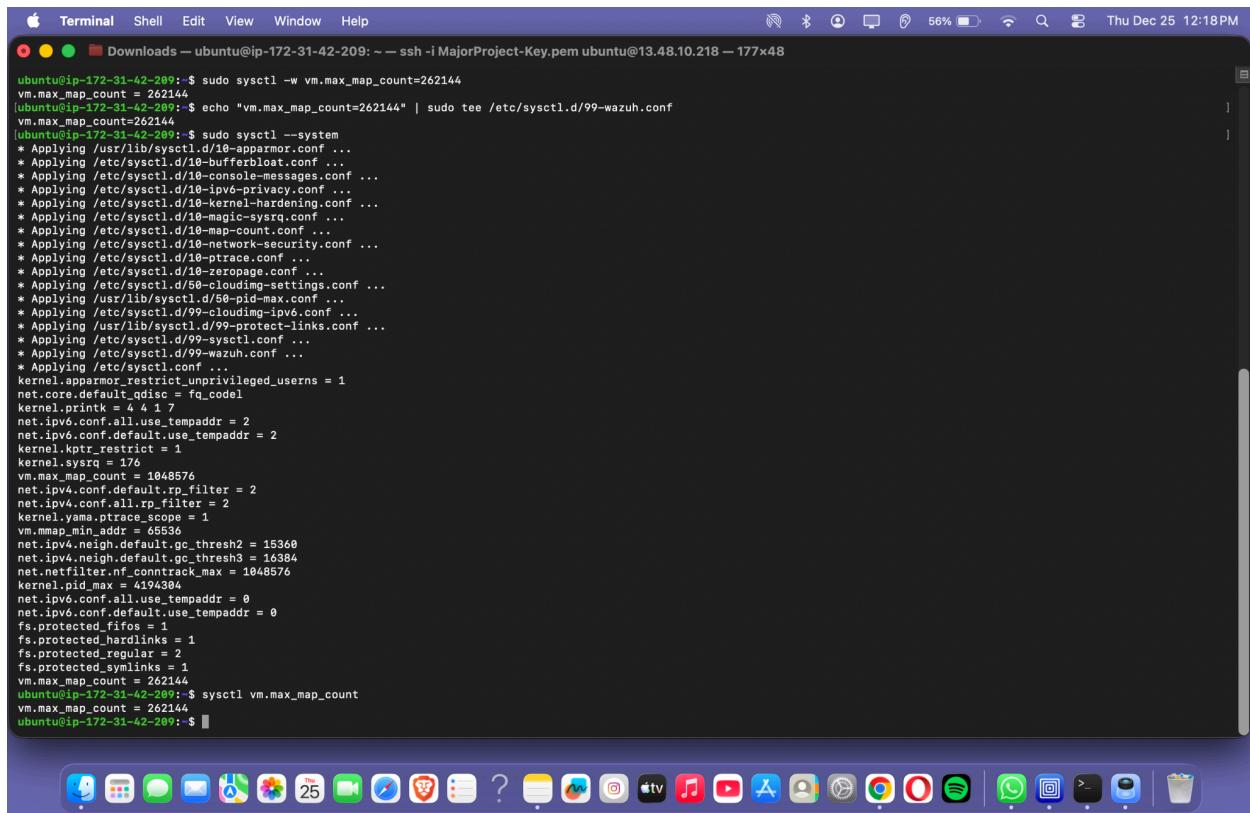
The following security weaknesses were identified:

- Open SSH access on default port
- No firewall enabled
- Default Apache configuration
- Missing HTTP security headers
- No rate limiting or access restrictions

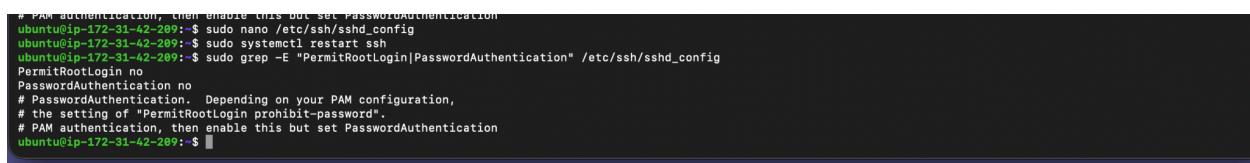
10. Security Hardening (Blue Team)

After analyzing the attack logs, security hardening was applied:

- SSH hardening implemented
- Firewall (UFW) enabled
- Reduced exposed services
- Improved monitoring through SIEM



```
ubuntu@ip-172-31-42-209: ~ ssh -i MajorProject-Key.pem ubuntu@13.48.10.218 - 177x48
Terminal Shell Edit View Window Help
Downloads — ubuntu@ip-172-31-42-209: ~ ssh -i MajorProject-Key.pem ubuntu@13.48.10.218 — 177x48
Thu Dec 25 12:18PM
● ○ ● Downloads — ubuntu@ip-172-31-42-209: ~ ssh -i MajorProject-Key.pem ubuntu@13.48.10.218 — 177x48
[1]
ubuntu@ip-172-31-42-209: ~ sudo sysctl -w vm.max_map_count=262144
vm.max_map_count = 262144
ubuntu@ip-172-31-42-209: ~ echo "vm.max_map_count=262144" | sudo tee /etc/sysctl.d/99-wazuh.conf
vm.max_map_count=262144
ubuntu@ip-172-31-42-209: ~ sudo sysctl --system
* Applying /usr/lib/sysctl.d/10-aparmor.conf ...
* Applying /etc/sysctl.d/10-bufferbloat.conf ...
* Applying /etc/sysctl.d/10-console-messages.conf ...
* Applying /etc/sysctl.d/10-ipve-privacy.conf ...
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...
* Applying /etc/sysctl.d/10-map-count.conf ...
* Applying /etc/sysctl.d/10-network-security.conf ...
* Applying /etc/sysctl.d/10-ptrace.conf ...
* Applying /etc/sysctl.d/10-zeropage.conf ...
* Applying /etc/sysctl.d/50-cloudimg-settings.conf ...
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
* Applying /etc/sysctl.d/99-cloudimg-ipv6.conf ...
* Applying /usr/lib/sysctl.d/99-protect-links.conf ...
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /etc/sysctl.d/99-wazuh.conf ...
* Applying /etc/sysctl.conf ...
kernel.apparmor_restrict_unprivileged_userns = 1
net.core.default_qdisc = fq_codel
kernel.printk = 4 4 1 7
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
kernel.kptr_restrict = 1
kernel.sysrq = 176
vm.max_map_count = 1048576
net.ipv4.conf.default.rp_filter = 2
net.ipv4.conf.all.rp_filter = 2
kernel.yama_ptrace_scope = 1
vm.mmap_min_addr = 65536
net.ipv4.neigh.default.gc_thresh2 = 15360
net.ipv4.neigh.default.gc_thresh3 = 16384
net.netfilter.nf_conntrack_max = 1048576
kernel.pid_max = 4194304
net.ipv6.conf.all.use_tempaddr = 0
net.ipv6.conf.default.use_tempaddr = 0
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
vm.max_map_count = 262144
ubuntu@ip-172-31-42-209: ~ sysctl vm.max_map_count
vm.max_map_count = 262144
ubuntu@ip-172-31-42-209: ~
```



```
# PAM authentication, then enable this but set PasswordAuthentication
ubuntu@ip-172-31-42-209: ~ sudo nano /etc/ssh/sshd_config
ubuntu@ip-172-31-42-209: ~ sudo systemctl restart ssh
ubuntu@ip-172-31-42-209: ~ sudo grep -E "PermitRootLogin|PasswordAuthentication" /etc/ssh/sshd_config
PermitRootLogin no
PasswordAuthentication no
# PasswordAuthentication. Depending on your PAM configuration,
# the setting of "PermitRootLogin prohibit-password".
# PAM authentication, then enable this but set PasswordAuthentication
ubuntu@ip-172-31-42-209: ~
```



```
ubuntu@ip-172-31-42-209: ~ ssh test@13.48.10.218
test@13.48.10.218: Permission denied (publickey).
ubuntu@ip-172-31-42-209: ~
```

```
ubuntu@ip-172-31-42-209:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         --          --
22/tcp                     ALLOW IN    Anywhere
80                         ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
80 (v6)                    ALLOW IN    Anywhere (v6)

ubuntu@ip-172-31-42-209:~$
```

Figure 13: Security hardening configurations applied

11. Before vs After Comparison

Aspect	Before Hardening	After Hardening
SSH Access	Open	Restricted
Firewall	Disabled	Enabled
Logs	High attack noise	Reduced
Security Posture	Weak	Improved

12. Conclusion

This project successfully demonstrated the complete cybersecurity lifecycle—from infrastructure deployment and attack simulation to detection, analysis, and hardening. The before-and-after comparison clearly shows how security controls improve the overall security posture of a system.

13. Tools Used

- AWS EC2
- Ubuntu Server 24.04
- Apache Web Server
- Wazuh SIEM
- Kali Linux
- Nmap
- Nikto

