A Project Report On

Automated SSH Brute-Force Attack Detection & Prevention using Wazuh (SIEM) and Fail2Ban

Submitted in partial fulfillment of the requirement for the award of the degree

MASTER OF SCIENCE (CS & CL)
from
Marwadi University

Academic Year 2025 – 26

Aditya Dewani (92400565060)
Jaykumar Sapra (92400565061)

Internal Guide
Dr.Pankaj Mudholkar

Rajkot-Morbi Road, At &PO :Gauridad, Rajkot 360 003. Gujarat. India.

Marwadi University
Marwadi Chandarana Group
NAAC A+

Faculty of Computer Applications (FoCA)

Certificate

This is to certify that the project work entitled

Automated SSH Brute-Force Attack Detection & Prevention using Wazuh (SIEM) and Fail2Ban

submitted in partial fulfillment of the requirement for the award of the degree of

Master of Science (CS & CL)

of the

Marwadi University

is a result of the bonafide work carried out by

Aditya Dewani (92400565060)
Jaykumar Sapra (92400565061)

during the academic year 2025 – 2026

Faculty Guide	HOD	Dean

DECLARATION

We hereby declare that this project work entitled Automated SSH Brute-Force Attack Detection & Prevention using Wazuh (SIEM) and Fail2Ban is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place:

Date:

Aditya Dewani (92400565060)	Signature:___________
Jaykumar Sapra (92400565061)	Signature:___________

ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement or success one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our guide **Dr.Pankaj Mudholkar,** the person who makes us to follow the right steps during our project work. We express our deep sense of gratitude to for his /her guidance, suggestions and expertise at every stage. A part from that his/her valuable and expertise suggestion during documentation of our report indeed help us a lot.

Thanks to our friend and colleague who have been a source of inspiration and motivation that helped to us during our project work.

We are heartily thankful to the Dean of our department **Dr. R. Sridaran** sir and HoD **Dr. Sunil Bajeja** sir for giving us an opportunity to work over this project and for their end-less and great support to all other people who directly or indirectly supported and help us to fulfil our task.

Aditya Dewani (92400565060)	Signature:___________
Jaykumar Sapra (92400565061)	Signature:___________

CONTENTS

FIGURE INDEX

Chapter 1
Introduction

Secure Shell (SSH) is a widely used protocol for remote server management, but it is highly vulnerable to brute-force attacks, where attackers attempt numerous username and password combinations to gain unauthorized access. Such attacks can compromise system confidentiality, integrity, and availability if not properly mitigated. To address this challenge, this project implements an automated SSH brute-force attack detection and prevention system by integrating Wazuh, an open-source security monitoring platform, with Fail2Ban, a log-parsing intrusion prevention tool. Wazuh provides centralized monitoring, threat detection, and alerting, while Fail2Ban proactively bans malicious IP addresses after repeated failed login attempts. Together, they create a robust defense mechanism that enhances system security by combining real-time detection with automated prevention.

1.1 Objective

The primary objective of this project is to design and implement an Intrusion Detection and Prevention System (IDPS) using Wazuh and Fail2Ban to defend against SSH brute-force attacks. The system will:

1. Real-Time Attack Detection
    - Monitor SSH authentication logs using Wazuh's SIEM capabilities.
    - Implement custom detection rules to identify brute-force patterns (e.g., multiple failed login attempts).
    - Generate security alerts for immediate SOC awareness.
2. Automated Attack Prevention
    - Integrate Fail2Ban to dynamically block malicious IPs via firewall rules.
    - Configure active response in Wazuh to trigger Fail2Ban bans automatically.
    - Reduce manual intervention while improving threat containment speed.
3. Security Visibility & Reporting
    - Visualize attack trends, top attacker IPs, and threat metrics in Wazuh's dashboard.
    - Generate incident reports for compliance and analysis.
    - Demonstrate a fully functional, open-source IDPS for small to medium enterprises.

This solution aims to improve detection accuracy, reduce response time, and lower dependency on commercial security tools while providing hands-on experience in SIEM and automated threat response.

1.2 Problem Definition

SSH brute-force attacks are a major security risk for any internet-exposed server. Attackers use automated tools to repeatedly guess login credentials, which can lead to unauthorized access if weak passwords are in use. Traditional security measures often rely on manual log monitoring, which is inefficient and slow to respond to rapid, large-scale attacks. Without real-time detection and automated blocking, systems remain vulnerable to compromise.

This project addresses the challenge of automating SSH attack detection and prevention using Wazuh (SIEM) and Fail2Ban. The goal is to create a lightweight, open-source solution that monitors SSH logs for suspicious activity, triggers alerts, and automatically blocks malicious IPs—reducing the burden on security teams while improving response times. By implementing this system, organizations can enhance their defenses against one of the most common attack vectors without expensive commercial tools.

1.3 Core Components

1. Wazuh Manager

- Acts as the central monitoring system, collecting and analyzing log data from the monitored hosts.
- Uses pre-defined and custom rules to detect suspicious authentication patterns such as multiple failed login attempts in a short period.
- Triggers an active response when a brute-force attempt is detected, ensuring timely action.

2. Wazuh Agent

- Installed on the target host (e.g., SSH server) to forward system logs, particularly /var/log/auth.log, to the Wazuh Manager.
- Ensures that all authentication events are transmitted in real time for analysis.
- Operates with minimal resource overhead to avoid impacting system performance.

3. Detection Rules (Custom Rules in Wazuh)

- XML-based rules that define thresholds for identifying brute-force attempts (e.g., more than 5 failed logins within 60 seconds).
- Provide flexibility to adjust detection sensitivity and reduce false positives.
- Each detected event is tagged with a unique rule ID and severity level.

4. Fail2Ban

- A host-based intrusion prevention tool that automatically blocks IP addresses generating repeated authentication failures.
- Integrates with the Linux firewall (iptables or nftables) to enforce bans.
- Configured with custom filters and jails to respond to Wazuh alerts and prevent malicious IPs from accessing the SSH service.

5. Active Response Script

- A custom Bash script that acts as the bridge between Wazuh alerts and Fail2Ban actions.
- Executes ban and unban commands based on Wazuh's detection output.
- Example: banning the attacker's IP using fail2ban-client when brute-force attempts are detected.

6. Testing Environment (Simulation Lab)

- Implemented in a virtualized environment using Ubuntu virtual machines.
- Consists of three roles:
    - Wazuh Manager VM – runs the SIEM and detection logic.
    - Target/Agent VM – SSH server monitored for brute-force attempts.
    - Attacker VM – simulates brute-force login attempts using tools like hydra
- Provides a controlled and ethical environment for experimentation and evaluation.

7. Monitoring and Visualization

- Wazuh Dashboard is used to visualize alerts and detection events.
- Displays real-time graphs of failed login attempts, triggered rules, and blocked IP addresses.

1.4 Project Profile

This project focuses on securing Linux-based servers against SSH brute-force attacks, which remain one of the most common methods attackers use to gain unauthorized access. The solution integrates Wazuh, an open-source SIEM platform, with Fail2Ban, a log-based intrusion prevention tool, to provide both real-time detection and automated prevention. Wazuh continuously monitors authentication logs, applies custom rules to detect suspicious activity, and raises alerts when brute-force patterns are observed. In parallel, Fail2Ban dynamically blocks malicious IP addresses by modifying firewall rules, thereby preventing further intrusion attempts. This combined approach offers a robust, automated defense system with minimal manual intervention.

The outcome of this project is a fully functional, open-source intrusion detection and prevention system tailored for SSH security. It will provide real-time monitoring, automated blocking of attackers, and comprehensive reporting, making it a cost-effective security solution for small to medium enterprises.

1.5 Assumptions and Constraints

Assumptions

- The project is developed and tested in a controlled lab environment with virtual machines.
- Attack simulations are limited to SSH brute-force attempts via log injection, not real-world attacks.
- System logs are assumed to be accurate, complete, and untampered.
- Legitimate users will not exceed reasonable failed login attempts (e.g., >5 within 1 minute).

Constraints

- Limited resources (CPU, RAM, storage) in virtual machines may affect scalability results.
- The system depends on Wazuh and Fail2Ban; if either fails, detection/prevention will fail.
- The system focuses only on SSH brute-force attacks, excluding other intrusion methods.
- Trade-off between false positives and detection accuracy — tightening thresholds may block legitimate users, while loosening them may allow attacks.

Chapter 2
Requirement Determination & Analysis

2.1 Requirement Determination

The requirements for the project were identified through multiple approaches:

1 Literature Review

- Analyzed existing studies and reports on intrusion detection and prevention systems.
- Identified SSH brute-force attacks as one of the most prevalent and critical threats to server security.

2 Tool Analysis

- Studied the features of Wazuh and Fail2Ban to evaluate their suitability
- Mapped their capabilities (log monitoring, rule-based detection, automated banning) to the project objectives.

3 Testing Scenarios

- Designed simulated brute-force attacks in a controlled lab environment.
- Used these simulations to define realistic detection thresholds (e.g., number of failed login attempts, timeframe) to balance between accuracy and false positives.

4 User Expectations

- Automation (to reduce manual log monitoring)
- Low false positives (to avoid blocking legitimate users)
- Fast response (to minimize the window of attack).

2.2 Targeted Users

The project is primarily targeted toward:

1. System Administrators – who need automated defense mechanisms to reduce manual monitoring.
2. Security Analysts – who require detection and logging for incident response and auditing.
3. Academic / Research Community – for demonstrating host-based intrusion prevention

methods in lab setups.

4. Organizations running Linux servers – particularly those with exposed SSH services (e.g., cloud VMs, hosting providers).

2.3 Details of Tools and Techniques Used / Implemented

Tools

1. Wazuh
    - Open-source Security Information and Event Management (SIEM).
    - Provides log monitoring, rule-based alerting, and active response capabilities.
    - Used for detection of brute-force attempts by analyzing repeated "Failed password" log entries.

2. Fail2Ban
    - Host-based intrusion prevention system.
    - Parses logs using regex filters to identify malicious activity.
    - Executes firewall rules (iptables/nftables) to block offending IP addresses.
    - Used for prevention (blocking attackers).

3. Bash Scripting
    - Used to create the active response script connecting Wazuh alerts with Fail2Ban actions (ban/unban IP).

4. Linux (Ubuntu 22.04)
    - Operating system for both Wazuh Manager and Agent nodes.
    - Provides SSH service, system logs, and firewall integration.

5. Virtualization (VMware/VirtualBox)
    - Enables safe, isolated lab environment for testing attacks and defenses.

Techniques

1. Log Monitoring & Parsing – Continuous monitoring of /var/log/auth.log to extract failed login attempts.
2. Rule-based Detection – Wazuh rules define thresholds (frequency, timeframe) for identifying brute-force attempts.
3. Automated Active Response – On detection, Wazuh triggers a custom script to communicate with Fail2Ban.
4. IP Banning via Firewall – Fail2Ban enforces bans by updating iptables/nftables to drop packets from attacker IP.
5. Safe Attack Simulation – Instead of real brute-force attacks, log injection (logger command) is used to simulate events.

2.4 Advantages and Limitations of the Used Security Tools

Wazuh

Advantages:

- Centralized log monitoring and alerting.
- Highly customizable rule engine.
- Scalable with multiple agents.
- Integrates well with dashboards (Kibana).

Limitations:

- Can be complex to configure for beginners.
- Rule tuning required to reduce false positives.
- Relies on log integrity; if logs are tampered, detection may fail.

Fail2Ban

Advantages:

- Lightweight and easy to deploy.
- Automatically blocks malicious IPs via firewall.

- Highly customizable filters and jails.
- Reduces system load by stopping repeated brute-force attempts.

Limitations:

- Only detects what matches regex patterns — may miss stealthy attacks.
- Banned IPs can change (attackers often use botnets).
- Overly strict thresholds may block legitimate users (false positives).
- Not designed for distributed or cloud-scale attacks without tuning.

Chapter 3
System Design

3.1Flow Chart

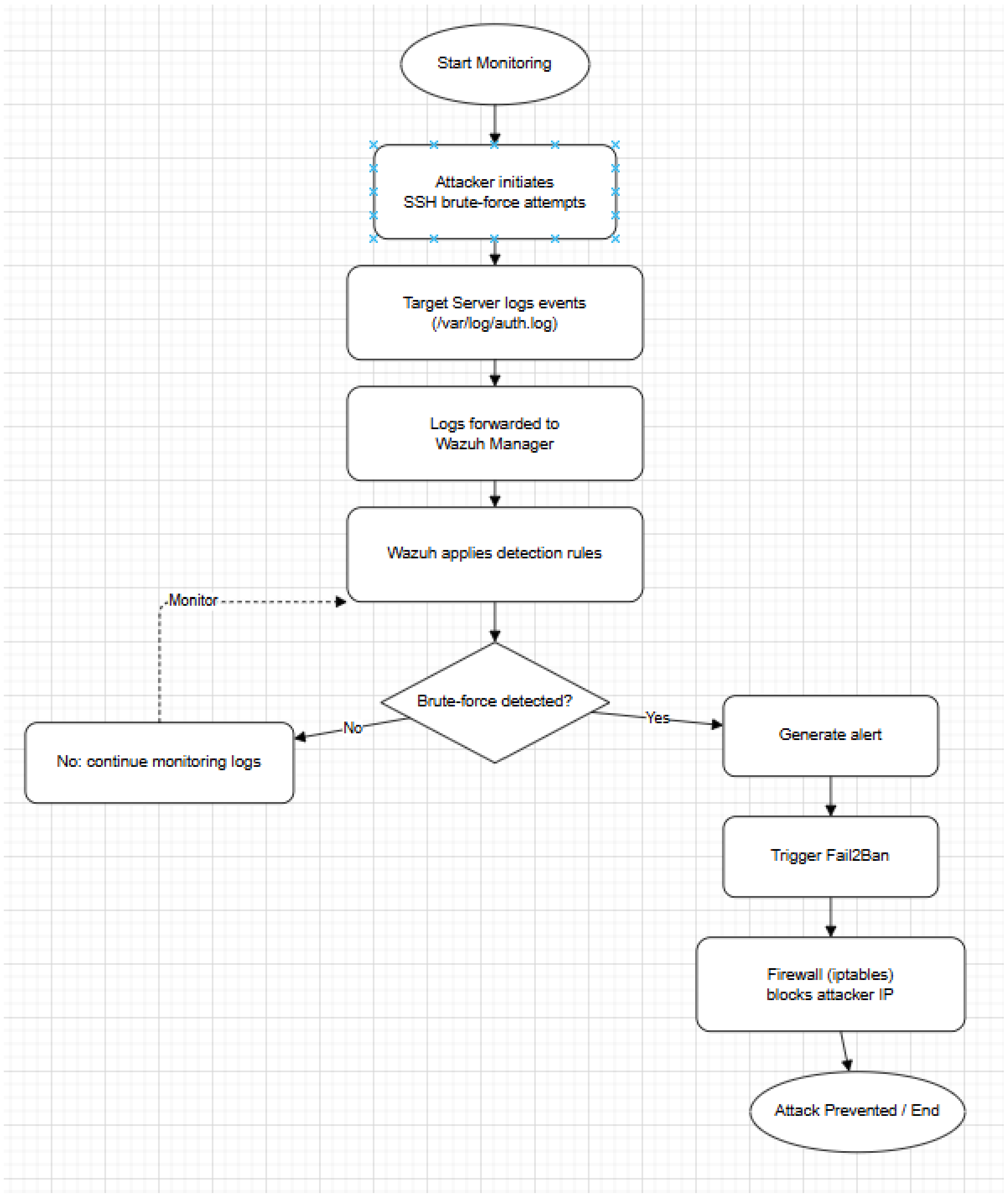

Start Monitoring
Attacker initiates SSH brute-force attempts
Target Server logs events (/var/log/auth.log)
Logs forwarded to Wazuh Manager
Wazuh applies detection rules
Monitor
Brute-force detected?
No
Yes
No: continue monitoring logs
Generate alert
Trigger Fail2Ban
Firewall (iptables) blocks attacker IP
Attack Prevented / End

Figure 3.1 .Flow chart

3.2Architecture Diagram

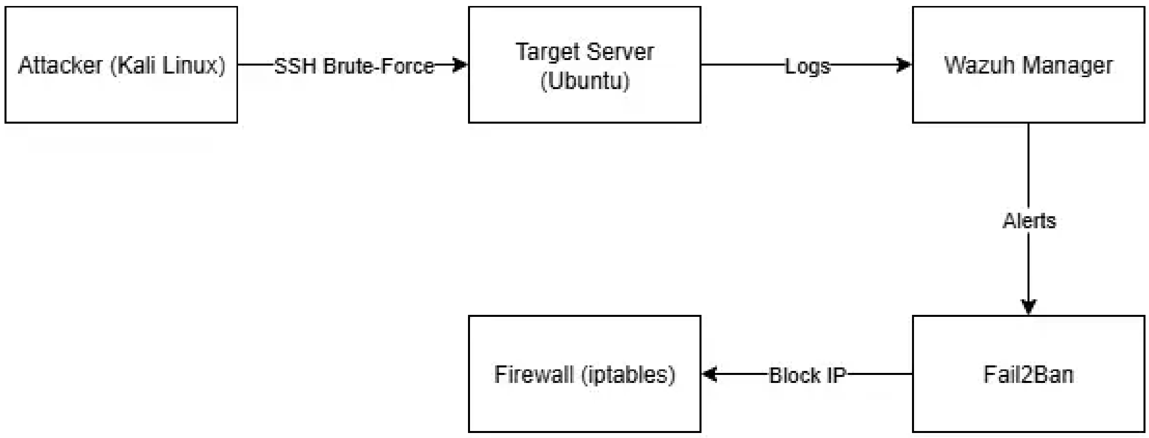


Figure 3.2 Workflow: Automated Detection and Blocking of SSH Attacks

The system architecture follows a log-driven detection and automated response workflow. The process is shown in the flow chart above.

1. Attacker (Kali Linux)
   ○ The attacker initiates an SSH brute-force attack against the target Ubuntu server by repeatedly attempting to guess usernames and passwords.

2. Target Server (Ubuntu)
   ○ All login attempts (successful or failed) are recorded in /var/log/auth.log.
   ○ These logs are continuously monitored by the Wazuh agent installed on the server.

3. Wazuh Manager
   ○ Collects logs from the target server.
   ○ Applies detection rules to identify brute-force patterns (e.g., multiple failed logins from the same IP within a short timeframe).
   ○ Generates an alert when suspicious behavior is detected.

4. Fail2Ban
   ○ Receives the alert from Wazuh's active response mechanism.
   ○ Executes predefined ban actions (using iptables/nftables) to block the

attacker’s IP.

5. Firewall (iptables)
    - Updates firewall rules to block incoming connections from the attacker’s IP address.
    - Prevents further brute-force attempts while reducing system log noise and resource usage.

3.3 Implementation Modules

Modules:

1. Log Monitoring & Detection (Wazuh)
2. Automated Prevention (Fail2Ban)
3. Attack Simulation & Testing (Kali Linux)

Module 1: Log Monitoring & Detection (Wazuh)
This module continuously monitors SSH login attempts on the target server, analyzes logs in real-time, and triggers alerts for brute-force attacks.

Key Components
Log Collection:

- Wazuh agent reads /var/log/auth.log (Ubuntu) or /var/log/secure (CentOS).

Rule-Based Detection:

- Custom rules detect:
- Single failed login (Rule ID: 100100).
- Multiple failures from the same IP (Rule ID: 100101).

Alerting:

- Sends alerts to the Wazuh dashboard for visualization.

Output
Real-time alerts on the Wazuh dashboard when brute-force attacks occur.

Module 2: Automated Prevention (Fail2Ban)
This module automatically blocks malicious IPs that exceed a threshold of failed SSH attempts.

Key Components

Jail Rules:

- Bans IPs after 3 failed attempts in 5 minutes.

Firewall Integration:

- Updates iptables/nftables to drop traffic from banned IPs.

Whitelisting:

- Allows trusted IPs (e.g., admin networks).

Output

Malicious IPs are automatically blocked and appear in fail2ban-client status sshd.

Module 3: Attack Simulation & Testing (Kali Linux)

This module simulates SSH brute-force attacks to validate the detection and prevention system.

Key Components

Tools Used:

- hydra: For password spraying attacks.
- nmap: For scanning open SSH ports.

Test Cases:

- Slow attacks (1 attempt/sec).
- Rapid attacks (10+ attempts/sec).

Output

Confirmation that the system detects and blocks attacks.

Chapter 4
Development

4.1 System Architecture and Environment Setup

Environment Setup

The implementation was carried out using virtual machines on VirtualBox. Each virtual machine was provisioned with sufficient hardware resources to support Wazuh, Fail2Ban, and SSH services without performance degradation.

- Host Machine (Lab Computer):
    - CPU: Intel i5/i7 (Quad-core, 2.0+ GHz)
    - RAM: 16 GB
    - OS: Windows/Linux host
- Virtual Machines:
    - Wazuh Manager VM
        - Ubuntu 22.04 LTS
        - 4 GB RAM, 2 vCPUs, 20 GB storage
        - Installed components: Wazuh Manager, Kibana, Filebeat
    - Target/Agent VM (SSH Server)
        - Ubuntu 22.04 LTS
        - 2 GB RAM, 1 vCPU, 15 GB storage
        - Installed components: OpenSSH, Wazuh Agent, Fail2Ban
    - Attacker VM
        - Kali Linux or Ubuntu 22.04
        - 2 GB RAM, 1 vCPU, 15 GB storage
        - Installed tools: Hydra / log injection scripts

4.2 Configuration and Rule Implementation

Configuration for monitoring the Auth log files

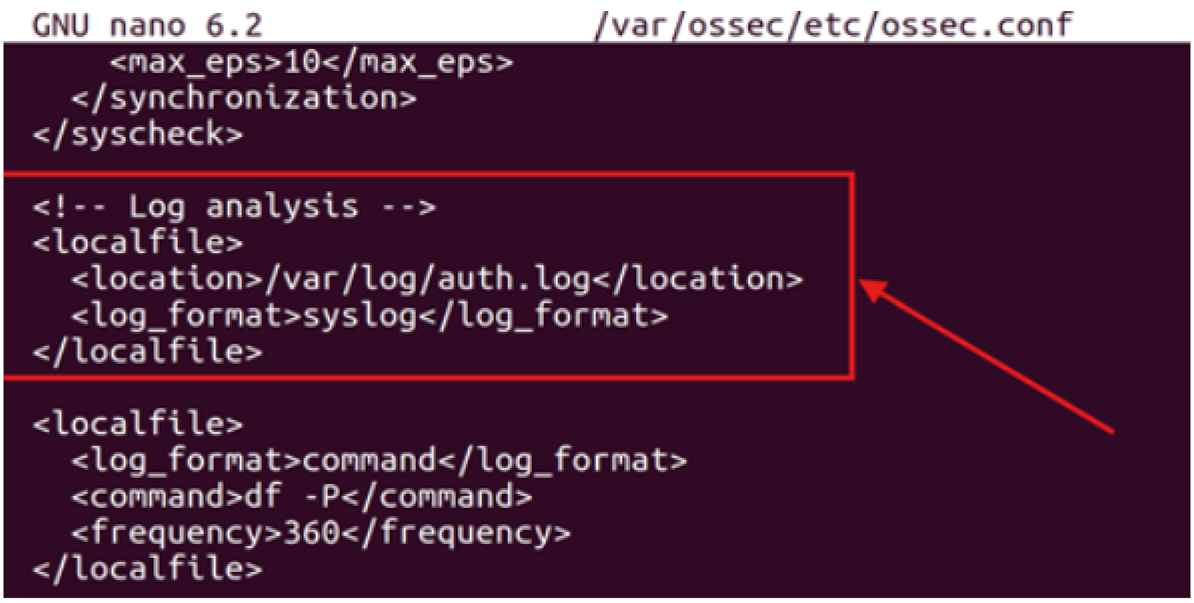
To monitor the auth logs of a client, configure the Wazuh agent ossec.conf file on Linux (Ubuntu) endpoint

- Edit the Wazuh agent configuration file.

```
sudo nano /var/ossec/etc/ossec.conf
```

- Add the following lines between <ossec_config> tags in the config file.

```
<localfile>
  <location>/var/log/auth.log</location>
  <log_format>syslog</log_format>
</localfile>
```



```
  GNU nano 6.2                    /var/ossec/etc/ossec.conf
      <max_eps>10</max_eps>
    </synchronization>
  </syscheck>

  <!-- Log analysis -->
  <localfile>
    <location>/var/log/auth.log</location>
    <log_format>syslog</log_format>
  </localfile>

  <localfile>
    <log_format>command</log_format>
    <command>df -P</command>
    <frequency>360</frequency>
  </localfile>
```

Figure.4.1 Wazuh configure file

- Restart the Wazuh agent.

```
sudo systemctl restart wazuh-agent
```

Setting up an SSH server

- Installing OpenSSH using apt.

```
sudo apt install openssh-server
```

- Enable and start SSH.

```
sudo systemctl enable ssh
sudo systemctl start ssh
```

- Go to the attacker machine and check if SSH is working.

```
ssh user@ip
```

Accept the fingerprint and enter the password.

Configuring wazuh Active response capabilities

We are configuring Wazuh to drop the brute force requests and block the requests for a certain period.

- Log in to the Wazuh dashboard.
- Click on menu -> server management -> settings.

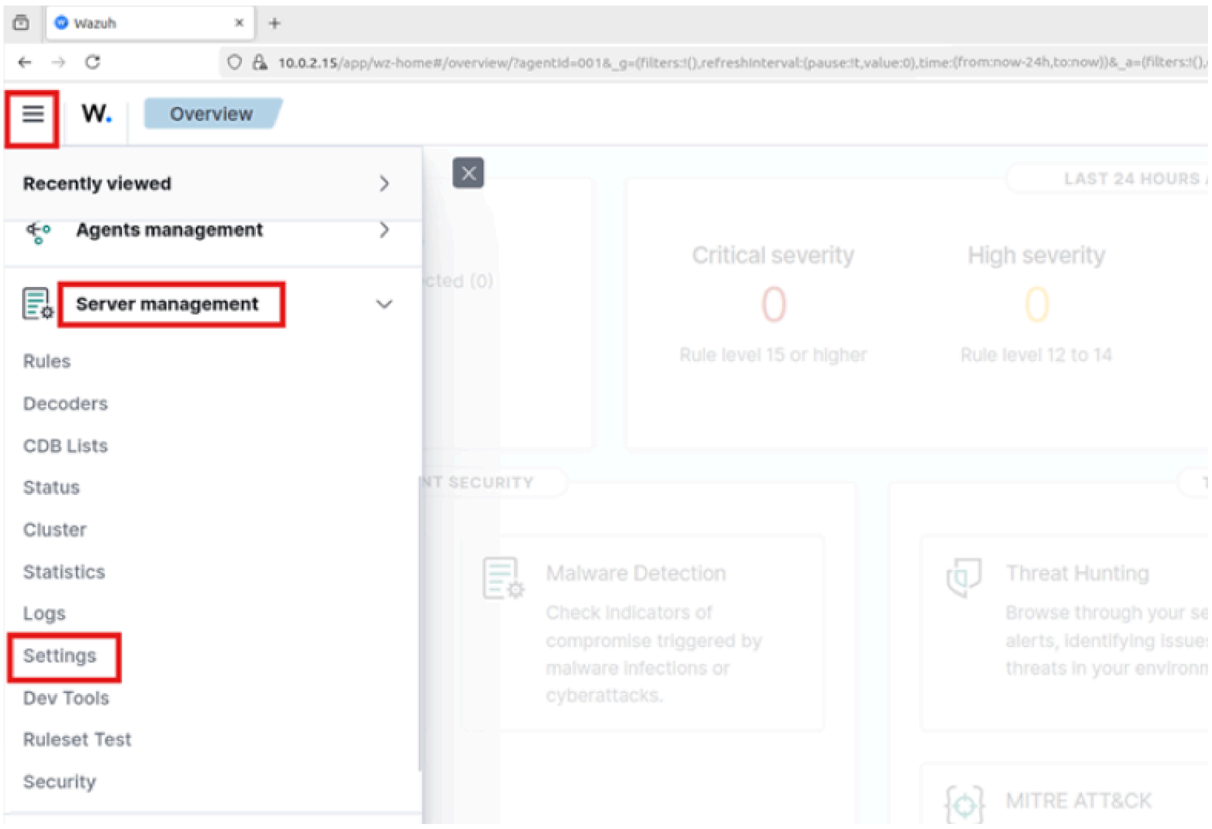
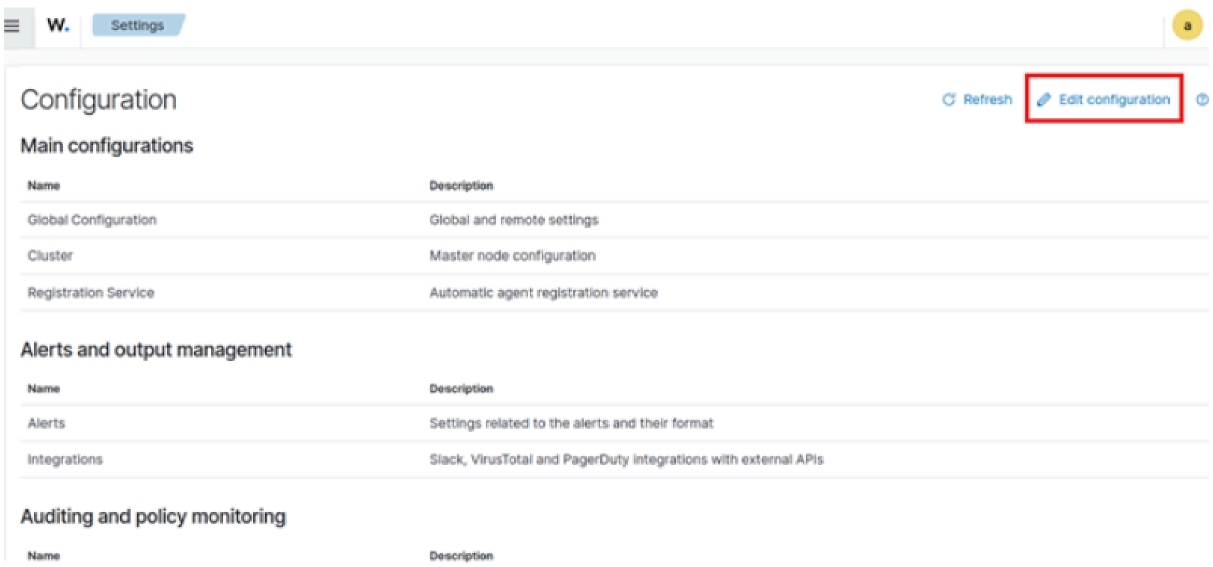


Figure 4.2 Wazuh dashboard

- Click on edit configuration.



Configuration

Main configurations

Name	Description
Global Configuration	Global and remote settings
Cluster	Master node configuration
Registration Service	Automatic agent registration service

Alerts and output management

Name	Description
Alerts	Settings related to the alerts and their format
Integrations	Slack, VirusTotal and PagerDuty integrations with external APIs

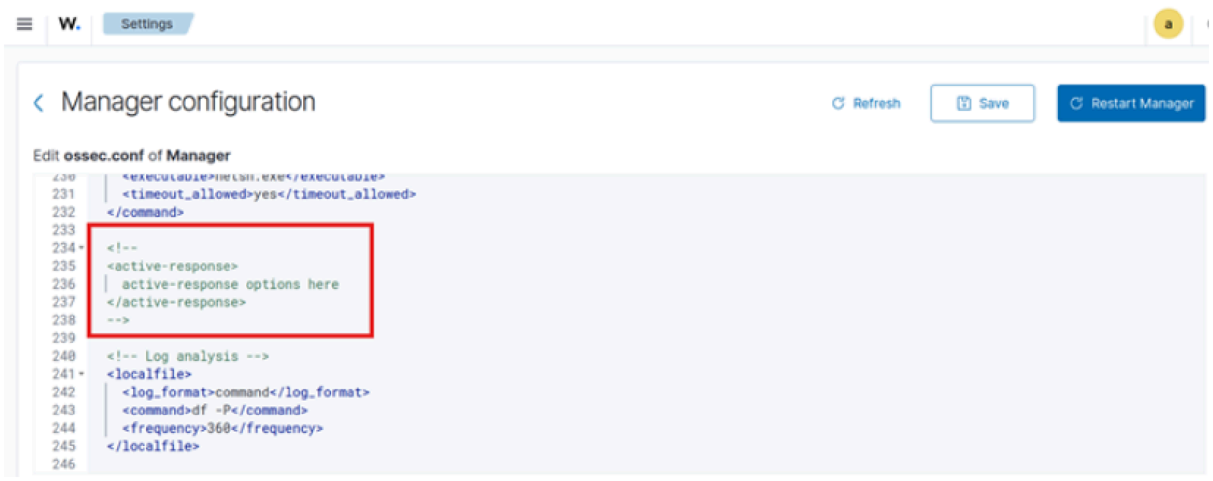
Auditing and policy monitoring

Name	Description

Figure 4.3 Wazuh configure active response

In the manager configuration, look for this block

```
<!--
  <active-response>
                                    active-response options here
  </active-response>
-->
```



Manager configuration

Edit **ossec.conf** of **Manager**

```
    <executable>netsh.exe</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

  <!--
  <active-response>
    active-response options here
  </active-response>
  -->

  <!-- Log analysis -->
  <localfile>
    <log_format>command</log_format>
    <command>df -P</command>
    <frequency>360</frequency>
  </localfile>
```

Figure 4.4 Wazuh configure manager

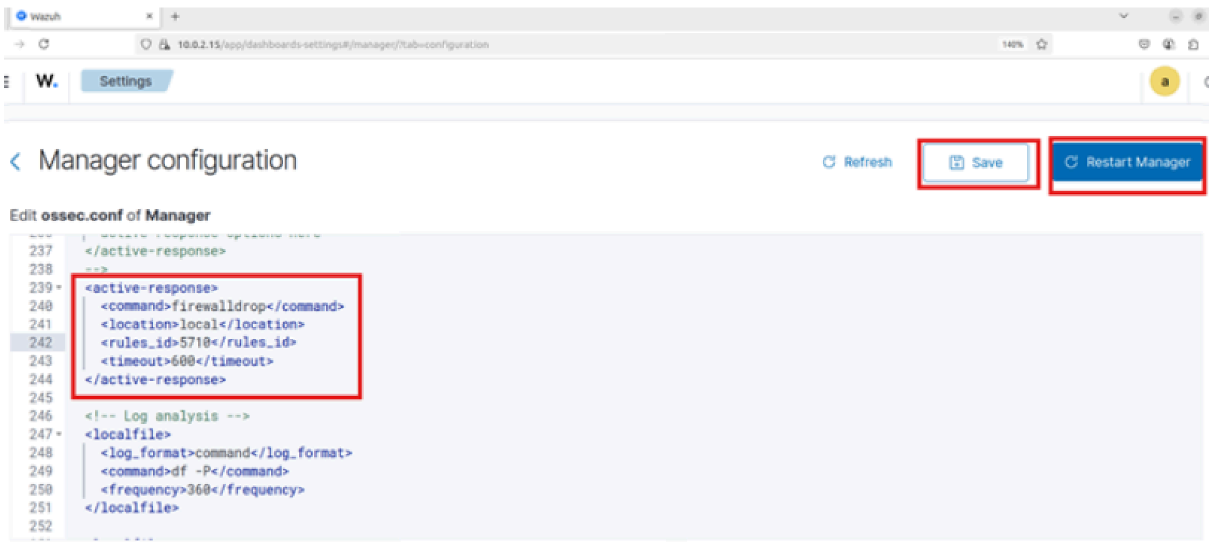
- Add this under the above block to configure the active response.

```
<active-response>
```

```
  <command>firewall-drop</command>

  <location>local</location>

  <rules_id>5710</rules_id>

  <timeout>600</timeout> <!-- Block IP for 10 minutes -->

</active-response>
```

Rule ID 5710: Attempt to log in using a non-existent user

- Click save, and restart the manager.


Wazuh
10.0.2.15/app/dashboards-settings#/manager/?tab=configuration
W.
Settings
Refresh
Save
Restart Manager

Manager configuration

Edit ossec.conf of Manager

```
</active-response>
-->
<active-response>
  <command>firewalldrop</command>
  <location>local</location>
  <rules_id>5710</rules_id>
  <timeout>600</timeout>
</active-response>

<!-- Log analysis -->
<localfile>
  <log_format>command</log_format>
  <command>df -P</command>
  <frequency>360</frequency>
</localfile>
```

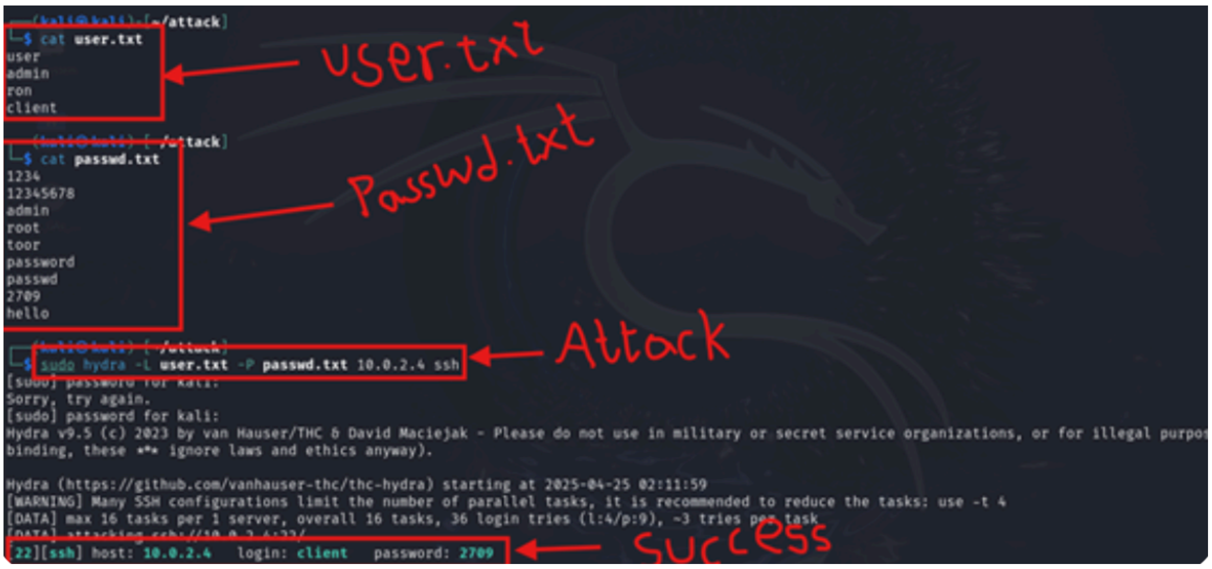
Figure 4.4 Wazuh configure manager

Attack simulation

- Start the Kali machine for attacking.
- Download or create common usernames and password files. Add the username and password of the client machine to the respective files.
- Run the following command to start the attack.

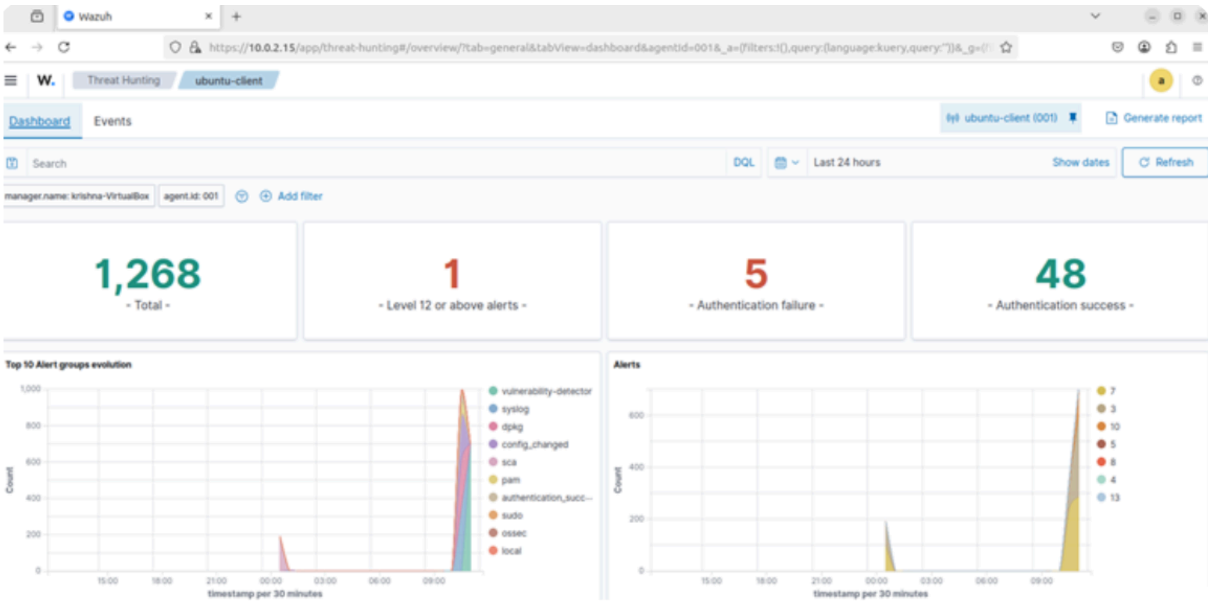
```
sudo hydra -L <USER_LIST.txt> -P <PASSWD_LIST.txt> <IP> ssh
```

The following image shows the usernames and passwords, as well as a successful SSH brute force attack



```
┌──(kali㉿kali)-[~/attack]
└─$ cat user.txt
user
admin
ron
client

┌──(kali㉿kali)-[~/attack]
└─$ cat passwd.txt
1234
12345678
admin
root
toor
password
passwd
2709
hello

┌──(kali㉿kali)-[~/attack]
└─$ sudo hydra -L user.txt -P passwd.txt 10.0.2.4 ssh
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purpos
binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-25 02:11:59
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 36 login tries (l:4/p:9), ~3 tries per task
[DATA] attacking ssh://10.0.2.4:22/
[22][ssh] host: 10.0.2.4   login: client   password: 2709
```

Figure 4.5 SSH Brute Force



Wazuh
https://10.0.2.15/app/threat-hunting#/overview/?tab=general&tabView=dashboard&agentId=001&_a=(filters:!(),query:(language:kuery,query:''))&_g=(
Threat Hunting
ubuntu-client
Dashboard
Events
ubuntu-client (001)
Generate report
Search
DQL
Last 24 hours
Show dates
Refresh
manager.name: krishna-VirtualBox
agent.id: 001
Add filter

1,268
- Total -

1
- Level 12 or above alerts -

5
- Authentication failure -

48
- Authentication success -

Figure 4.6 Wazuh Detect attack

This is the screenshot of the Wazuh dashboard before the attack, which shows 5 authentication failures.

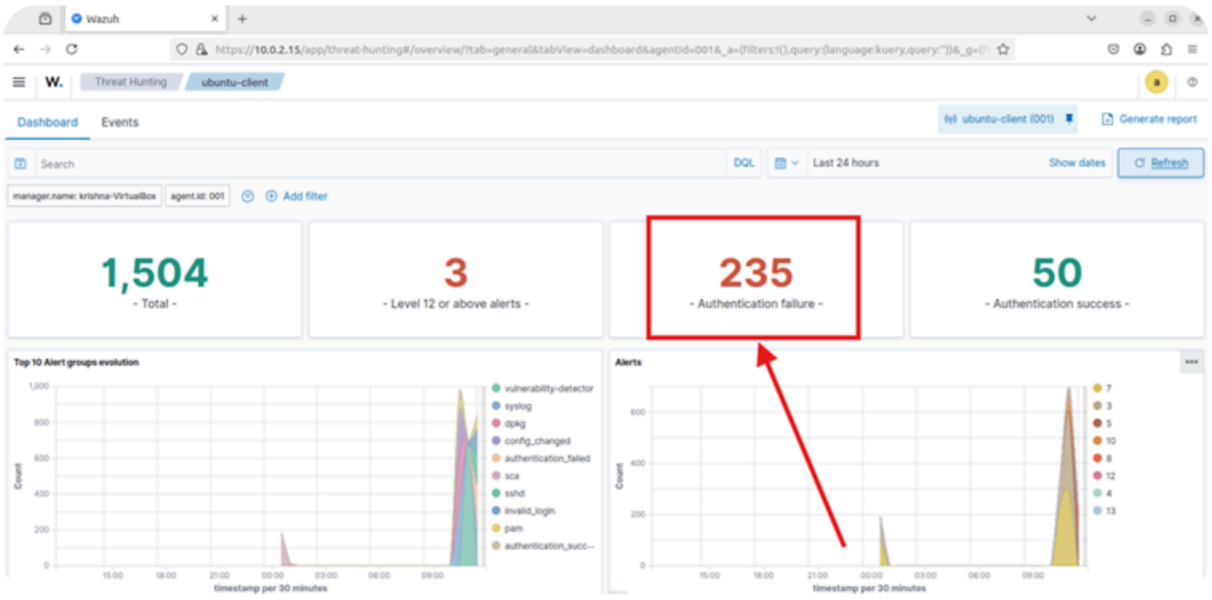


Figure 4.7 Wazuh Detect attack

This is the screenshot of the Wazuh dashboard before the attack, which shows 235 authentication failures.

Now, if you try to log in using SSH, you cannot access for 10 minutes.

Chapter 5
Proposed Enhancements

Implemented system demonstrates effective detection and prevention of SSH brute-force attacks, there are several opportunities for further improvement. One potential enhancement is to extend the system beyond SSH to other protocols that are frequently targeted, such as FTP, SMTP, and HTTP-based authentication services. By broadening the scope, the solution can evolve into a more comprehensive intrusion detection and prevention framework.

Another enhancement involves the integration of external threat intelligence feeds. This would allow the system to block known malicious IP addresses proactively, rather than waiting for an attack to occur. Such integration could significantly reduce exposure time and strengthen the system's overall resilience. In addition, the adoption of machine learning techniques could provide more dynamic and adaptive detection. Unlike static rules, machine learning models can identify unusual login behaviors and evolving attack strategies, further minimizing false positives and missed detections.

Chapter 6
Conclusion

This project successfully designed and implemented an automated detection and prevention mechanism for SSH brute-force attacks using Wazuh and Fail2Ban. Through the integration of Wazuh's rule-based log analysis and Fail2Ban's automated response capabilities, the system was able to detect brute-force attack attempts in real time and mitigate them effectively by banning malicious IP addresses. Testing within a controlled environment demonstrated that the solution achieves its objectives of reducing manual intervention, responding quickly to threats, and minimizing false positives through careful rule tuning.

The work carried out in this project addresses a significant real-world security challenge and provides a practical approach to intrusion detection and prevention at the host level. While the current scope was limited to SSH brute-force attacks, the results validate the feasibility of extending the approach to other protocols and environments.

In conclusion, the project demonstrates that open-source tools like Wazuh and Fail2Ban, when combined with proper configuration and customization, can provide an efficient, scalable, and automated solution to one of the most common security threats faced by modern servers.

Chapter 7
Bibliography

Offline References (Books)

[1] M. Al-Bassam, “Brute Force Attacks: Understanding and Mitigating,” *International Journal of Computer Applications*, 2019.

[2] A. Berton and J. García, “Host-based intrusion detection systems: A survey,” *Journal of Information Security and Applications*, 2021.

[3] Wazuh, “Wazuh Documentation: Security monitoring and threat detection,” Wazuh, 2023.

[4] S. Kumar and A. Gaur, “Enhancing system security using automated detection of brute-force SSH attacks,” *International Journal of Advanced Computer Science and Applications*, 2020.

[5] OWASP Foundation, “Authentication Cheat Sheet,” OWASP, 2023.

[6] SANS Institute, “Detecting and Defending Against SSH Brute Force Attacks,” SANS Whitepaper, 2022.

[7] MITRE, “ATT&CK Framework – Brute Force Technique (T1110),” MITRE ATT&CK, 2023.

[8] J. Allen, “Automated log analysis for intrusion detection,” *ACM SIGSAC Review*, 2020.

[9] DigitalOcean, “How To Protect SSH with Fail2Ban on Ubuntu,” DigitalOcean Community Tutorials, 2022.

Online References

Wazuh Documentation – Blocking SSH Brute-Force Attack with Active Response

URL:

https://documentation.wazuh.com/current/user-manual/capabilities/active-response/ar-use-cases/blocking-ssh-brute-force.html

Wazuh Brute-Force Detection Rules.

URL: https://documentation.wazuh.com/current/ruleset/ruleset-s005-ssh.html

Explains how Wazuh’s Active Response can be configured to block SSH brute-force attempts using scripts like firewall-drop.

URL: https://documentation.wazuh.com/current/proof-of-concept-guide/detect-brute-force-attack.html

Red Hat. (2022). *How to protect SSH with Fail2Ban.* Retrieved from https://www.redhat.com/sysadmin/secure-ssh-fail2ban

Shalaginov, A., Østby, J., & Huseby, A. (2021). *Intrusion detection and prevention systems: A survey of machine learning and rule-based techniques.* Retrieved from https://arxiv.org/abs/2106.00821

Ubuntu Community Help Wiki. (2023). *OpenSSH server security.* Retrieved from https://help.ubuntu.com/community/SSH