My Wazuh Monitoring & Security Journey by Scofield Lori Tosan

This document captures my hands-on journey exploring and setting up Wazuh, an open-source security monitoring platform. As a curious and self-motivated Computer Science student, I dove into the world of system monitoring, detection rules, and log analysis—without prior experience. It wasn't all smooth sailing, but that's exactly what made it worth documenting.

Overview: Brute Force Attack Simulation with Kali Linux & Monitoring with Wazuh OVA

As part of my cybersecurity learning journey, I conducted a **simulated brute force attack** using **Kali Linux** and monitored the activity using **Wazuh**, a powerful open-source security platform. This exercise helped me understand both offensive techniques and defensive monitoring in real time.

Tools Used

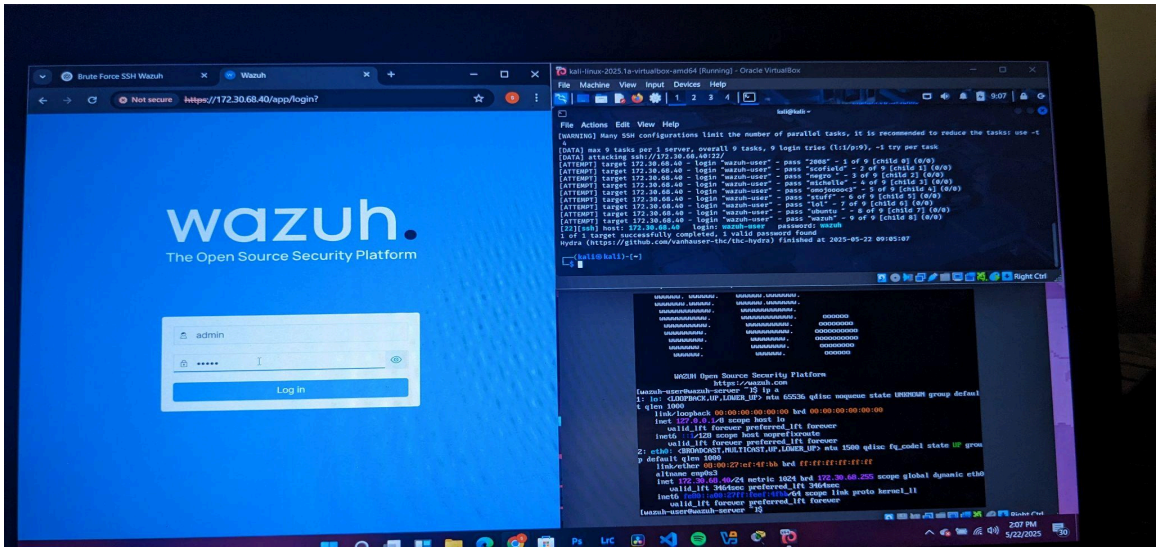
- **Kali Linux** (Attacker)
- **Wazuh OVA** (Manager + Web Interface)
- **Linux target system or SSH service** (Victim/Target)
- **VirtualBox** (for running VMs)

Step-by-Step Summary

1. Setting Up Wazuh OVA

- I downloaded and imported the official **Wazuh OVA file** into VirtualBox.
- After booting it up, I configured the **web interface** (accessible via browser on `https://localhost:5601`) and ensured the **Wazuh manager** was running.
- This OVA acts as the central monitoring hub, including the manager, Elasticsearch, and Kibana.

- **Photographic evidence**



2. Installing Wazuh Agent on Kali Linux

To simulate a monitored system, I installed the **Wazuh agent** directly on Kali Linux:

bash

CopyEdit

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo gpg
--dearmor -o /usr/share/keyrings/wazuh-archive-keyring.gpg

echo "deb
[signed-by=/usr/share/keyrings/wazuh-archive-keyring.gpg]
https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee
/etc/apt/sources.list.d/wazuh.list

sudo apt update && sudo apt install wazuh-agent
```

- I registered Kali as an **agent** in the Wazuh dashboard and configured it with the manager's IP.

- Then I edited the config file (`/var/ossec/etc/ossec.conf`) to ensure it was pointing to the right manager and logging correctly.

3. Simulating the Brute Force Attack :

Using Hydra, a password-cracking tool in Kali, I simulated a brute-force attack against an SSH service:

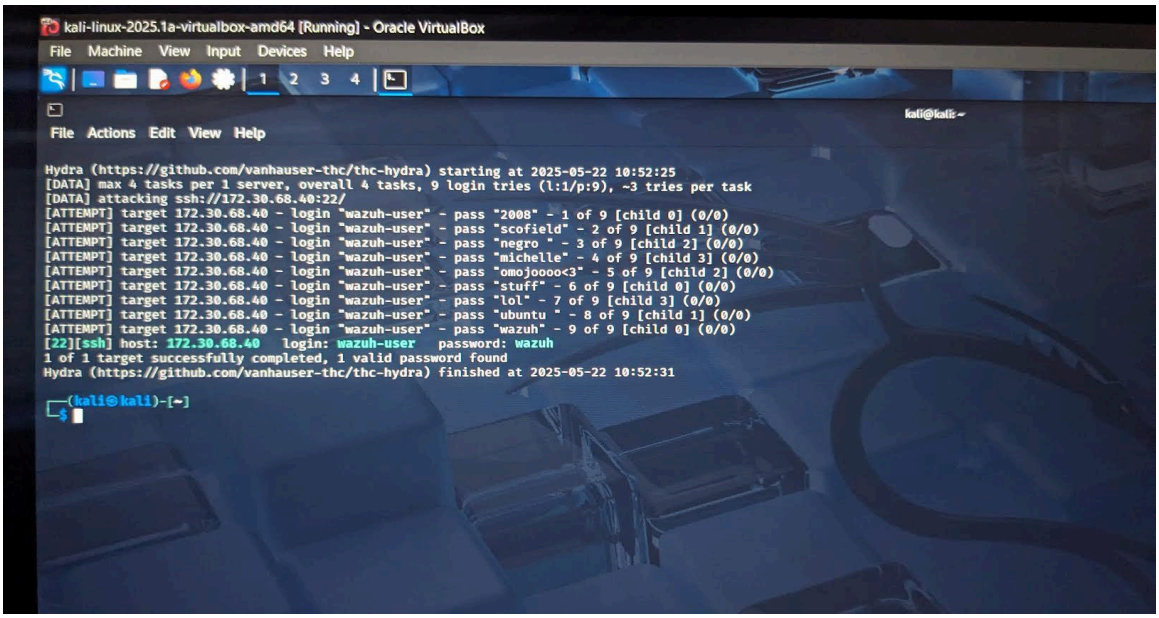
bash

CopyEdit

```
hydra -l wazuh-user -P /usr/share/wordlists/past.txt
ssh://172.30.68.18
```

- This simulated a malicious actor trying to guess SSH credentials.
- It generated multiple failed login attempts in the logs.

Photographic evidence

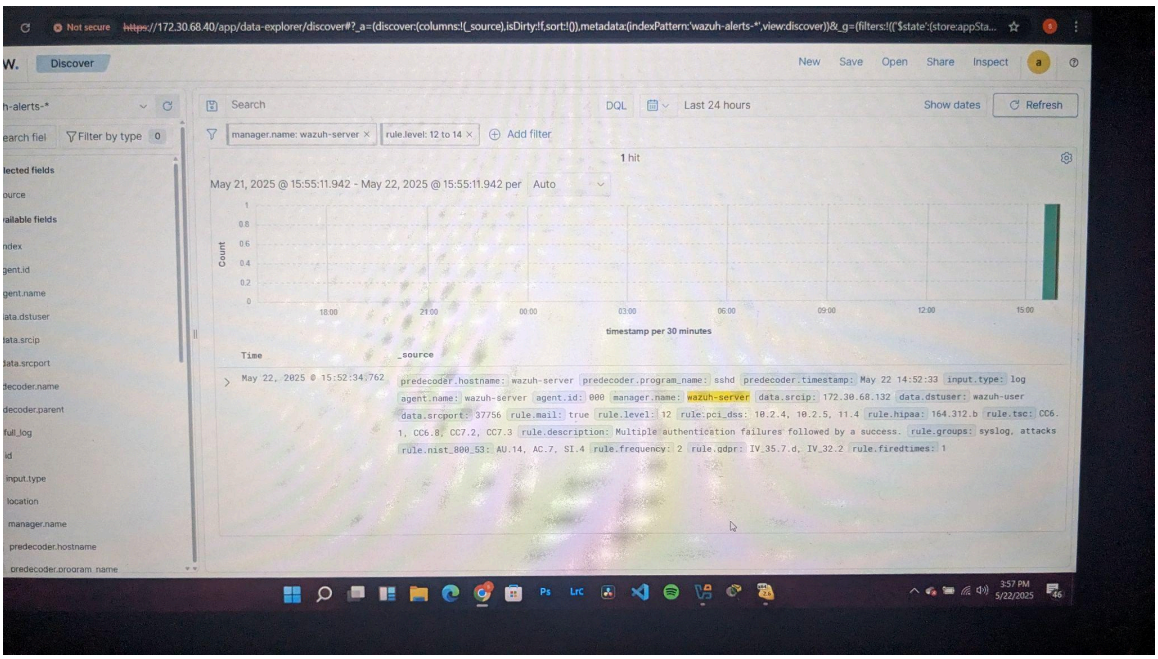


4. Monitoring with Wazuh

- Wazuh, now monitoring Kali, captured the brute-force behavior by analyzing log files.
- Alerts were triggered based on predefined **rules** for multiple failed SSH logins.
- In the **Kibana interface**, I could see:

- Alert name: `sshd: authentication failure`
- Rule group: `authentication_failures`
- Alert level: Critical

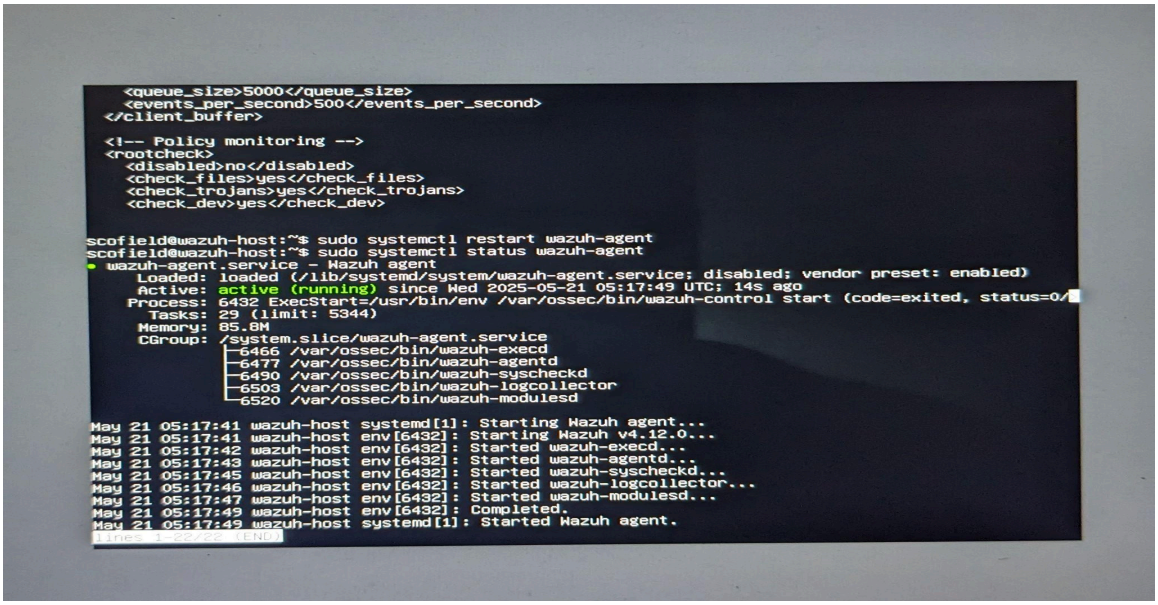
Photographic evidence



5. Customizing Detection

- I modified the `ossec.conf` file to better capture SSH logs and test different thresholds.
- I learned how **log-based intrusion detection** works and how you can fine-tune detection by adjusting rules and decoders.

Photographic evidence



Getting Started

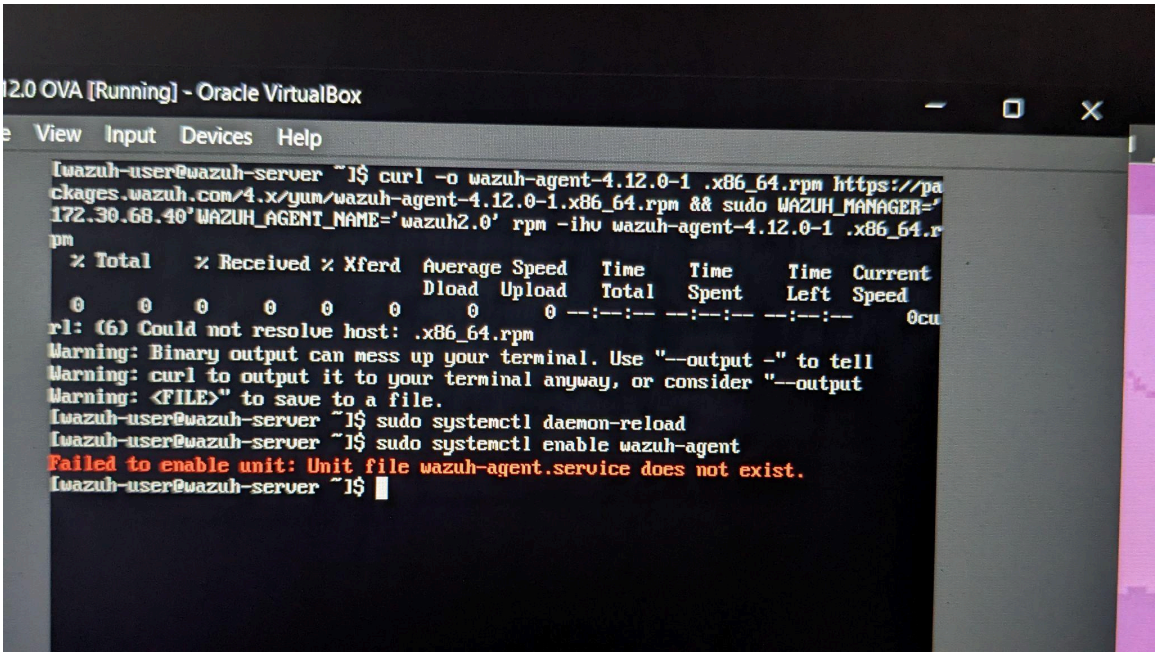
Initially, I was both excited and overwhelmed. Wazuh's documentation is vast, and setting things up on a Linux environment required a lot of reading and guessing. I began with the basics—installing Wazuh manager and agent, and making sure the services were up and running. I ran into minor issues here with systemd but quickly resolved them using community forums.

The Configuration Struggles

One of my biggest challenges was configuring the `ossec.conf` file. It took multiple tries to properly set up the `<localfile>` and `<command>` directives. At one point, Wazuh refused to start due to an XML syntax error. It turned out I had a missing closing tag. Another time, my custom command didn't execute because I had placed it outside the correct configuration block.

These moments were frustrating, especially when logs didn't show exactly what was wrong. But I kept testing, Googling, reading through the documentation, and even referencing similar issues discussed online.

Photographic evidence



Success with SSH Brute Force Detection

After fine-tuning my configuration and enabling the right rules, Wazuh finally detected brute-force login attempts on SSH. Seeing those alerts in the Discover tab felt like a huge win. I had successfully correlated logs, rules, and real system behavior. That clarity made all the errors and restarts feel worth it.

What I Learned

This experience reinforced my skills in Linux and cybersecurity, but even more importantly, it taught me resilience. I didn't give up when the platform didn't behave. Instead, I treated every error as a puzzle—an opportunity to grow. I now understand better how log analysis, custom detection rules, and system monitoring fit into real-world cybersecurity workflows.

I'm proud of the progress I've made with Wazuh, and I look forward to exploring more advanced features in the future. This project has definitely been a milestone in my learning journey.