*Soc lab with real-time threat monitoring*

IFT 593 – Applied Project

*Yogesh Rathod*

*ASU | MS IT (Cybersecurity)*

# Abstract

This report presents the design, implementation, and evaluation of a SOC homelab capable of real-time cyber threat detection and incident response. The project utilized open-source tools, including Wazuh for SIEM, Suricata for network intrusion detection, and pfSense for firewall management, to build a realistic security monitoring environment. The lab was configured to capture security events across endpoints and network devices, simulating attack scenarios such as brute-force attacks, malware infections, file integrity violations, and MITRE ATT&CK techniques. A notable aspect of this project was the structured development process, managed entirely using Jira for task tracking, planning, and milestone management, ensuring smooth and efficient project execution. Real-time alerting to Slack channels was integrated for enhanced visibility and faster response. Through single-handed development and execution, the project provided valuable insights into SOC operations, real-time threat monitoring, and incident handling workflows, while simultaneously strengthening project management and technical leadership skills crucial for cybersecurity professionals.

# Acknowledgements

I would like to express my sincere gratitude to Dayspring Johnson from Cyberwox ([LinkedIn](https://www.linkedin.com/in/dayspringjohnson/)) and Steven Rocks from MyDFIR ([LinkedIn](https://www.linkedin.com/company/mydfir/about/)) for providing valuable educational content and resources that greatly supported my learning throughout this project. I also extend my thanks to the broader cybersecurity community for continuously sharing knowledge, tools, and guidance that inspire and empower learners like me. This project has been a transformative experience, allowing me to apply and expand my knowledge through the resources, mentorship, and community support I received.

Ways to find me:

[LinkedIn](https://www.linkedin.com/in/yrathod/) | [Portfolio Website](https://yogeshrathod.framer.website/) | [GitHub](https://github.com/yogeshrathod2508/threat-dectection-lab/blob/main/README.md) | [Medium](https://yogesh-rathod.medium.com/)

Table of Contents

[Abstract 1](#_Toc196744230)

[Acknowledgements 1](#_Toc196744231)

[Introduction 4](#_Toc196744232)

[Hardware Requirements 5](#_Toc196744233)

[Environment Setup 5](#_Toc196744234)

[Components 5](#_Toc196744235)

[SIEM 5](#_Toc196744236)

[IDS 6](#_Toc196744237)

[Firewall 6](#_Toc196744238)

[Domain Controller 7](#_Toc196744239)

[VMware 7](#_Toc196744240)

[Understanding Network Diagram and Configuration of Components 8](#_Toc196744241)

[Establishing connection of components with pfSense in an isolated environment 9](#_Toc196744242)

[Configuring Internet Access to LAN VMs via pfSense Firewall 10](#_Toc196744243)

[Wazuh SIEM server and agent installation and configuration 13](#_Toc196744244)

[Configuring SYSLOG server in pfSense to send firewall logs 13](#_Toc196744245)

[Simulated Attack Scenarios 16](#_Toc196744246)

[Detecting File Modifications with Wazuh FIM (File Integrity Monitoring) 16](#_Toc196744247)

[Guide and POC to FIM with Wazuh 16](#_Toc196744248)

[Detecting Brute-force RDP attacks on Windows Server 18](#_Toc196744249)

[Allow connections to access RDP of windows server 18](#_Toc196744250)

[VirusTotal Integration to identify and remove malicious installations 19](#_Toc196744251)

[Abuse IP DB integration to identify malicious attackers and blocking them 19](#_Toc196744252)

[Detecting Malicious Commands in Linux Endpoints 20](#_Toc196744253)

[Attack Emulation using MITRE ATT&CK framework 21](#_Toc196744254)

[Setting up the lab environment 21](#_Toc196744255)

[T1053 – Scheduled Task/Job 21](#_Toc196744256)

[T1218.010 – Signed Binary Proxy Execution: Regsvr32 22](#_Toc196744257)

[T1518.001 – Software Discovery: Security Software Discovery 23](#_Toc196744258)

[T1548.003 – Create or Modify System Process: Windows Service 23](#_Toc196744259)

[T1123 – Audio Capture 24](#_Toc196744260)

[Setting up Slack to receive alerts in real-time 25](#_Toc196744261)

[Conclusion 26](#_Toc196744262)

[Future Work 26](#_Toc196744263)

[References 27](#_Toc196744264)

[Glossary 28](#_Toc196744265)

# Introduction

The rise in cyber threats has made it imperative for organizations to establish robust Security Operations Center (SOC) that can monitor, detect, and respond to incidents in real time. Understanding these complex environments requires not only theoretical knowledge but also hands-on practice with real-world tools and scenarios. This project focuses on building a comprehensive SOC homelab, designed entirely from the ground up, to simulate an enterprise-grade cybersecurity infrastructure. Using open-source technologies such as Wazuh SIEM, Suricata IDS, and pfSense Firewall, the lab aims to replicate realistic security operations including network monitoring, endpoint protection, alerting, and incident response.

The homelab integrates multiple layers of security, simulates various attack techniques based on the MITRE ATT&CK framework, and emphasizes proactive threat detection and mitigation strategies. Furthermore, the project involved meticulous planning and task management using Jira to ensure smooth execution, allowing a full end-to-end experience in designing, implementing, and operationalizing a SOC environment. This practical setup enhances the understanding of modern cybersecurity challenges and prepares for real-world SOC roles.

# Hardware Requirements

|  |  |  |
| --- | --- | --- |
| **Component** | **Memory (RAM)** | **Storage** |
| Wazuh SIEM VM | 12 GB | 50+ GB |
| PfSense Firewall VM | 4 GB | 20 GB |
| Windows Server VM | 8 GB | 10 GB |
| Windows Client VM | 4 GB | 10 GB |
| Kali Linux VM | 4 GB | 10 GB |

# Environment Setup

## Components

### SIEM

A security solution that collects, analyzes, and correlates logs for threat detection and incident response.

**Tool used:** Wazuh (Open source)

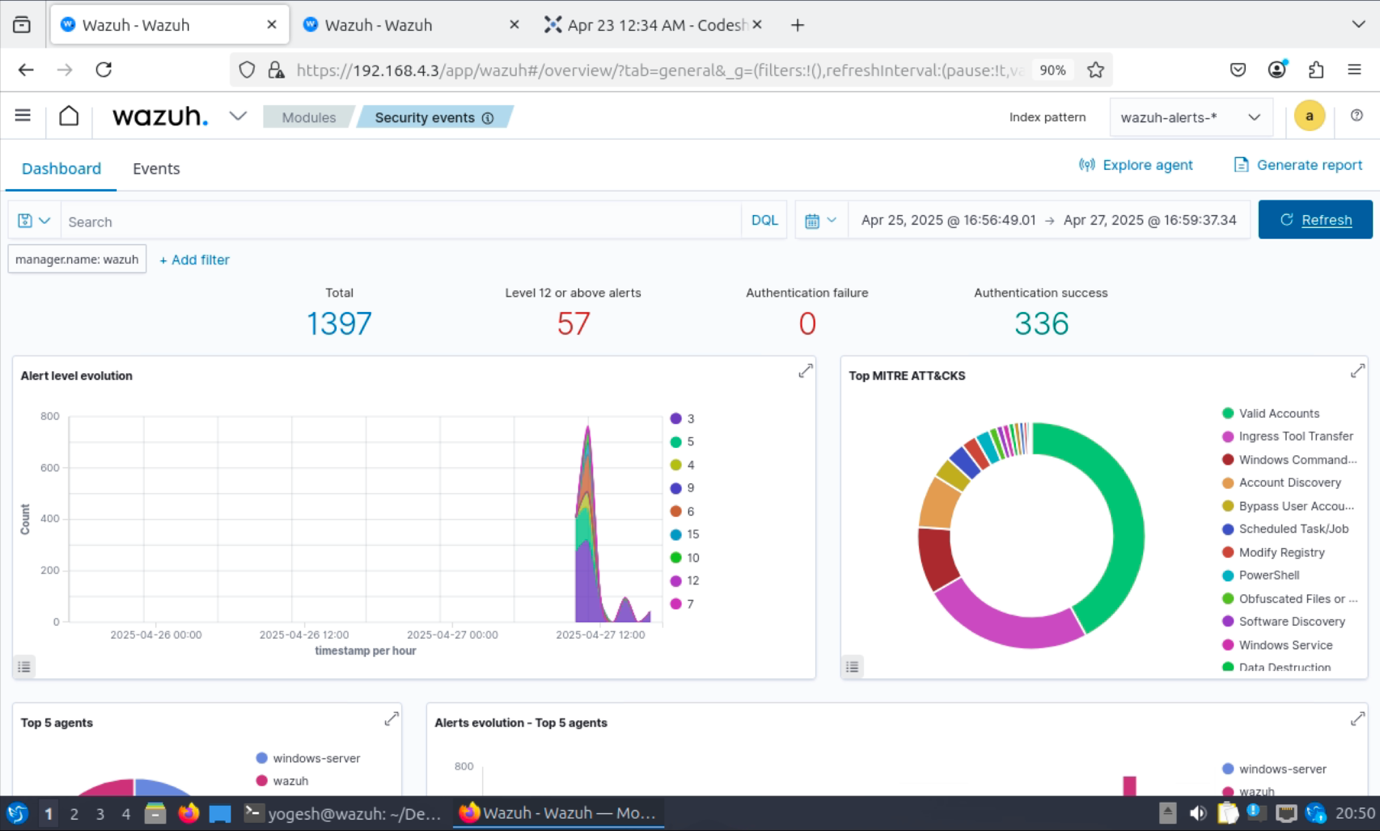


Figure 1 - Wazuh Dashboard

### IDS

A system that monitors network traffic for suspicious activity and alerts security teams.

**Tool Used:** Suricata (Open source)

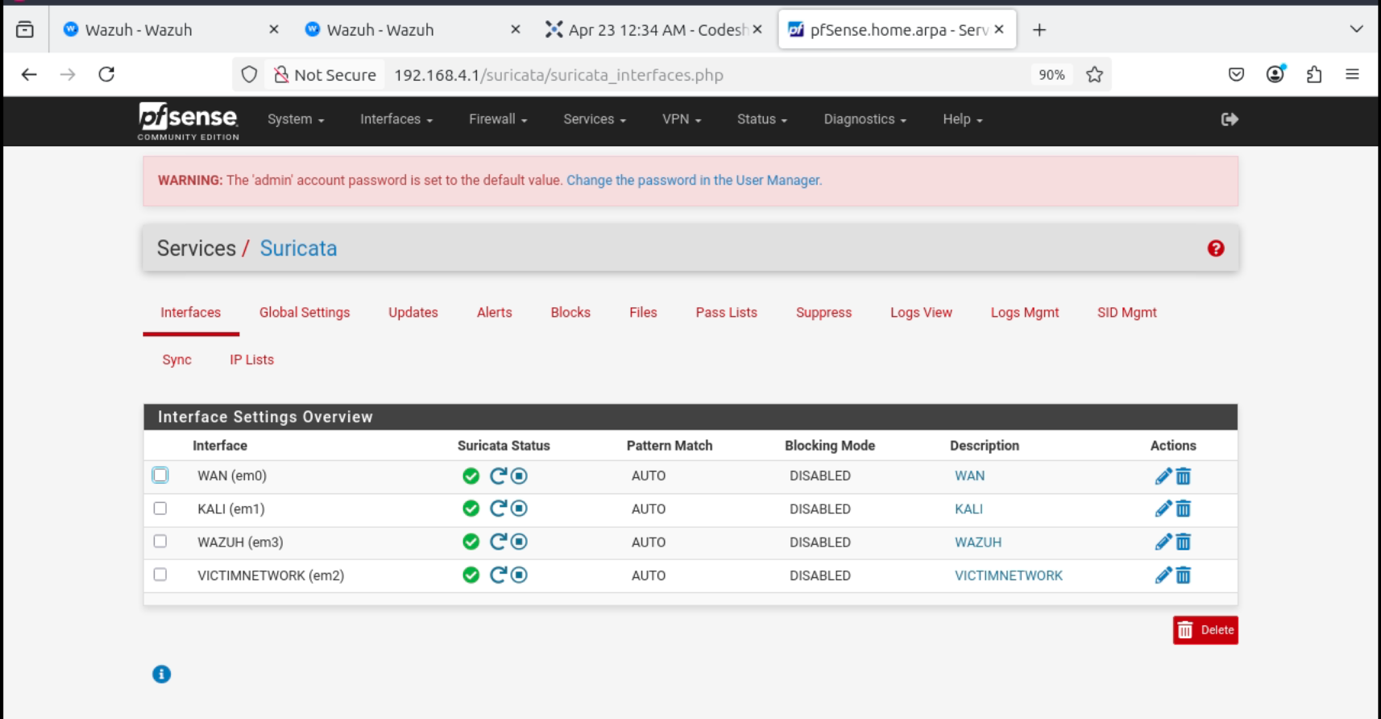


Figure 2 - Configured interfaces on Suricata IDS

### Firewall

A security device or software that filters incoming and outgoing network traffic based on predefined rules.

**Tool Used:** pfSense (Open source)

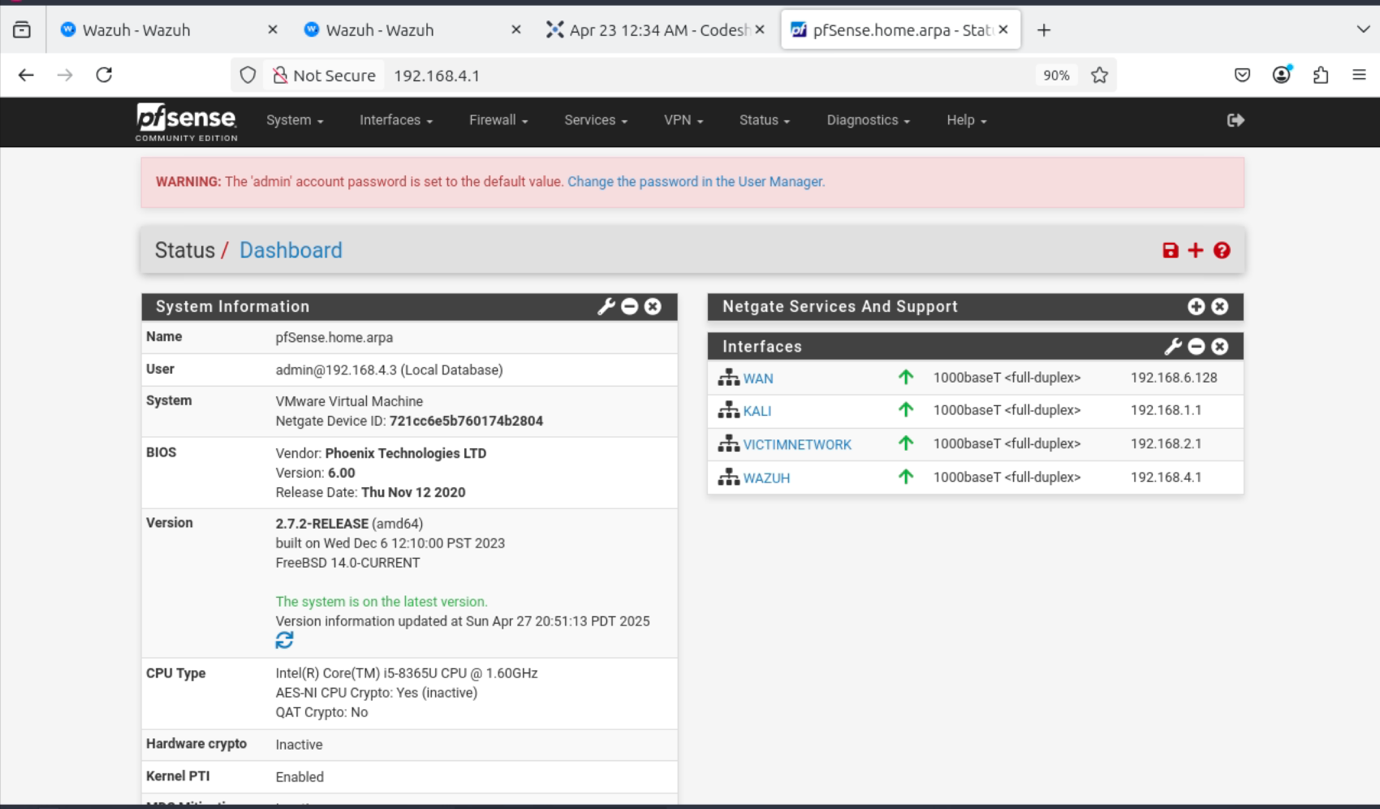


Figure 3 - Firewall Dashboard

### Domain Controller

A server in a Windows network that manages authentication, user access, and security policies.

**OS:** Windows server 2022 (Evaluation Server - Free)

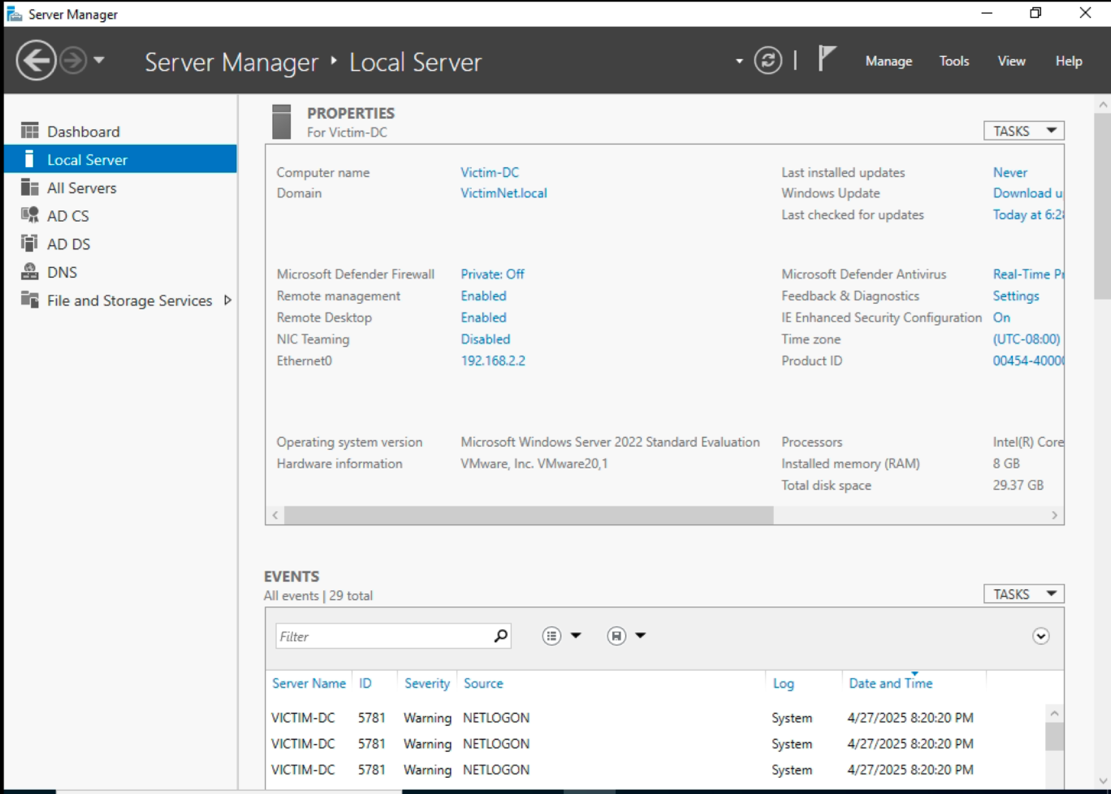


Figure 4 - Windows Server with configured AD

### VMware

A virtualization platform that allows multiple operating systems to run on a single physical machine.

**Tool Used:** VMWare Workstation Pro 17 (Free for personal use)

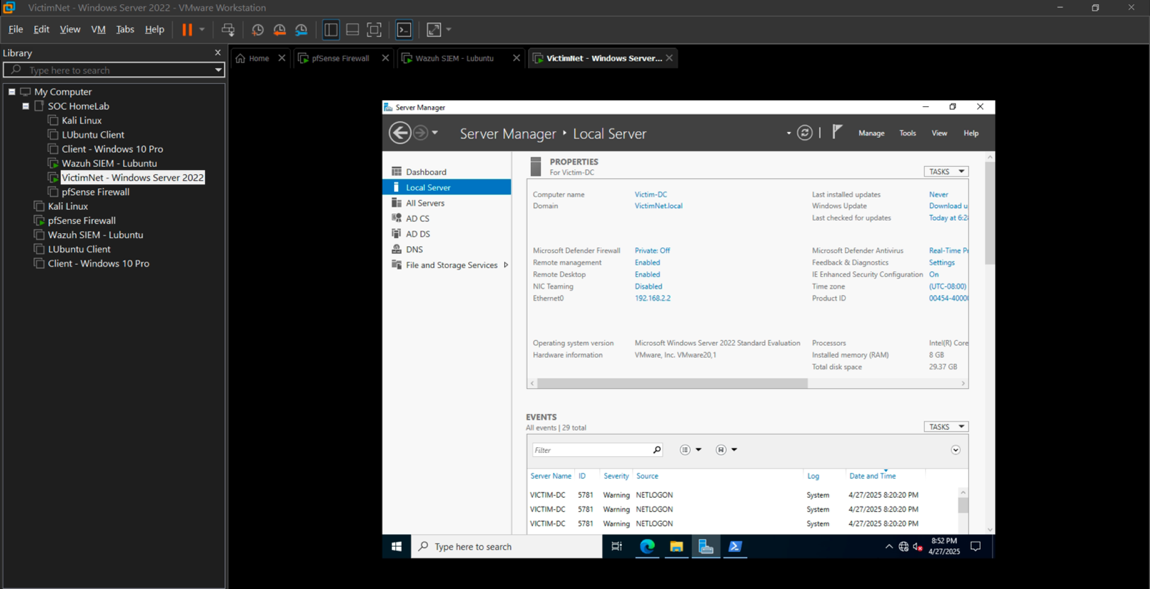


Figure – VMware Workstation Pro

## Understanding Network Diagram and Configuration of Components

A diagram of a computer network

AI-generated content may be incorrect.

Figure 6 - Homelab Network Diagram

**Note:** *Connect each VM excluding pfSense with only one custom vmnet adapter. The routing and internet access will be provided by the firewall.*

This homelab architecture is centered around a **pfSense firewall** integrated to control, monitor, and filter traffic across different segments. The entire setup is hosted on a **Windows Host Machine**, which runs a **VM Server** provided by VMWare Workstation Pro 17. All virtual machines, including the firewall, attack machine, Wazuh Server, domain controller, and victim machine, operate inside this VM server.

At the core of the lab, the **pfSense Firewall** serves as the gateway and primary security enforcement point. It is connected to three separate VM networks: **VMnet2** (for the Attack Machine), **VMnet3** (for the Domain Controller and Victim Machines), and **VMnet4** (for the Wazuh Server). All inbound and outbound traffic between these segments is routed through the firewall, where **it** actively monitors for all traffic.

The **Attack Machine**, running **Kali Linux** with IP 192.168.1.2, connects to the pfSense firewall via **VMnet2**. It is used to simulate real-world cyberattacks, such as SQL injection, brute-force attempts, and exploitation activities. Traffic originating from the attacker is monitored by firewall and logs are passed to the Wazuh SIEM setup for alerting and analysis.

On the victim side, the **Domain Controller** operates as the core Active Directory server, responsible for authentication and network management. It is connected to pfSense using **VMnet3**. It also forwards security logs to the SOC Analyst machine to enhance threat visibility. The **Victim Machine** , resides in the same victim network and simulates a standard workstation used in an enterprise environment, communicating mainly with the Domain Controller.

The **SOC Analyst Machine**, configured with **Wazuh**, has an IP address of 192.168.4.1 and connects to pfSense through **VMnet4**. It receives logs and alerts from the firewall and the domain controller through a **agent**, allowing real-time detection, incident monitoring, and threat hunting based on the MITRE ATT&CK framework.

All traffic—whether attack simulations, user activities, or admin operations—must pass through the **pfSense Firewall**, where it is inspected, filtered, and logged. This design ensures network segmentation, centralized traffic control, attack detection, and incident response analysis, creating a realistic environment for security monitoring and defense practice.

### Establishing connection of components with pfSense in an isolated environment

After connecting each VM to its custom network, we now need to ensure that they are centrally connected to the pfSense firewall and can communicate with the other interfaces of the firewall, but not with the host machine. First, we need to assign the interfaces in pfSense. Then, we will configure an IP address for each interface and set up the corresponding IP configuration on the devices connected to each specific interface.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7 - Configured interfaces in pfsense

**Reference:** Configuration of this can be found from CyberWox’s [‘Building a Cybersecurity Homelab’](https://www.cyberwoxacademy.com/post/building-a-cybersecurity-homelab) article.

### Configuring Internet Access to LAN VMs via pfSense Firewall

Here, I have developed an infrastructure which is isolated from the host network and only pfSense is allowed to have NAT connection with host and others have custom network setup. To connect each component with one another, we need to set something called as Outbound NAT connection.

#### Note

* **NAT:** Used for converting private IP addresses assigned (by pfSense or configured manually) to public IP address for internet access. This is necessary as private IPs are not routable to the internet.
* **Why need it?** NAT allows them to "borrow" the firewall’s WAN IP for external communication.
* **IP forwarding:** Enables pfSense to act as a **router**, allowing traffic to flow between different subnets (e.g., 192.168.1.0/24 and 192.168.2.0/24).
  + Without IP forwarding, traffic between LAN1 (Ubuntu) and LAN2 (Windows Server) would not route through pfSense.
  + **Why Need it?** By default, devices on separate subnets cannot communicate directly. IP forwarding allows pfSense to route traffic between them, enforcing firewall rules along the way

#### How to?

1. Add default gateway of pfSense to endpoints.

**Ubuntu:**

* Edit **sudo nano /etc/netplan/\*.yaml** (Enter your file name)

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 8 - Setting default gateway in ubuntu based linux

* Save the changes and run **sudo netplan apply.**

**Kali:**

* Go to Advanced Network Configuration in Kali linux, select your connection and edit the settings.
* Go to IPv4 settings and modify as shown

A screenshot of a computer

AI-generated content may be incorrect.

Figure 9 - Setting default gateway in kali

1. Configure Outbound NAT in pfSense as follows.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 10 - Manual Outbound NAT Configuration

A screenshot of a computer

AI-generated content may be incorrect.

Figure 11 – Adding NAT configuration to Wazuh Server interface

1. Change ip forwarding to 1 if 0.

* Navigate to pfsense VM CLI and open a shell and enter the following command: ***sysctl net.inet.ip.forwarding=1***

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 12 - Enabling IP forwarding in pfsense

**Note:** *The command executed above will not be persistent so you will need to enter each time after a reboot. So, make sure you find a persistent solution.*

### Wazuh SIEM server and agent installation and configuration

**Wazuh Server:**

* A quickstart installation guide for Wazuh server: <https://documentation.wazuh.com/current/quickstart.html>
* To install components in step-by-step basis: <https://documentation.wazuh.com/current/installation-guide/index.html>

**Wazuh Agent:**

* For every type of OS: <https://documentation.wazuh.com/current/installation-guide/wazuh-agent/index.html>

### Configuring SYSLOG server in pfSense to send firewall logs

**Note: Agent vs agentless** - Wazuh provides agent for deploying them to the client machines and extracting data from the clients and sending them to the server.

To install an agent, Wazuh provides a wget script which installs the agent in our linux machine but pfSense is installed in openBSD system and by default it prevents the users from installing any 3rd party packages so we cannot install the agent and the wget command doesn’t work as well.

I tried installing package ‘wget’ in my pfSense VM but was unable to install any package from the openBSD repository. I researched and below are the details:

* pfSense doesn’t pull packages from the FreeBSD repo, just their own. It’s not a general-purpose OS but rather significantly modified as an appliance.
* To send firewall traffic to SIEM, you can forward your logs through **syslog** server.

**Reference:** <https://opennix.org/en/docs/pfsense/pfsense-wazuh-integration/>

**Another great resource:** <https://benheater.com/integrating-pfsense-with-wazuh/>

Now to setup syslog server, we need to enable archiving in Wazuh. Steps are given below:

* <https://www.infopercept.com/blogs/enabling-archive-logs-in-wazuh>
* <https://documentation.wazuh.com/current/user-manual/manager/event-logging.html>

Now, we need to configure syslog in pfSense to send the firewall logs to the Wazuh. Wazuh has inbuilt decoders for pfSense as well. But the default syslog by pfSense is not decoded properly by Wazuh so we will use **syslog-ng package** for this. You can install syslog\_ng on pfSense by going to the GUI and System > Package Manager > Available Packages and search for syslog\_ng and install it.

Follow the guide from the link below.

**Setup Syslog-ng in pfsense:** <https://marceltc.com/sending-pfsense-syslogs-to-wazuh-siem>

**Note:** Use the Log Message Format as **syslog** and **NOT BSD.** The guide above asks us to set to BSD but I didn’t get any logs when I had set to BSD.

After configuring everything, I created a new rules file for overwriting the rule description as follows:

**Local\_rules.xml in Wazuh manager:**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 13 - Local rules conf file

**Wazuh Manager Conf file:**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 14 – Turning on the logging in ossec.conf

A screenshot of a computer

AI-generated content may be incorrect.

Figure 15 – Setting port for receiving logs in ossec.conf

# Simulated Attack Scenarios

## Detecting File Modifications with Wazuh FIM (File Integrity Monitoring)

The FIM module in Wazuh performs scheduled scans on designated paths and continuously monitors specific directories for any modifications. Users can configure which paths to track within the Wazuh agents and manager settings.

FIM maintains a local database where it stores file checksums and attributes. During each scan, the Wazuh agent detects alterations in monitored files and reports them to the Wazuh server. The module identifies changes by comparing the current checksum and attributes of a file with the previously stored values. If discrepancies are found, it triggers an alert.

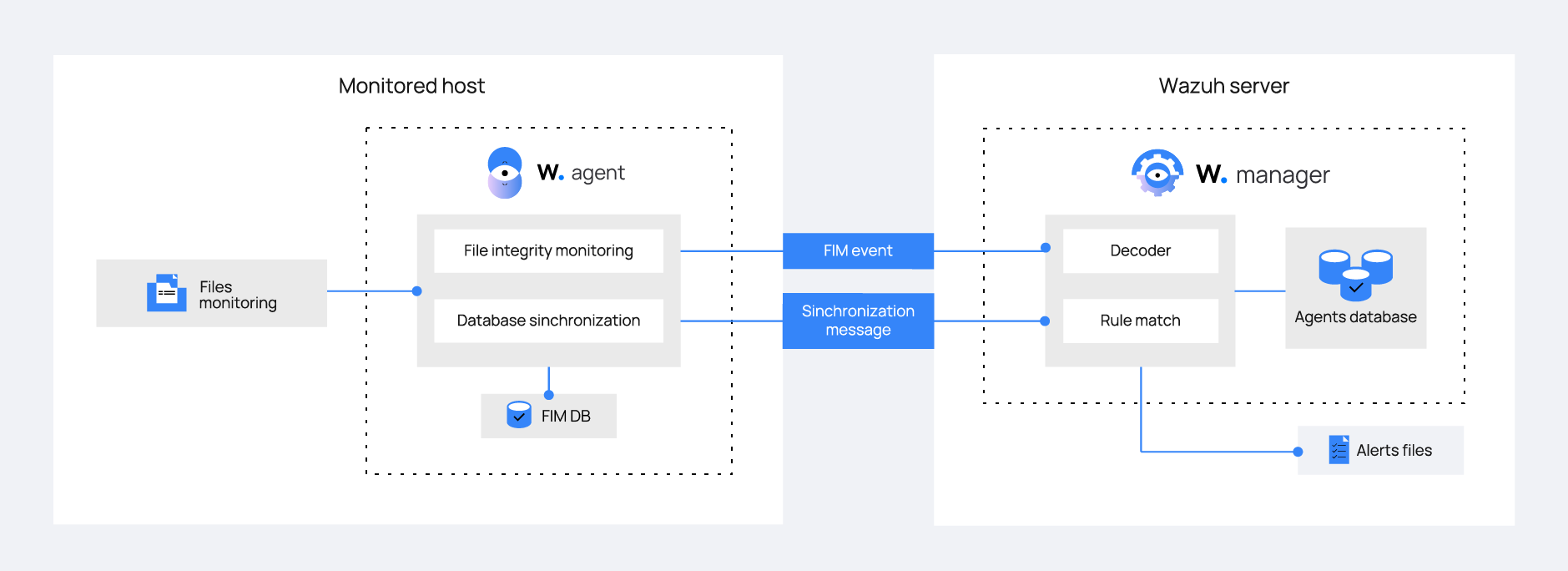


Figure 16 - FIM Working ([Source](https://documentation.wazuh.com/current/user-manual/capabilities/file-integrity/how-it-works.html))

### Guide and POC to FIM with Wazuh

We can configure File Integrity Management with Wazuh by just editing ossec config file from our endpoint machine. The full guide is given below.

**Reference:** <https://documentation.wazuh.com/current/proof-of-concept-guide/poc-file-integrity-monitoring.html>

A computer screen with white text

AI-generated content may be incorrect.

Figure 17 - Ubuntu Client ossec.conf file

A screenshot of a computer

AI-generated content may be incorrect.

Figure 18 - File Addition log

## Detecting Brute-force RDP attacks on Windows Server

Brute-force attacks are a common method used by attackers to gain unauthorized access, often targeting services like SSH on Linux and RDP on Windows. Wazuh detects these attacks by correlating multiple failed login attempts. This use case demonstrates how to configure Wazuh to detect and block brute-force attacks on RHEL and Windows endpoints using Active Response.

### Allow connections to access RDP of windows server

* Start the RDP listening from Server Manager > Remote Desktop > Uncheck allow only from computer with NLA > Apply

In our attacking machine, we will use a tool called hydra for simulating brute-force attack on our RDP client. Given below is the command.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 19 - Attacking using Hydra

A screenshot of a phone

AI-generated content may be incorrect.

Figure 20 - Auth Failure Logs on Wazuh

## VirusTotal Integration to identify and remove malicious installations

VirusTotal is a free online service that scans files and URLs for malware using multiple antivirus engines and provides an API key that can be integrated with Wazuh SIEM to monitor downloaded or installed files for anomalies or potential threats.

First, enable file monitoring if it's not already active. VirusTotal will identify malicious files and trigger an alert in Wazuh, after which a configured active response script on the Linux endpoint will automatically execute and remove the threat. The complete guide is provided below.

**Reference:** <https://documentation.wazuh.com/current/proof-of-concept-guide/detect-remove-malware-virustotal.html>

A screenshot of a computer

AI-generated content may be incorrect.

Figure 21 - Active response in Action

## Abuse IP DB integration to identify malicious attackers and blocking them

This use case shows how to block malicious IPs from accessing web servers. Apache is set up on Ubuntu and Windows machines, while an Kali Linux system acts as the attacker. Its IP is added to a public IP reputation list, and Wazuh is configured to block it from reaching

A screenshot of a computer

AI-generated content may be incorrect.

Figure 22 - Blocking known malicious actor

**Reference:** <https://documentation.wazuh.com/current/proof-of-concept-guide/block-malicious-actor-ip-reputation.html>

## Detecting Malicious Commands in Linux Endpoints

This integration ensures comprehensive tracking of user-executed commands, including sudo and root actions, for enhanced security monitoring. It helps detect suspicious activity by alerting on malicious commands in real time. Combining Auditd's granular logging with Wazuh's analysis strengthens Linux endpoint visibility and threat response.

**Steps:**

* Install auditd in endpoint
* Then we add a rule in our endpoint **which logs all commands (execve) run by user UID 1000 which is the current logged in user(including sudo/su), filtering out system noise (GID 994) and tagging logs for Wazuh analysis (audit-wazuh-c). They cover both 32-bit (b32) and 64-bit (b64) processes to ensure full command monitoring.**
* **Add the list of suspicious programs.**
* **Add rule and conf in Wazuh server to detect and create an alert.**

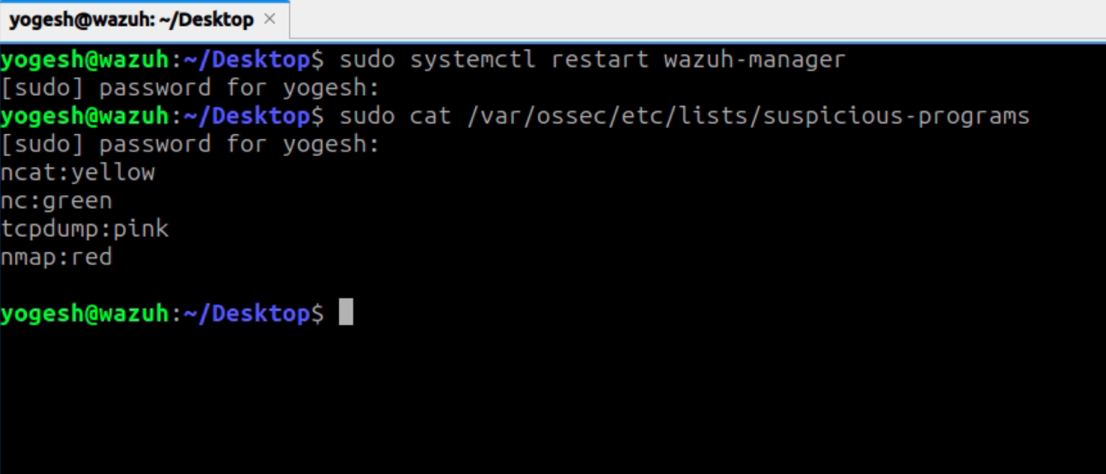


Figure 23 - List of suspicious programs

**A screenshot of a computer

AI-generated content may be incorrect.**

Figure 24 - Malicious command execution

**Reference:** <https://documentation.wazuh.com/current/proof-of-concept-guide/audit-commands-run-by-user.html>

## Attack Emulation using MITRE ATT&CK framework

This module focuses on attack emulation using Atomic Red Team (ART) mapped to the MITRE ATT&CK® framework and detection through Wazuh SIEM. ART automates adversary technique testing, while Sysmon captures system activities via Windows Event Logging. Wazuh provides real-time detection and alerting across environments. This report details the emulation of selected ATT&CK techniques on a Windows system, configuration of Wazuh for event collection, custom rule creation, and analysis of generated alerts.

### Setting up the lab environment

1. Install Sysmon from Sysinternals tools and install it in the desired windows endpoint.
2. Install Atomic Red Team Suite
3. Configure Wazuh agent to capture Sysmon events.
4. Create custom rules for detecting specific tactics from MITRE ATT&CK.
5. Invoke the command to trigger specific tactics with command.
6. Visualize the alerts in the Wazuh MITRE ATT&CK Dashboard.

### **T1053 – Scheduled Task/Job**

Attackers create or modify scheduled tasks to execute malicious programs at system startup or on a recurring basis for persistence.

Attack Emulation on the endpoint

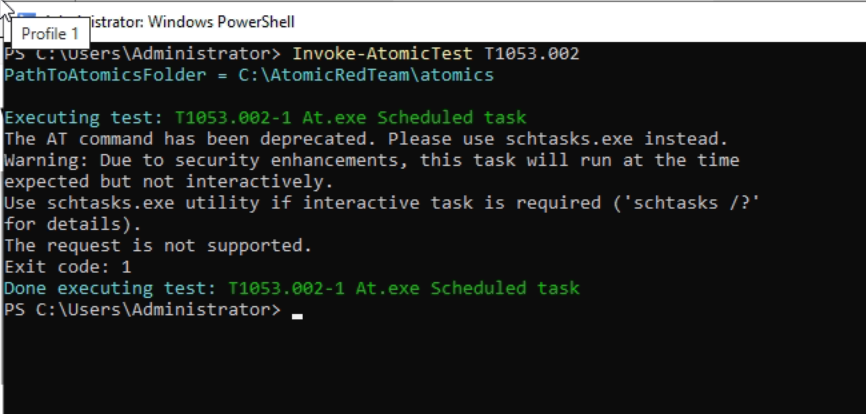


Figure - Running Test

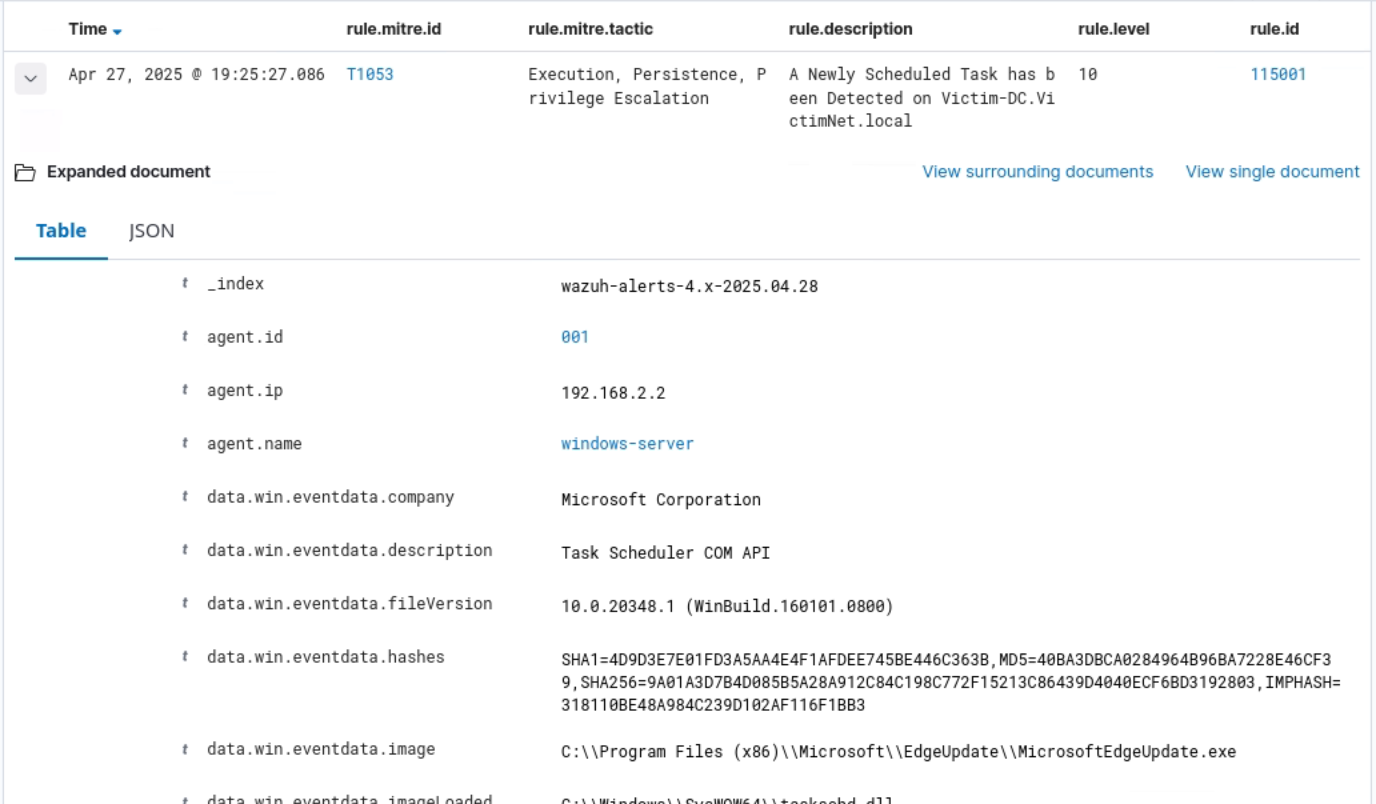


Figure - Capturing attack alert for T1053

### T1218.010 – Signed Binary Proxy Execution: Regsvr32

Attackers abuse regsvr32.exe, a signed Windows tool, to proxy execution of malicious DLLs, bypassing security controls.

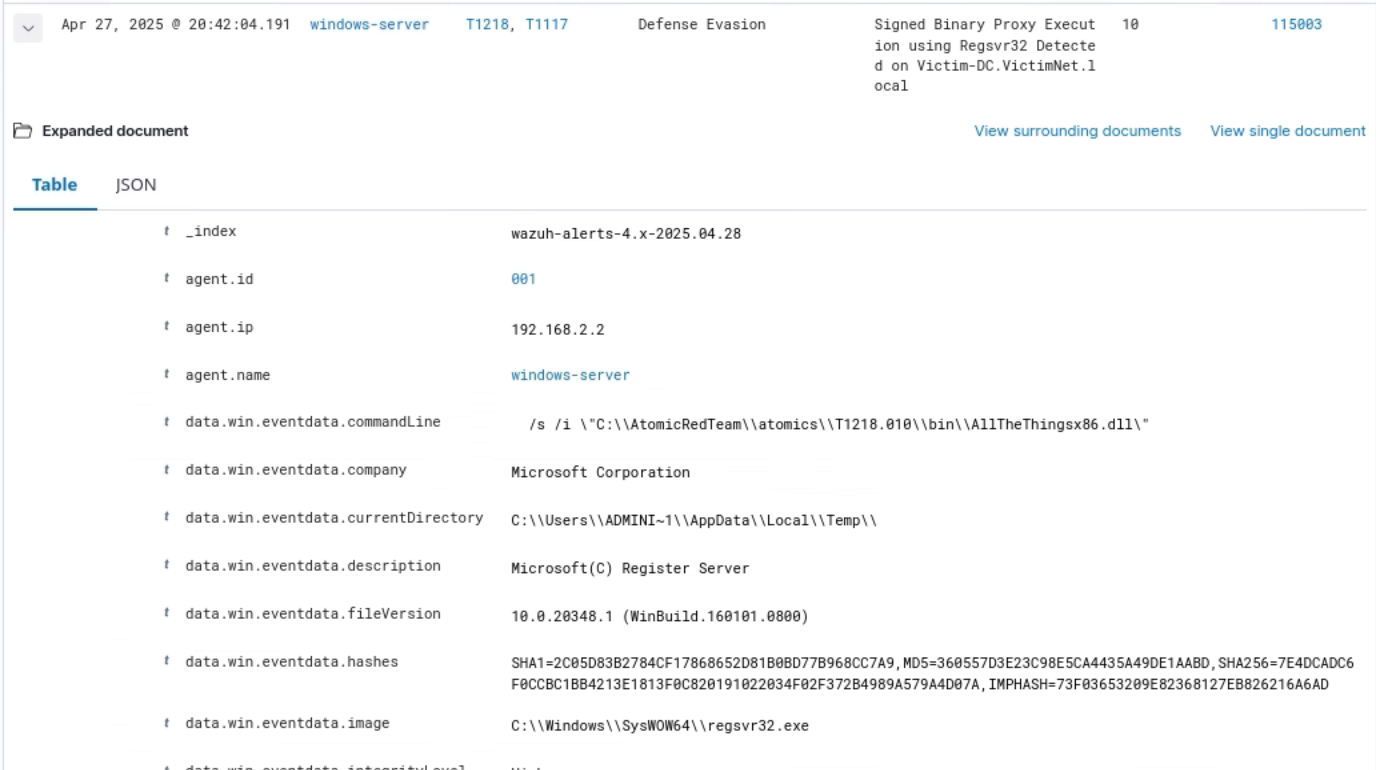


Figure - Capturing attack alert for T1218

### T1518.001 – Software Discovery: Security Software Discovery

Adversaries identify installed security products like antivirus or EDR solutions to plan evasion techniques or tailor attacks.

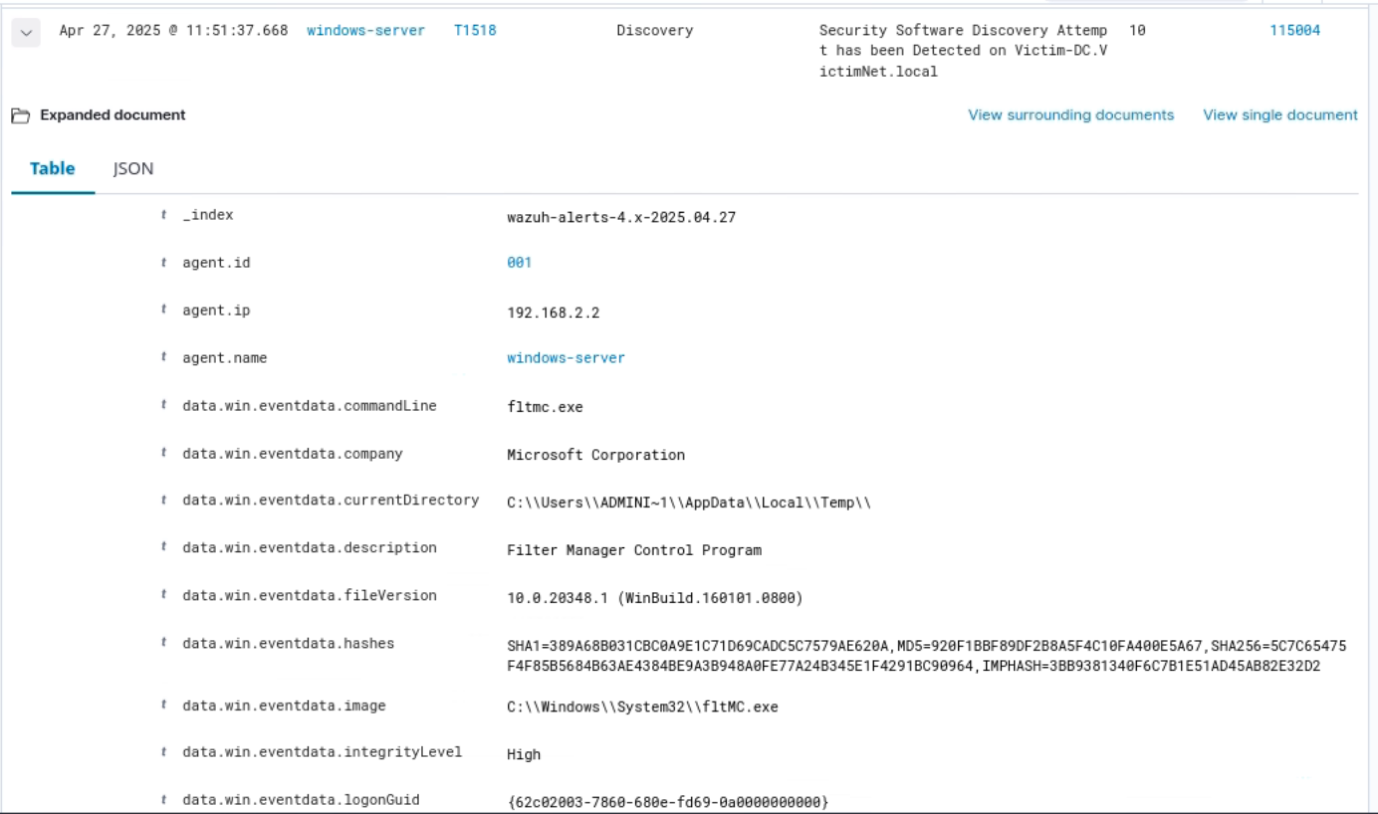


Figure - Capturing attack alert for T1518

### T1548.003 – **Create or Modify System Process: Windows Service**

Adversaries create or modify Windows services to achieve persistence or escalate privileges, allowing their payloads to run automatically with high-level permissions during system startup or service execution.

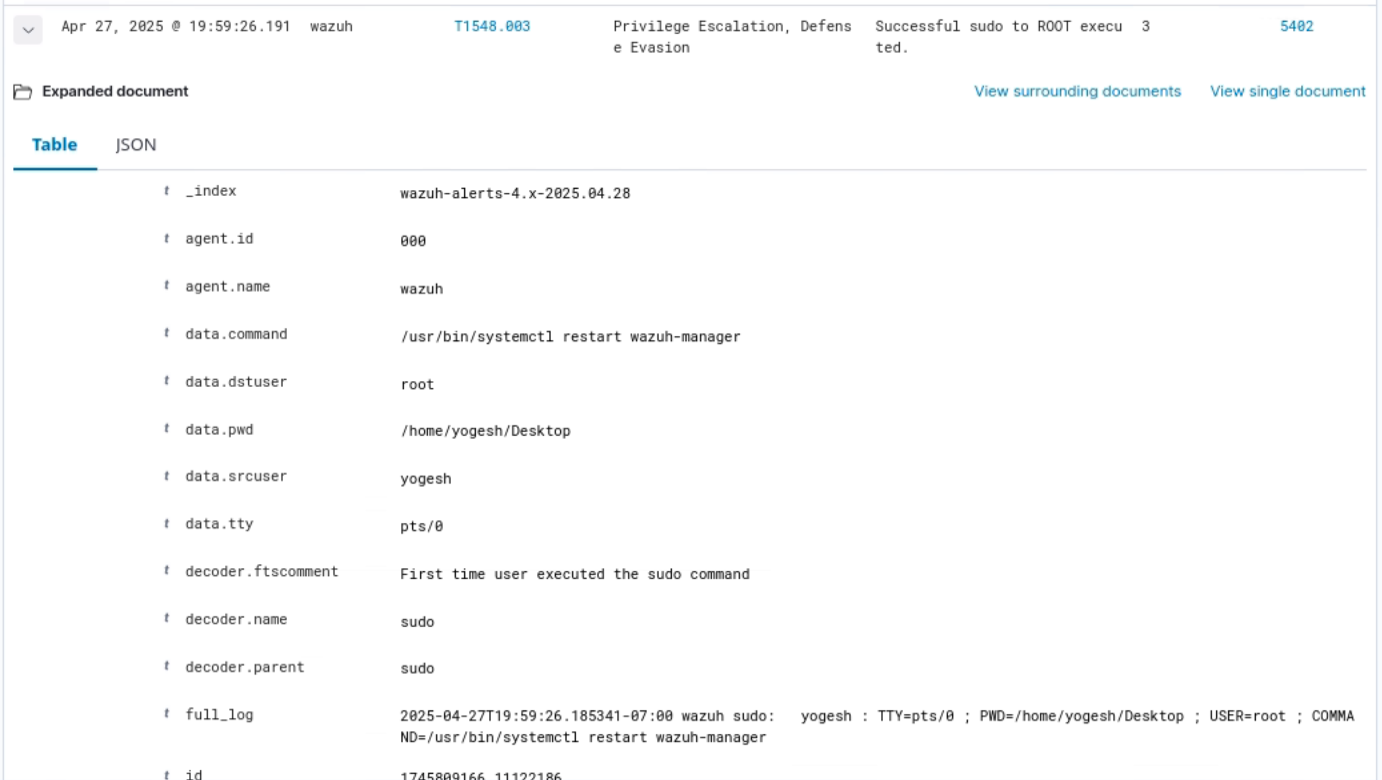


Figure - Capturing attack alert for T1548

### T1123 – Audio Capture

Malware captures audio from a device’s microphone to gather sensitive conversations or environmental sounds.

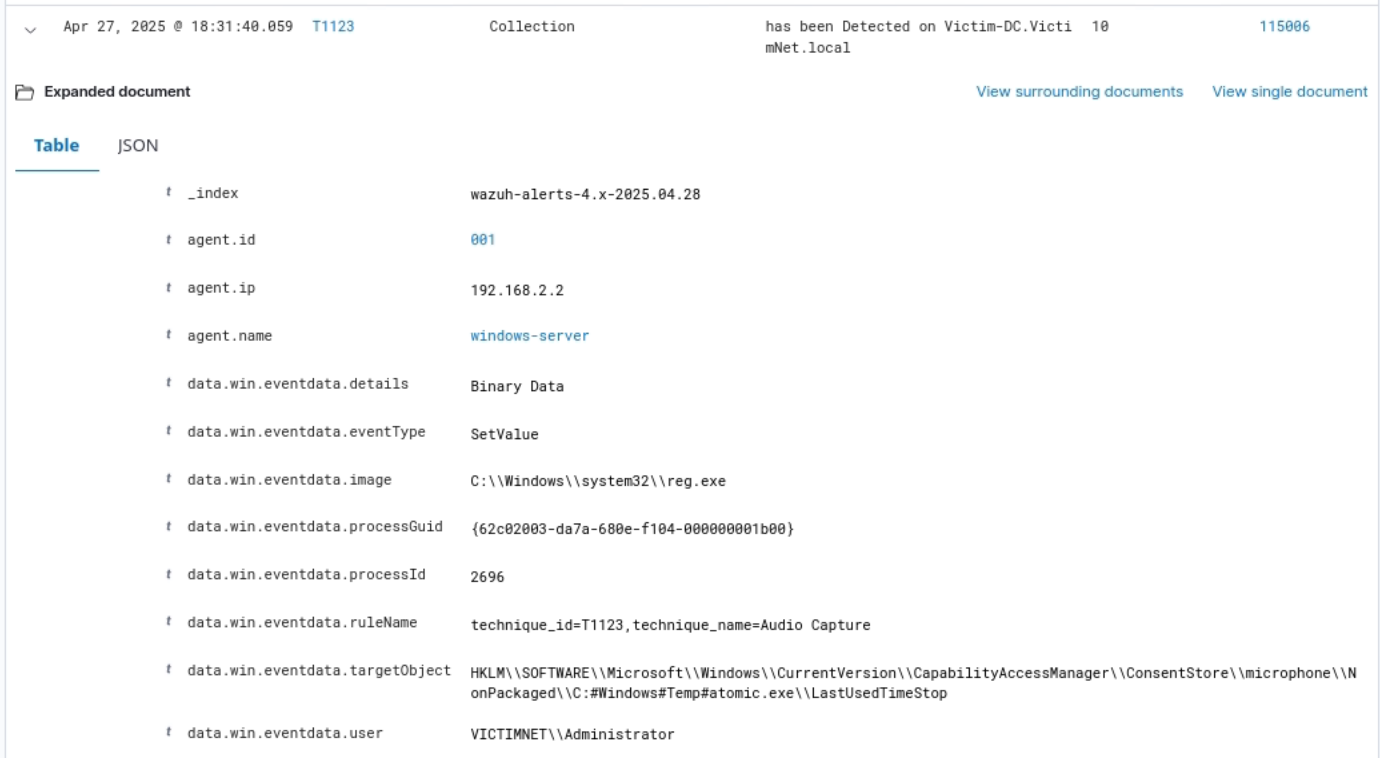


Figure - Capturing attack alert for T1123

#### T1135 – Network Share Discovery

Attackers scan for shared folders and drives over a network to find sensitive data or lateral movement targets.

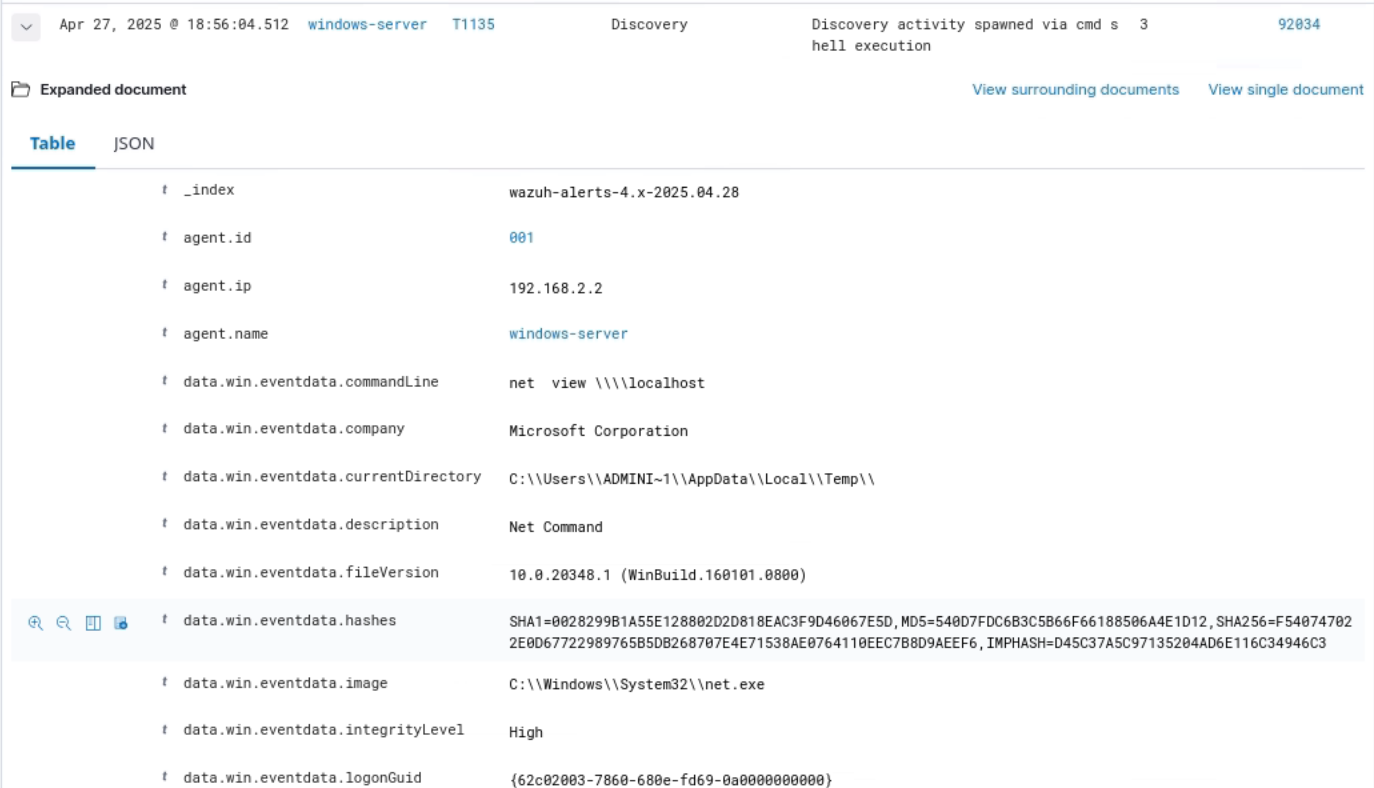


Figure - Capturing attack alert for T1135

# Setting up Slack to receive alerts in real-time

Setting up real-time alerting with Slack is straightforward using Wazuh’s built-in alerting integrations. I configured Slack to receive alerts directly from the Wazuh server based on specific alert levels. This setup ensures immediate notification of critical security events. The integration requires minimal configuration using Wazuh’s Slack module and webhook URL. This enhances incident response by keeping the security team promptly informed.

#### Wazuh Configuration

A screenshot of a computer

AI-generated content may be incorrect.

Figure 32 - Slack Webhook conf in Wazuh server

#### Slack Notifications

A screenshot of a computer

AI-generated content may be incorrect.

Figure 33 - Slack Notification from Wazuh Server

# Conclusion

The development of this SOC homelab offered not only technical insights but also strengthened my project management and organizational skills. Building the entire environment single-handedly required detailed planning, task prioritization, and agile tracking to ensure steady progress. I utilized Jira as a project management tool to break down the work into manageable tasks, track milestones, and monitor dependencies, which streamlined the development cycle and reduced delays. Managing all technical and administrative aspects independently sharpened my decision-making and problem-solving abilities, fostering a strong balance between execution and strategic oversight. Overall, this project not only deepened my expertise in real-time threat detection, incident response, and cybersecurity architecture, but also enhanced my capability to manage complex technical projects efficiently from initiation to completion — a crucial skill for real-world cybersecurity operations.

# Future Work

Going forward, one major improvement will be configuring and fine-tuning the Network Intrusion Detection System (NIDS) using Suricata on the pfSense firewall to enhance real-time attack detection. We also plan to test detection rules by simulating attacks from resources like 3CORESec [TestMyNIDS](https://github.com/3CORESec/testmynids.org) to validate the effectiveness of the IDS setup.

Further work will involve customizing and automating incident responses through SIEM playbooks in Wazuh and expanding attack emulation across different MITRE ATT&CK techniques. Future upgrades may also include integrating a central threat intelligence feed for scaling the homelab into a full Security Operations Center (SOC) simulation environment.

# References

1. SOC Homlelab playlist by CyberWox: <https://www.youtube.com/playlist?list=PLDqMNdDvMsRkmtiKcZwbhOz7MeLQE0r3G>
2. Cybersecurity Analyst Projects by MyDFIR: <https://www.youtube.com/playlist?list=PLG6KGSNK4PuBWmX9NykU0wnWamjxdKhDJ>
3. Integrating pfsense: <https://benheater.com/integrating-pfsense-with-wazuh/>

# Glossary

1. **SIEM (Security Information and Event Management)**: A system that collects, analyzes, and correlates security events from multiple sources to detect and respond to threats.
2. **IDS (Intrusion Detection System)**: A security tool that monitors network or system activities for malicious actions or policy violations.
3. **Threat Intelligence**: Information about current or emerging threats that helps organizations detect, prevent, and respond to cyberattacks.
4. **VirusTotal**: A free online service that analyzes files and URLs for viruses, worms, trojans, and other kinds of malicious content using multiple antivirus engines.
5. **Abuse IPDB**: A project that collects and shares reports about malicious IP addresses, helping organizations block bad traffic.
6. **Syslog**: A standard protocol used to send system logs or event messages to a centralized server for monitoring and analysis.
7. **FIM (File Integrity Monitoring)**: A security technology that detects unauthorized changes to critical system files, configurations, and content.