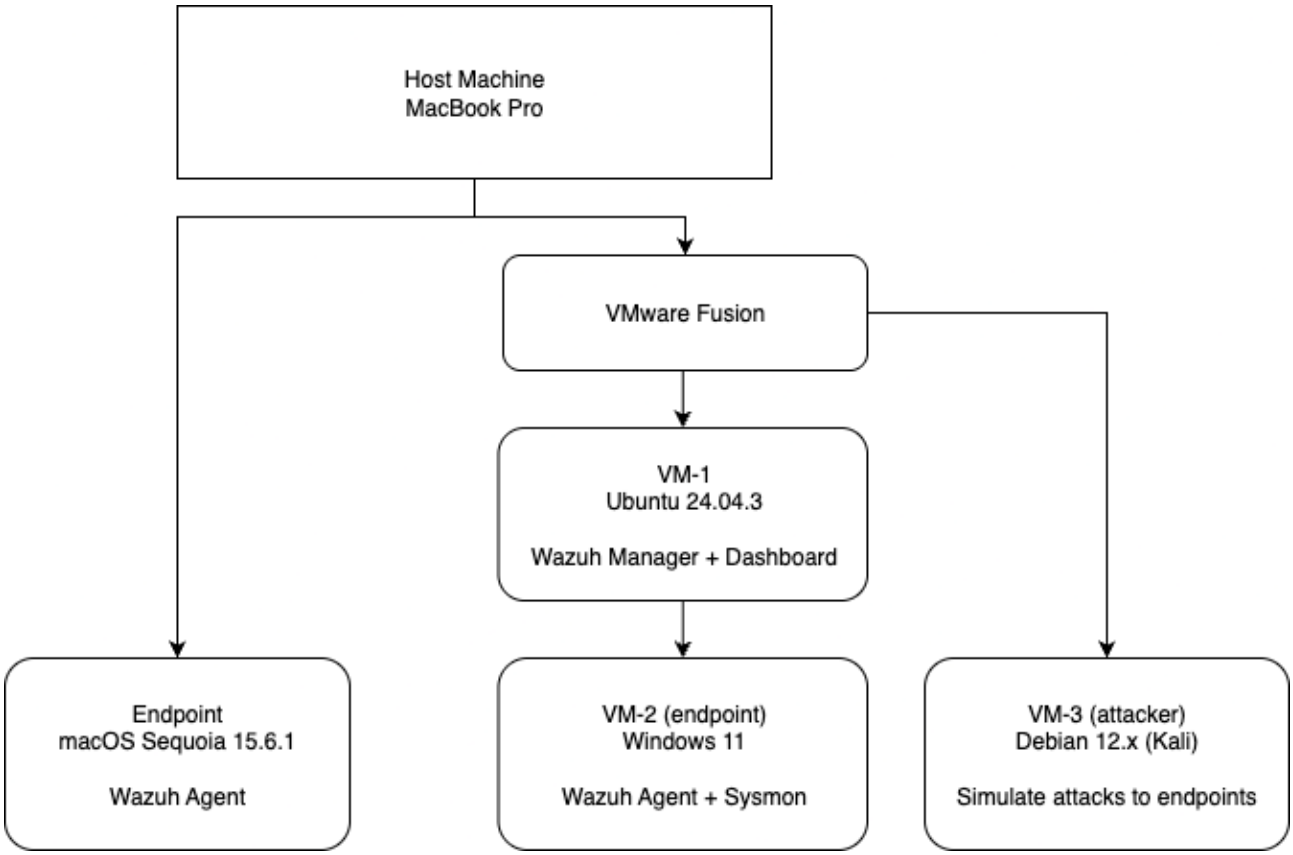
SOC Homelab Project

Objective
Build a SOC environment to practice endpoint monitoring, log analysis, and incident detection.

Technologies
OS: Linux, Windows, macOS
Tools: VMware Fusion, Wazuh (SIEM), Sysmon, nmap, hydra

Architecture Diagram



Implementation (overview)
1. Set up environment
2. Generate test events + verify alerts on Wazuh dashboard
3. Hardening endpoints

Implementation (detailed)

1. Set up environment

Instance 1 - Mac (host machine)

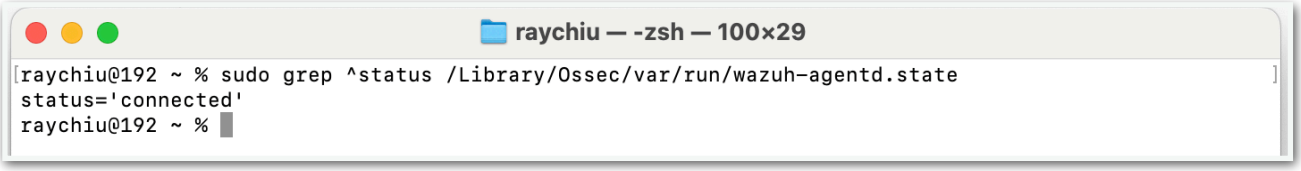
- Installed VMware Fusion
- Installed Wazuh Agent (also act as an endpoint itself)

Troubleshoot

Issue	• agent did not show in dashboard
Diagnosis	• ping was successful • port 1514 was configured correctly • however, server IP was dashboard's IP (127.0.0.1) instead of VM's IP (192.168.x.x)
Solution	• updated server IP and restart agent

Screenshot - VMware Fusion on Mac Host

Screenshot - Wazuh agent status on Mac Endpoint



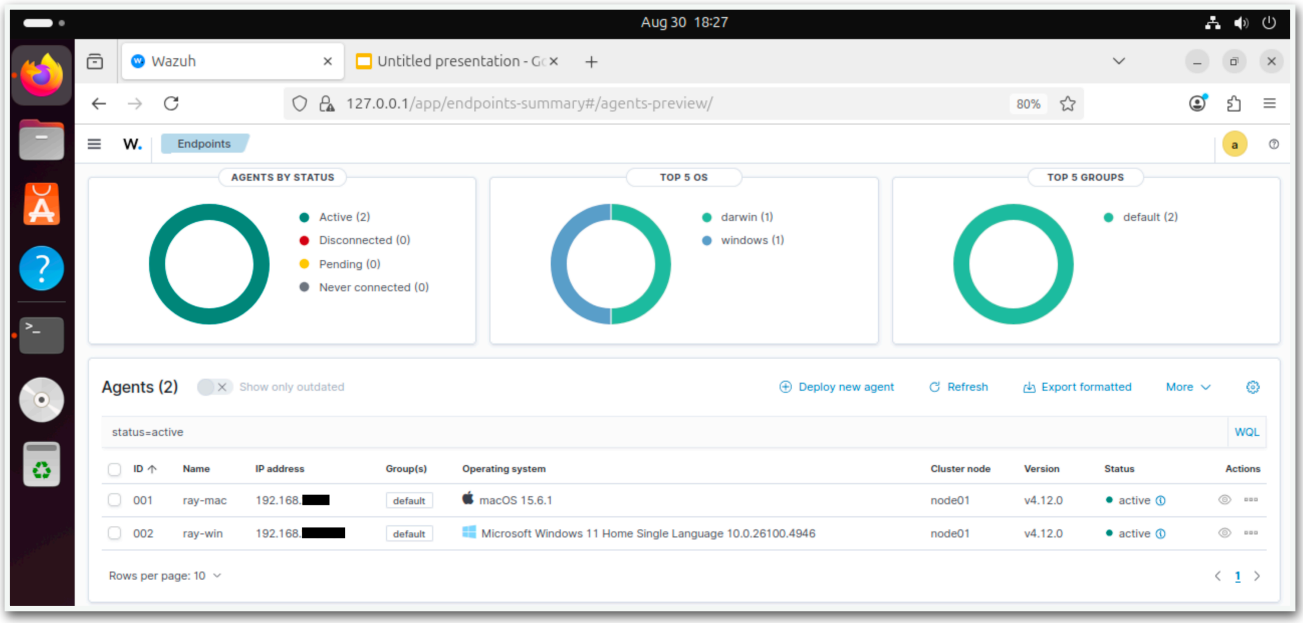
Instance 2 - Ubuntu VM

- Installed Wazuh Manager + Dashboard

Troubleshoot

Issue	• Wazuh installation failed multiple times
Diagnosis 1	• Wazuh did not support latest Ubuntu 25.04 as of August 2025
Solution 1	• re-installed Ubuntu 24.04 instead
Diagnosis 2	• did not allocate enough disk space for VM
Solution 2	• allocated 100GB for this VM (minimum is 80GB)

Screenshot - Connected agents on Wazuh Dashboard on Ubuntu



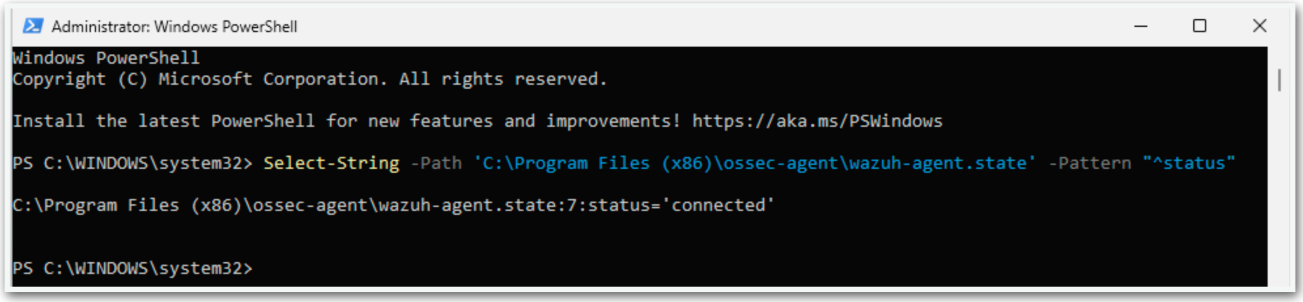
Instance 3 - Windows VM

- Installed Wazuh Agent
- Installed Sysmon with SwiftOnSecurity config file
- Forwarded Sysmon logs to Wazuh dashboard

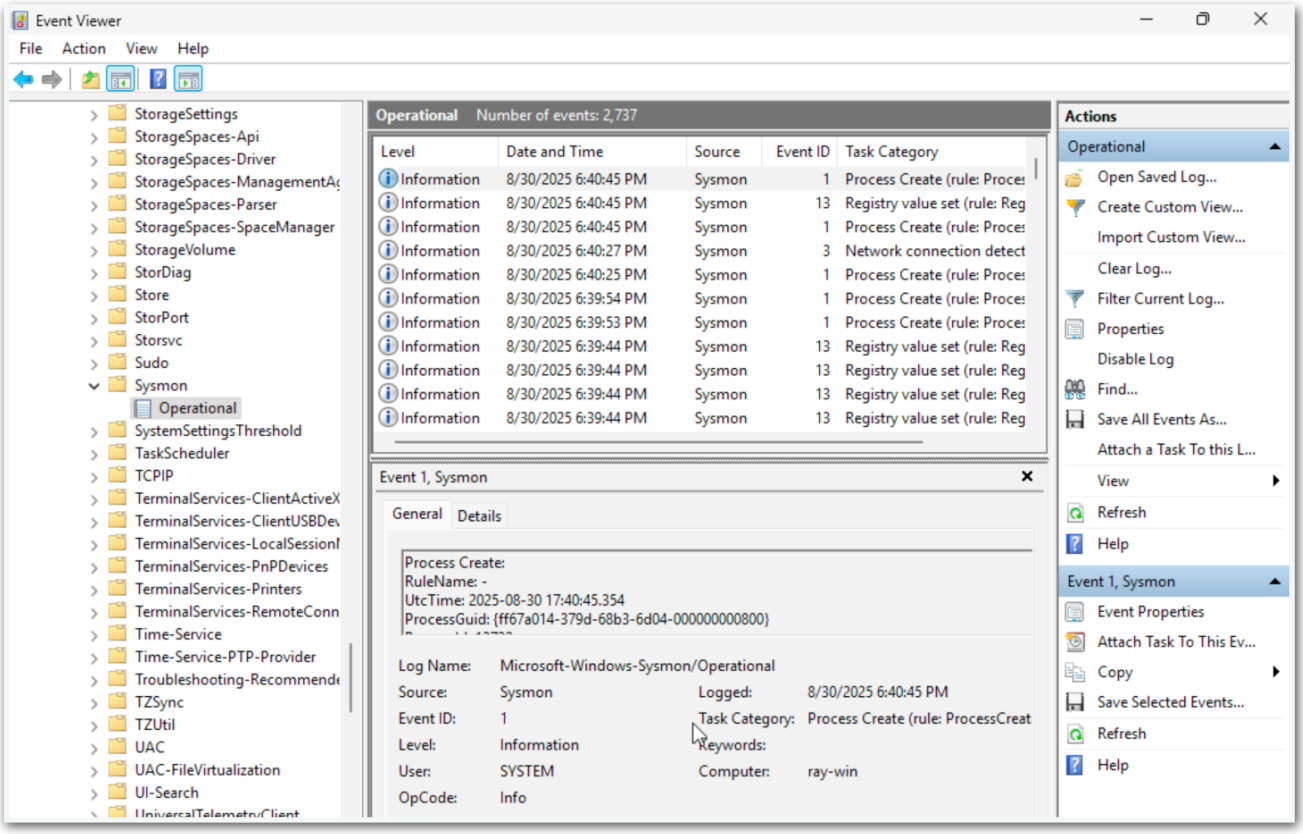
Troubleshoot

Issue	• sysmon driver failed
Diagnosis	• wrong .exe were used - sysmon.exe and sysmon64.exe do not work in ARM architecture
Solution	• used sysmon64a.exe

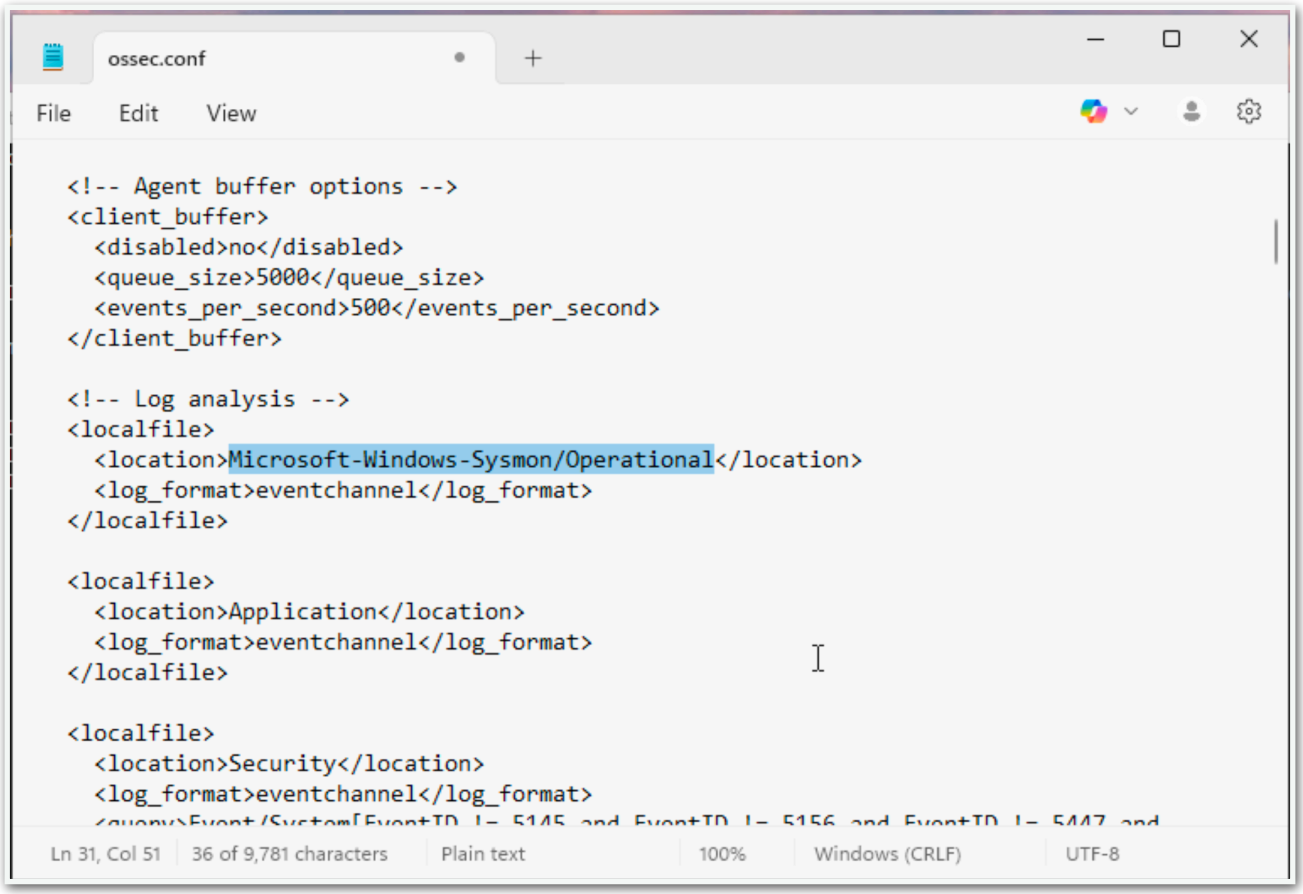
Screenshot - Wazuh agent status on Windows Endpoint



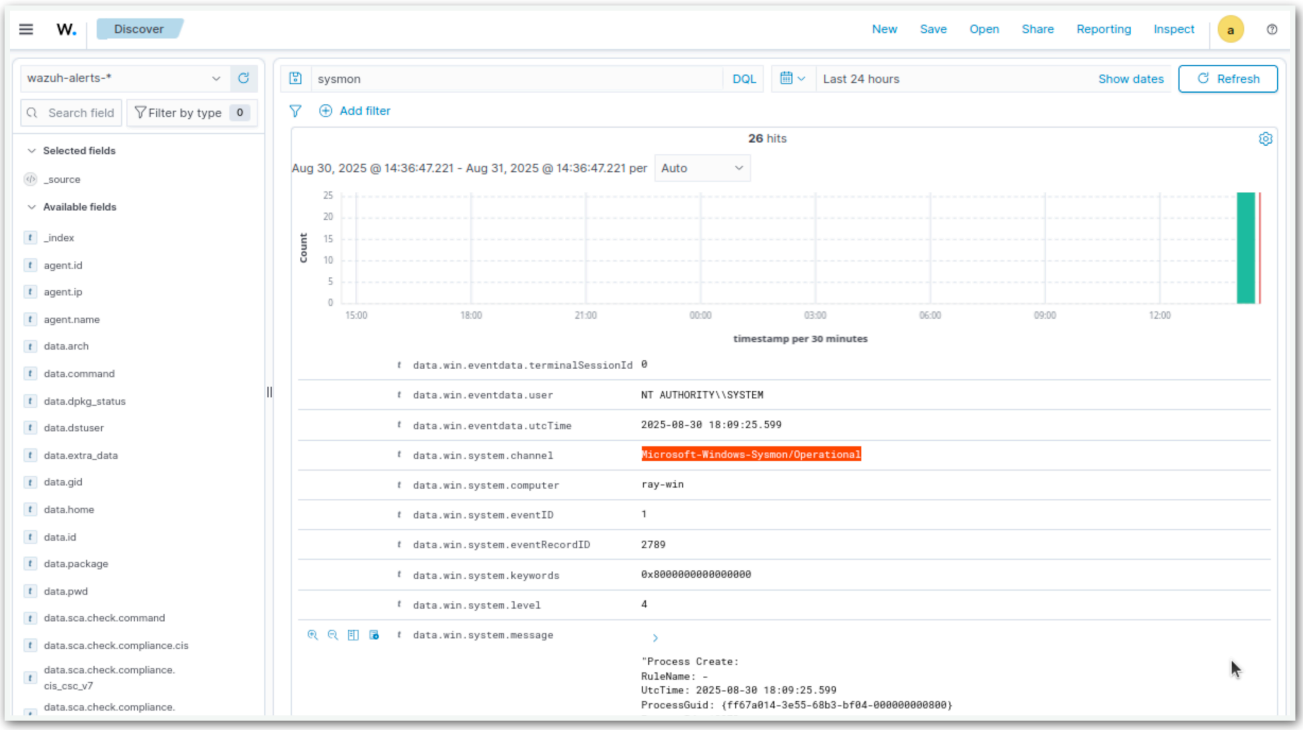
Screenshot - Sysmon logs in Windows Event Viewer on Windows Endpoint



Screenshot - Config to forward Sysmon logs to Wazuh dashboard



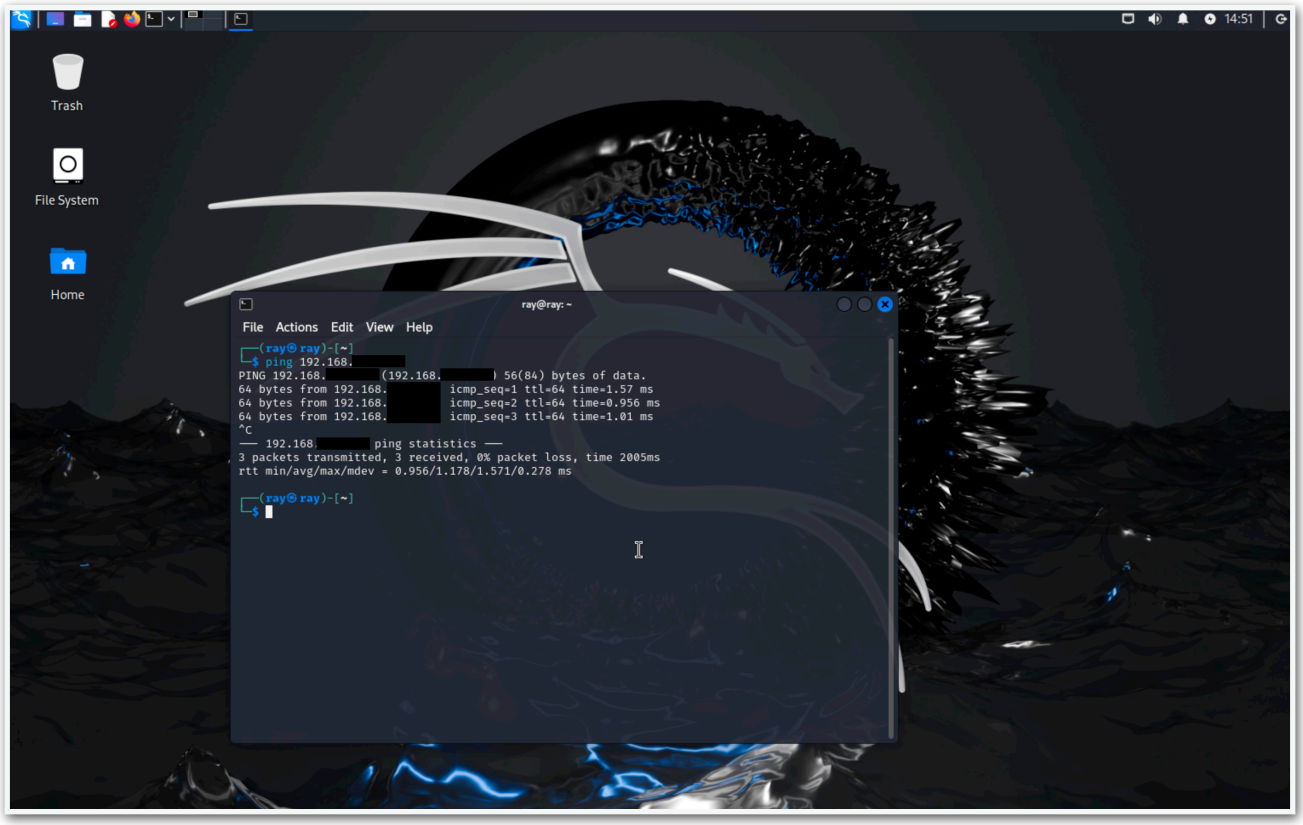
Screenshot - Verify Sysmon logs collection in Wazuh dashboard



Instance 4 - Kali Linux VM

- Deployed for role playing attacker

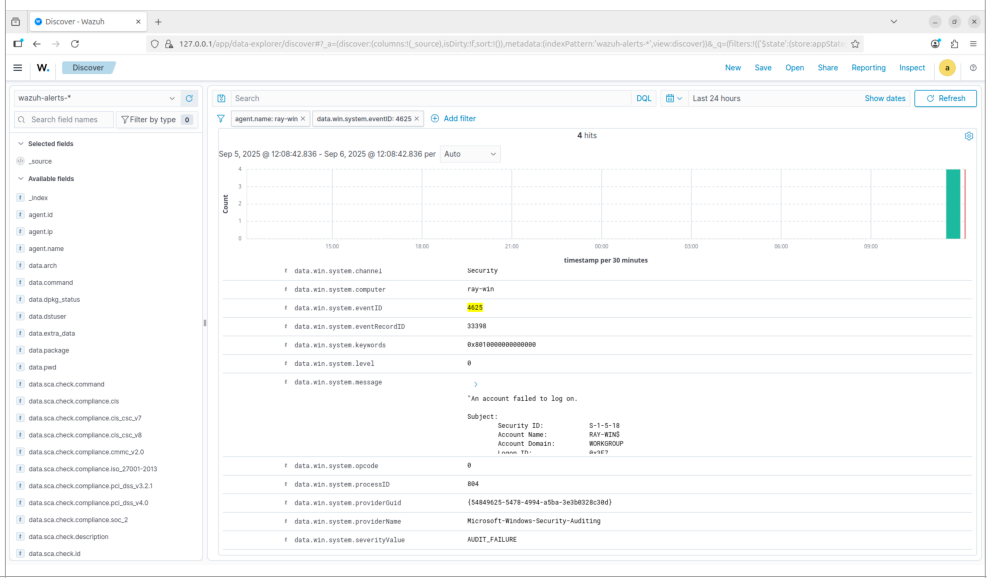
Screenshot - Ping Windows Endpoint on Kali for testing



2. Generate test events + verify alerts on Wazuh dashboard (ongoing)

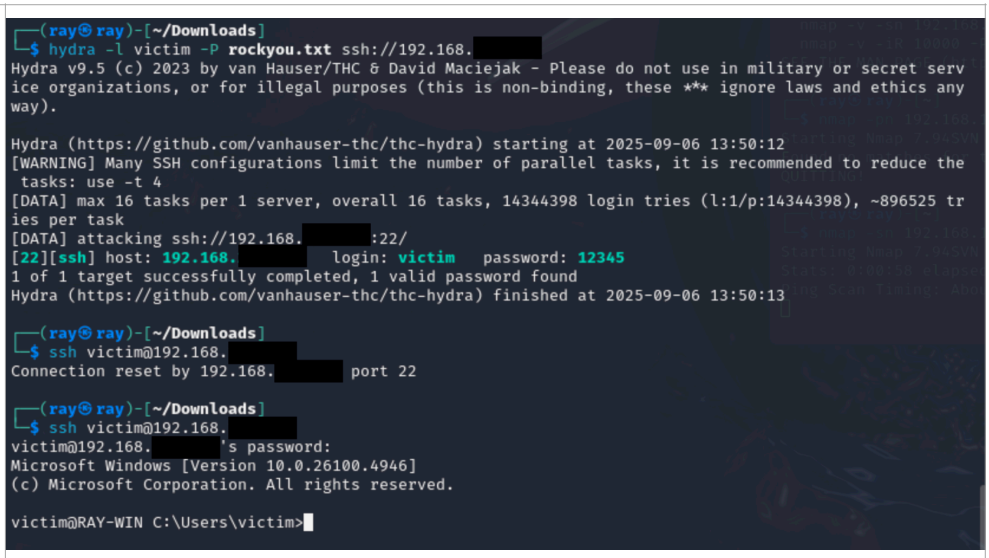
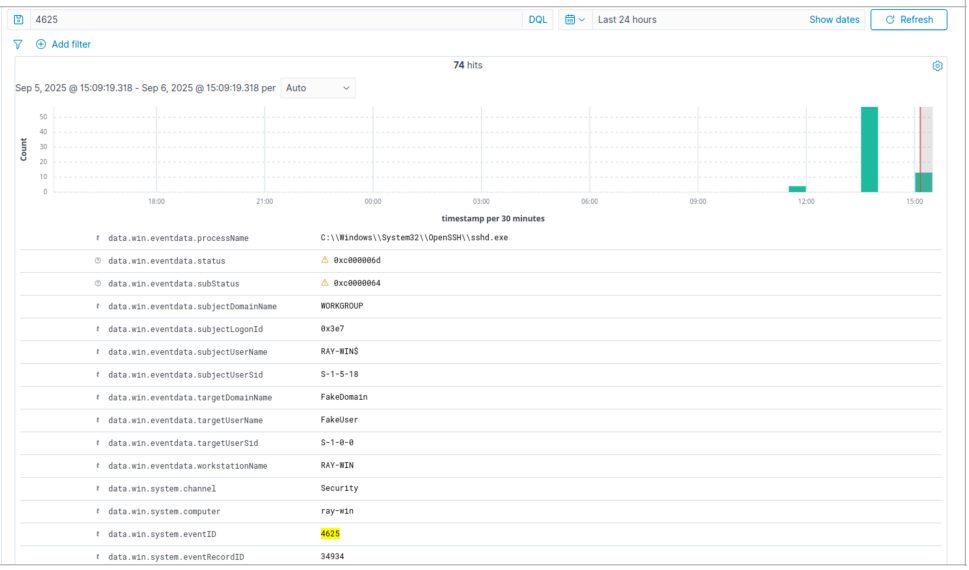
Detect Failed Logon

Test 1

Date	2025-09-06
Target	Windows Endpoint
Method	• Enter wrong password in Windows Lock Screen
Result	• Event detected by agent and shown on Wazuh dashboard • Filtered logs by “data.win.system.eventID: 4625”
Screenshot	

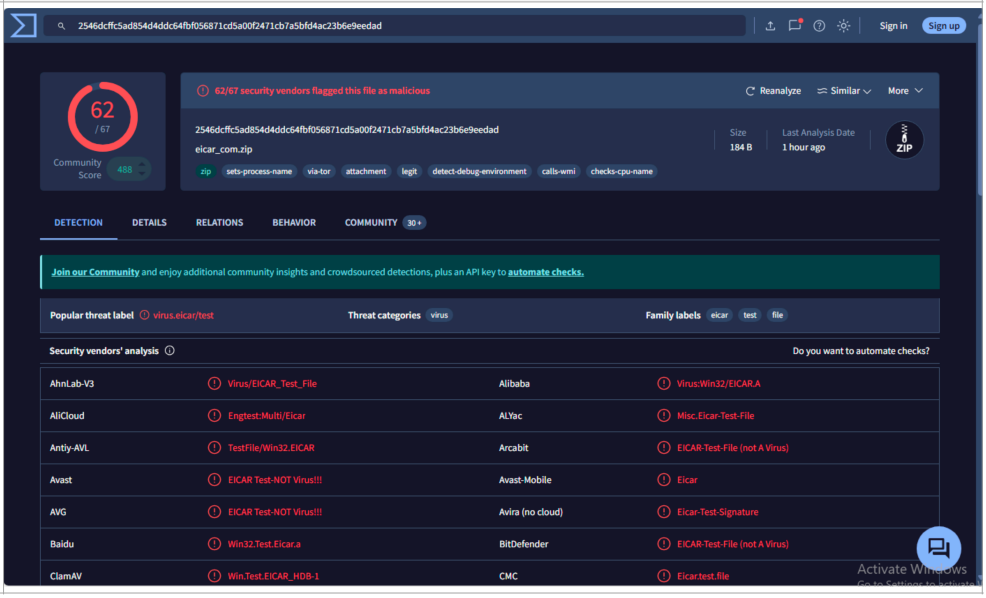
Test 2

Date	2025-09-06
Target	Windows Endpoint
Method	• Simulate brute-force attack on Windows Endpoint • Created “victim” user on Windows with weak password • Use Hydra on Kali Linux to attempt brute-force
Details	# attempt with known user name and password via smb Command: hydra -l victim -p 12345 smb://192.168.x.x Not responding - may be blocked by Windows # attempt nmap scan on Windows endpoint Command: nmap -sV 192.168.x.x No host found - may be blocked by Windows # tried same nmap scan on Ubuntu Command: nmap -sV 192.168.x.x 1 Host up, port 443 and 3389 open # attempt hydra on Ubuntu with known username and password via rdp Command: hydra -l ray -p xxxxx rdp://192.168.x.x [3389][rdp] host: 192.168.x.x login: ray password: xxxxxxx 1 of 1 target successfully completed, 1 valid password found # attempt hydra on Ubuntu with password list Command: hydra -l ray -P rockyou.txt rdp://192.168.x.x [3389][rdp] host: 192.168.x.x login: ray password: 123456789 [3389][rdp] host: 192.168.x.x login: ray password: 123456 [3389][rdp] host: 192.168.x.x login: ray password: 12345 [3389][rdp] host: 192.168.x.x login: ray password: password 1 of 1 target successfully completed, 4 valid passwords found # turn off Windows firewall and nmap scan again Command: nmap -sV 192.168.x.x Host is up (0.00030s latency). Not shown: 997 closed tcp ports (conn-refused) PORT STATE SERVICE VERSION 135/tcp open msrpc Microsoft Windows RPC 139/tcp open netbios-ssn Microsoft Windows netbios-ssn 445/tcp open microsoft-ds? Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows # install and enable OpenSSH on Windows Endpoint - success # connect to Windows via ssh from Kali - success # attempt hydra with password list on Windows using ssh [DATA] attacking ssh://192.168.x.x:22/ [22][ssh] host: 192.168.x.x login: victim password: 12345 1 of 1 target successfully completed, 1 valid password found # use brute-forced password to connect Windows victim Command: ssh victim@192.168.x.x Password: 12345 Success

Screenshot Enumerate password list Found password SSH using found password	
Screenshot Wazuh detected brute-force on dashboard	

Detect Malicious File

Test 1

Date	2025-09-06
Target	Windows Endpoint
Method	Download a test malicious file
Details	# Download malicious file File name: eicar_com.zip File hash: 2546DCFFC5AD854D4DDC64FBF056871CD5A00F2471CB7A5BFD4AC23B6E9EEDAD # Check file hash on VirusTotal 62/67 flagged as malicious # Integrate VirusTotal with Wazuh dashboard <integration> <name>virustotal</name> <api_key>**********</api_key> <group>syscheck</group> <alert_format>json</alert_format> </integration> # Config Wazuh agent to monitor “Downloads” <syscheck> <directories check_all="yes" realtime="yes">C:\Users*\Downloads</directories> </syscheck> # Test download and flagged by Wazuh - success
Screenshot File hash check on VirusTotal	

Screenshot VirusTotal API integration in config file	<add_labels>yes</add_labels> </wodle> <!-- VirusTotal integration --> <integration> <name>virustotal</name> <api_key>1b[redacted]e5</api_key> <group>syscheck</group> <alert_format>json</alert_format> </integration> <!-- System inventory --> <wodle name="syscollector"> <disabled>no</disabled> <interval>1h</interval>
Screenshot VirusTotal sys check config	<!-- Directories to check (perform all possible verifications) --> <directories check_all="yes" realtime="yes">C:\Users*\Downloads</directories> <directories>/etc,/usr/bin,/usr/sbin</directories> <directories>/bin,/sbin,/boot</directories>
Screenshot Malicious file download alert on dashboard from VirusTotal API integration	

3. Hardening endpoints (ongoing)

Configuration Scanning

Date	2025-08-30
Endpoint	MacBook
Benchmark	CIS_Apple_macOS_15.0_Sequoia_Benchmark_v1.0.0
Result	Passed 36 Failed 23 NA 2 Score 61%
Vulnerability 1	CVE-2022-40898 An issue discovered in Python Packaging Authority (PyPA) Wheel 0.37.1 and earlier allows remote attackers to cause a denial of service via attacker controlled input to wheel cli.
Remediation	# Before pip ver 21.2.4 ; wheel ver 0.37.0 # After pip ver 25.2 ; wheel ver 0.45.1 # Command used /Library/Developer/CommandLineTools/usr/bin/python3 -m pip install --upgrade pip python3 -m pip install --upgrade wheel
Vulnerability 2	CVE-2022-40899 An issue discovered in Python Charmers Future 0.18.2 and earlier allows remote attackers to cause a denial of service via crafted Set-Cookie header from malicious web server.
Remediation	# Before future ver 0.18.2 # After future ver 1.0.0 # Command used python3 -m pip install --upgrade future
Vulnerability 3	CVE-2024-6345 A vulnerability in the package_index module of pypa/setuptools versions up to 69.1.1 allows for remote code execution via its download functions. These functions, which are used to download packages from URLs provided by users or retrieved from package index servers, are susceptible to code injection. If these functions are exposed to user-controlled inputs, such as package URLs, they can execute arbitrary commands on the system. The issue is fixed in version 70.0.

Remediation	# Before setuptools ver 58.0.4 # After setuptools 80.9.0 #Command used: python3 -m pip install --upgrade setuptools
Vulnerability 4	CVE-2024-44142 The issue was addressed with improved bounds checks. This issue is fixed in GarageBand 10.4.12. Processing a maliciously crafted image may lead to arbitrary code execution.
Remediation	removed GarageBand

— — — — — — — — — — — — — —

Date	2025-08-30
Endpoint	Windows VM
Benchmark	CIS Microsoft Windows 11 Enterprise Benchmark v3.0.0
Result	Pass 124 Failed 348 NA 10 Score 26%