Information Assurance Project Report

Title: Implementing Automated File Integrity Monitoring (FIM) for Critical Asset Protection using Wazuh

Submitted By:
Abdul Rafay (Roll No: **243705**)

Course & Instructor:
Subject Instructor: Dr. Ahmed Naeem
Subject: Information Assurance

Institution Details:
Air University, Multan Campus
Spring 2025

Table of Contents

1. Abstract 3
2. Introduction 3
3. System Architecture & Lab Environment 4
4. Implementation Methodology 4
   4.1. Phase 1: Wazuh Server Installation & Configuration 4
   4.2. Phase 2: Wazuh Agent Installation & Configuration on Windows 7
   4.3. Phase 3: Kali Linux Installation & Configuration 10
   4.4. Phase 4: Establishing Connections (SSH & RDP) 11
   4.5. Phase 5: Connecting Windows & Kali for Simulation 13
5. Configuration of Security Policies 13
   5.1. Enabling File Integrity Monitoring (FIM) 13
   5.2. Windows Security Adjustments 14
6. Attack Simulation & Execution 15
   6.1. Scenario 1: RDP Brute-Force Attack 15
   6.2. Scenario 2: Malicious File Injection for FIM Test 16
   6.3. Scenario 3: File Modification and Deletion via SSH from Kali Linux (Simulated Remote Access) 17
7. Results & Log Analysis 21
   7.1. RDP Attack Logs 21
   7.2. FIM Alert Analysis 21
8. Challenges & Lessons Learned 21
9. Conclusion 22
10. Future Work 22
11. References 23
12. List of Figures 24

1. Abstract

This project focuses on deploying a Security Information and Event Management (SIEM) solution using Wazuh to monitor, detect, and respond to security threats in a simulated networked environment. The scope includes installing and configuring the Wazuh Server, deploying the Wazuh Agent on a Windows machine for File Integrity Monitoring (FIM), installing Kali Linux, establishing secure connections (including SSH and RDP), simulating attacks between Windows and Kali machines, analyzing generated logs, and monitoring file integrity changes. The demonstration highlights Wazuh's ability to correlate logs for threat identification, such as attacker IP and hostname during brute-force attempts. This setup emulates real-world cybersecurity practices, emphasizing the importance of intrusion detection and integrity assurance.

2. Introduction

In today's digital landscape, organizations face increasing cyber threats, including unauthorized access and file tampering. Host-Based Intrusion Detection Systems (HIDS) like Wazuh provide essential tools for real-time monitoring and alerting. This project simulates an internal network attack scenario using virtual machines to showcase Wazuh's capabilities in log collection, analysis, and FIM.

Objectives:

- Deploy and configure Wazuh Server using OVA appliance.
- Install and connect Wazuh Agent on a Windows endpoint with File Integrity Monitoring.
- Set up Kali Linux as an attacker machine and establish connectivity via RDP and SSH.
- Simulate realistic attacks (RDP brute-force and file injection).
- Analyze generated logs and alerts in the Wazuh dashboard to identify threats.

The project uses open-source tools to ensure reproducibility and educational value.

3. System Architecture & Lab Environment

A virtualized lab environment was created to isolate the simulation and prevent real-world risks. The setup uses Oracle VirtualBox as the hypervisor for all machines.

- **SIEM Server:** Wazuh Server (version 4.x) deployed via OVA file. IP: 192.168.56.102
- **Victim Machine:** Windows 10/11. IP: 192.168.56.1 (Host-Only Network).
- **Attacker Machine:** Kali Linux Rolling Edition. IP: Dynamic (Host-Only Network)
- **Network Configuration**: All machines on a Host-Only Adapter for internal communication.
- **Hypervisor:** Oracle VirtualBox (version 7.x).

This architecture allows controlled interactions, such as RDP connections from Kali to Windows and log forwarding to Wazuh.

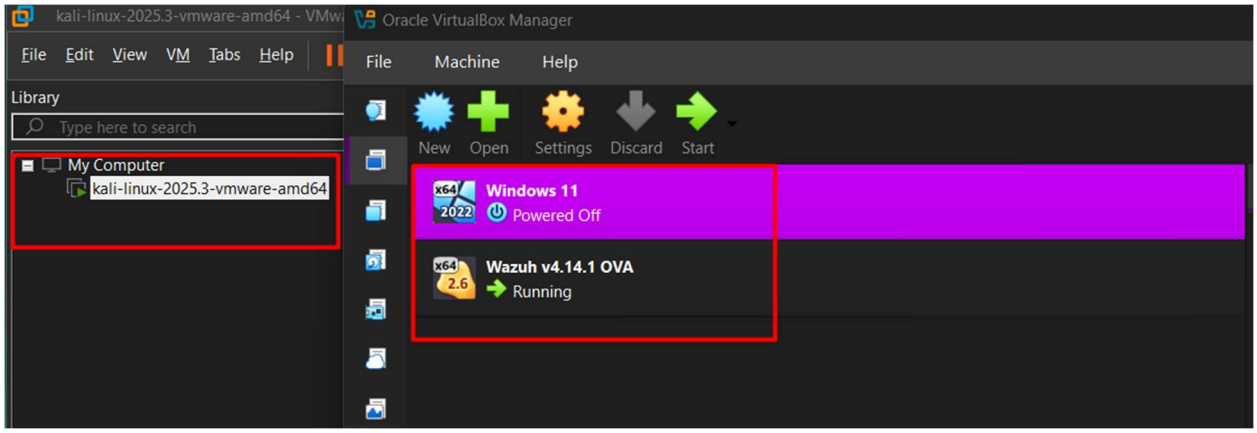


Figure 1: Overview of VirtualBox VM List Showing Wazuh, Windows, and Kali Machines

4. Implementation Methodology

4.1. Phase 1: Wazuh Server Installation & Configuration

The Wazuh Server was deployed using a pre-configured OVA appliance for efficiency and stability.

Steps Performed:

1. Downloaded the Wazuh OVA file **wazuh-4.x.ova** from the official Wazuh website.
2. Opened VirtualBox and imported the OVA: Selected File > Import Appliance, browsed to the downloaded OVA file, reviewed the settings (e.g., CPU: 2, RAM: 4GB), and clicked Import.
3. Configured the imported VM: Right-clicked the VM > Settings > System > Allocated 4GB RAM and 2 CPUs; Network > Adapter 1 > Enabled and set to Host-Only Adapter (e.g., vboxnet0).
4. Powered on the VM.
5. Logged in via the console with default credentials: Username: wazuh-user, Password: wazuh.
6. Verified the IP address using the command:

   : Ip a

   - Noted the inet address under eth0 or similar interface (e.g., 192.168.56.102).
7. Switched to root if needed: sudo -i.
8. Accessed the Wazuh web dashboard via HTTPS (SSL-enabled by default) at https://192.168.56.102 using a web browser on the host machine.
9. Logged into the dashboard with default or generated credentials: Username: admin, Password: admin
10. Verified cluster status and components (Wazuh manager, indexer, dashboard) in the dashboard overview.

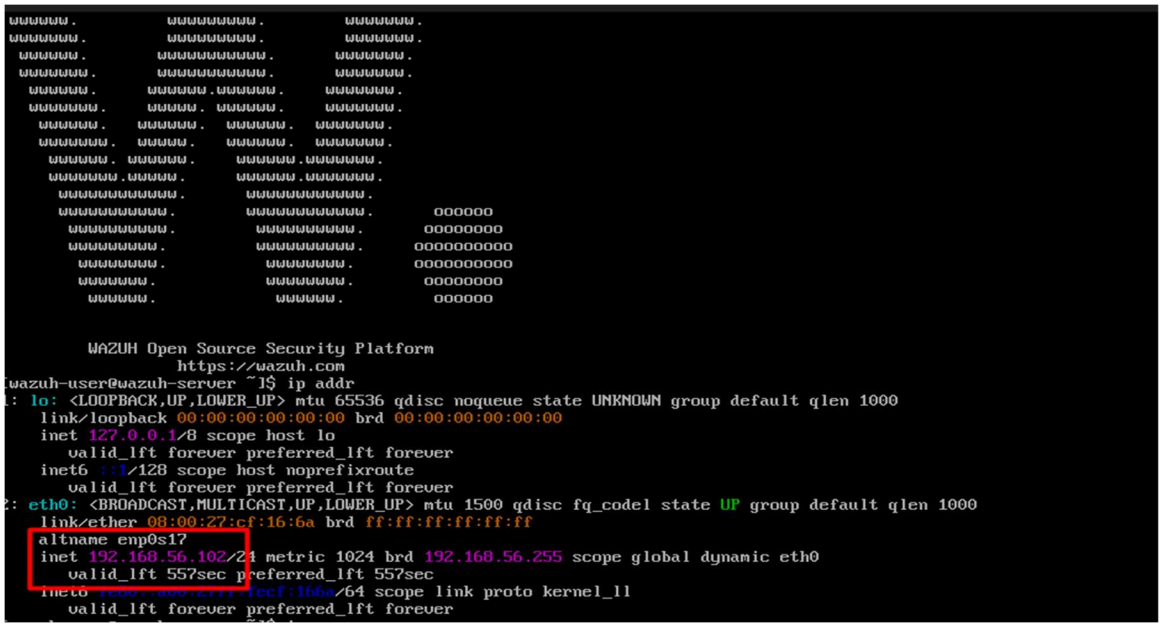


Figure 2: Wazuh Server Console Login and IP Address Verification (ip a command)

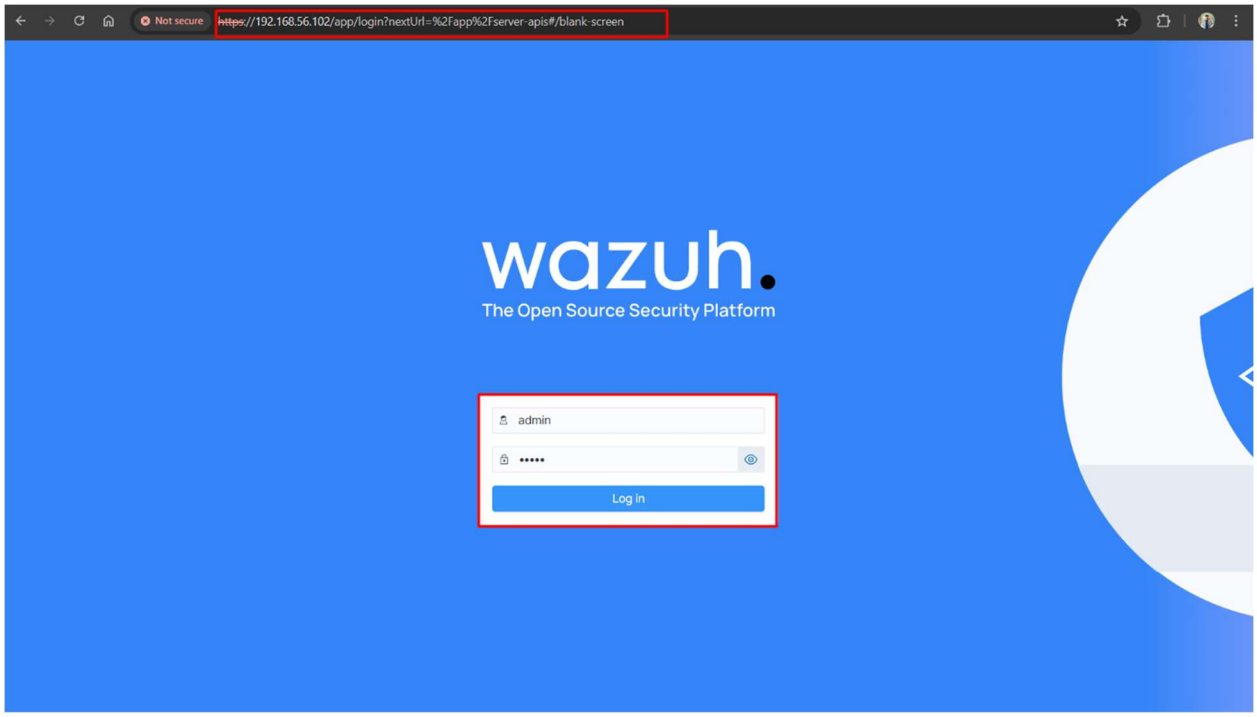


Figure 3: Wazuh Dashboard Login Page

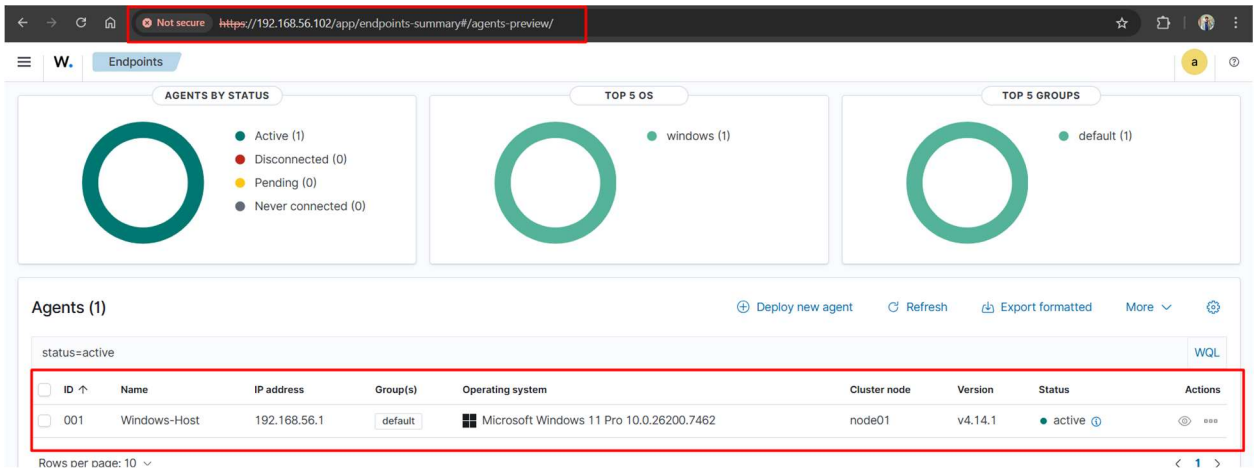


Figure 4: Wazuh Dashboard Overview Showing Cluster Status and Agents

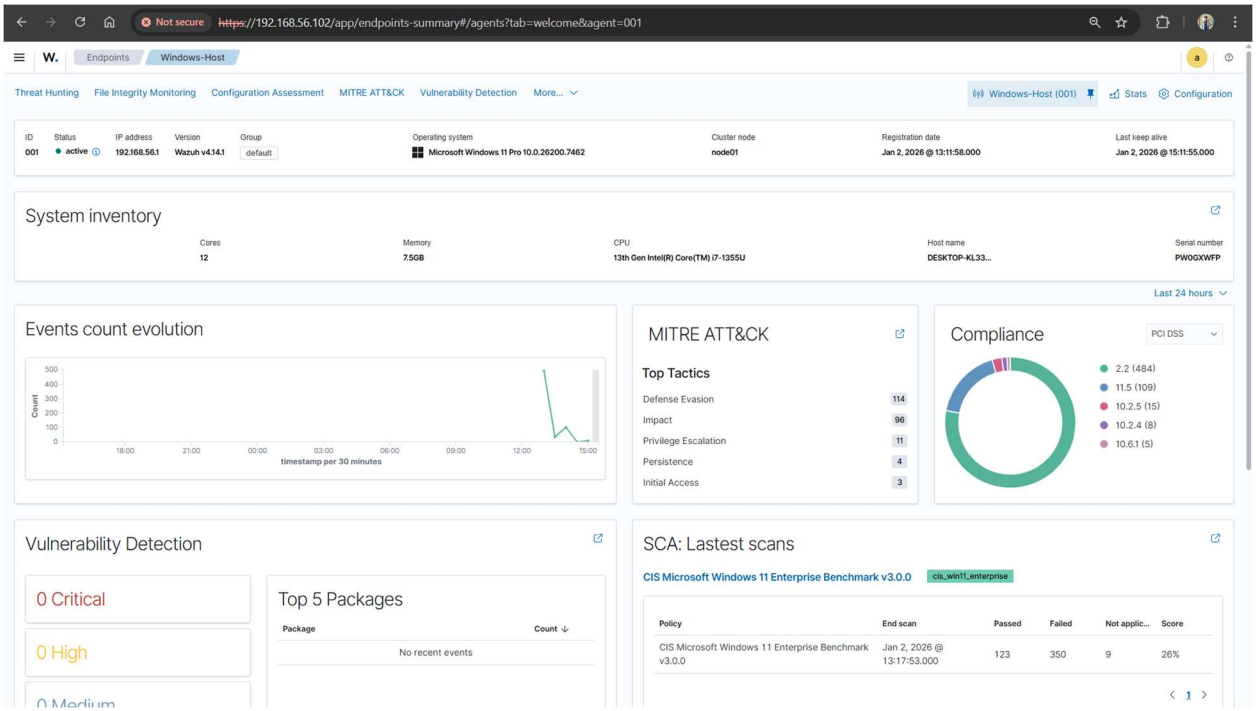


Figure 5: Wazuh Dashboard General Alerts Overview

4.2. Phase 2: Wazuh Agent Installation & Configuration on Windows

The Wazuh Agent was installed on the Windows victim machine to enable log collection and FIM.

Steps Performed:

1. On the Wazuh Server, accessed the CLI (via SSH or console) and generated an agent authentication key:
    - Ran sudo /var/ossec/bin/manage_agents.
    - Selected 'A' to add a new agent.
    - Entered agent name: Windows-Host.
    - Entered IP: any (or a specific IP like 192.168.56.1).
    - Confirmed and extracted the unique authentication key (e.g., a long base64 string).
    - Exited the tool with 'Q'.
2. On the Windows machine:

- Downloaded the Wazuh Agent MSI installer (e.g., wazuh-agent-4.x.msi) from https://packages.wazuh.com/4.x/windows/.
- Opened PowerShell as Administrator.
- Navigated to the download directory.
- Installed the agent using the command:
  `.\wazuh-agent-4.x.msi /q WAZUH_MANAGER="192.168.56.102"`
    - For additional options: Added WAZUH_AGENT_NAME="Windows-Host", WAZUH_AGENT_GROUP="default" if needed.

3. Launched the Wazuh Agent Manager: Navigated to C:\Program Files (x86)\ossec-agent\win32ui.exe, ran as Administrator.
4. In the Agent Manager: Selected Manage > Authentication Key > Imported the extracted key from the server.
5. Verified the connection: Selected View > Status, ensured it showed "Connected".
6. Restarted the Wazuh service: Opened Services.msc, found "WazuhSvc", right-clicked > Restart (or via PowerShell: Restart-Service -Name wazuhsvc).
7. On the Wazuh dashboard: Navigated to Agents > Verified the Windows-Host agent appeared as "Active."

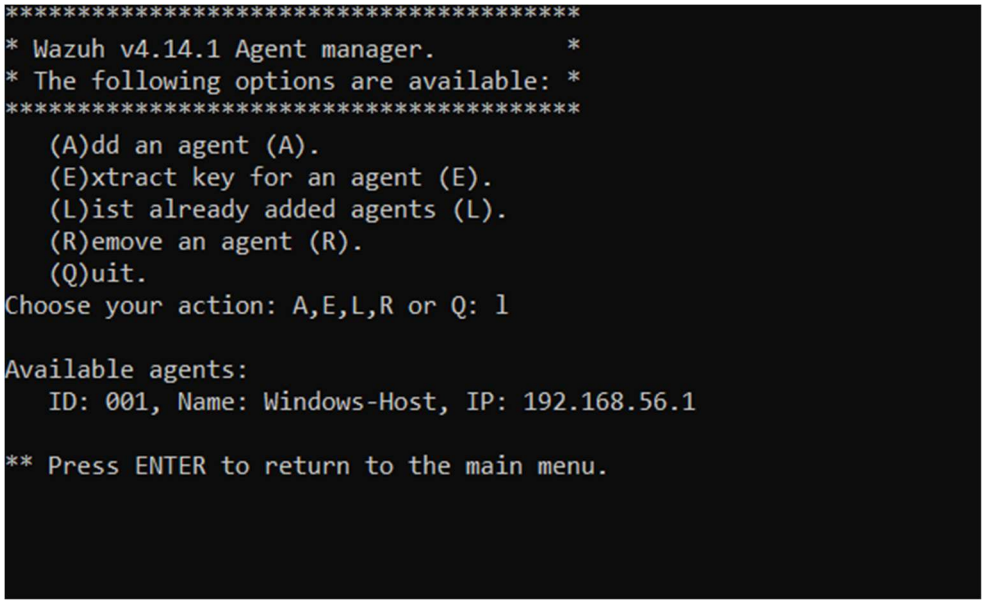


Figure 6: Wazuh manage_agents Tool Showing Added Windows Agent and Key

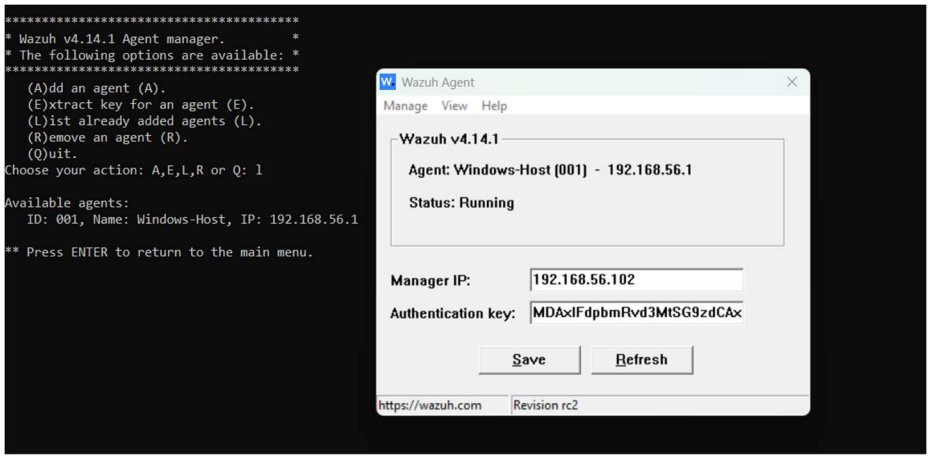


Figure 7: Windows Wazuh Agent Manager Showing Imported Key and Connected Status

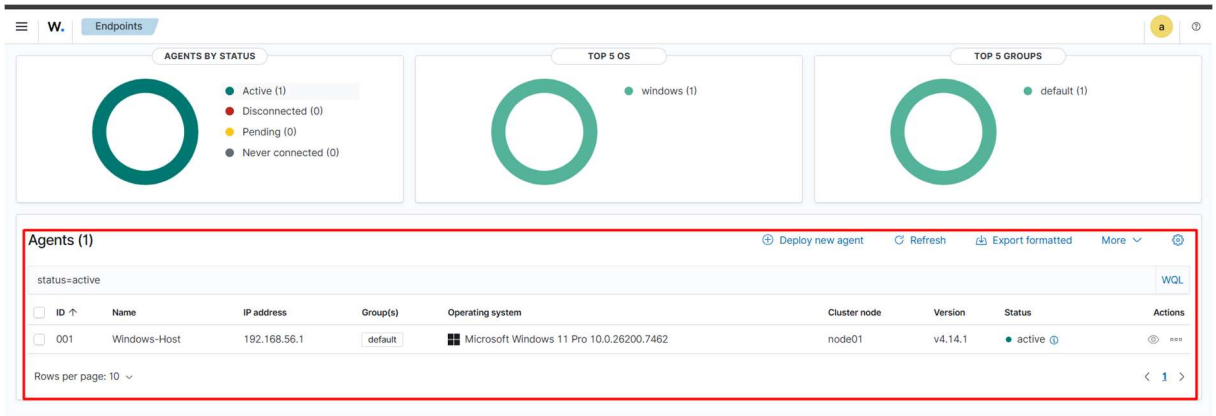


Figure 8: Wazuh Dashboard Agents List with Windows Agent Active

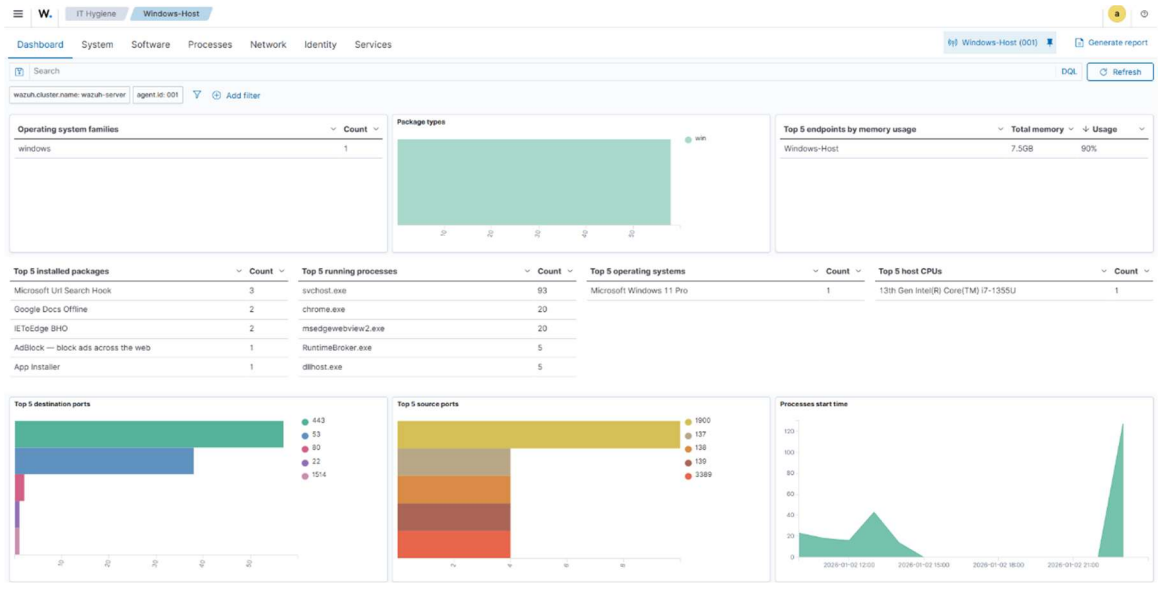


Figure 9: Detailed View of Windows Agent Status in Wazuh Dashboard

4.3. Phase 3: Kali Linux Installation & Configuration

Kali Linux was installed as the attacker machine to simulate threats.

Steps Performed:

1. Downloaded the Kali Linux ISO (Rolling Edition, e.g., kali-linux-202x.x-installer-amd64.iso) from https://www.kali.org/get-kali/.
2. In VirtualBox: Selected Machine > New.
3. Configured the VM: Name: Kali-Attacker, Type: Linux, Version: Debian (64-bit), Memory: 2048 MB, Hard Disk: 20GB, CPUs: 2, Network: Host-Only Adapter.
4. Mounted the ISO and started installation (Graphical Install).
5. Configured language, hostname (kali), partitions, GRUB, etc.
6. Post-installation updates and installed freerdp3-x11.

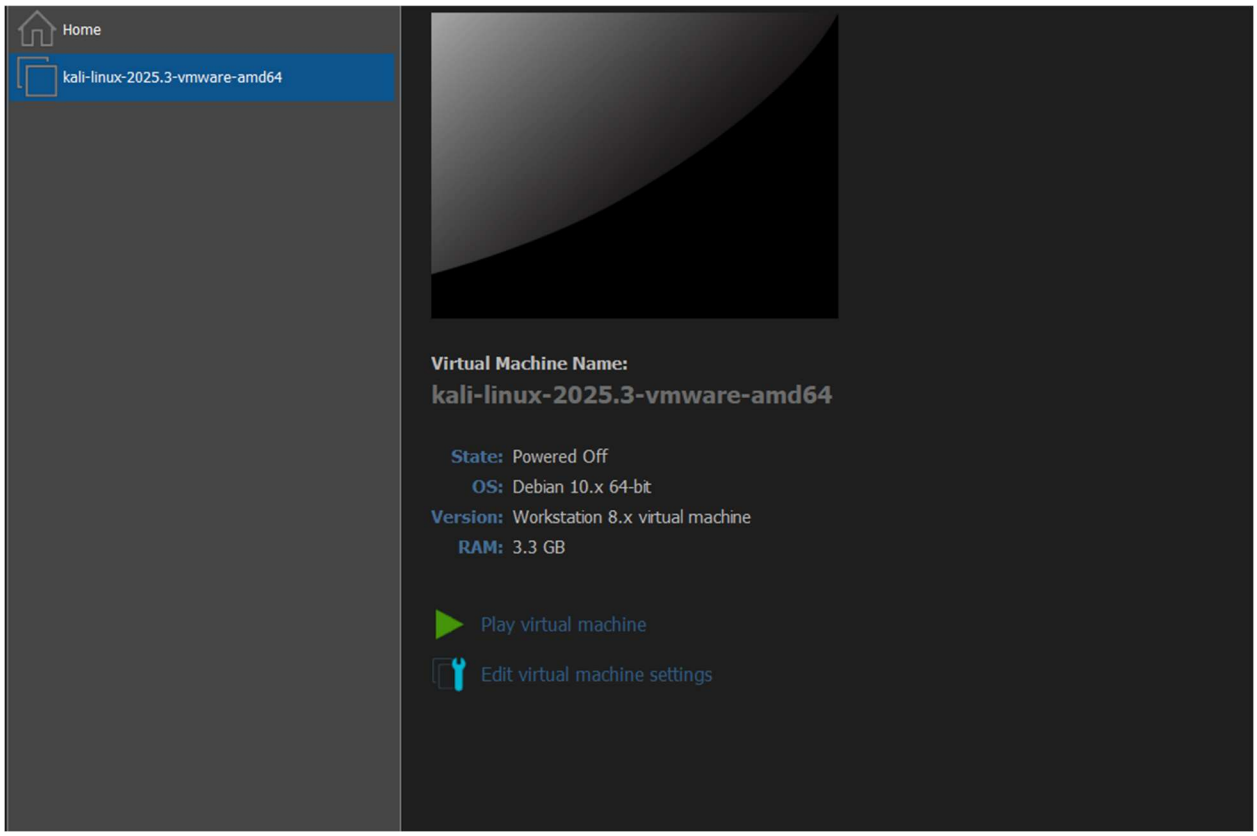


Figure 10: Kali Linux Boot Menu Selecting Graphical Install

Figure 11: Kali Linux Installation Wizard (Language/Hostname Setup)

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ca:fd:44 brd ff:ff:ff:ff:ff:ff
    inet 192.168.142.128/24 brd 192.168.142.255 scope global dynamic noprefixroute eth0
       valid_lft 1452sec preferred_lft 1452sec
    inet6 fe80::4f40:a6ef:3604:d8ed/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=128 time=1.20 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=128 time=1.36 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=128 time=2.24 ms
^C
── 192.168.56.1 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 1.199/1.599/2.236/0.455 ms
```

Figure 12: Kali Linux Terminal After Installation Showing ip a and Ping to Windows

4.4. Phase 4: Establishing Connections (SSH & RDP)

- SSH Connection to Wazuh Server: From Windows PowerShell.
- RDP Connection Setup: From Kali to Windows.

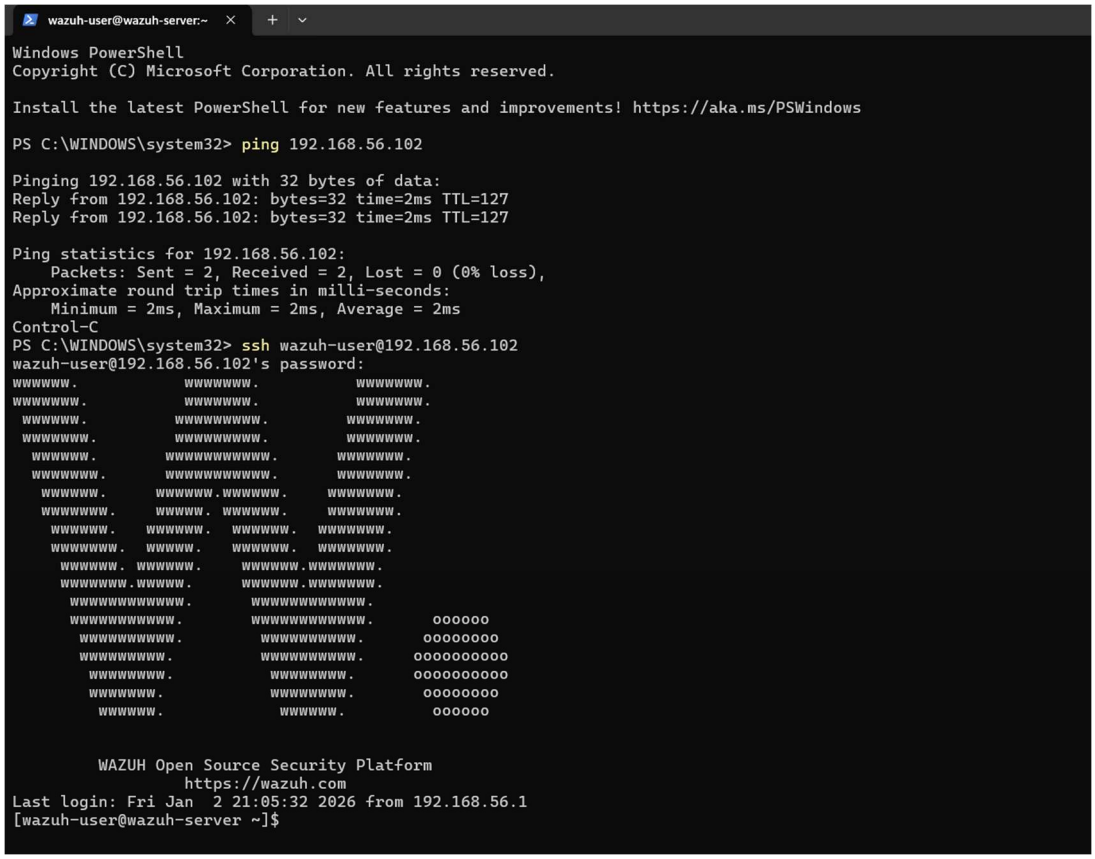


Figure 13: Windows PowerShell SSH Connection to Wazuh Server

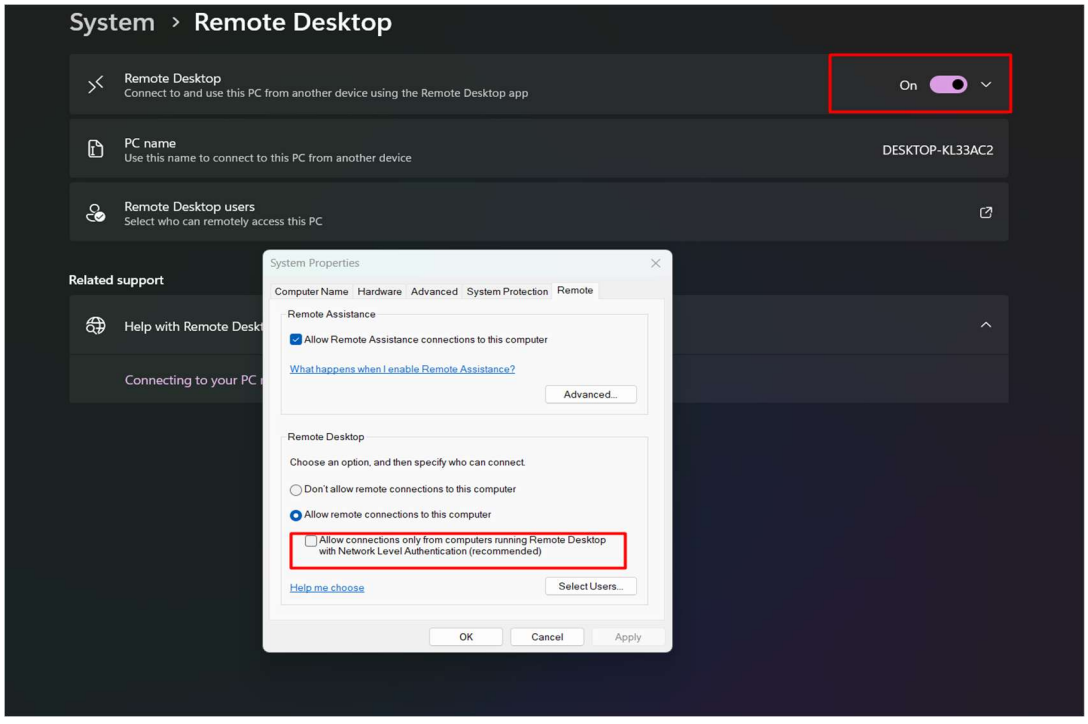


Figure 14: Windows Remote Desktop Settings with RDP Enabled and NLA Disabled

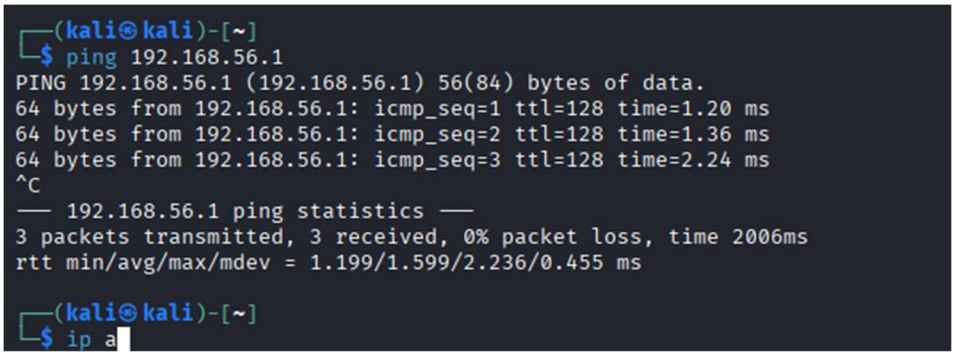


Figure 15: Kali Terminal Ping to Windows Machine Confirming Connectivity

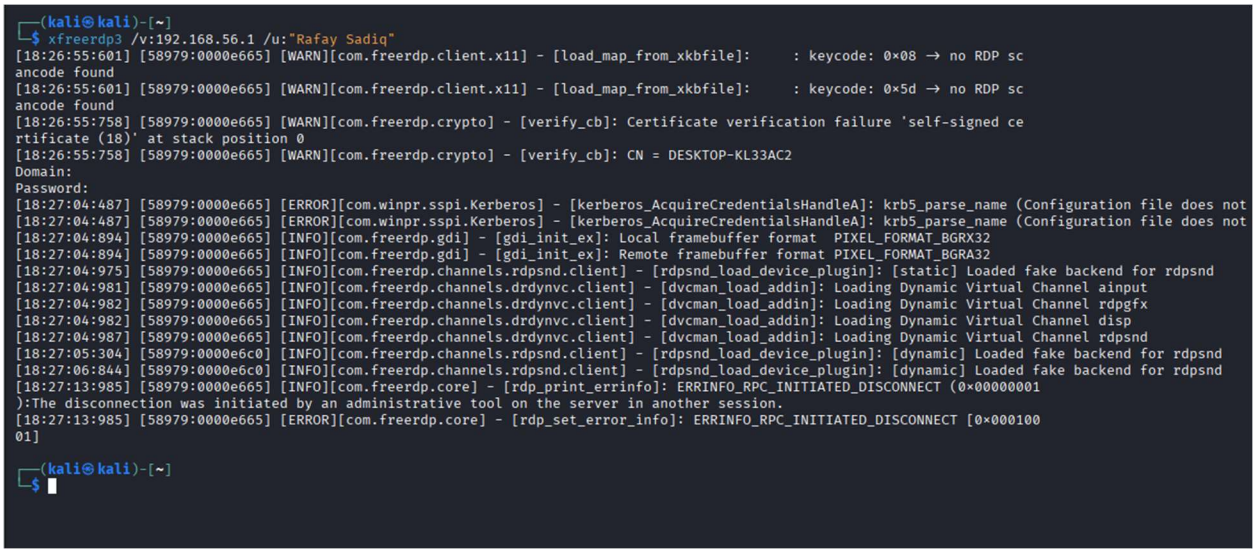


Figure 16: Kali Terminal Successful RDP Test Connection (Non-Attack)

4.5. Phase 5: Connecting Windows & Kali for Simulation

Connectivity verified via ping and test RDP.

5. Configuration of Security Policies

5.1. Enabling File Integrity Monitoring (FIM)

Steps Performed:

1. Edited ossec.conf on Windows agent.
2. Added directory monitoring for Desktop.

3. Restarted agent.

```
  <max_eps>50</max_eps>

  <!-- Database synchronization settings -->
  <synchronization>
    <enabled>yes</enabled>
    <interval>5m</interval>
    <max_eps>10</max_eps>
  </synchronization>


  <frequency>3600</frequency>
  <auto_ignore>yes</auto_ignore>
  <scan_on_start>yes</scan_on_start>

  <directories check_all="yes" realtime="yes">C:\Users\Rafay Sadiq\Downloads</directories>
  <directories check_all="yes" realtime="yes">C:\Users\Rafay Sadiq\Desktop</directories>
  <directories check_all="yes" realtime="yes">C:\Users\Rafay Sadiq\Documents</directories>

</syscheck>
```

Figure 17: Edited ossec.conf File Showing FIM Directory Configuration

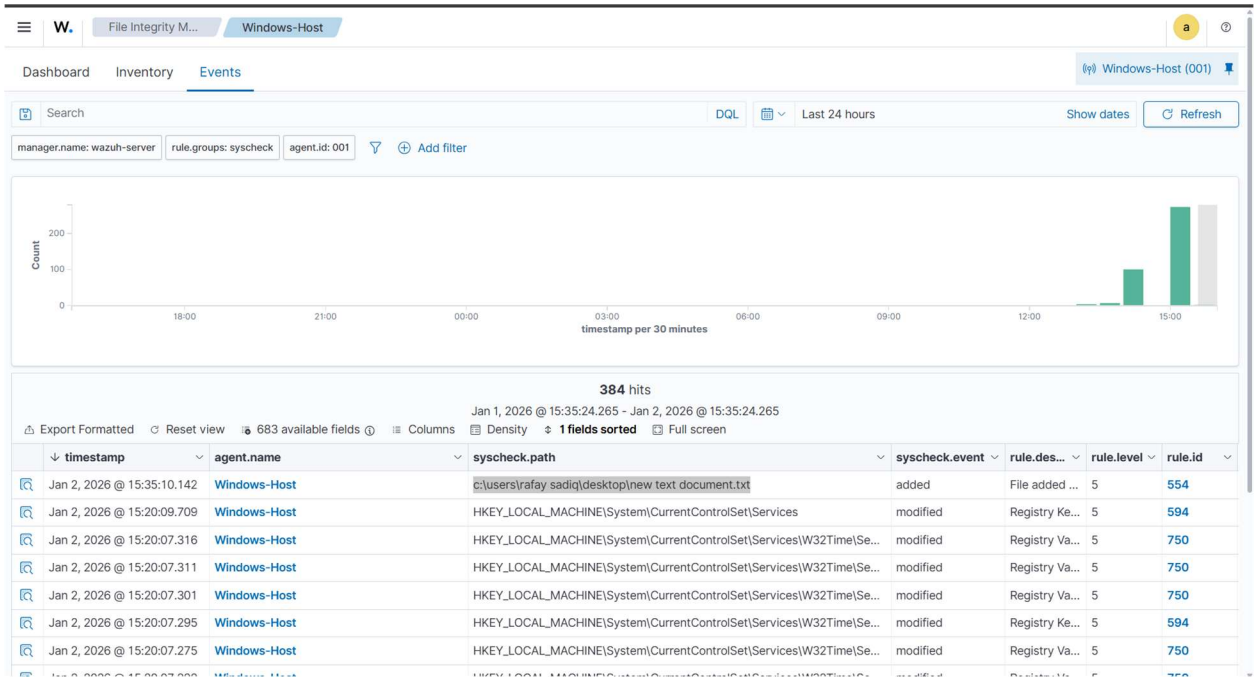


Figure 18: Wazuh Dashboard Agent Configuration View Confirming FIM Settings

5.2. Windows Security Adjustments

Enabled RDP and disabled NLA

6. Attack Simulation & Execution

6.1. Scenario 1: RDP Brute-Force Attack

Multiple failed **login** attempts from Kali.

```
┌──(kali㉿kali)-[~]
└─$ xfreerdp3 /v:192.168.56.1 /u:"Rafay Sadiq"
[18:36:06:755] [63454:0000f7e0] [WARN][com.freerdp.client.x11] - [load_map_from_xkbfile]:     : keycode: 0×08 → no RDP scancode found
[18:36:06:755] [63454:0000f7e0] [WARN][com.freerdp.client.x11] - [load_map_from_xkbfile]:     : keycode: 0×5d → no RDP scancode found
[18:36:06:848] [63454:0000f7e0] [WARN][com.freerdp.crypto] - [verify_cb]: Certificate verification failure 'self-signed certificate (18)' at stack pos
ition 0
[18:36:06:848] [63454:0000f7e0] [WARN][com.freerdp.crypto] - [verify_cb]: CN = DESKTOP-KL33AC2
Domain:
Password:
[18:36:10:193] [63454:0000f7e0] [ERROR][com.winpr.sspi.Kerberos] - [kerberos_AcquireCredentialsHandleA]: krb5_parse_name (Configuration file does not
specify default realm [-1765328160])
[18:36:10:194] [63454:0000f7e0] [ERROR][com.winpr.sspi.Kerberos] - [kerberos_AcquireCredentialsHandleA]: krb5_parse_name (Configuration file does not
specify default realm [-1765328160])
[18:36:10:211] [63454:0000f7e0] [ERROR][com.freerdp.core] - [nla_recv_pdu]: ERRCONNECT_LOGON_FAILURE [0×00020014]
[18:36:10:211] [63454:0000f7e0] [ERROR][com.freerdp.core.rdp] - [rdp_recv_callback_int][0×55a19f3c0790]: CONNECTION_STATE_NLA - nla_recv_pdu() fail
[18:36:10:211] [63454:0000f7e0] [ERROR][com.freerdp.core.rdp] - [rdp_recv_callback_int][0×55a19f3c0790]: CONNECTION_STATE_NLA status STATE_RUN_FAILED
[-1]
[18:36:10:211] [63454:0000f7e0] [ERROR][com.freerdp.core.transport] - [transport_check_fds]: transport_check_fds: transport→ReceiveCallback() - STATE
_RUN_FAILED [-1]
```

Figure 19: Kali Terminal Executing xfreerdp3 Brute-Force Attempts with Errors

Figure 20: Wazuh Dashboard Security Events Showing RDP Authentication Failures

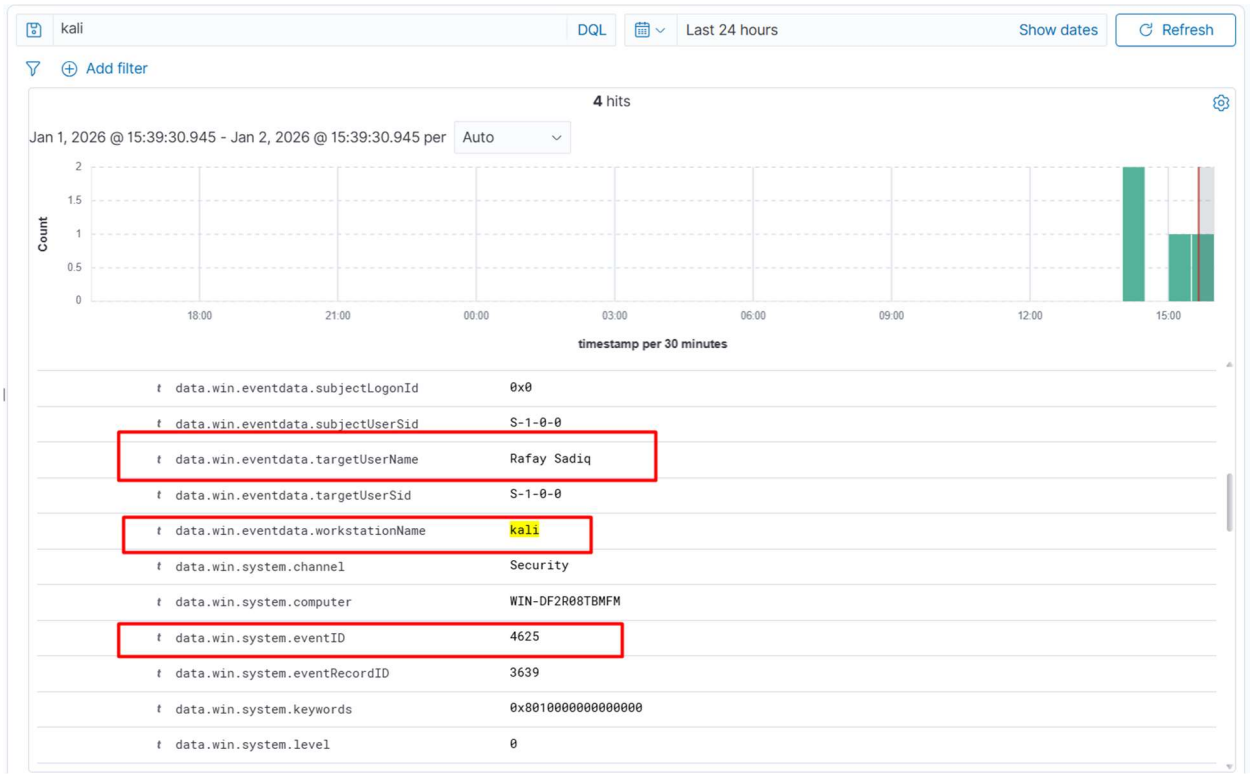


Figure 21: Wazuh Dashboard Alert Details for Brute-Force Attack with Attacker Source (kali)

6.2. Scenario 2: Malicious File Injection for FIM Test

Created rafay.txt on monitored Desktop.

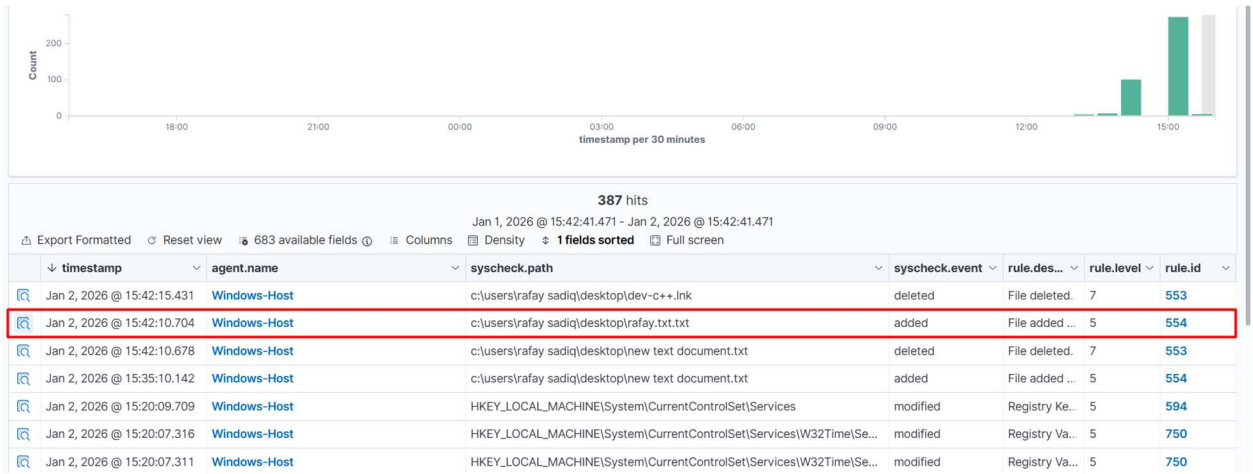


Figure 22: Windows Desktop Showing Newly Created rafay.txt File

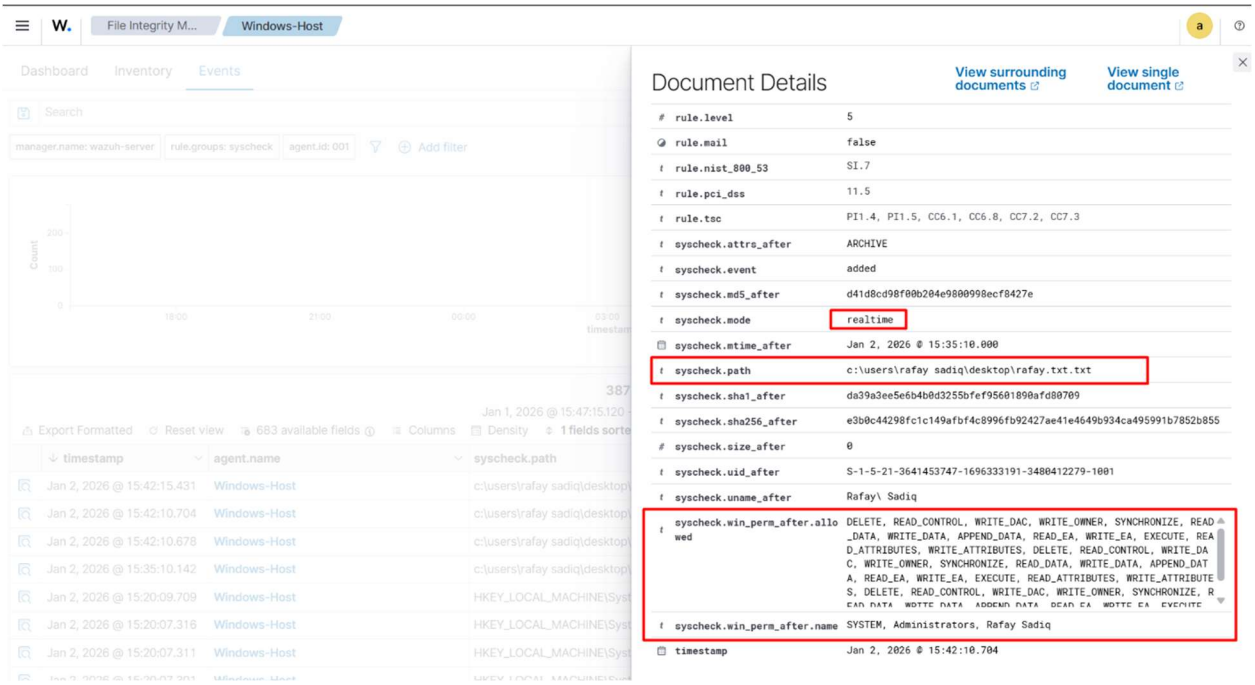


Figure 23: Wazuh Dashboard FIM Alert for File Addition

6.3. Scenario 3: File Modification and Deletion via SSH from Kali Linux (Simulated Remote Access)

To further test the robustness of File Integrity Monitoring, OpenSSH was enabled on the Windows victim machine to allow remote command execution from Kali Linux (simulating an attacker gaining SSH access after initial compromise). Once connected via SSH, the monitored Desktop directory was accessed remotely, and file tampering actions were performed using Windows command-line tools executed over the SSH session.

Prerequisites (Performed on Windows):

- Enabled OpenSSH Server via Settings > Apps > Optional Features > Add "OpenSSH Server."
- Started the service and configured the firewall to allow port 22.
- Added the user "Rafay Sadiq" to allowed SSH logins.

Steps Performed:

1. Successfully logged into Windows via SSH from Kali Linux using valid credentials:
   `ssh "Rafay Sadiq”192.168.56.1`
   (Entered password when prompted)
2. After a successful SSH login, modified the existing file using Windows command syntax:
   `echo Modified malicious payload >> "C:\Users\Rafay Sadiq\Desktop\rafay.txt"`
3. Deleted the file:
   `del "C:\Users\Rafay Sadiq\Desktop\rafay.txt"`

These actions simulated a real attacker who has gained SSH access to the Windows endpoint and is tampering with monitored files remotely.

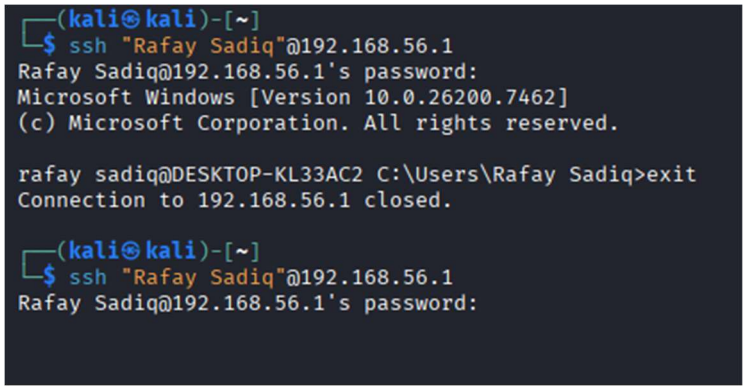


Figure 24: Kali Terminal Showing Successful SSH Connection to Windows

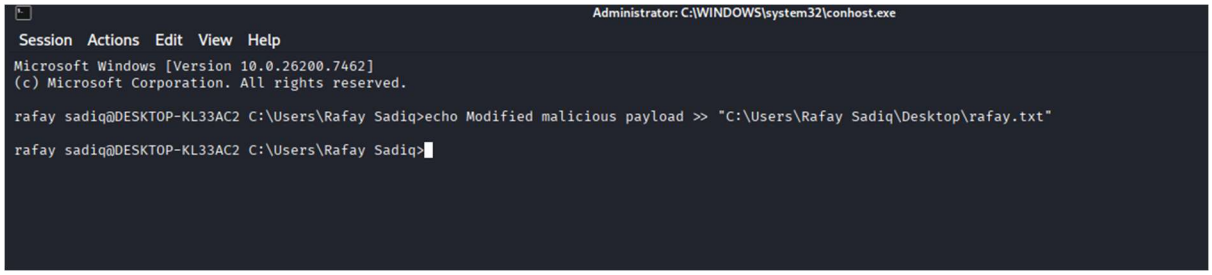


Figure 25: Kali Terminal (SSH Session) Executing File Modification Command on Windows

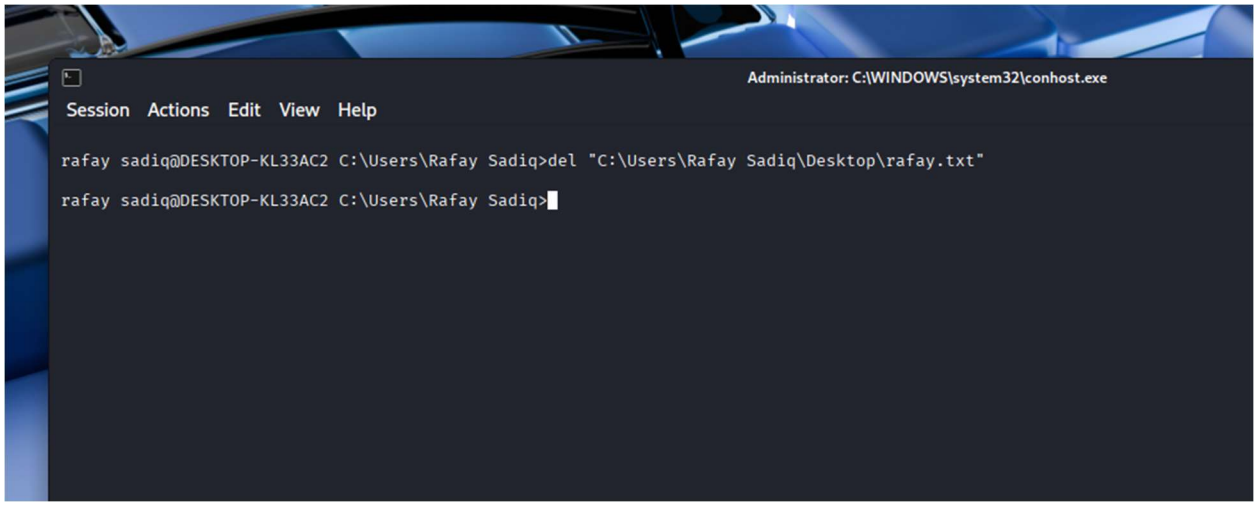
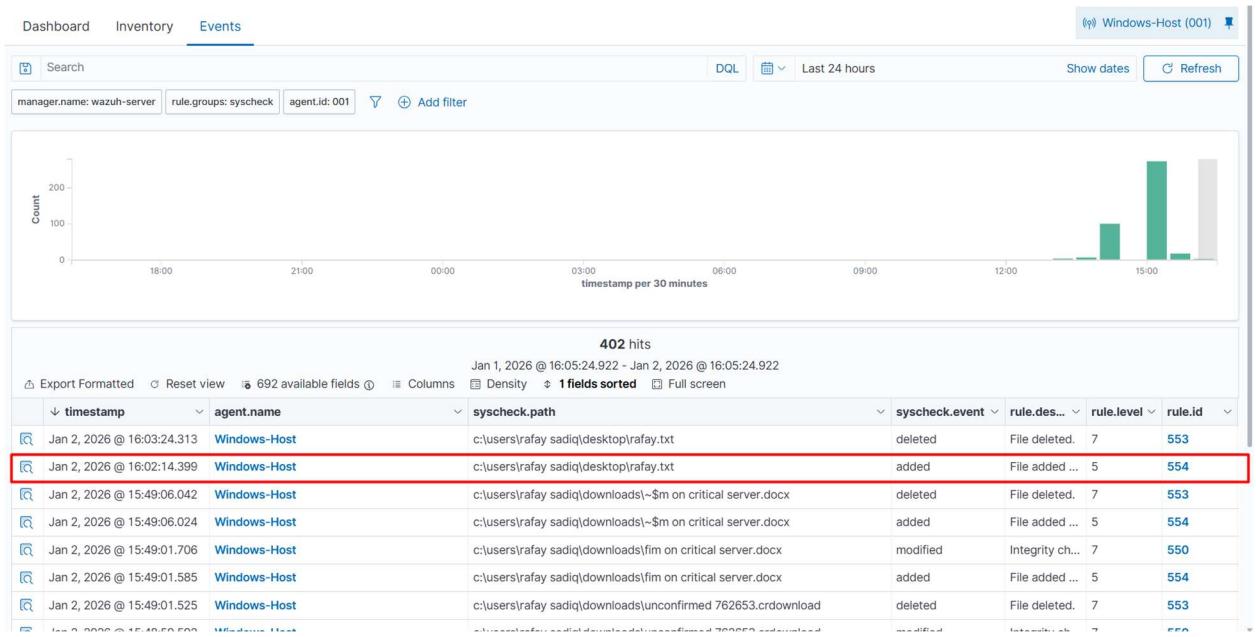


Figure 26: Kali Terminal (SSH Session) Executing File Deletion Command on Windows



timestamp	agent.name	syscheck.path	syscheck.event	rule.des...	rule.level	rule.id
Jan 2, 2026 @ 16:03:24.313	Windows-Host	c:\users\rafay sadiq\desktop\rafay.txt	deleted	File deleted.	7	553
Jan 2, 2026 @ 16:02:14.399	Windows-Host	c:\users\rafay sadiq\desktop\rafay.txt	added	File added ...	5	554
Jan 2, 2026 @ 15:49:06.042	Windows-Host	c:\users\rafay sadiq\downloads\~$m on critical server.docx	deleted	File deleted.	7	553
Jan 2, 2026 @ 15:49:06.024	Windows-Host	c:\users\rafay sadiq\downloads\~$m on critical server.docx	added	File added ...	5	554
Jan 2, 2026 @ 15:49:01.706	Windows-Host	c:\users\rafay sadiq\downloads\fim on critical server.docx	modified	Integrity ch...	7	550
Jan 2, 2026 @ 15:49:01.585	Windows-Host	c:\users\rafay sadiq\downloads\fim on critical server.docx	added	File added ...	5	554
Jan 2, 2026 @ 15:49:01.525	Windows-Host	c:\users\rafay sadiq\downloads\unconfirmed 762653.crdownload	deleted	File deleted.	7	553

Figure 27: Wazuh Dashboard FIM Alert for File Modification

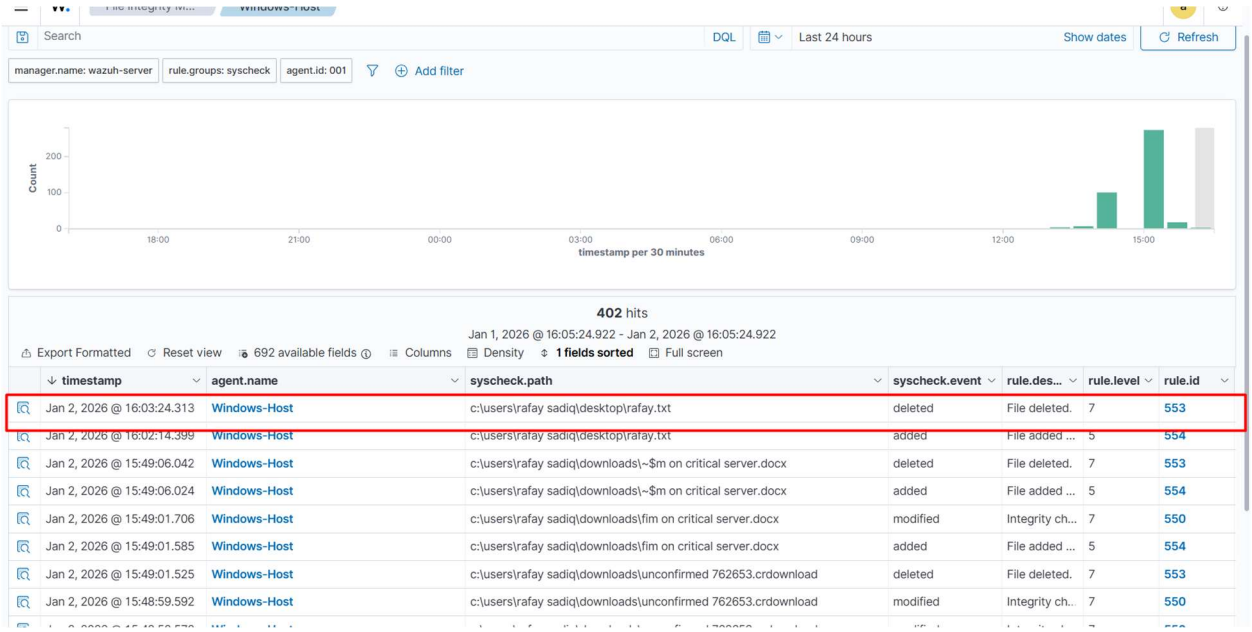


Figure 28: Wazuh Dashboard FIM Alert for File Deletion

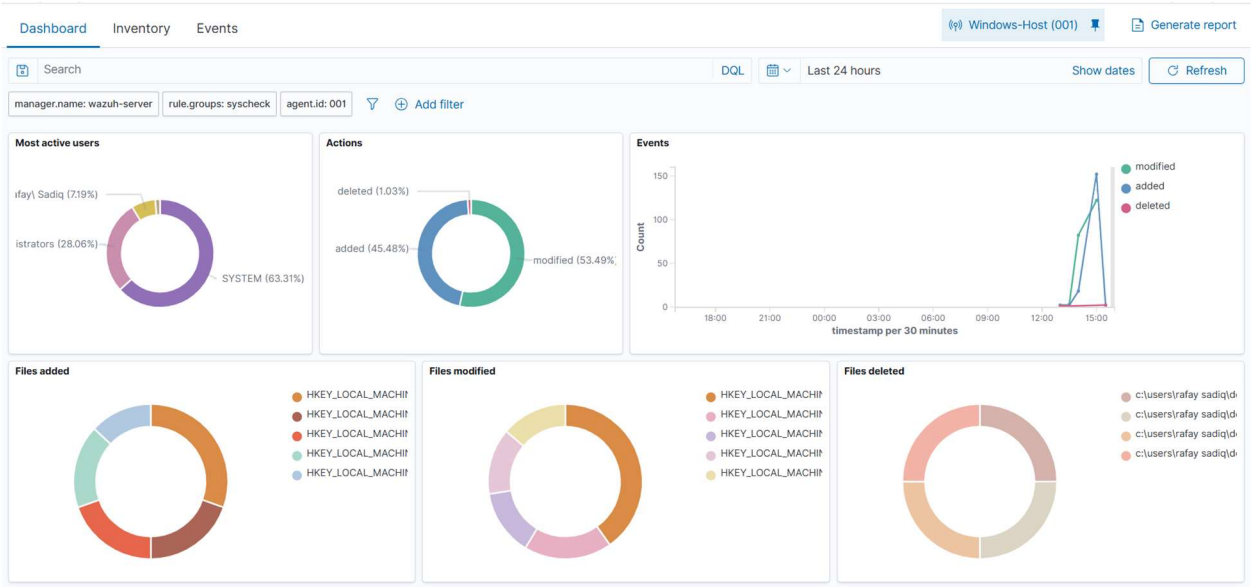


Figure 29: Wazuh Integrity Monitoring Module Overview Showing Sequence of Addition, Modification, and Deletion Alerts

7. Results & Log Analysis

The Wazuh dashboard effectively displayed all alerts, agent status, and event details as shown in the figures above.

7.1. RDP Attack Logs

- Event ID 4625 captured in Windows Security logs, forwarded to Wazuh:
    - Workstation: **kali**
    - Source IP: Kali's IP (e.g., 192.168.56.103)
    - Failure: Unknown username or bad password.

7.2. FIM Alert Analysis

- Alert: File added at c:\users\rafay sadiq\desktop\rafay.txt.
- Details: Action: added, Severity: Level 5, Attributes: Size, permissions, hashes.

8. Challenges & Lessons Learned

During the implementation and testing of this project, several challenges were encountered:

- Network Configuration Issues: Initial connectivity problems arose due to VirtualBox network settings (NAT vs Host-Only). Switching to Host-Only Adapter resolved IP reachability but required careful IP verification on all machines.
- RDP Compatibility: The default freerdp tool on Kali had issues with modern Windows RDP encryption. Installing freerdp3-x11 and disabling Network Level Authentication (NLA) on Windows was necessary for simulation, though this reduces security in real environments.
- Agent Connection Delays: The Windows agent occasionally took time to register and appear as "Active" in the dashboard, requiring service restarts and key re-imports.

- Alert Tuning: Initial FIM alerts were noisy; fine-tuning the monitored directories improved relevance.

Key Lessons Learned:

- Strong authentication mechanisms like NLA and complex passwords are critical to prevent brute-force attacks.
- Real-time FIM is highly effective for detecting unauthorized changes but should be configured selectively to avoid alert fatigue.
- Proper network isolation in labs is essential for safe attack simulation.
- Log correlation in Wazuh provides excellent forensic value, revealing attacker hostname and source details even in simulated internal threats.

9. Conclusion

This project successfully implemented a Wazuh-based SIEM for threat detection and File Integrity Monitoring. The dashboard screenshots illustrate real-time monitoring capabilities, agent connectivity, and alert generation. The simulation demonstrated Wazuh's effectiveness in detecting RDP brute-force attempts (via Event ID 4625) and file modifications. Overall, the setup reinforced the importance of proactive monitoring and secure configurations in defending against internal and external threats.

10. Future Work

To extend this project and enhance its capabilities, the following improvements can be implemented:

- Active Response Integration: Configure Wazuh's Active Response module to automatically block attacker IPs (e.g., via firewall rules) upon detecting brute-force attempts.
- Vulnerability Detection: Enable and test Wazuh's Vulnerability Detection module to scan the Windows endpoint for known CVEs.

- Multi-Agent Environment: Deploy additional agents on Linux endpoints (e.g., Ubuntu server) and monitor a heterogeneous network.
- Integration with External Tools: Connect Wazuh to external SIEM (like Elastic Stack) or ticketing systems for automated incident response.
- Realistic Threat Simulation: Incorporate tools like Metasploit from Kali for advanced exploit simulation and observe Wazuh's detection rates.

These enhancements would transform the setup into a more comprehensive Security Operations Center (SOC) simulation.

11. References

1. Wazuh Documentation: https://documentation.wazuh.com/
2. Kali Linux Documentation: https://www.kali.org/docs/
3. Microsoft RDP Guide: https://learn.microsoft.com/en-us/windows-server/remote/remote-desktop-services/
4. Oracle VirtualBox Documentation: https://www.virtualbox.org/manual/

12. List of Figures

Figure 1: Overview of VirtualBox VM List Showing Wazuh, Windows, and Kali Machines:
Figure 2: Wazuh Server Console Login and IP Address Verification (ip a command):
Figure 3: Wazuh Dashboard Login Page:
Figure 4: Wazuh Dashboard Overview Showing Cluster Status and Agents:
Figure 5: Wazuh Dashboard General Alerts Overview:
Figure 6: Wazuh manage_agents Tool Showing Added Windows Agent and Key:
Figure 7: Windows Wazuh Agent Manager Showing Imported Key and Connected Status:
Figure 8: Wazuh Dashboard Agents List with Windows Agent Active:
Figure 9: Detailed View of Windows Agent Status in Wazuh Dashboard:
Figure 10: Kali Linux Boot Menu Selecting Graphical Install:
Figure 11: Kali Linux Installation Wizard (Language/Hostname Setup):
Figure 12: Kali Linux Terminal After Installation Showing Ip a and Ping to Windows:
Figure 13: Windows PowerShell SSH Connection to Wazuh Server:
Figure 14: Windows Remote Desktop Settings with RDP Enabled and NLA Disabled:
Figure 15: Kali Terminal Ping to Windows Machine Confirming Connectivity:
Figure 16: Kali Terminal Successful RDP Test Connection (Non-Attack):
Figure 17: Edited ossec.conf File Showing FIM Directory Configuration:
Figure 18: Wazuh Dashboard Agent Configuration View Confirming FIM Settings:
Figure 19: Kali Terminal Executing xfreerdp3 Brute-Force Attempts with Errors:
Figure 20: Wazuh Dashboard Security Events Showing RDP Authentication Failures:
Figure 21: Wazuh Dashboard Alert Details for Brute-Force Attack with Attacker Source (kali):
Figure 22: Windows Desktop Showing Newly Created rafay.txt File:
Figure 23: Kali Terminal Showing Successful SSH Connection to Windows:
Figure 24: Kali Terminal (SSH Session) Executing File Modification Command on Windows:
Figure 25: Kali Terminal (SSH Session) Executing File Deletion Command on Windows:
Figure 26: Wazuh Dashboard FIM Alert for File Modification:
Figure 27: Wazuh Dashboard FIM Alert for File Deletion:
Figure 28: Wazuh Integrity Monitoring Module Overview Showing Sequence of Addition, Modification, and Deletion Alerts: