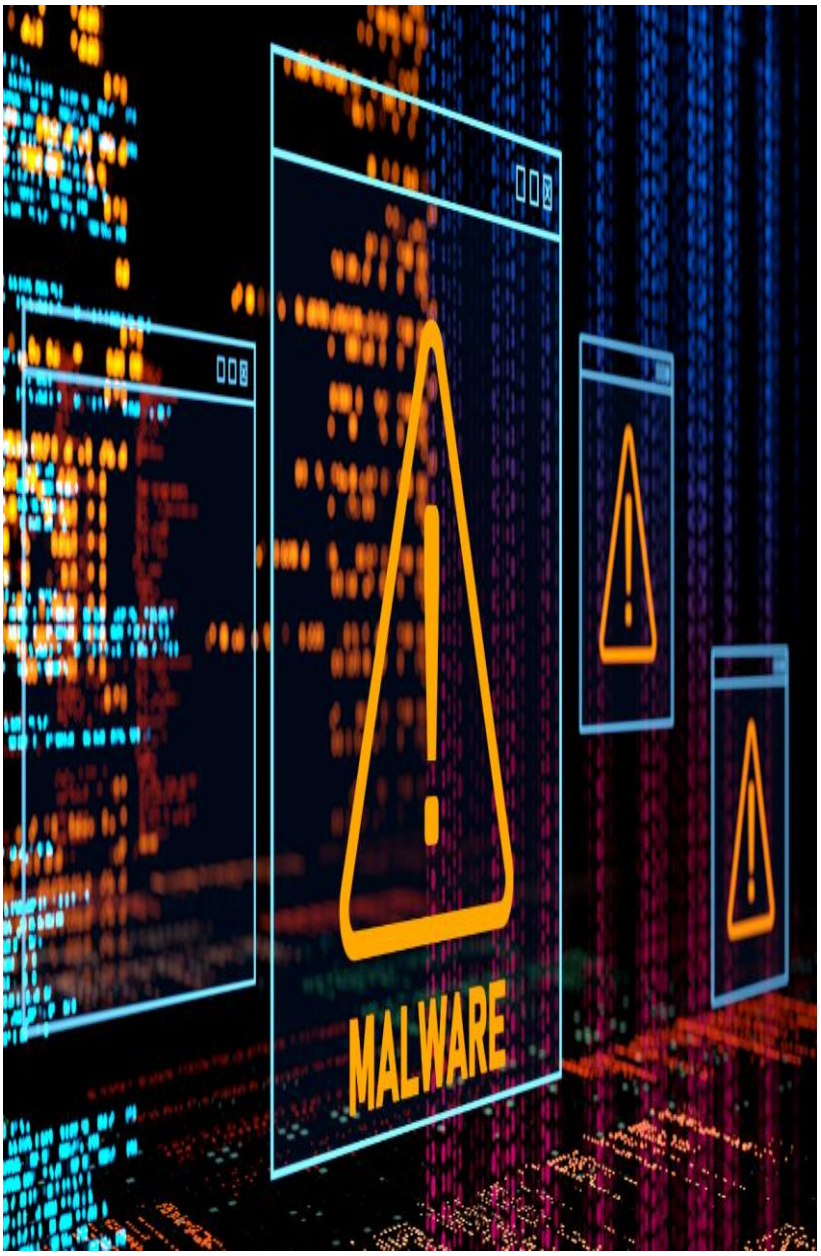
رواد مصر الرقمية



Ahmed Sameh

Ahmed Mohamed

Hanaa Marzouk

Mohamed Khaled

"Malware Analysis and Prevention Strategy Project Documentation"

رواد مصر الرقمية

Chapter 1 – Malware Analysis

1.1 Introduction to Malware Analysis

In the modern cybersecurity landscape, malware remains one of the most critical threats targeting individuals, enterprises, and government infrastructures. Attackers continuously develop new techniques to bypass security controls, making malware analysis an essential discipline for understanding malicious behavior, identifying infection methods, and building efficient defensive strategies.

This chapter focuses on examining malware from multiple angles: its operational traits, classification criteria, methods of propagation, techniques used to hide its activity, and the types of payloads executed once a system is compromised. By studying these aspects, security professionals can enhance detection mechanisms, support threat hunting activities, and design better prevention strategies.

Malware analysis is not only about identifying the malicious file; it involves understanding how it spreads, why it hides, and what it intends to achieve. These elements represent the foundation for effective defense systems.

1.2 Malware Classification According to Core Behaviors

Malware is not categorized randomly; its classification mainly depends on four technical behaviors that describe how it operates inside a computing environment:

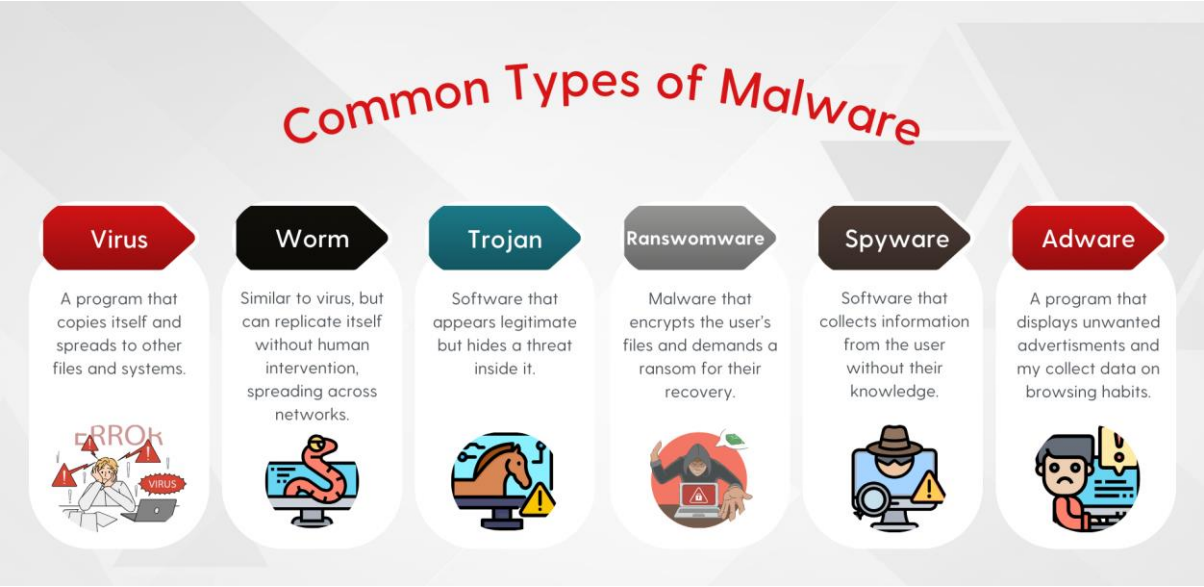
1. **Circulation**
   This trait focuses on how malware moves from one device to another. Some malicious programs are designed to replicate aggressively, using email, networks, shared storage, or user actions to spread.
2. **Infection**
   After reaching a system, the malware must gain persistence or embed itself within files, applications, or system components. This is what allows it to remain active after reboots and continue executing malicious operations.

3. **Concealment**
   To remain undetected, attackers implement evasion techniques such as code obfuscation, stealth processes, or hiding inside legitimate applications. Concealment allows the malware to operate silently without raising alerts.
4. **Payload Capabilities**
   This defines the purpose of the malware — whether it is stealing data, encrypting files, damaging systems, logging keystrokes, or establishing remote control.

Understanding these four behaviors simplifies the learning process, allowing analysts to categorize new malware samples based on actions rather than names.

1.3 Malware Circulation

Two types of malware are primarily designed for spreading: viruses and worms.



1.3.1 Viruses

A computer virus behaves similarly to biological viruses — it attaches itself to a host and replicates. Unlike worms, a virus cannot spread without user interaction. It needs a file, document, or program to insert its code into.

Common Infection Methods of Viruses:

• **Executable File Infection:**
Attaches the virus code to apps (.exe, .dll) so it activates once the program is launched.

• **Macro Infection:**
Targets Office documents that contain macros (Word/Excel). When the document opens, the embedded malicious macro executes.

• **Appender Technique:**
The virus inserts its code at the end of a file and modifies the beginning to jump to the malicious section.

• **Armored Viruses:**
Designed specifically to resist analysis by adding layers of obfuscation, anti-debugging, and misleading code patterns.

Mutation in Viruses:
• Oligomorphic viruses change between a small number of predefined versions.
• Polymorphic viruses generate new encrypted forms with each execution.
• Metamorphic viruses rewrite their entire code, producing logically similar but structurally different new copies.

1.3.2 Worms

Worms are self-propagating malware that do not require user assistance. Instead, they exploit weaknesses in applications or operating systems to jump between machines across a network.

Key Traits of Worms:
• Spread automatically from one device to others.
• Consume system and network resources due to fast replication.
• Modern worms often carry destructive payloads such as deleting files or installing backdoors.
• They typically scan the network continuously for devices with the same vulnerabilities.

1.4 Malware Infection Techniques

Not all malware spreads; some aim to silently compromise the system after arriving through email, downloads, or malicious websites.

1.4.1 Trojans

Trojans disguise themselves as legitimate applications. They trick users into installing them and operate silently while performing harmful tasks.

Capabilities of Trojans:

- Stealing credentials
- Logging keystrokes
- Opening backdoors
- Allowing attackers remote control

1.4.2 Ransomware

Ransomware locks devices or files and demands payment. Modern ransomware focuses on encryption rather than screen-locking.

Two Main Generations:

1. Locker Ransomware – blocks access to the device.
2. Crypto-Ransomware – encrypts documents, drives, and network shares.

1.4.3 Crypto-Malware

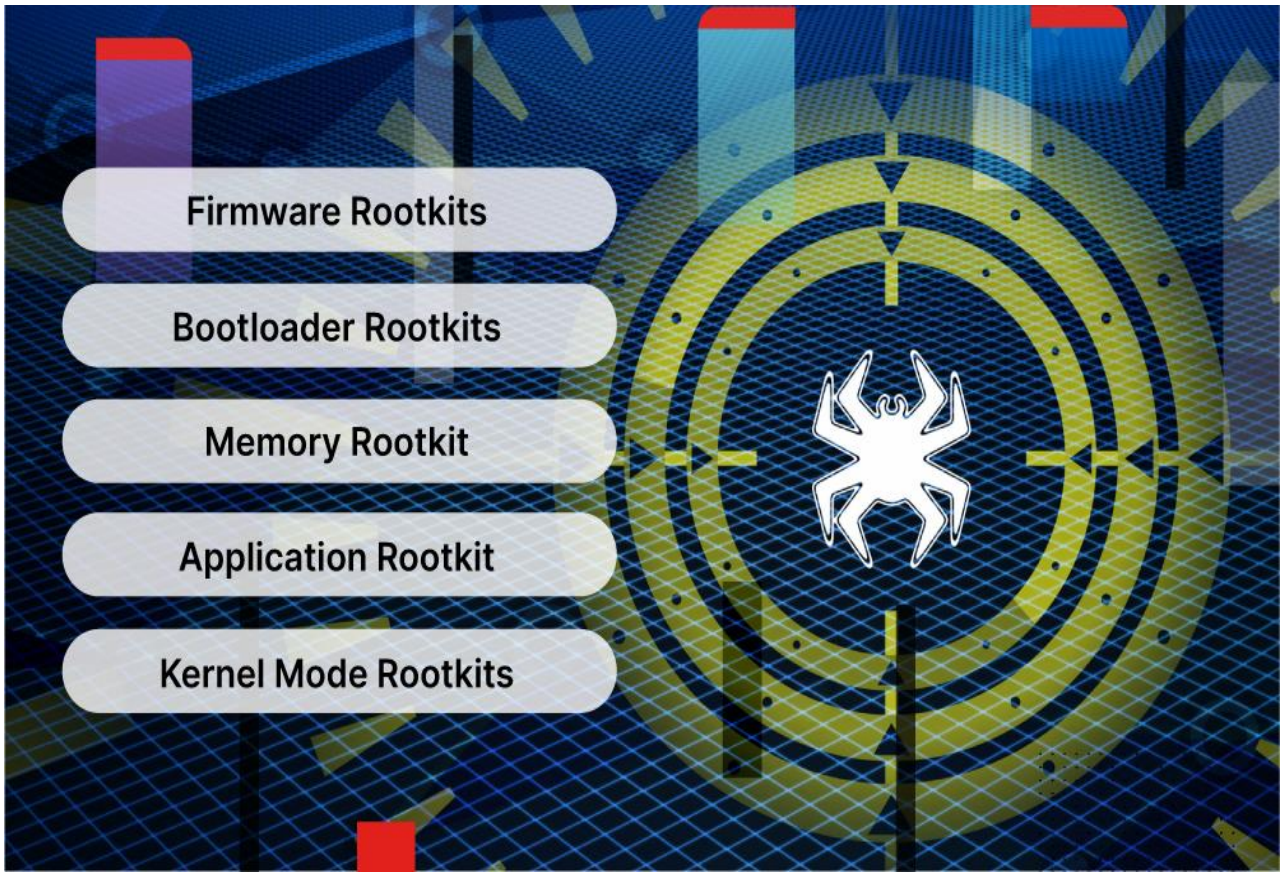
Crypto-malware is an advanced form of ransomware that aggressively encrypts user files, network shares, NAS devices, cloud storage, and sometimes mobile devices.
It uses strong cryptographic algorithms, making decryption impossible without the attacker's private key. Many large-scale incidents in recent years fall under this category.

1.5 Concealment Techniques Used by Malware

Concealment is a crucial operational stage that enables malware to persist inside the system.

Types of Concealment:

1. **Physical Concealment**
   Malicious components can be introduced through infected USBs, hardware keyloggers, or rogue devices that appear harmless but contain embedded attacks.
2. **Informational Concealment**
   Involves hiding malicious code through:
   • Obfuscation
   • Encryption
   • Packing
   • Fileless execution (using PowerShell, WMI, registry)
3. **Legal/Logical Concealment**
   Some malware mimics legitimate system files, modifies logs, or uses spoofed filenames to avoid suspicion.
4. **Stealth Malware (Rootkits)**
   Rootkits hide processes, drivers, and files by modifying system-level functions. They operate at different layers (user-mode, kernel-mode, bootkits, firmware rootkits).



1.6 Malware Payload Capabilities

The payload represents the real intention of the malware — the damaging or malicious activity executed after infection.

1.6.1 Types of Payloads

1. **Data Theft Payloads**
   Used by spyware, banking Trojans, and credential stealers.
2. **Destructive Payloads**
   Used to delete files, wipe disks, or corrupt boot sectors (e.g., NotPetya).
3. **Remote Control Payloads**
   Allows attackers to manage systems remotely through RATs or botnet clients.
4. **Encryption Payloads**
   Encrypt files or entire drives as part of ransomware.

1.6.2 Spyware and Keyloggers

Spyware collects sensitive data silently.
Keyloggers record every keystroke typed by the victim. They can be:
• Software Keyloggers: run in the background with no interface.
• Hardware Keyloggers: inserted between keyboard and computer ports.

1.6.3 Adware

Adware automatically displays advertisements on the infected system. While not always destructive, it:
• Slows device performance
• Shows unwanted or inappropriate content
• Can redirect users to malicious sites

1.6.4 Logic Bombs

A logic bomb stays dormant until a specific condition triggers it (date, username, specific action). Once triggered, it performs harmful operations such as deleting data or corrupting systems.

1.6.5 Backdoors & Botnets

Backdoors bypass security controls to grant unauthorized access.

Botnets, controlled by a botmaster, consist of compromised machines performing large coordinated attacks such as:

- DDoS
- Spam campaigns
- Credential stuffing
- Crypto-mining

Botnet commands can be sent via:

- Websites
- Social media posts
- Encrypted emails
- Hidden C&C channels

1.7 Static vs Dynamic Malware Analysis

These are the two primary methods for studying malware samples.

Static Analysis

Involves examining the file without running it.

Useful for:

- Extracting strings
- Identifying signatures
- Recognizing known patterns
- Reviewing file structure

Dynamic Analysis

Runs the malware inside a controlled environment (sandbox) to observe:

- Network communication
- File changes
- Registry modifications
- Process behavior

Combining both methods provides the most accurate results.

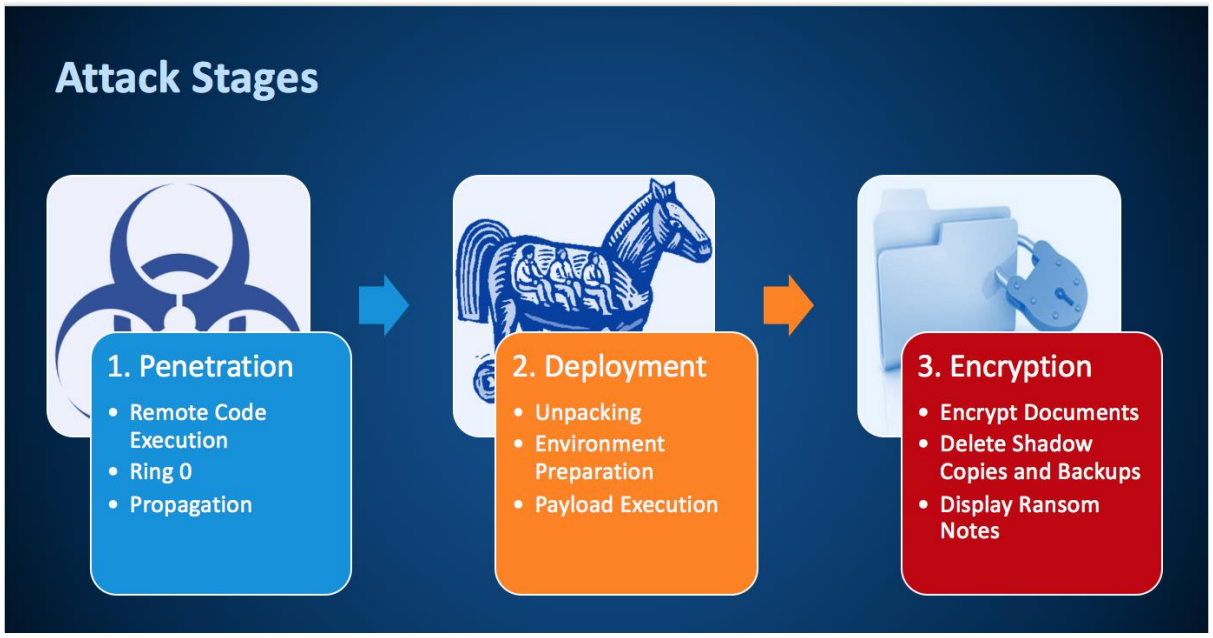
1.8 Real-World Malware Case Studies

1.8.1 Target Breach (2013)

Attackers gained access via a third-party vendor account, deployed memory-scraping malware on POS systems, and extracted millions of credit card details.

1.8.2 WannaCry (2017)

A worm-based ransomware using the EternalBlue exploit, spreading globally and impacting hospitals, telecom companies, and critical sectors.



1.8.3 Twitter Hack (2020)

A large social engineering attack targeting employees, allowing unauthorized access to internal admin tools and posting cryptocurrency scams.

1.8.4 Stuxnet (2010)

A highly sophisticated worm targeting industrial control systems, modifying PLC behavior to damage nuclear centrifuges.

1.8.5 Emotet (2014)

Started as a banking Trojan, evolved into a botnet that delivered other

malware families, and became one of the world's most expensive cyber threats.

1.8.6 Zeus (2007)
A credential-stealing Trojan responsible for billions in financial theft, using keylogging and form-grabbing techniques.

1.8.7 Agent Tesla (2014)
Advanced spyware focused on stealing credentials and exfiltrating data through multiple channels.

1.8.8 NotPetya (2017)
A destructive wiper disguised as ransomware, permanently damaging systems and causing global financial losses.

Chapter 2 – SIEM Configuration and Monitoring

CHAPTER 2

SIEM Configuration and Monitoring (Wazuh Deployment)

2.1 Introduction to SIEM

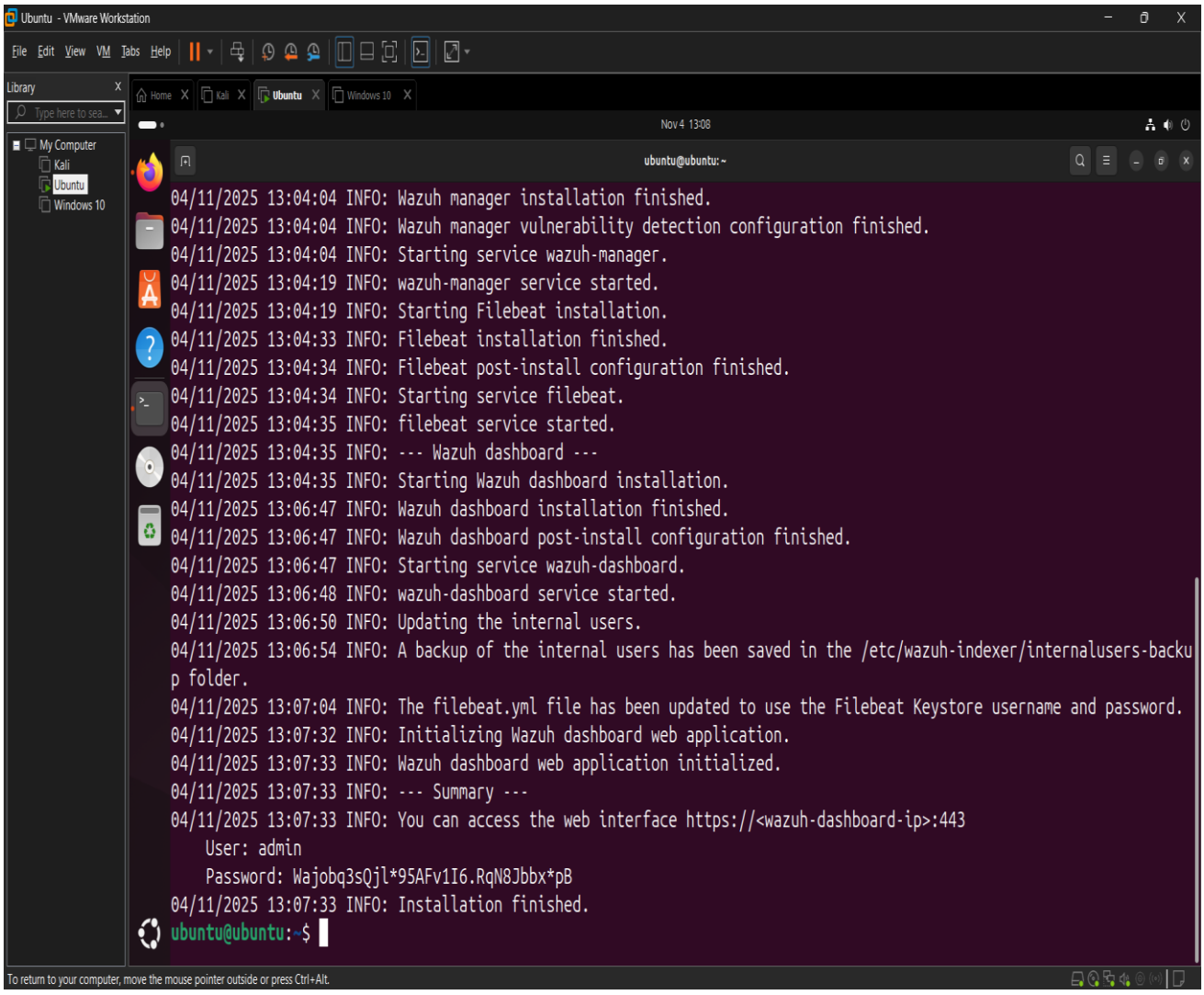
Security Information and Event Management (SIEM) platforms play a critical role in modern cybersecurity operations by collecting, correlating, and analyzing logs and security events from distributed systems. A SIEM provides real-time threat detection, historical event investigation, automated alerting, and centralized visibility across all endpoints. As cyber threats grow more sophisticated, organizations rely on SIEM solutions to shorten detection time, enforce compliance, and gain deep insight into system behavior.
Wazuh, an open-source SIEM and XDR platform, was chosen in this project because it combines log analysis, File Integrity Monitoring (FIM), vulnerability assessment, malware detection, agent-based monitoring, and integration with external threat intelligence services like VirusTotal. This chapter documents the complete deployment workflow of Wazuh—from preparing the environment, to installing

the server and agents, enabling modules, testing detection, and validating alert responses—supported by 35 technical screenshots.

2.2 Preparing the Environment

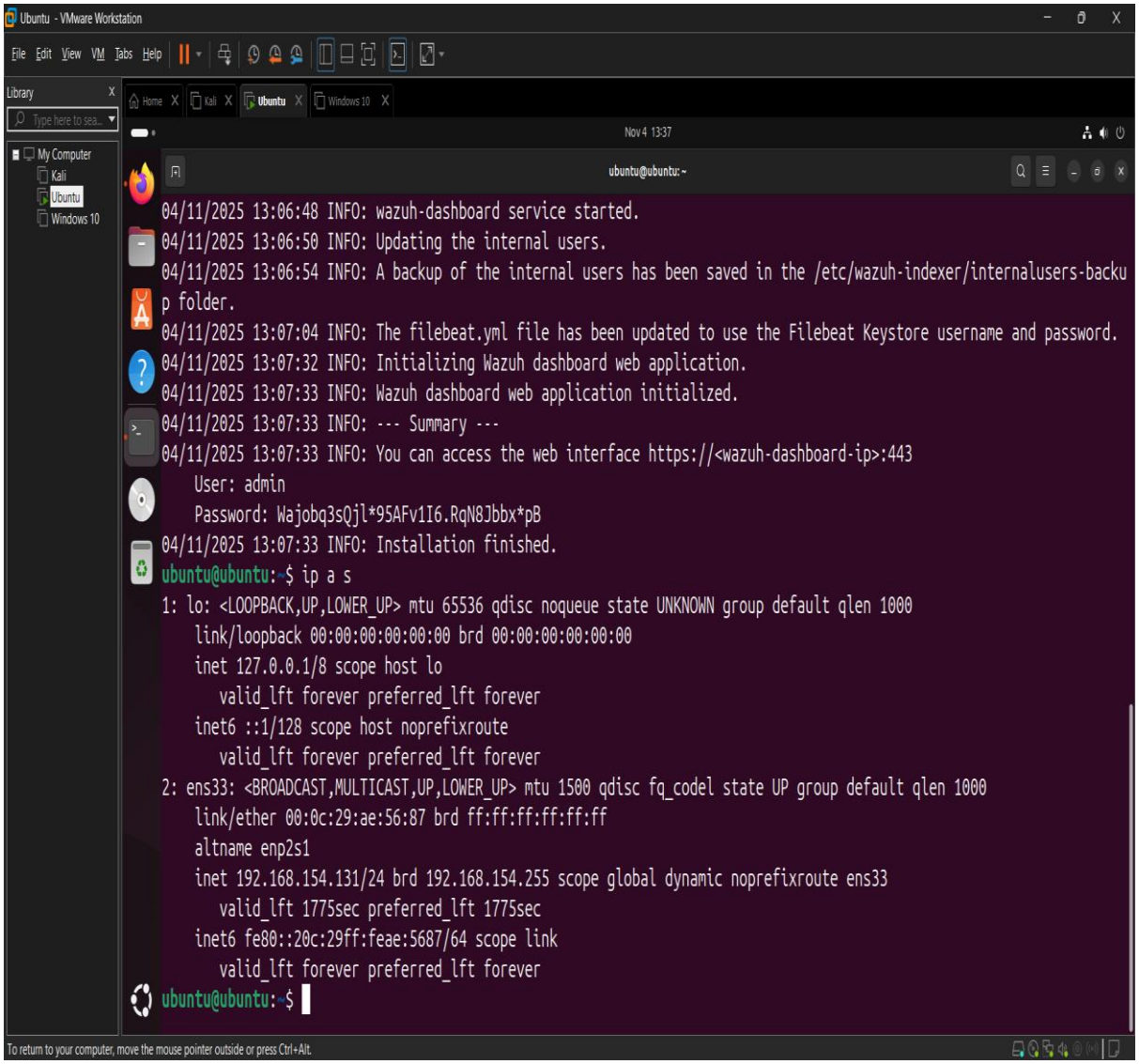
Before installing Wazuh, both the Ubuntu server and the Windows endpoint were prepared with the necessary software components and dependencies. These preparation steps ensure that all Wazuh modules—such as vulnerability scanning, real-time monitoring, and active response—operate smoothly without compatibility issues. The following figures illustrate the preparation process and highlight essential configurations required before deploying the SIEM.

2.2.1 Wazuh Installation Log Output



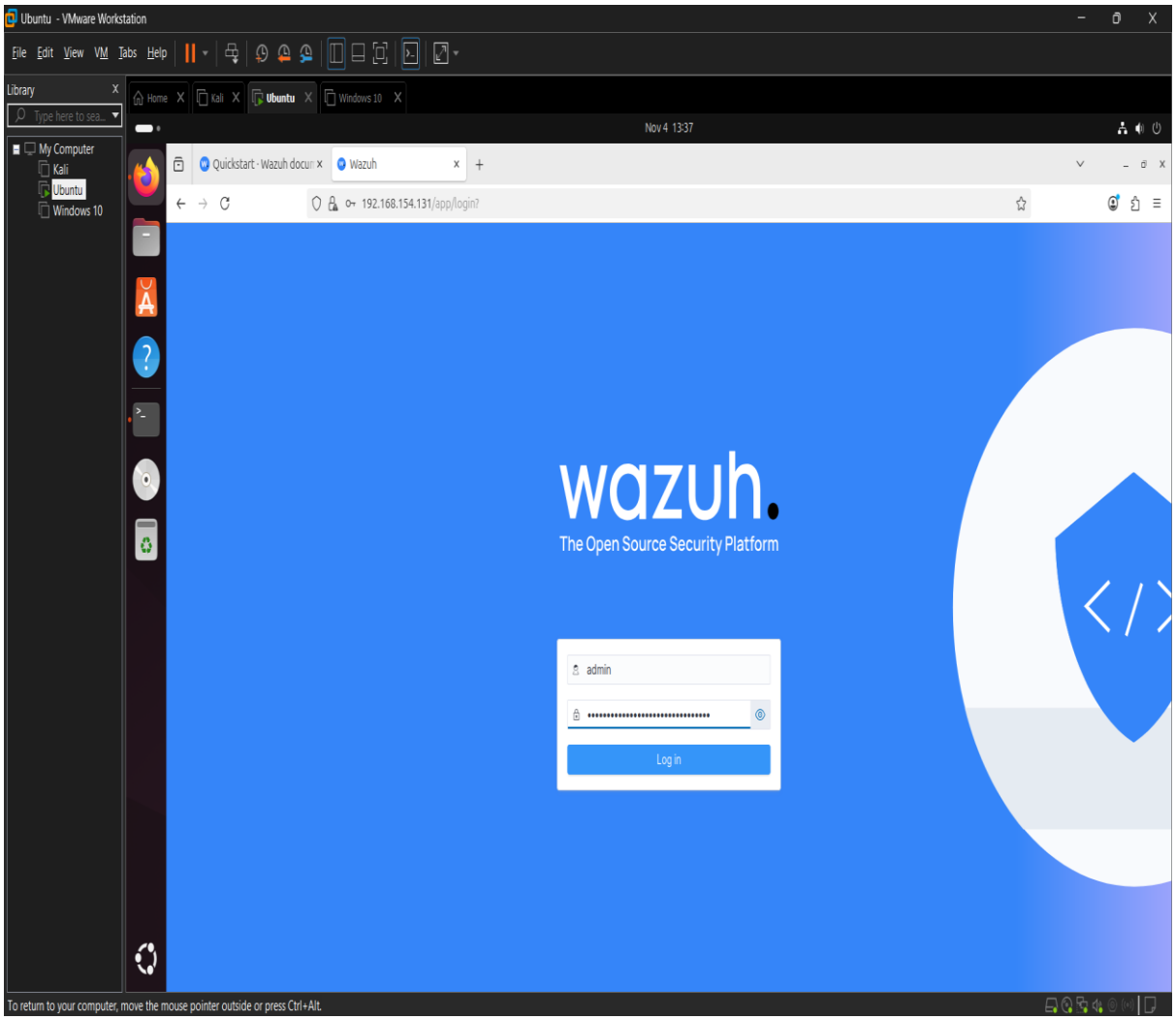
The image shows the final installation log of a full Wazuh deployment on Ubuntu, including the Manager, Filebeat, and Wazuh Dashboard.
It confirms that all services were installed, configured, and started successfully without errors.
The log also shows that internal user data was backed up, and Filebeat configurations were updated automatically.
At the end, the system provides the web interface URL along with the default admin username and generated password, indicating that the installation is fully completed.

2.2.2 **Wazuh Install Log & Network IP**



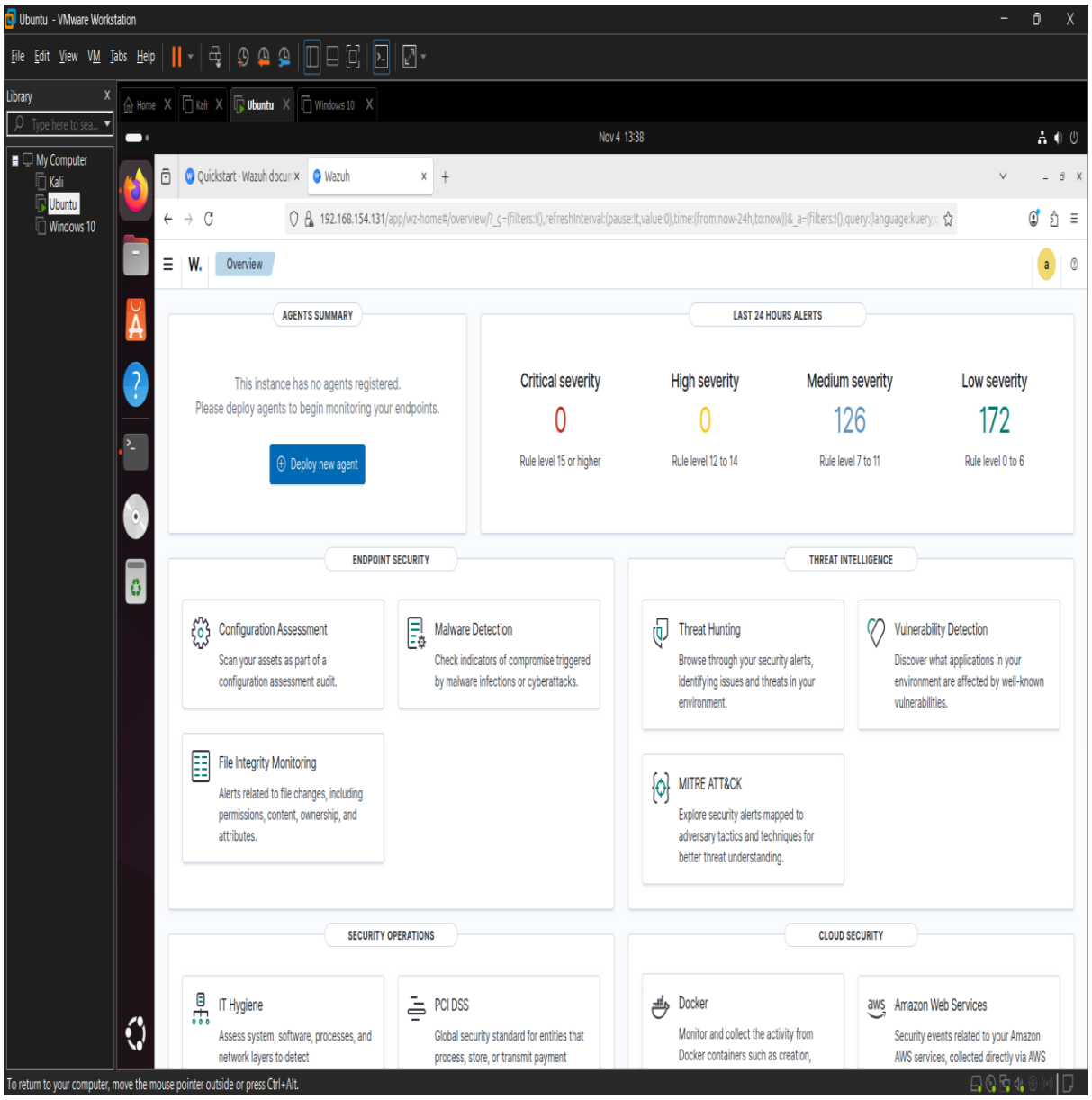
The image shows that the Wazuh installation finished successfully and all services started without errors.
It also provides the default admin username and the generated password for accessing the dashboard.
Below the logs, the output of **ip a s** displays the network interfaces and their assigned IP addresses.
The main interface **ens33** has IP **192.168.154.131**, which will be used to access the Wazuh web dashboard.

2.2.3 **Wazuh Dashboard Login Page**



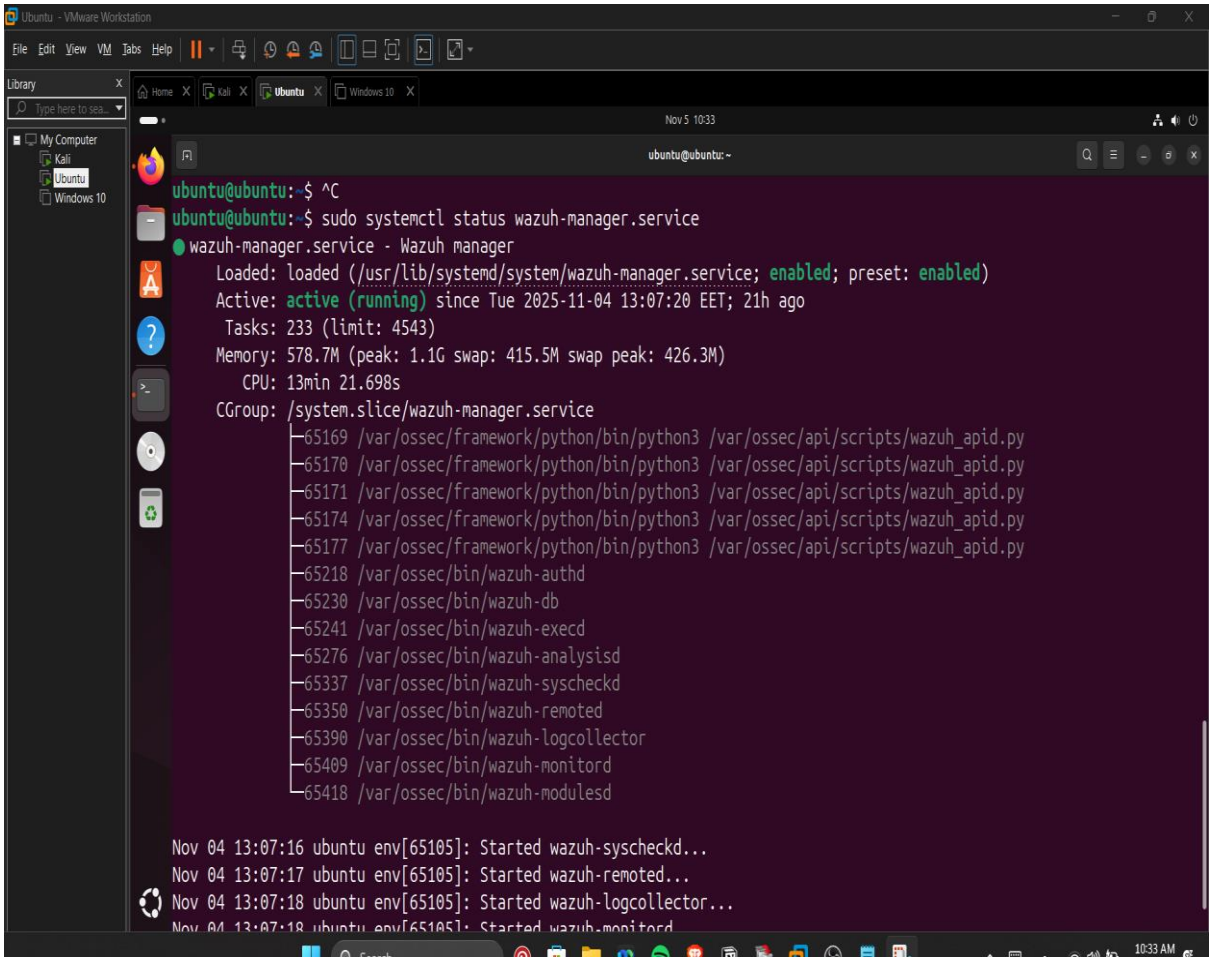
The image shows the Wazuh web dashboard login screen accessed through the IP address **192.168.154.131**.
The page displays the Wazuh logo and the login form where the username “admin” is already entered.
The password field is filled with the generated admin password from the installation logs.
This confirms that the Wazuh dashboard is running successfully and ready for the first login.

2.2.4 **Wazuh Dashboard – Overview Page**



The image shows the Wazuh dashboard home screen after successful login, displaying system status and security features.
It indicates that no agents are registered yet and provides a button to deploy new agents for monitoring endpoints.
The dashboard shows alert statistics for the last 24 hours, including medium- and low-severity alerts.
Below, different security modules appear, such as Configuration Assessment, Malware Detection, Threat Hunting, File Integrity Monitoring, and Vulnerability Detection.

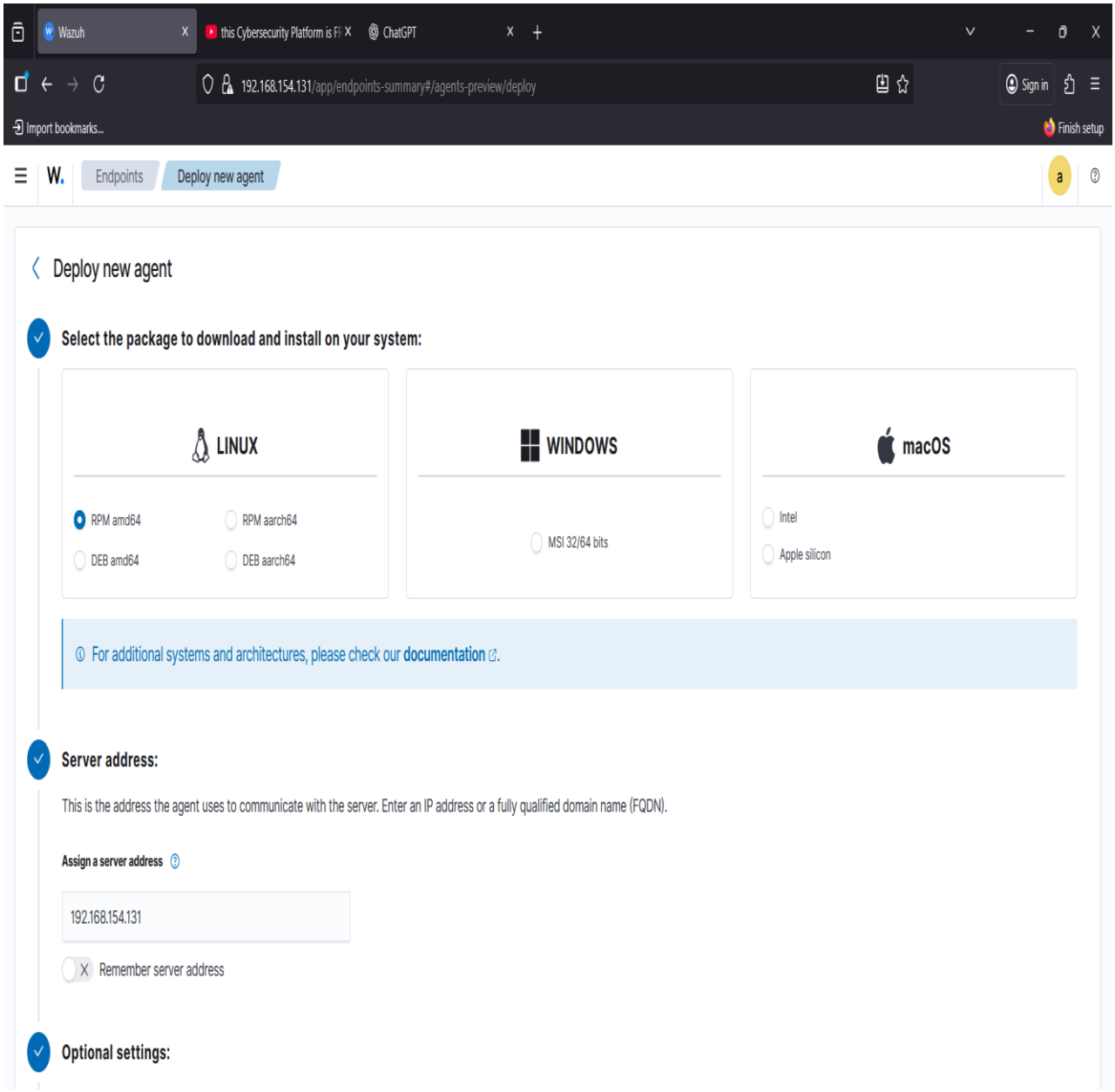
2.2.5 Wazuh Manager Service Status



The image shows the result of the command sudo systemctl status wazuh-manager.service, confirming that the Wazuh manager service is **active (running)**.
It displays the service details, including memory usage, CPU time, and the list of active Wazuh components like wazuh-authd, wazuh-db,

and wazuh-analysisd.
The logs at the bottom confirm that multiple Wazuh modules started successfully without errors.
This indicates that the Wazuh manager is fully operational and functioning correctly on the system.

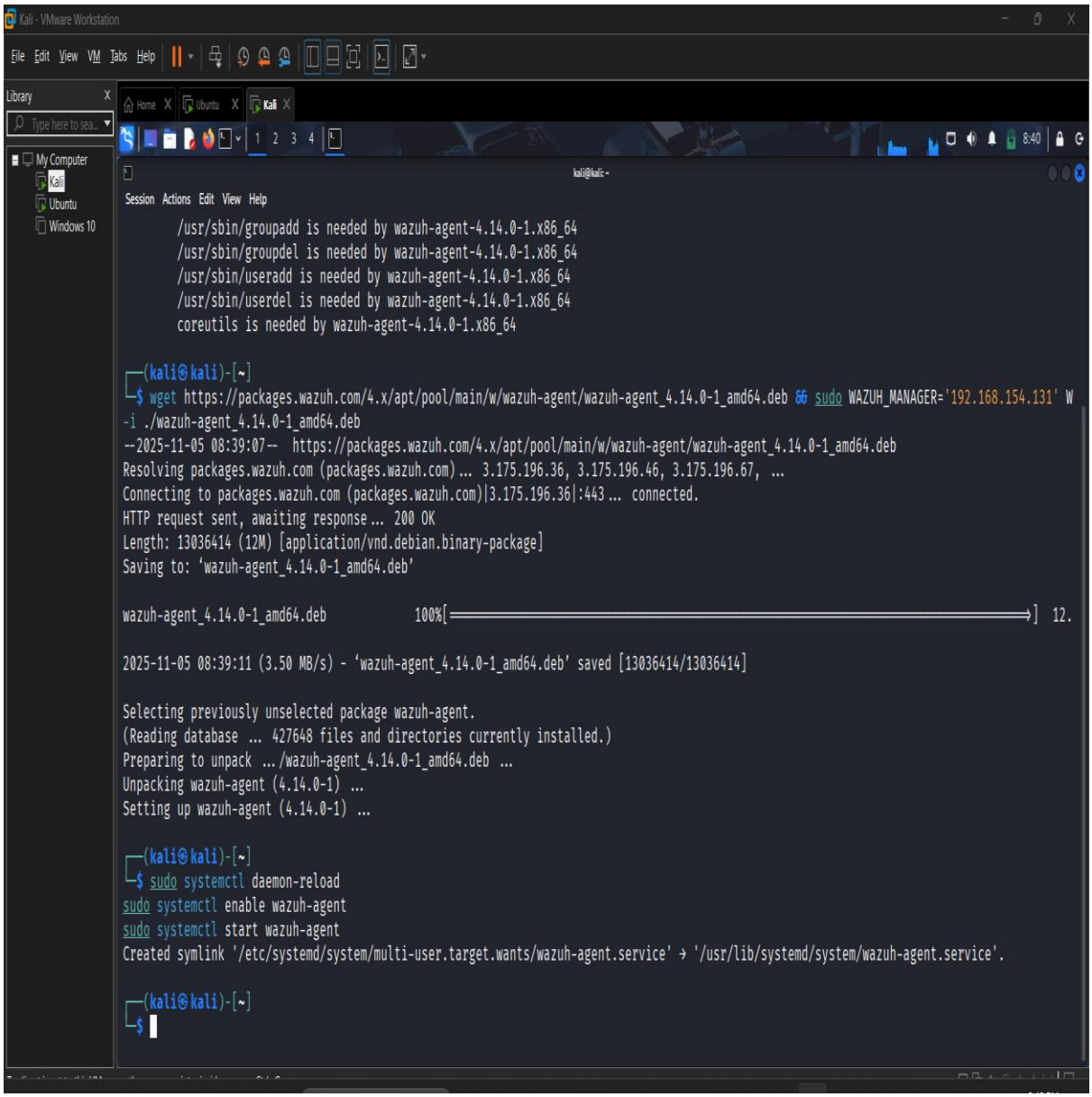
2.2.6 Wazuh – Deploy New Agent Page



The image shows the “Deploy new agent” section where you select the operating system to install the Wazuh agent.
Linux, Windows, and macOS options are available, with Linux (RPM

amd64) currently selected.
The server address field is automatically filled with **192.168.154.131**, which is the Wazuh manager IP.
This page prepares the correct agent package and configuration needed to connect endpoints to the Wazuh server.

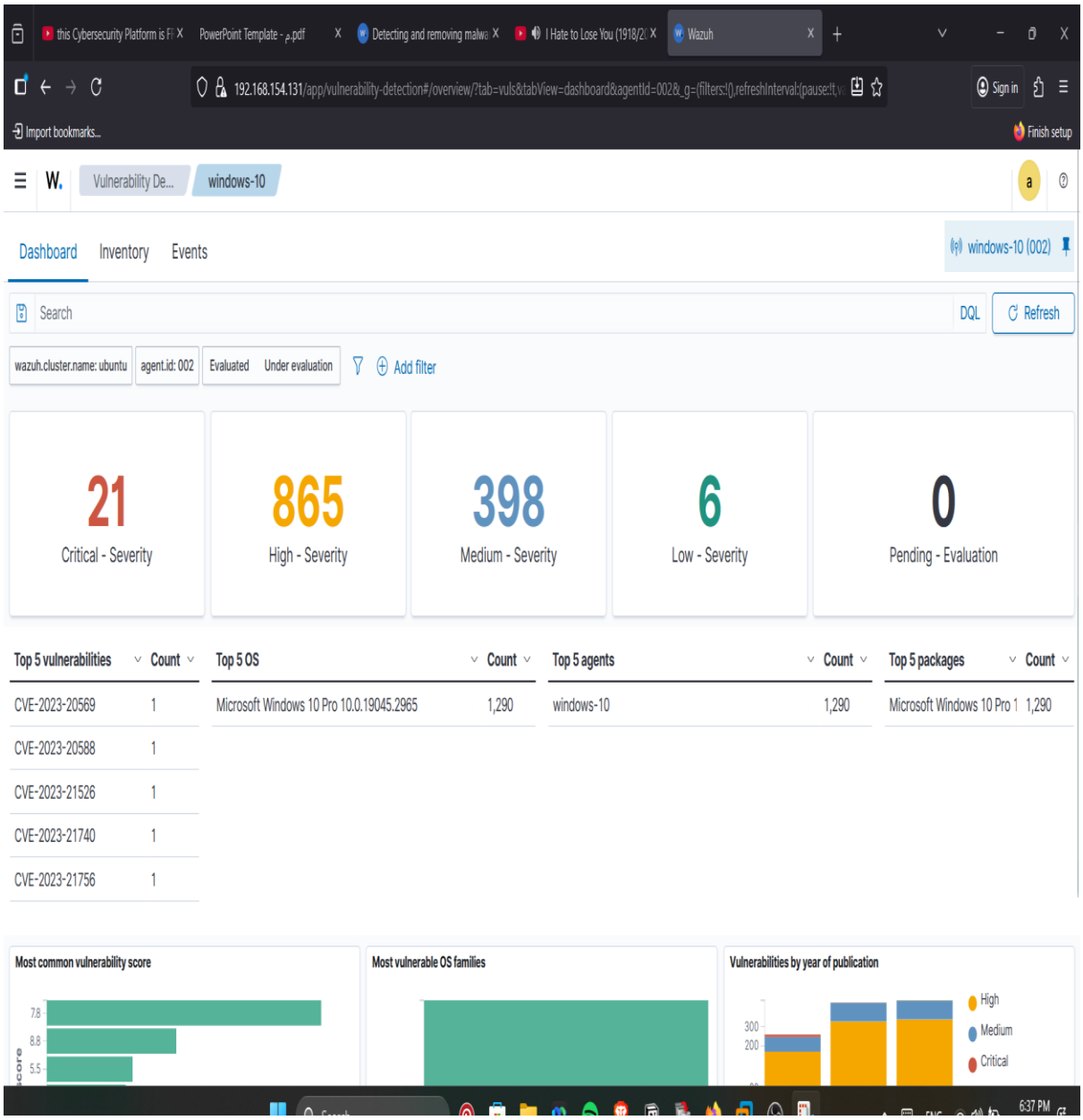
2.2.7 Installing Wazuh Agent on Kali Linux



The image shows the Kali terminal downloading the Wazuh agent package using wget from the official Wazuh repository.

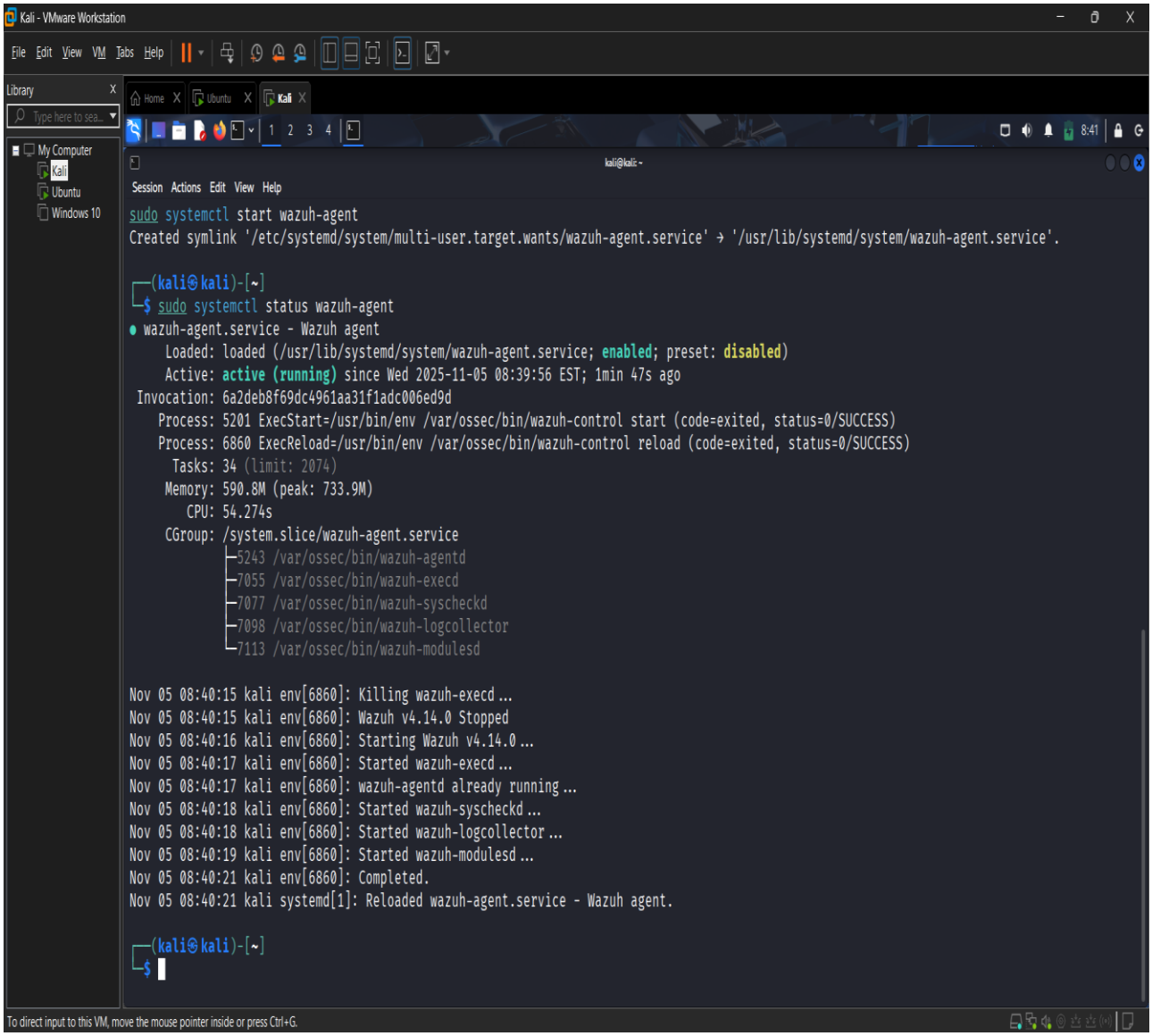
The package wazuh-agent_4.14.0-1_amd64.deb is successfully downloaded and installed with sudo WAZUH_MANAGER='192.168.154.131' dpkg -i.
Systemd commands are then executed to reload services, enable the wazuh-agent service at startup, and start it immediately.
This confirms that the Wazuh agent is now installed, configured, and running on the Kali machine, ready to communicate with the Wazuh manager.

2.2.8 Wazuh Vulnerability Detection – Windows 10 Agent



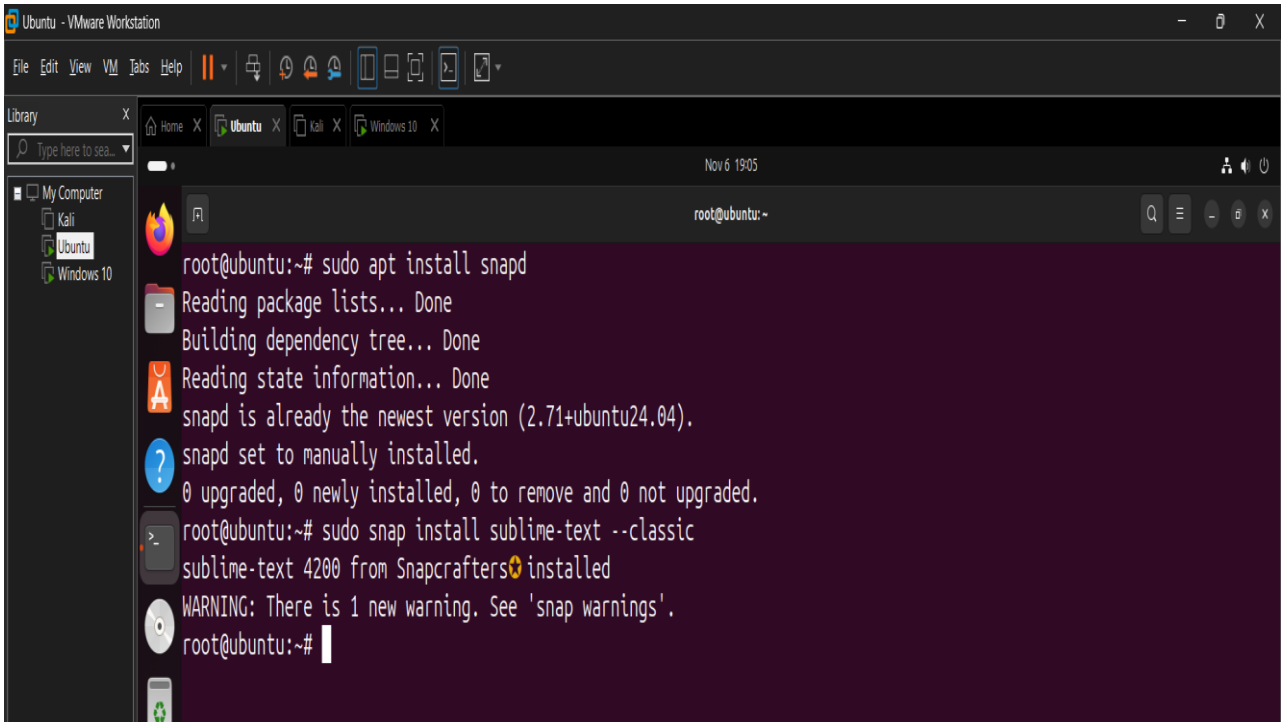
The image shows the vulnerability dashboard for the Windows-10 agent, displaying detected security issues.
It lists **21 critical**, **865 high**, **398 medium**, and **6 low** severity vulnerabilities on the monitored Windows 10 system.
The dashboard highlights top vulnerabilities, OS details, and package information, confirming that the agent is actively scanning and reporting results.
This view helps identify high-risk areas on the endpoint and prioritize remediation actions.

2.2.9 Wazuh Agent Status on Kali Linux



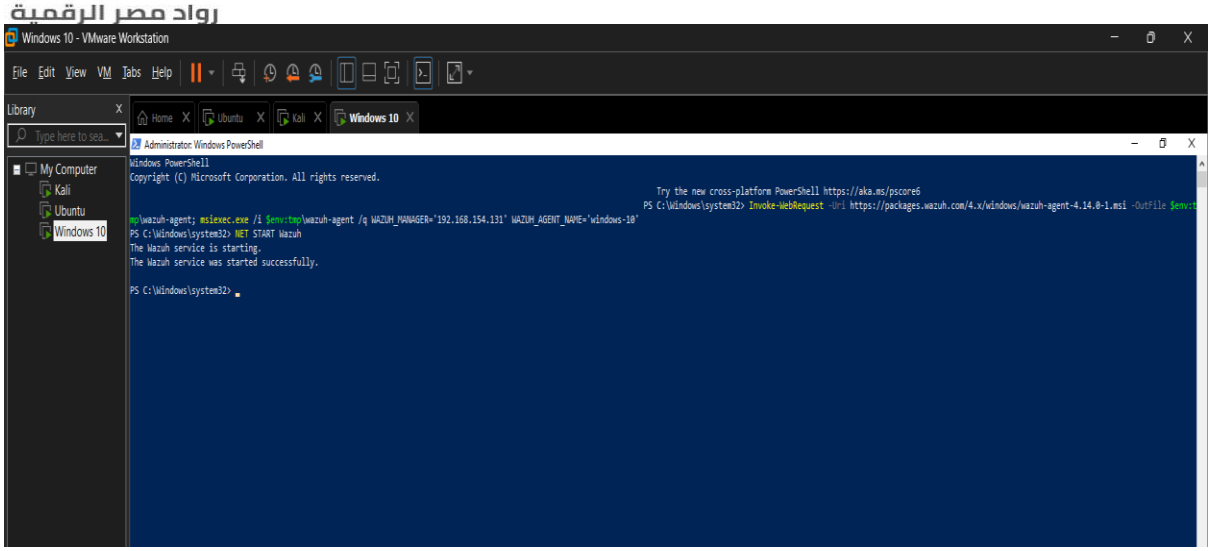
The image shows the result of the command sudo systemctl status wazuh-agent, confirming that the agent is **active (running)**.
The service is enabled, consuming around **590MB of memory**, and has multiple active subprocesses such as wazuh-agentd, wazuh-syscheckd, and wazuh-logcollector.
The logs at the bottom indicate that the agent was reloaded successfully, and all modules started without any errors.
This confirms that the Kali machine is properly connected and reporting to the Wazuh manager.

2.3 Installing Sublime Text via Snap on Ubuntu



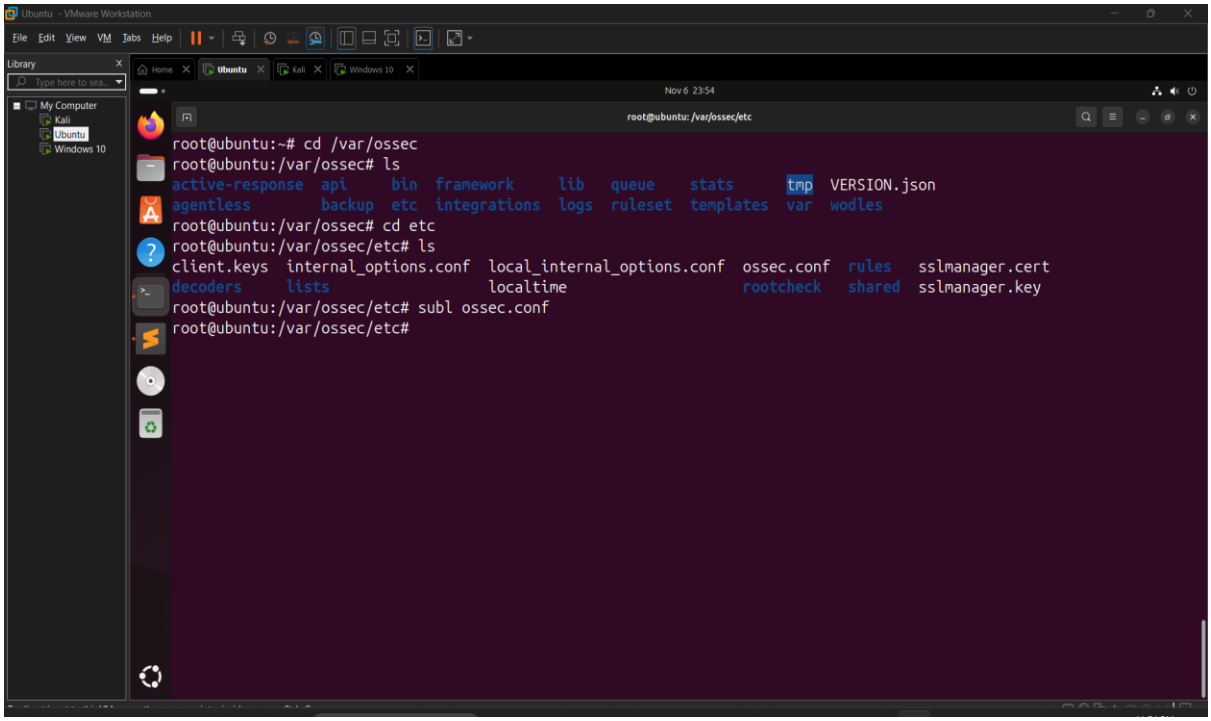
The image shows the terminal installing Sublime Text on Ubuntu.
First, the command sudo apt install snapd confirms that Snap is already installed and up to date.
Then the command sudo snap install sublime-text --classic installs Sublime Text version 4200 successfully.
A warning appears about Snap, but the installation completes without any errors.

2.3.1 Installing and Starting Wazuh Agent on Windows 10



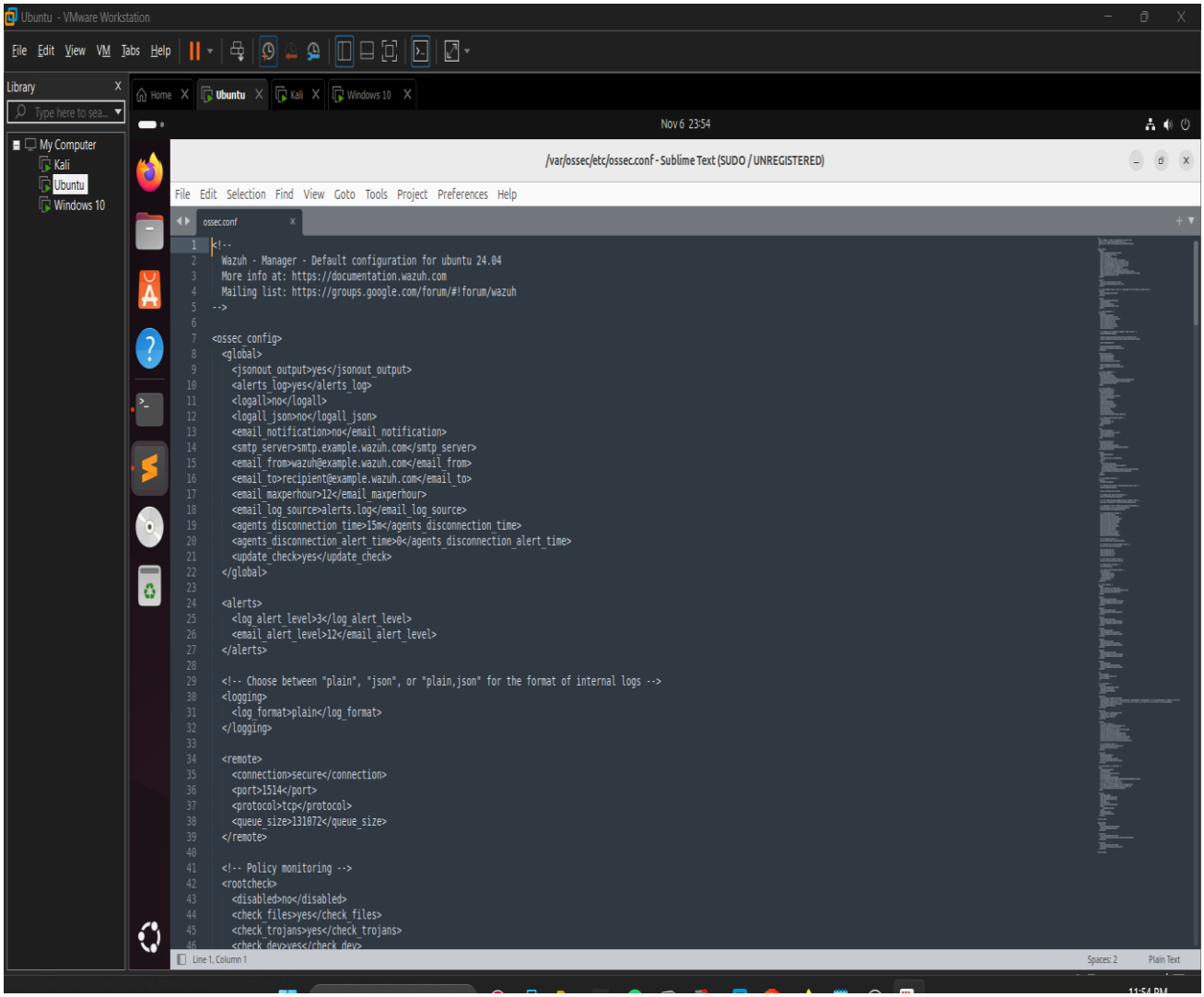
The image shows a Windows PowerShell session running as Administrator where the Wazuh agent is being installed.
The agent installer is executed with parameters specifying the Wazuh Manager IP (192.168.154.131) and the agent name “windows-10”.
After installation, the command NET START wazuh is used, and the output confirms that **“The Wazuh service was started successfully.”**
This indicates that the Windows machine is now connected and reporting to the Wazuh manager.

2.3.2 Accessing Wazuh Configuration Files on Ubuntu



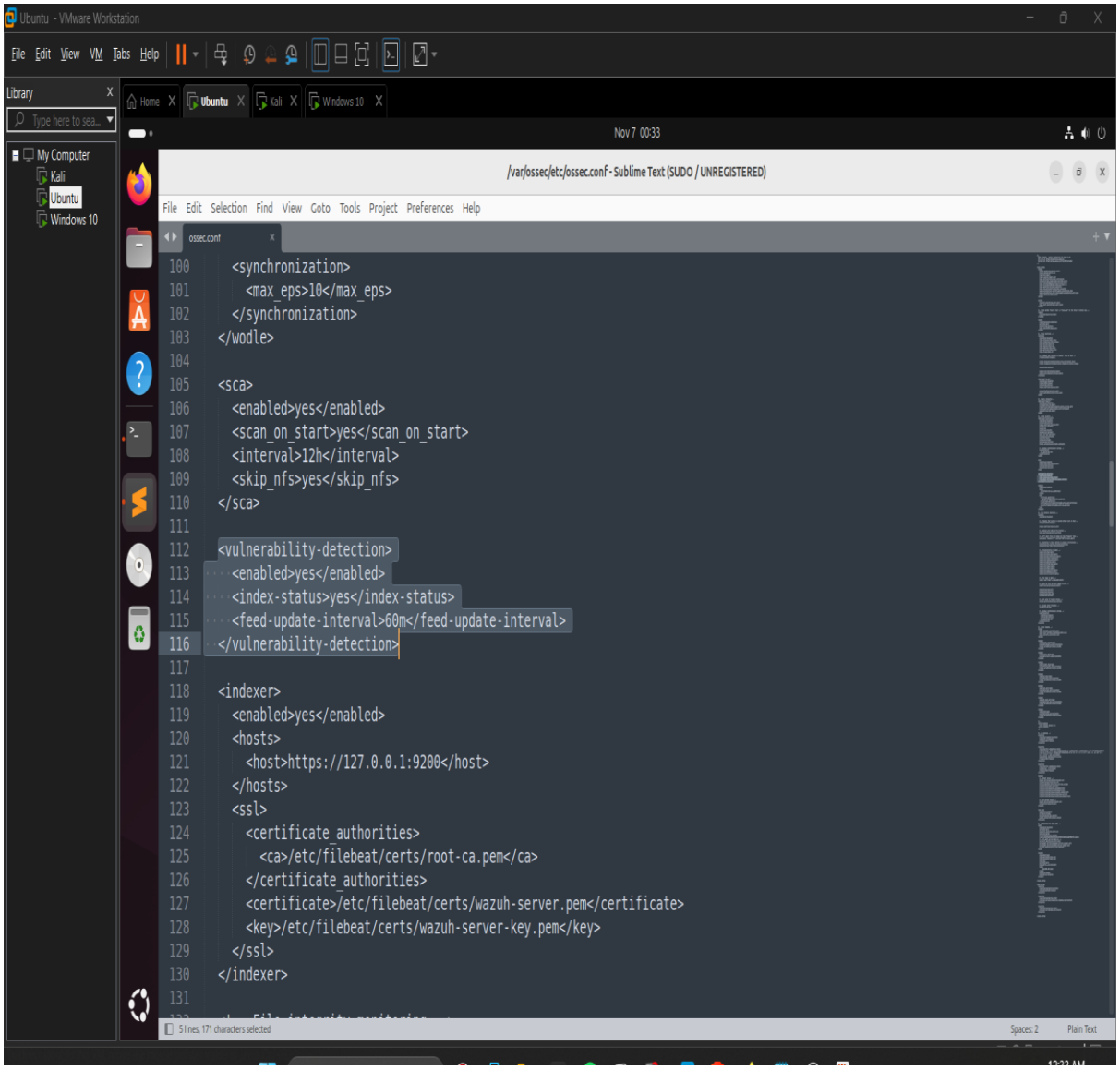
The image shows navigation inside the Wazuh installation directory /var/ossec on Ubuntu.
The user lists directories such as active-response, api, framework, logs, ruleset, and tmp.
Inside /var/ossec/etc, configuration files appear, including ossec.conf, client.keys, and decoder/rule folders.
Finally, the command subl ossec.conf is used to open the main Wazuh configuration file in Sublime Text.

2.3.4 **Wazuh Manager Main Configuration File (ossec.conf)**



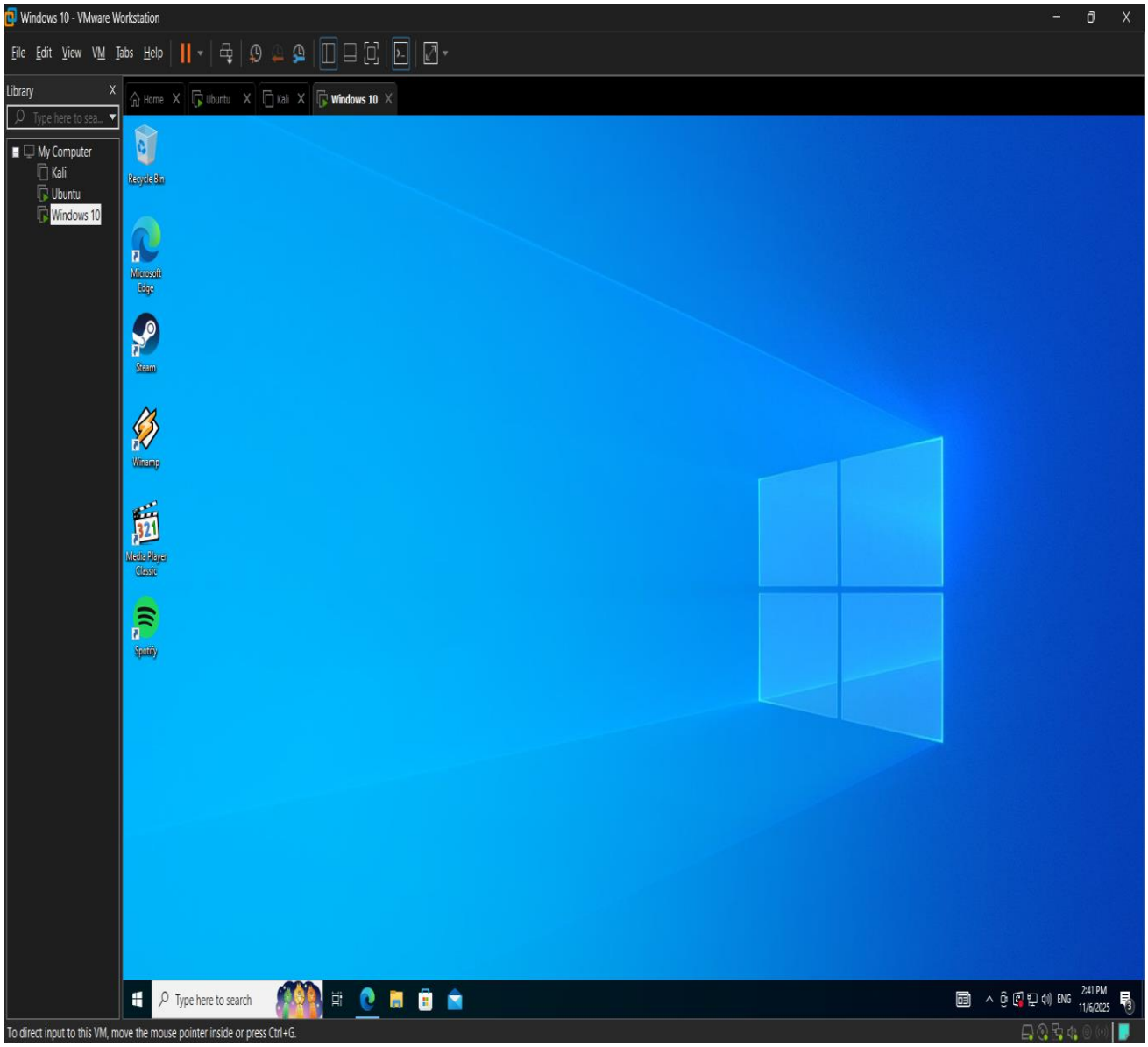
The image displays the ossec.conf file opened in Sublime Text on Ubuntu.
This file contains the core configuration for the Wazuh Manager, including global settings, alert levels, email notifications, logging format, and remote communication parameters.
Sections like <global>, <alerts>, <logging>, and <remote> define how logs are processed, how alerts are generated, and how agents connect to the manager.
This configuration file is essential for customizing Wazuh's behavior, integrations, and security policies.

2.3.5 Vulnerability Detection Configuration in ossec.conf



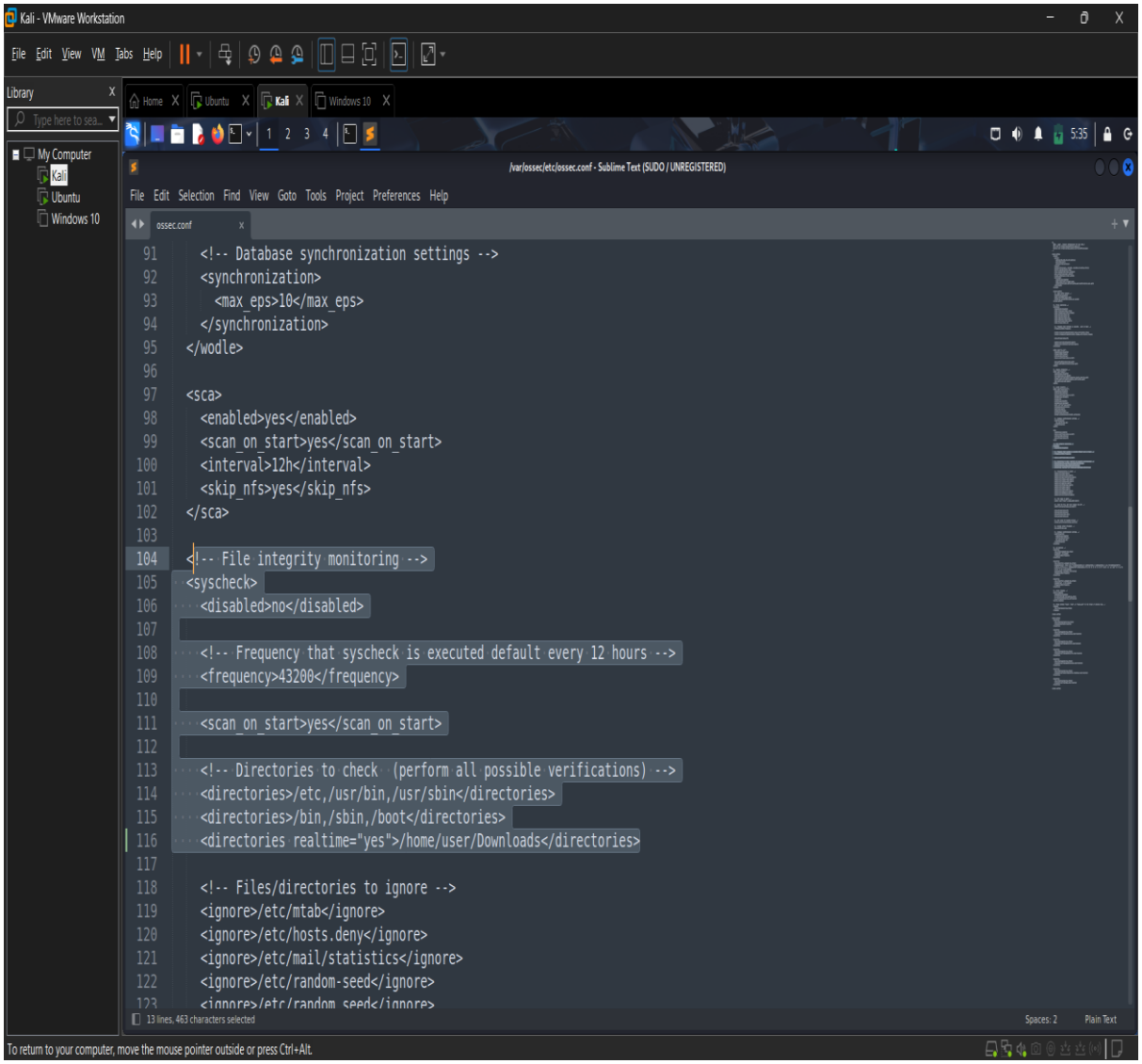
The image shows a highlighted section of the ossec.conf file that configures Wazuh's vulnerability detection module.
The <vulnerability-detection> block is enabled, with options that activate indexing and set the vulnerability feed update interval to **60 minutes**.
This configuration ensures continuous scanning of agents for known vulnerabilities and keeps the detection database updated automatically.
It is a key component for maintaining real-time security awareness across all connected endpoints.

2.3.6 Windows 10 Desktop After Agent Deployment



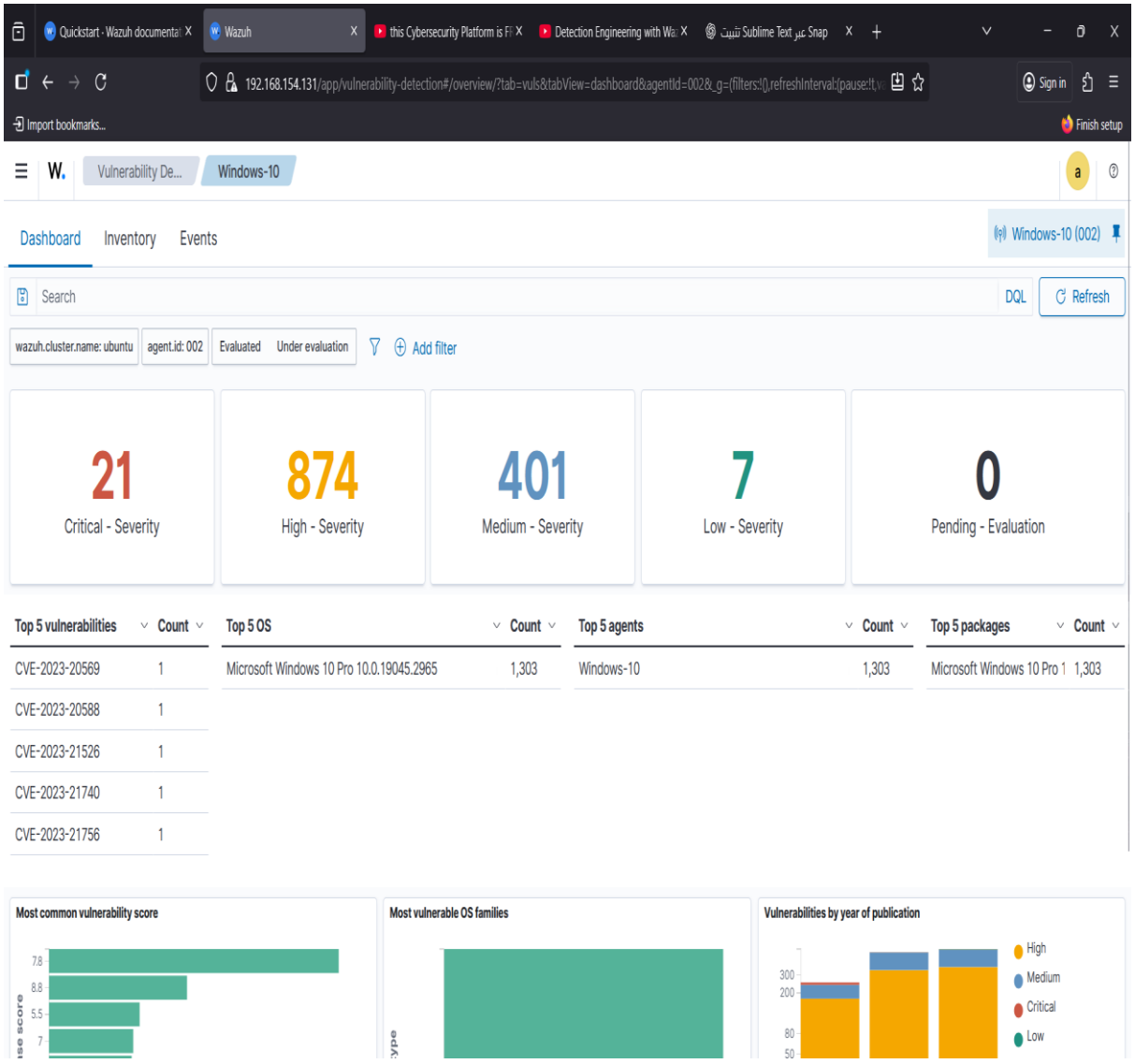
The image shows the Windows 10 virtual machine’s desktop after completing the Wazuh agent installation.
The system appears idle and functioning normally, with common applications such as Edge, Steam, Winamp, Media Player Classic, and Spotify visible.
This confirms that the endpoint is stable and ready for monitoring by the Wazuh server.
The installation did not interfere with the operating system’s performance or desktop environment.

2.3.7 File Integrity Monitoring Configuration in ossec.conf



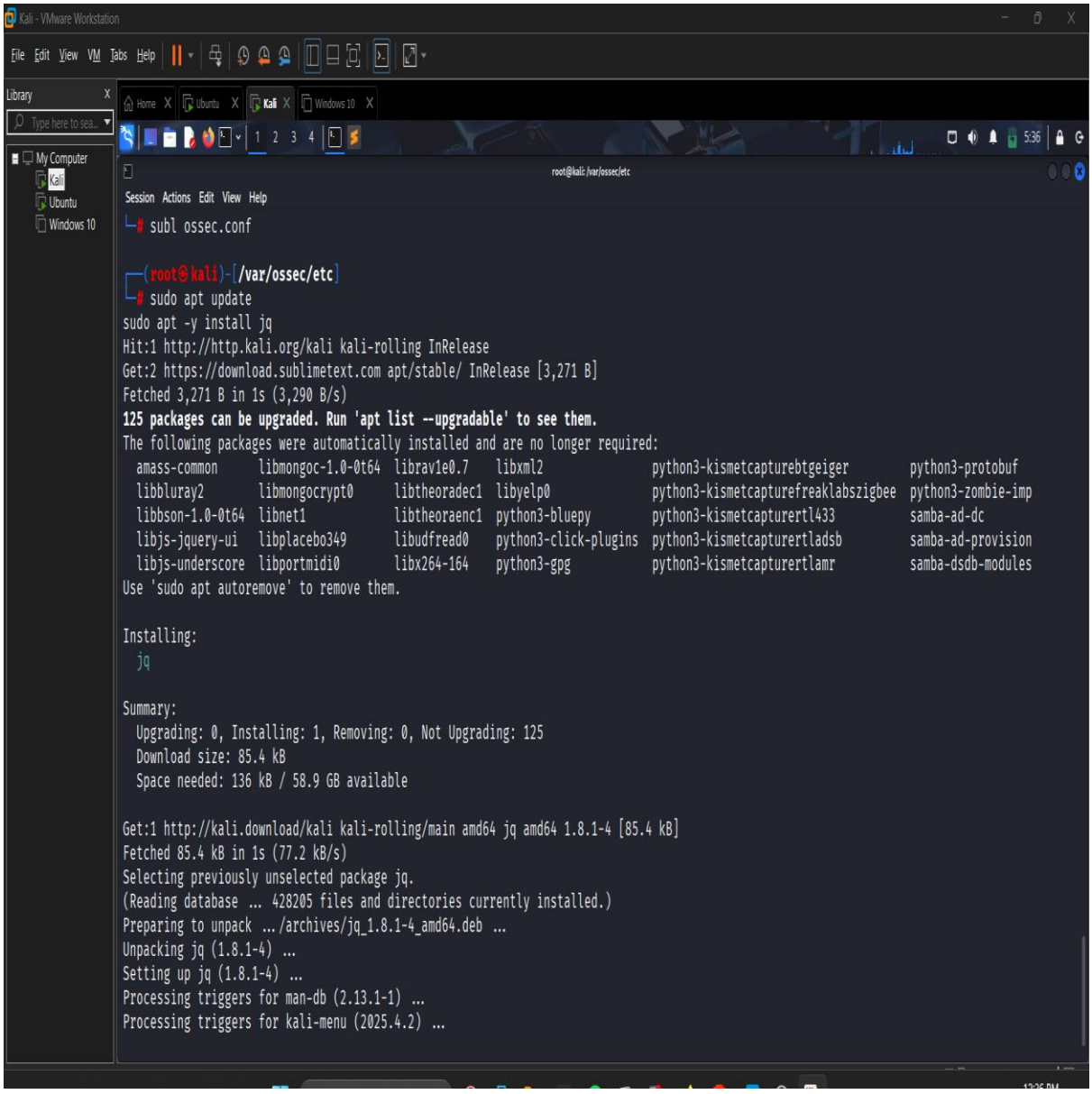
The image shows the syscheck section of the ossec.conf file, which controls File Integrity Monitoring (FIM) in Wazuh.
The module is **enabled**, with a scan frequency of **43,200 seconds** (12 hours) and a scan triggered on startup.
Wazuh is configured to monitor key system directories such as /etc, /usr/bin, /usr/sbin, /bin, /sbin, and /boot, as well as the user's Downloads folder.
Ignore rules exclude temporary or irrelevant paths, ensuring efficient integrity monitoring and reducing false alerts.

2.3.8 Updated Vulnerability Detection Results – Windows 10 Agent



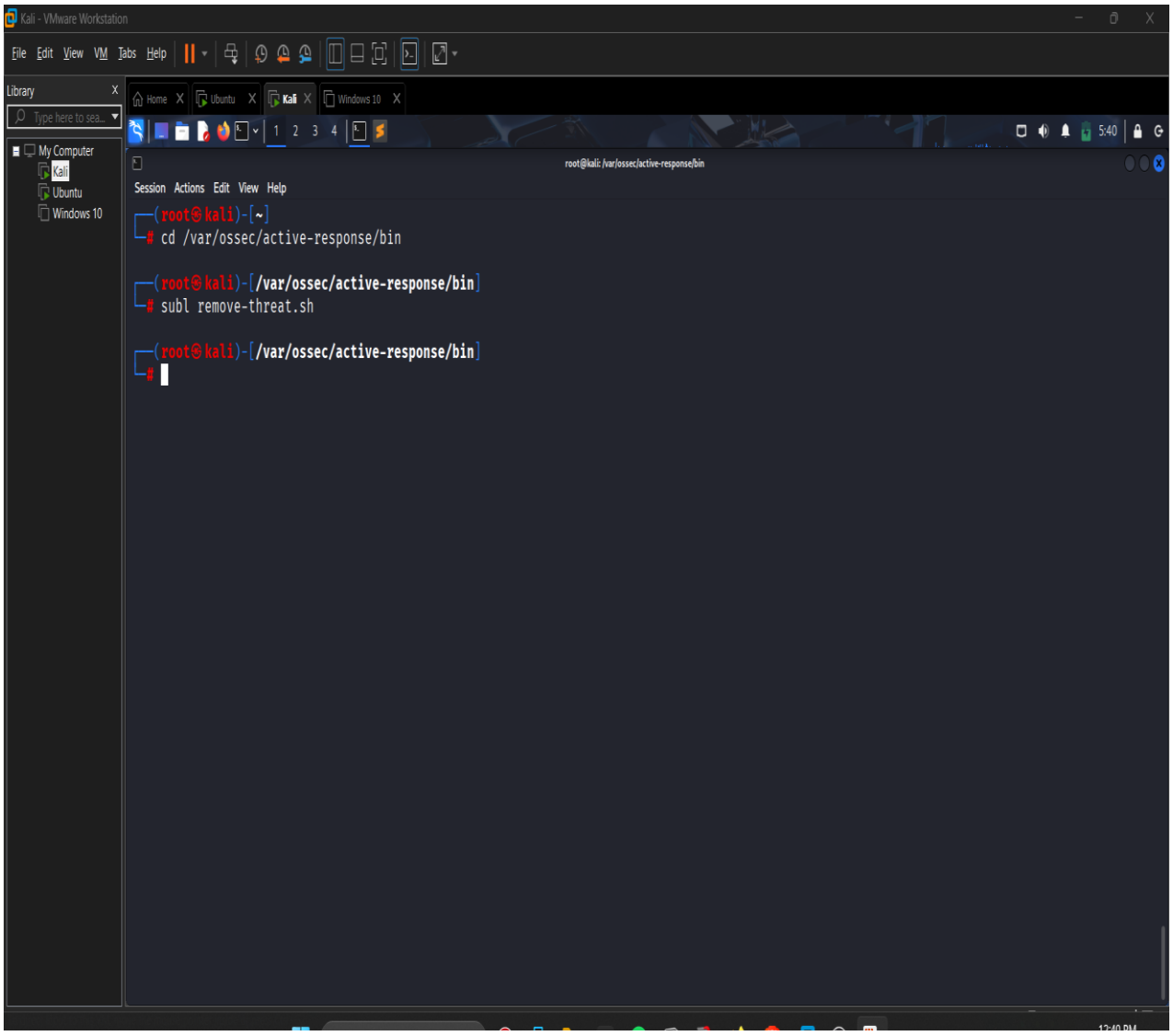
The image shows the latest vulnerability scan results for the Windows-10 agent in the Wazuh dashboard.
The system reports **21 critical**, **874 high**, **401 medium**, and **7 low** severity vulnerabilities, showing a slight increase from the previous scan.
It identifies the Windows 10 Pro OS build and lists recent CVEs detected on the endpoint.
This dashboard helps monitor changes in vulnerability levels over time and guides remediation priorities.

2.3.9 Updating Kali Linux and Installing jq Utility



The image shows a Kali Linux terminal running system update commands.
sudo apt update checks available package updates, showing **125 packages** that can be upgraded.
Next, the command sudo apt install jq installs the jq JSON processing tool, which is often required for parsing vulnerability or log data.
The installation completes successfully, preparing the system for advanced log or API analysis tasks.

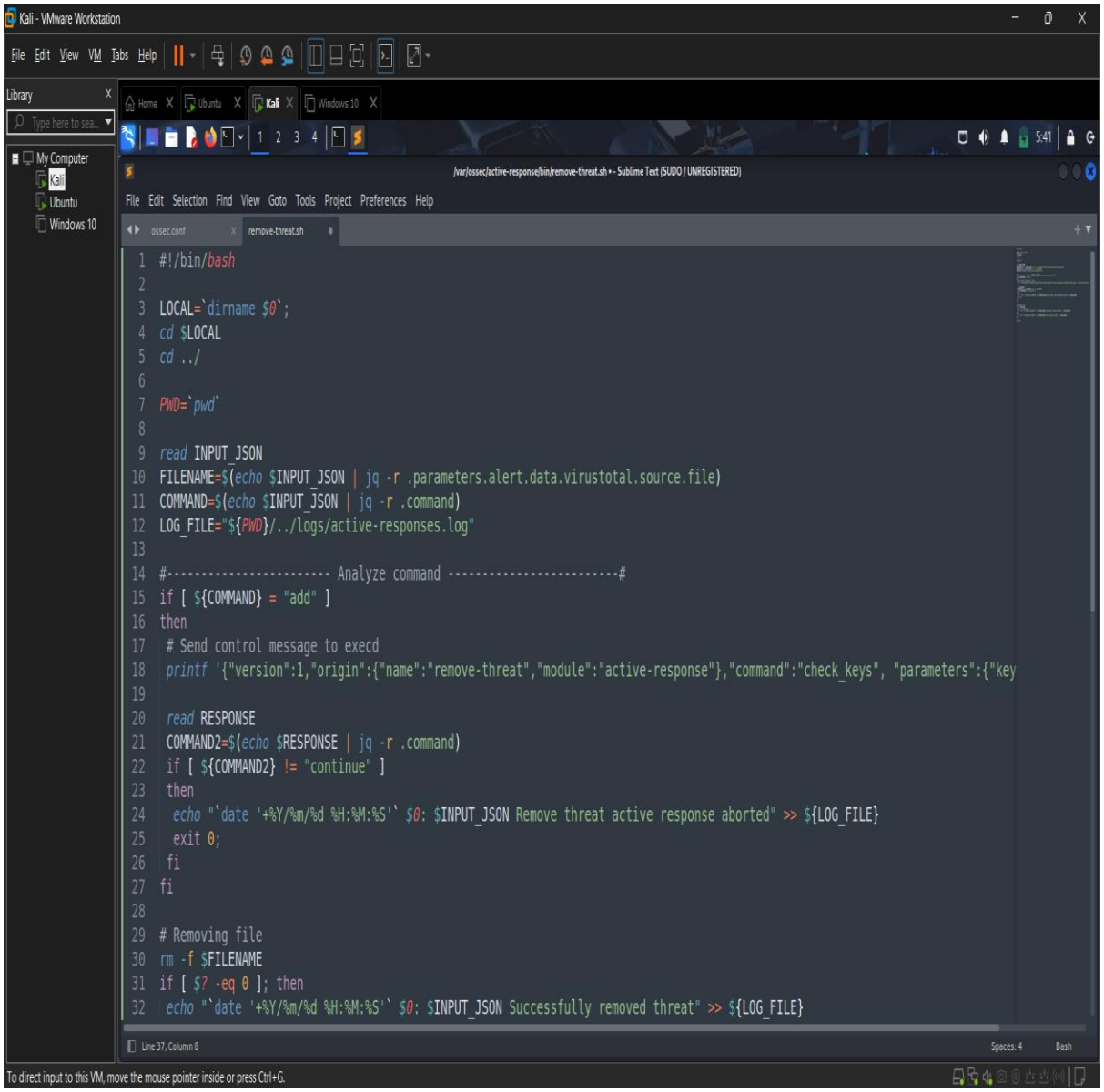
2.4 Accessing Active Response Scripts in Wazuh Agent



The image shows navigation into the directory /var/ossec/active-response/bin on a Kali Linux system.

This folder contains executable scripts used by Wazuh's Active Response module to take automated actions against threats.
The command subl remove-threat.sh opens the script in Sublime Text for editing or review.
This script can be customized to automatically mitigate or block malicious activity detected by the Wazuh agent.

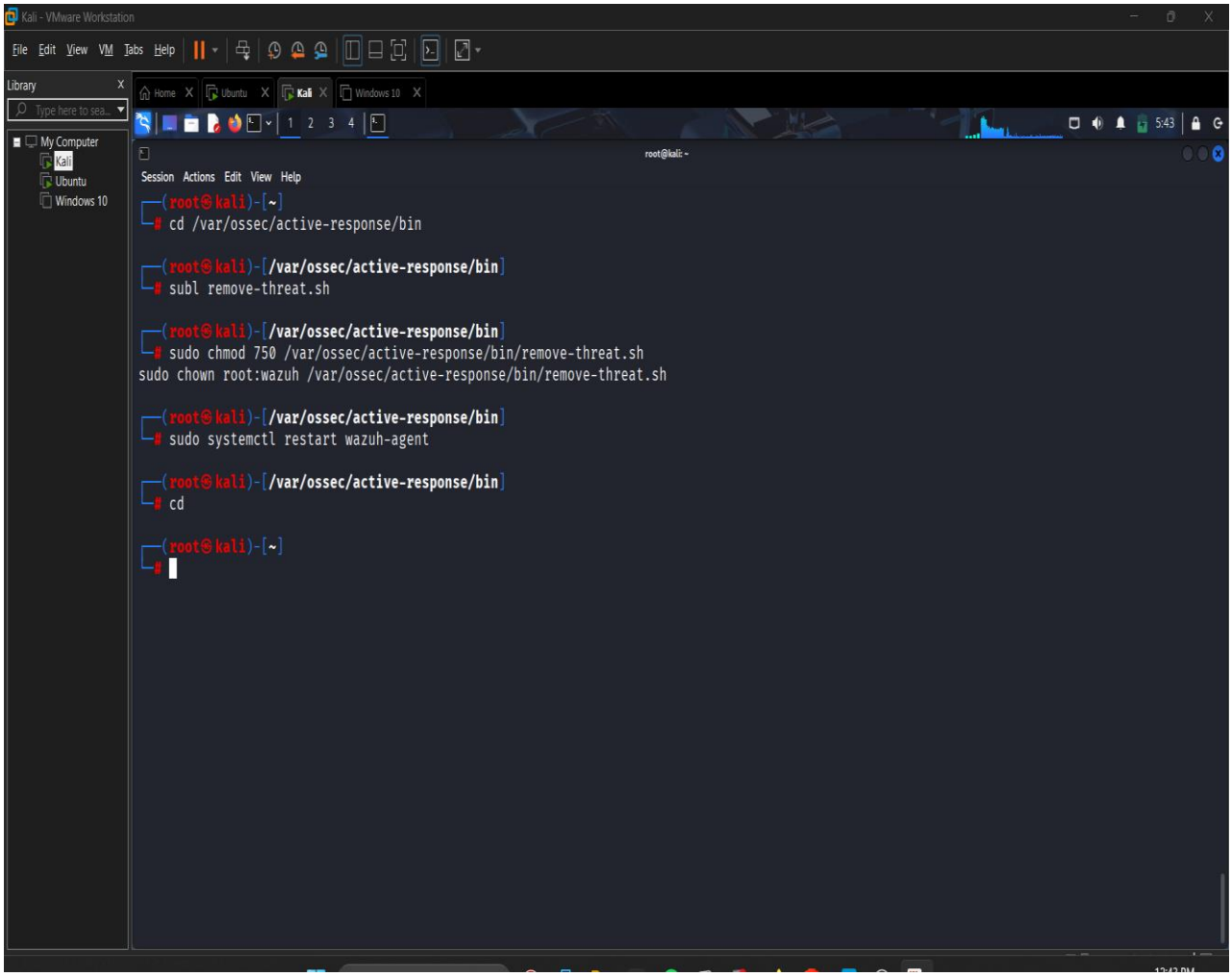
2.4.1 Remove-threat.sh – Active Response Script Logic



The image shows the contents of the remove-threat.sh script used by Wazuh's Active Response module.
The script reads JSON input, extracts the malicious file path and

command using jq, and logs activity to active-responses.log.
If the command is “add”, it communicates with execd to validate the action; if approved, the script proceeds to remove the detected threat file.
Successful or aborted actions are logged with timestamps, enabling automated and traceable threat response on the endpoint.

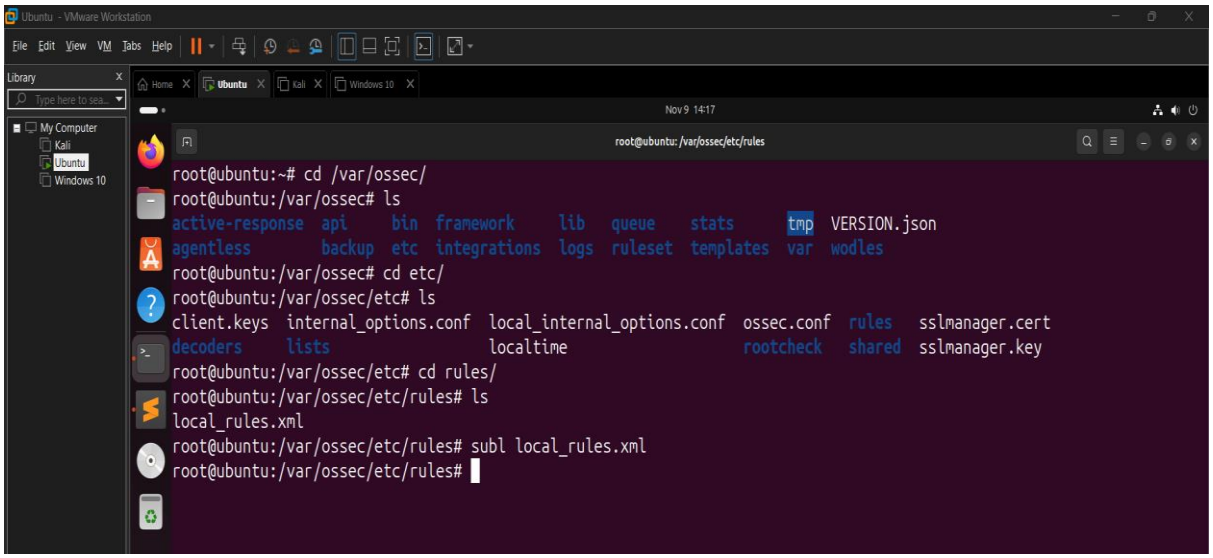
2.4.2 Updating Script Permissions and Restarting Wazuh Agent



The image shows commands executed to activate and apply the custom remove-threat.sh active response script.
chmod 750 sets secure execution permissions, and chown root:wazuh assigns ownership to the correct user and group.
The command sudo systemctl restart wazuh-agent reloads the agent so it can recognize and use the updated script.

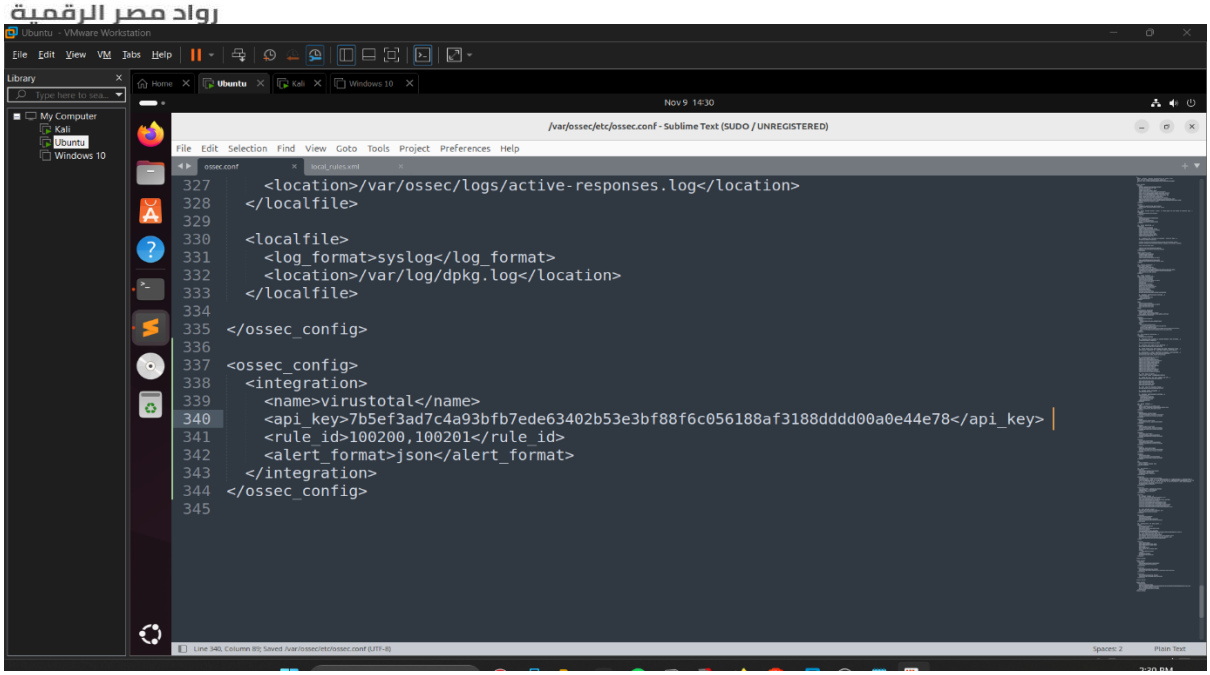
This finalizes the setup of automated threat removal functionality on the Kali endpoint.

2.4.3 Accessing Custom Wazuh Rules (local_rules.xml)



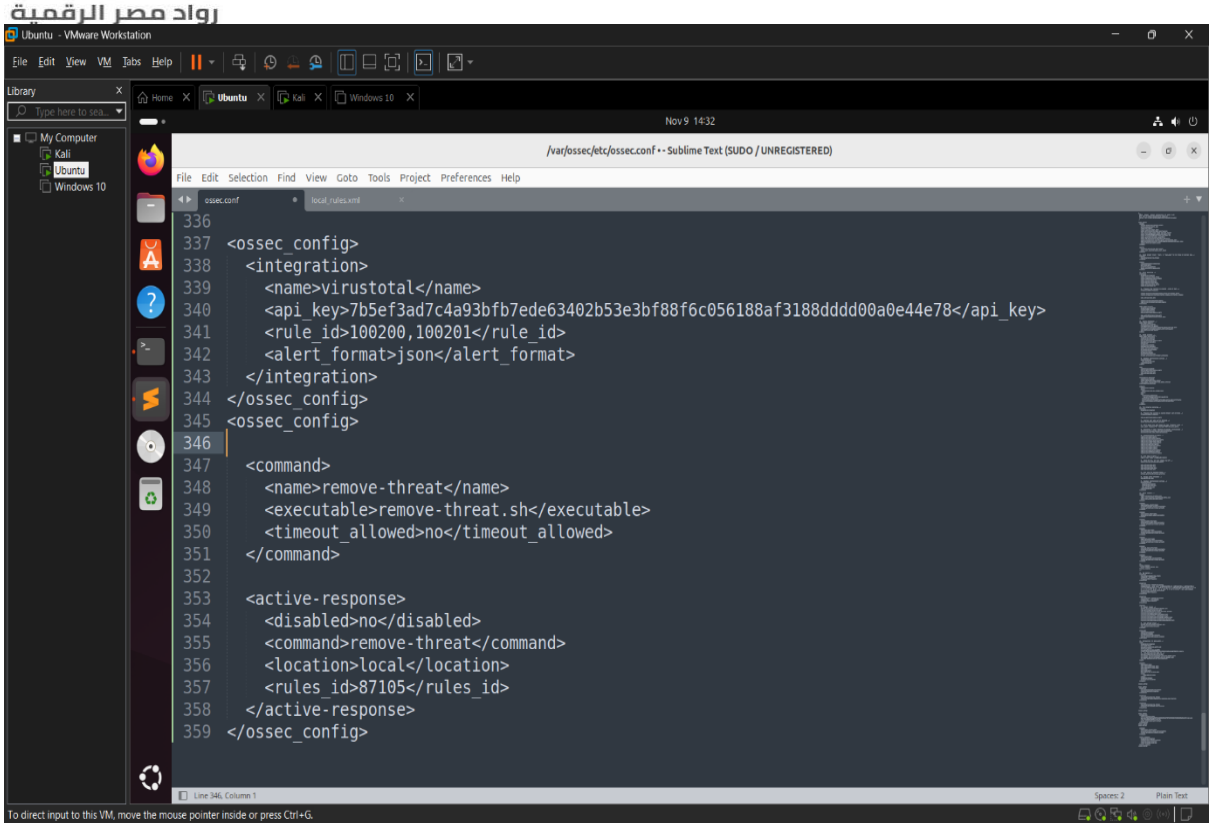
The image shows the navigation to the Wazuh custom rules directory at /var/ossec/etc/rules on Ubuntu.
Inside the folder, the file local_rules.xml is listed, which is used to create or modify custom correlation rules.
The command subl local_rules.xml opens this file in Sublime Text for editing.
This allows the administrator to define custom alerts, tune false positives, and integrate new detection logic into Wazuh.

2.4.5 Enabling VirusTotal Integration in Wazuh



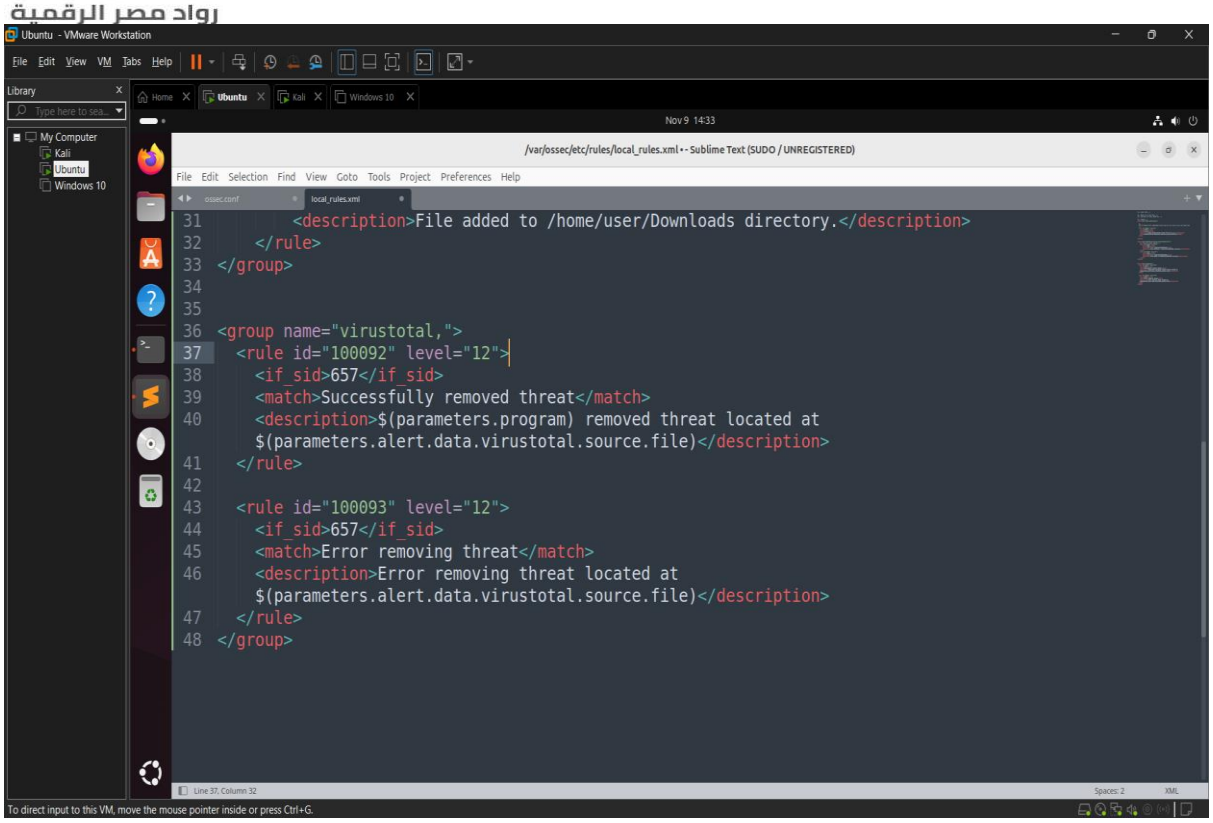
The image shows the ossec.conf file where a VirusTotal integration block has been added.
Inside this <integration> section, the name is set to **virustotal**, an API key is provided, and it is linked to custom rules **100200** and **100201**.
This configuration allows Wazuh to automatically send file hashes detected by those rules to VirusTotal for reputation analysis.
The result is an automated workflow that enriches alerts with external threat intelligence.

2.4.6 Adding the Custom Active Response Command to Wazuh



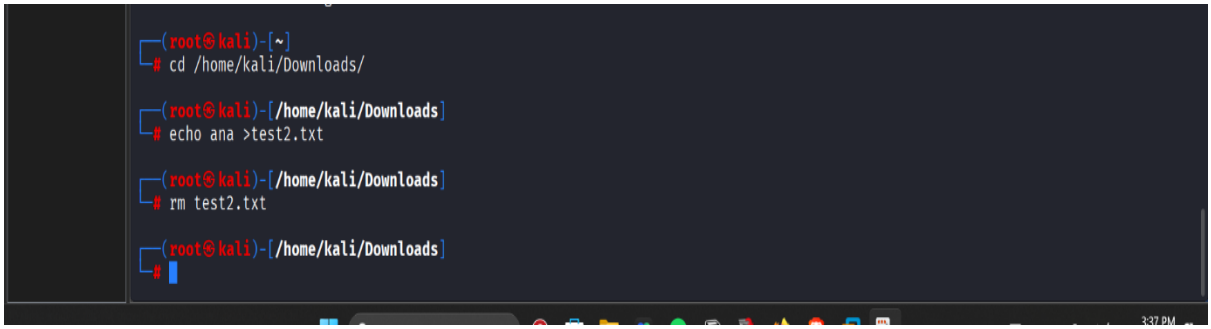
This section of the ossec.conf file defines and activates the custom **remove-threat** script.
Inside the <command> block, the script name remove-threat.sh is registered so the agent can call it when needed.
The <active-response> block then links this command to a specific rule (ID **87105**), ensuring the script runs automatically when that alert is triggered.
This configuration enables automated threat removal directly from Wazuh's alerting engine.

2.4.6 Custom Rules for VirusTotal Active Response Output



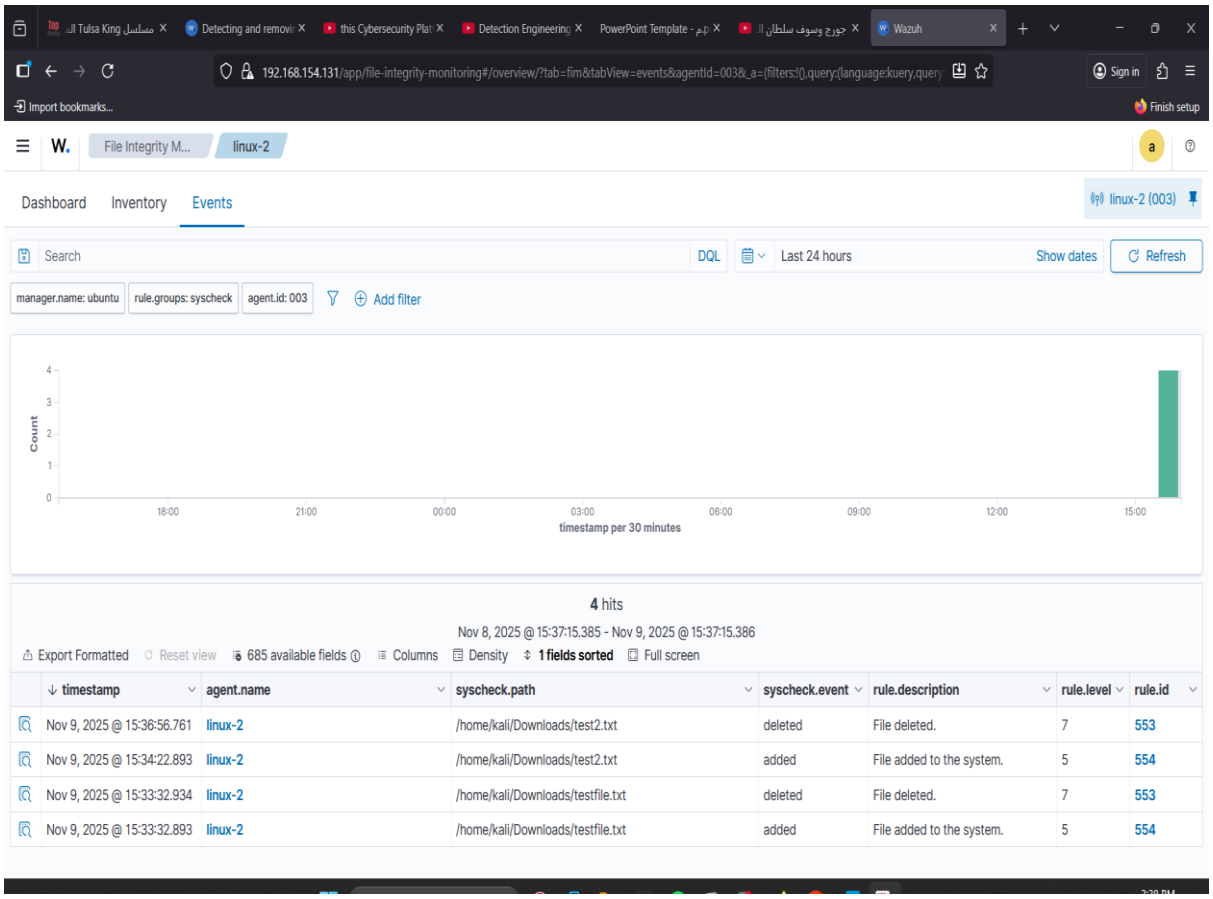
This part of local_rules.xml defines two high-severity rules (level 12) that interpret the results of your active-response script.
The first rule (100092) matches the message **"Successfully removed threat"** and creates a detailed alert including the file path returned by VirusTotal integration.
The second rule (100093) handles **failure cases**, generating an alert when the script reports an error while removing the threat.
These rules ensure clear visibility and logging of both successful and failed automated threat-removal actions.

2.4.7 Testing File Integrity Monitoring (Syscheck) on Kali



In this step, we navigated to the *Downloads* directory on the Kali machine to verify whether Wazuh's File Integrity Monitoring (Syscheck) is successfully detecting changes. I created a new file named **test2.txt** to trigger a "File added" alert, and then removed the same file to generate a "File deleted" alert. These actions confirm that the custom Syscheck rules—mapped to rule IDs **100200** and **100201**—are functioning correctly and that Wazuh is actively monitoring file activity inside the /home/kali/Downloads directory.

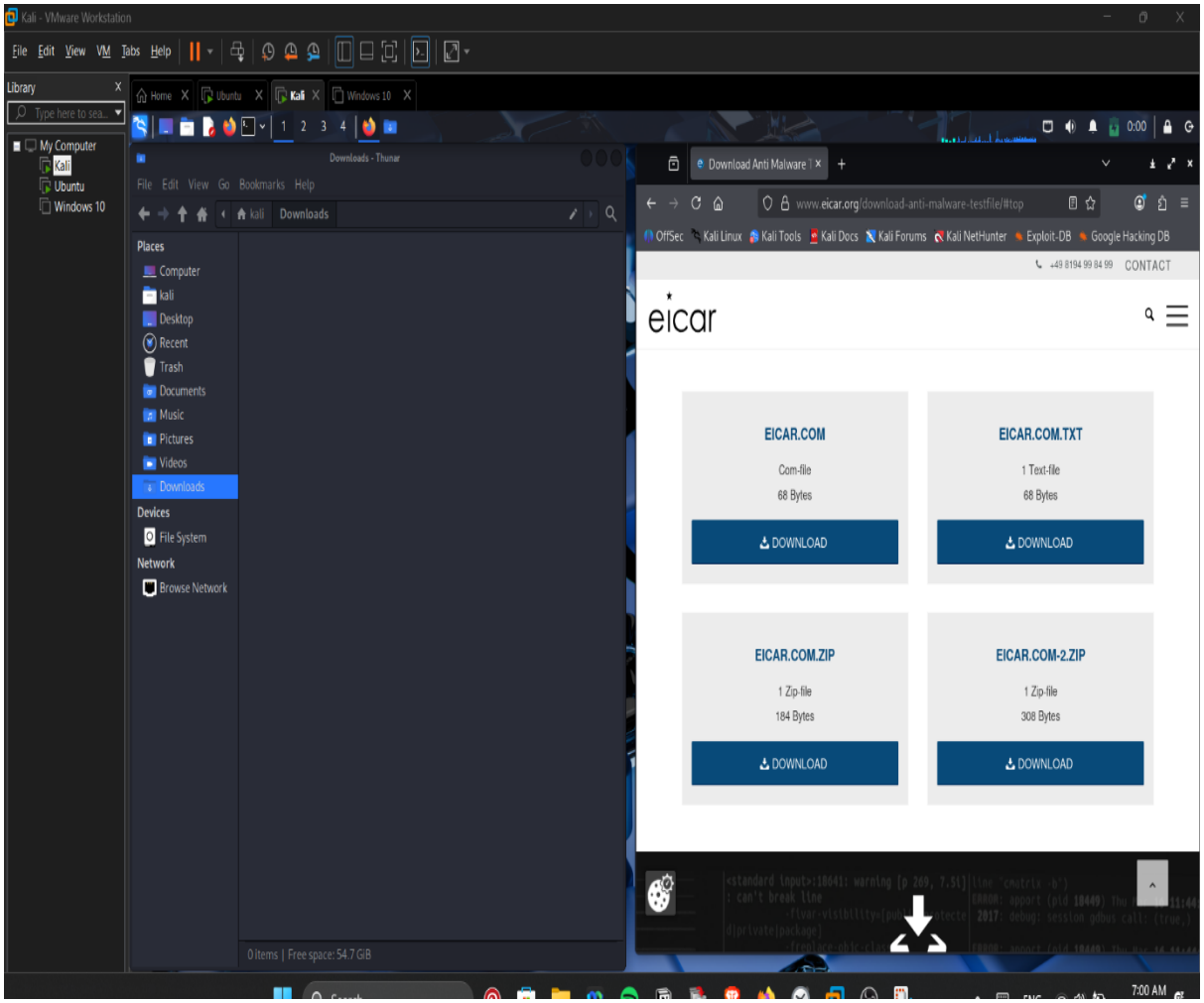
2.4.8 Syscheck Events Detected on the Linux Agent



timestamp	agent.name	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
Nov 9, 2025 @ 15:36:56.761	linux-2	/home/kali/Downloads/test2.txt	deleted	File deleted.	7	553
Nov 9, 2025 @ 15:34:22.893	linux-2	/home/kali/Downloads/test2.txt	added	File added to the system.	5	554
Nov 9, 2025 @ 15:33:32.934	linux-2	/home/kali/Downloads/testfile.txt	deleted	File deleted.	7	553
Nov 9, 2025 @ 15:33:32.893	linux-2	/home/kali/Downloads/testfile.txt	added	File added to the system.	5	554

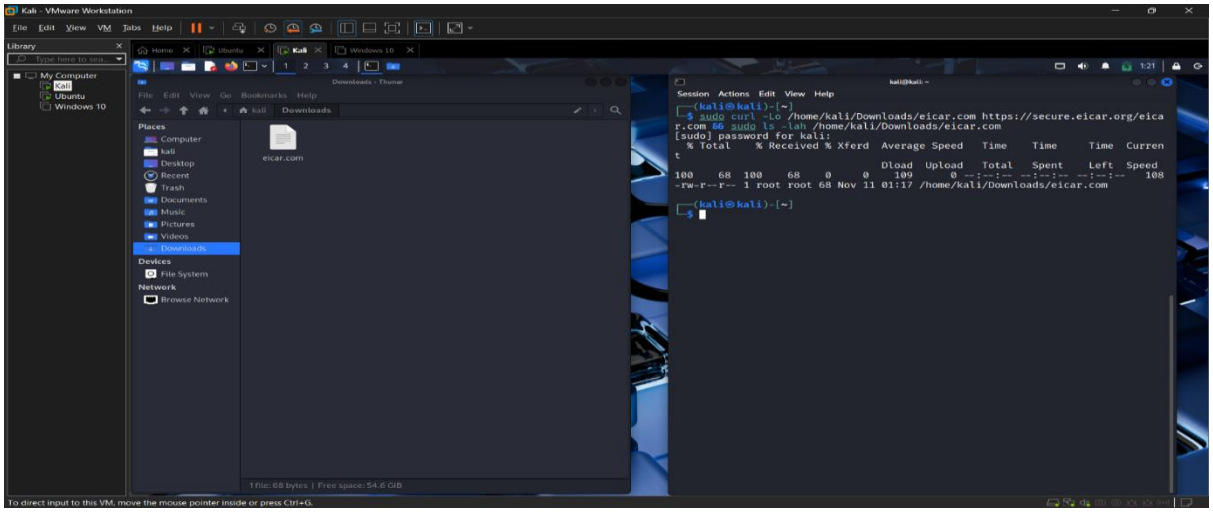
This screenshot shows the **Wazuh File Integrity Monitoring (FIM)** results for the Linux agent after performing file creation and deletion tests in the /home/kali/Downloads directory. The Events tab displays four alerts generated by Syscheck, indicating two files being **added** and **deleted**. Each event includes the file path, event type, rule triggered, and severity level, confirming that the custom monitoring rules and Syscheck module are functioning correctly.

2.4.9 Downloading the EICAR Test File on Kali Linux



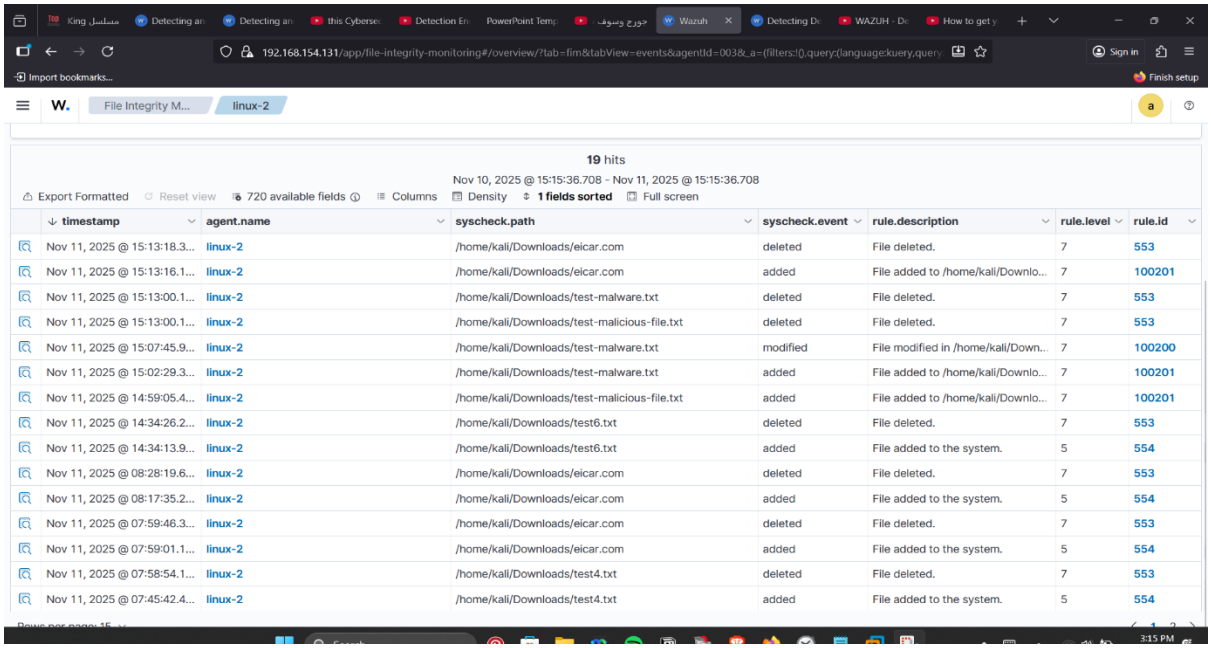
The screenshot shows the **Kali Linux Downloads folder** open on the left and the **official EICAR test file download page** on the right. The EICAR file is a harmless, standardized test file used worldwide to safely trigger malware detection systems. This setup indicates that you are preparing to **download the EICAR test file** to verify that your **Wazuh File Integrity Monitoring + VirusTotal integration + Active Response script** detects, analyzes, and removes the file as expected.

2.5 Downloading the EICAR Test File via Terminal on Kali Linux



The screenshot shows the EICAR test file successfully downloaded into the **Kali Linux Downloads folder** using a curl command. On the left, the file manager displays the newly created **eicar.com** file. On the right, the terminal confirms the download with correct file permissions and size (68 bytes). This step is part of testing your Wazuh FIM + VirusTotal + Active Response workflow by introducing a safe malware test file into the monitored directory.

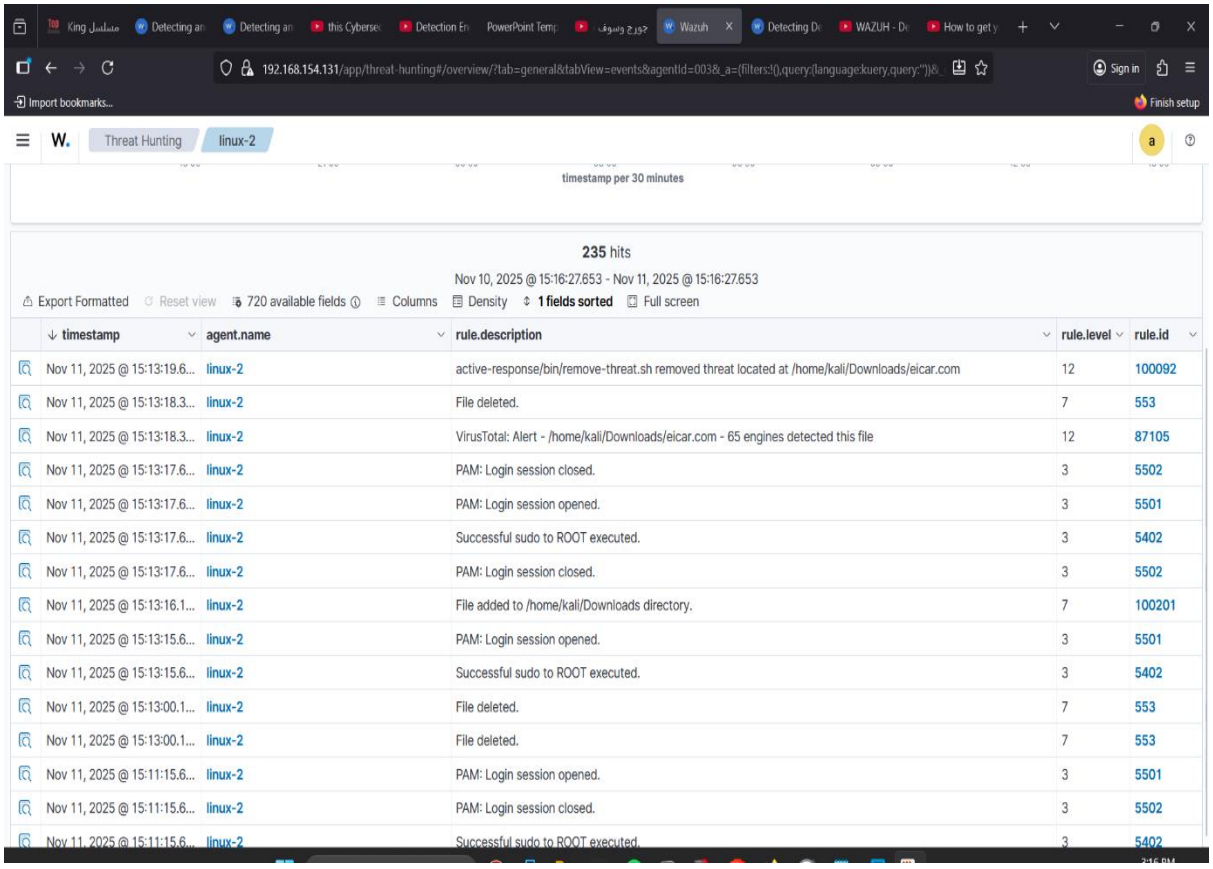
2.5.1 Full File Integrity Activity Log for the Linux Agent



timestamp	agent.name	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
Nov 11, 2025 @ 15:13:18.3...	linux-2	/home/kali/Downloads/eicar.com	deleted	File deleted.	7	553
Nov 11, 2025 @ 15:13:16.1...	linux-2	/home/kali/Downloads/eicar.com	added	File added to /home/kali/Downlo...	7	100201
Nov 11, 2025 @ 15:13:00.1...	linux-2	/home/kali/Downloads/test-malware.txt	deleted	File deleted.	7	553
Nov 11, 2025 @ 15:13:00.1...	linux-2	/home/kali/Downloads/test-malicious-file.txt	deleted	File deleted.	7	553
Nov 11, 2025 @ 15:07:45.9...	linux-2	/home/kali/Downloads/test-malware.txt	modified	File modified in /home/kali/Down...	7	100200
Nov 11, 2025 @ 15:02:29.3...	linux-2	/home/kali/Downloads/test-malware.txt	added	File added to /home/kali/Downlo...	7	100201
Nov 11, 2025 @ 14:59:05.4...	linux-2	/home/kali/Downloads/test-malicious-file.txt	added	File added to /home/kali/Downlo...	7	100201
Nov 11, 2025 @ 14:34:26.2...	linux-2	/home/kali/Downloads/test6.txt	deleted	File deleted.	7	553
Nov 11, 2025 @ 14:34:13.9...	linux-2	/home/kali/Downloads/test6.txt	added	File added to the system.	5	554
Nov 11, 2025 @ 08:28:19.6...	linux-2	/home/kali/Downloads/eicar.com	deleted	File deleted.	7	553
Nov 11, 2025 @ 08:17:35.2...	linux-2	/home/kali/Downloads/eicar.com	added	File added to the system.	5	554
Nov 11, 2025 @ 07:59:46.3...	linux-2	/home/kali/Downloads/eicar.com	deleted	File deleted.	7	553
Nov 11, 2025 @ 07:59:01.1...	linux-2	/home/kali/Downloads/eicar.com	added	File added to the system.	5	554
Nov 11, 2025 @ 07:58:54.1...	linux-2	/home/kali/Downloads/test4.txt	deleted	File deleted.	7	553
Nov 11, 2025 @ 07:45:42.4...	linux-2	/home/kali/Downloads/test4.txt	added	File added to the system.	5	554

This screenshot shows the **complete list of File Integrity Monitoring (FIM) events** detected by the Wazuh agent on *linux-2*. The table records every action performed inside the /home/kali/Downloads directory— including adding, modifying, and deleting files such as **eicar.com**, **test-malware.txt**, and other test files. Each event is associated with a specific rule ID and severity level, confirming that Syscheck is actively tracking all file operations for security analysis and threat detection.

2.5.2 Threat Hunting Alerts Showing VirusTotal Detection & Active Response Execution

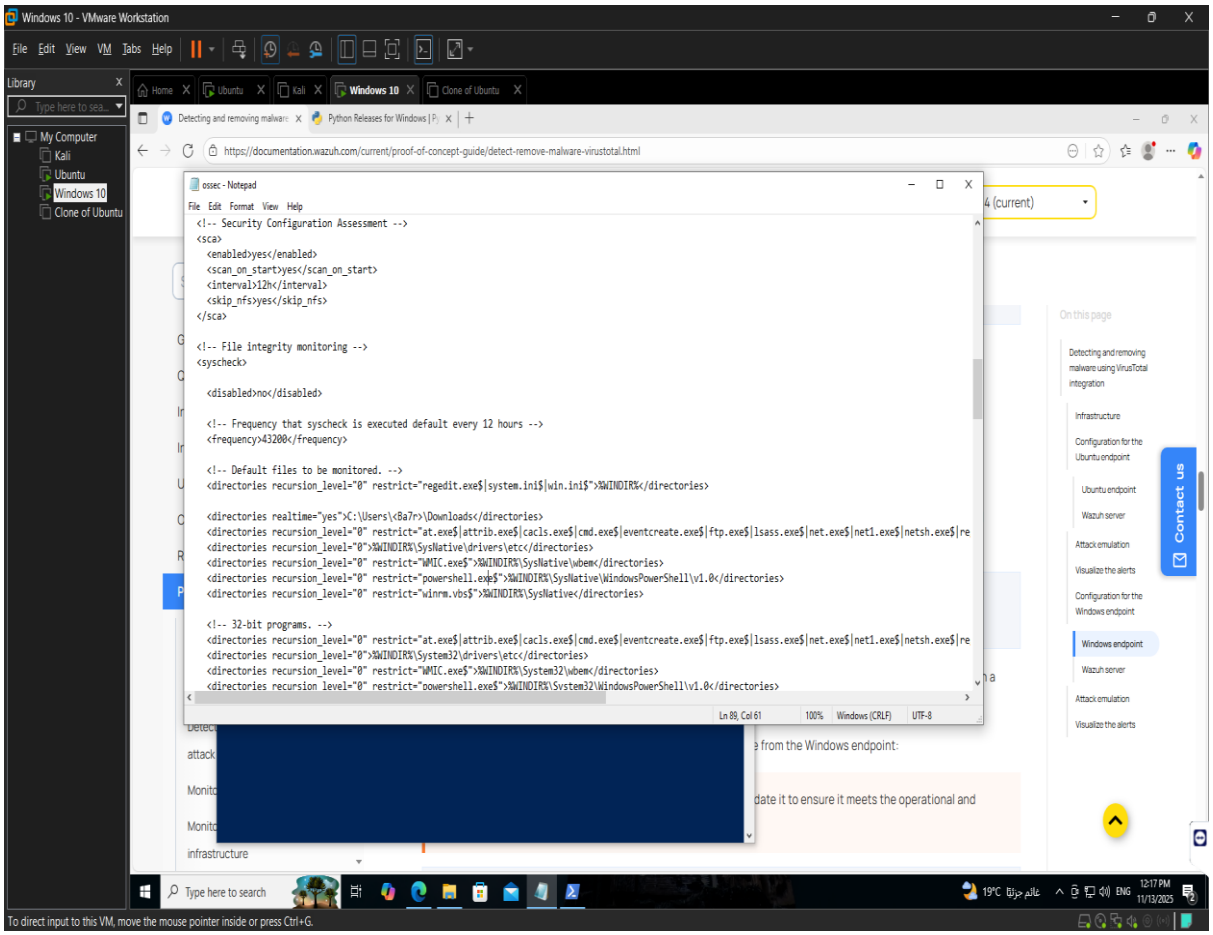


timestamp	agent.name	rule.description	rule.level	rule.id
Nov 11, 2025 @ 15:13:19.6...	linux-2	active-response/bin/remove-threat.sh removed threat located at /home/kali/Downloads/eicar.com	12	100092
Nov 11, 2025 @ 15:13:18.3...	linux-2	File deleted.	7	553
Nov 11, 2025 @ 15:13:18.3...	linux-2	VirusTotal: Alert - /home/kali/Downloads/eicar.com - 65 engines detected this file	12	87105
Nov 11, 2025 @ 15:13:17.6...	linux-2	PAM: Login session closed.	3	5502
Nov 11, 2025 @ 15:13:17.6...	linux-2	PAM: Login session opened.	3	5501
Nov 11, 2025 @ 15:13:17.6...	linux-2	Successful sudo to ROOT executed.	3	5402
Nov 11, 2025 @ 15:13:17.6...	linux-2	PAM: Login session closed.	3	5502
Nov 11, 2025 @ 15:13:16.1...	linux-2	File added to /home/kali/Downloads directory.	7	100201
Nov 11, 2025 @ 15:13:15.6...	linux-2	PAM: Login session opened.	3	5501
Nov 11, 2025 @ 15:13:15.6...	linux-2	Successful sudo to ROOT executed.	3	5402
Nov 11, 2025 @ 15:13:00.1...	linux-2	File deleted.	7	553
Nov 11, 2025 @ 15:13:00.1...	linux-2	File deleted.	7	553
Nov 11, 2025 @ 15:11:15.6...	linux-2	PAM: Login session opened.	3	5501
Nov 11, 2025 @ 15:11:15.6...	linux-2	PAM: Login session closed.	3	5502
Nov 11, 2025 @ 15:11:15.6...	linux-2	Successful sudo to ROOT executed.	3	5402

This screenshot displays the **Threat Hunting** page for the *linux-2* Wazuh agent, where multiple security events have been recorded. The key highlight is that Wazuh detected the **EICAR test malware file**, triggered the VirusTotal integration, and automatically executed the **active response script** (remove-threat.sh) to delete the file. The table also shows related FIM events, PAM login activity, sudo escalations,

and full rule metadata — confirming successful malware detection, verification by VirusTotal, and automated threat removal.

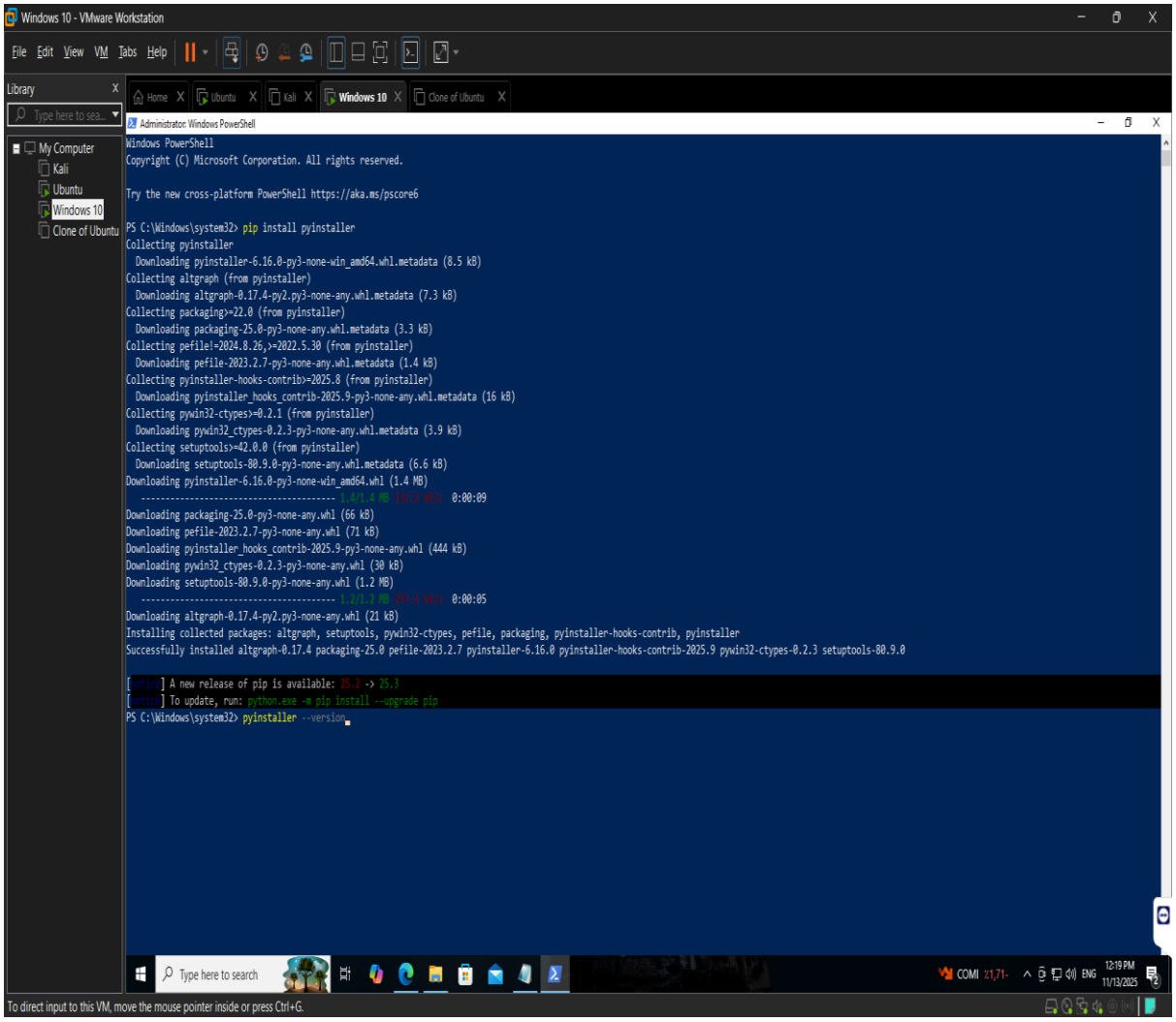
2.5.3 Windows Wazuh Agent Configuration for Real-Time Malware and File Integrity Monitoring



The screenshot shows the **Windows Wazuh Agent ossec.conf file** opened in Notepad, where the agent is being configured to monitor system activity and detect malicious files. In this configuration, **File Integrity Monitoring (FIM)** is enabled, along with **Security Configuration Assessment (SCA)**, both running at 12-hour intervals and also scanning on startup. The most important part is the line enabling **real-time monitoring** on the Downloads directory, meaning any file added, modified, or deleted in C:\Users\<User>\Downloads will immediately generate an alert. This is essential for detecting malware dropped by the user—such as the EICAR test file—allowing Wazuh to trigger VirusTotal evaluation and active responses

automatically. The screenshot also shows the agent monitoring important Windows system executables like cmd.exe, regedit.exe, PowerShell, WMI folders, and system32 drivers, ensuring that any suspicious modification to critical tools is detected.

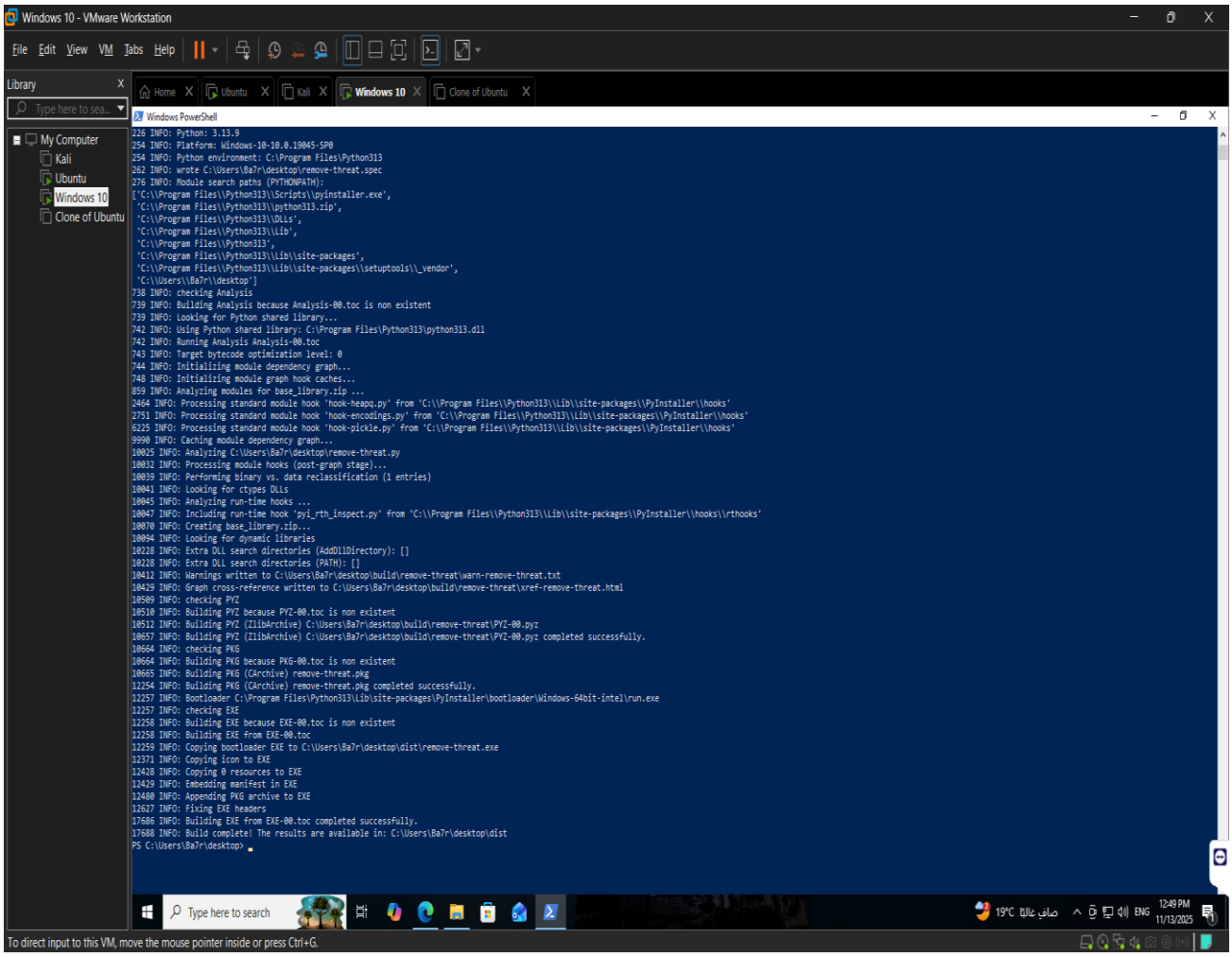
2.5.4 Installing PyInstaller on Windows via PowerShell



The image shows an **elevated Windows PowerShell session** where PyInstaller is being installed using the command pip install pyinstaller. The terminal output displays the process of downloading required Python packages such as *altgraph*, *pywin32-ctypes*, *pefile*, and *pyinstaller-hooks-contrib*. After all dependencies are downloaded and installed successfully, PowerShell returns to the prompt, where the user runs pyinstaller --version to verify that the installation is complete. This step confirms that Python packaging is ready, often

used in malware analysis or for building executable files in a security testing environment.

2.5.5 Building the Malware-Removal Script into an EXE Using PyInstaller



In this image, the Windows PowerShell console shows the full process of **compiling a Python script (remove-threat.py) into a standalone Windows executable** using PyInstaller. The output logs reveal several internal stages: analyzing the Python environment, loading module hooks, checking dependencies, collecting Python libraries, and packaging all required components into a distributable EXE. PyInstaller then generates supporting files (like PYZ archives, bootloader, and resources), and finally creates the executable remove-threat.exe inside the dist directory on the Desktop. The completion message confirms that the build succeeded, meaning the malware-

removal script is now ready to be executed on Windows systems without needing Python installed.

Chapter 3 – Malware Prevention Strategy

3.1 Introduction

Malware prevention is one of the cornerstones of enterprise security. Even with advanced SIEM systems, strong endpoint protection, and continuous monitoring, cyberattacks often succeed because they exploit human behavior rather than technical vulnerabilities. Organizations that rely only on detection and response without strengthening prevention expose themselves to unnecessary risk, operational disruption, and potential financial loss.

This chapter introduces a comprehensive malware prevention framework designed to reduce the likelihood of infection within the enterprise environment. The strategy focuses on:

- User awareness and behavioral defense
- Email-based threat identification
- Secure handling of attachments and hyperlinks
- Recognition of malicious patterns in communication
- Password hygiene and identity protection
- Social engineering countermeasures
- Proper SOC escalation workflow
- Real-world phishing examples
- Additional learning resources

Every section is rewritten with deeper explanations and more practical guidance to match enterprise-grade cybersecurity needs.

3.2 User Awareness and Cyber Hygiene

User awareness forms the first and most effective line of defense against malware. Most successful cyberattacks begin with a user interacting with a malicious message, opening an unsafe file, responding to a fraudulent request, or using weak authentication.

Strengthening user awareness requires regular training, simulation exercises, clear communication, and reinforcing safe digital habits. This section covers the elements users must understand and practice consistently.

Core Principles of User Awareness

- Always verify the source of communication
- Treat unexpected emails, links, and attachments with suspicion
- Avoid sharing sensitive information over email or messaging apps
- Follow organizational security policies and reporting procedures
- Use strong, unique passwords for every account
- Enable multi-factor authentication whenever possible
- Report any suspicious behavior to the security team immediately

Effective awareness programs significantly reduce infection rates and help detect attacks early, before malware spreads internally.

3.2.1 Identifying Phishing Attempts

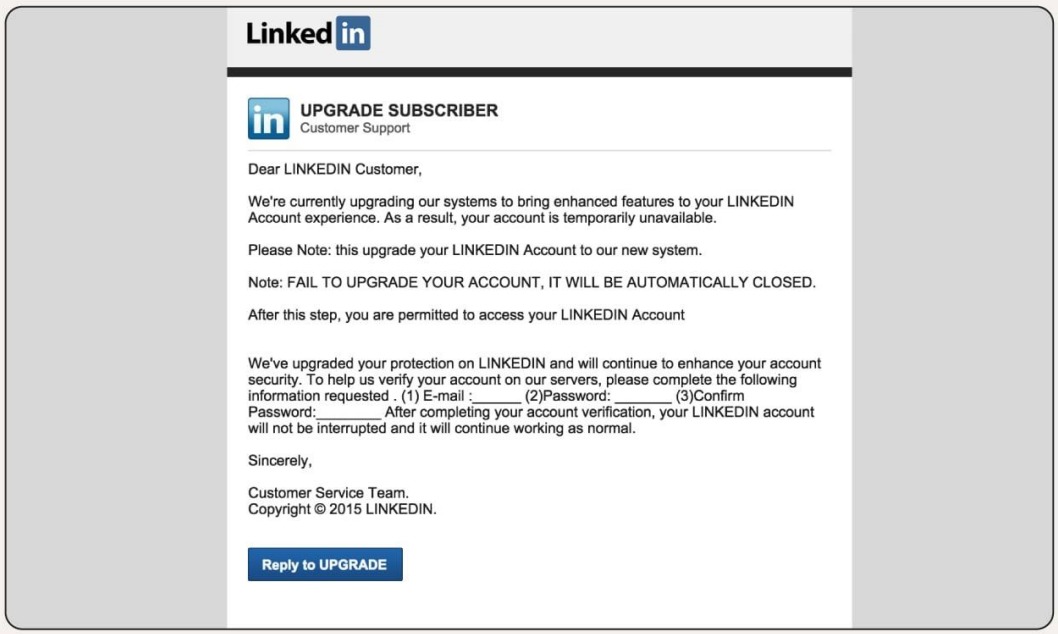
Phishing emails are responsible for the majority of malware infections, credential theft incidents, and unauthorized access events. Attackers rely heavily on phishing because it works — users are more likely to trust emails that resemble official communication, especially when urgency or authority is used to pressure compliance.

Common Indicators of Phishing Emails

1. **Unusual Sender Address**
   The attacker uses an address similar to a legitimate domain, such as:
   - support@micros0ft-secure.com
   - hr-department@payrolI-system.com
2. **Unexpected Attachments or Links**
   Emails claiming you have an invoice, shipment update, or unpaid bill.
3. **Requests for Credentials**
   Messages asking you to "confirm password," "verify identity," or "unlock account."
4. **Urgency or Threatening Language**
   Examples include:
   - "Your account will be disabled within 6 hours"
   - "Final warning: verify now"
5. **Generic Greetings**
   "Dear customer" instead of your actual name.
6. **Grammar Mistakes and Poor Formatting**
   Attackers often use automated translators or rushed templates.

How to Evaluate a Suspicious Email

- Hover over hyperlinks to preview the real URL
- Inspect sender domain carefully
- Compare message format to previous legitimate messages
- Ask yourself: "Did I expect this email?"
- Report the message before taking any action



3.2.2 Identifying Malicious Attachments

Malware is frequently delivered through attachments disguised as legitimate files. Attackers use socially convincing subjects such as invoices, government forms, salary notifications, or security updates.

High-Risk File Types

- Executables: .exe, .scr, .bat
- Scripts: .js, .vbs, .ps1
- Office documents with macros: .docm, .xlsm
- Compressed malware: .zip, .rar, .7z
- PDF files containing embedded exploit code

How Attackers Hide Malicious Files

- Double extensions:
  Report2025.pdf.exe
- Fake icons mimicking Word, PDF, or Excel

- Encrypted ZIP files to bypass mail scanning
- Files named after well-known companies like DHL, Amazon, and Microsoft

Safe Attachment Handling Practices

1. Verify the sender and purpose of the file
2. Check the file extension carefully
3. Do not enable macros unless required and verified
4. Upload suspicious files to SOC for analysis
5. Avoid decompressing unknown ZIP/RAR archives
6. Use sandboxing tools for unsafe files

3.2.3 Password Hygiene and Credential Protection

Passwords remain a high-value target for attackers. Weak, reused, or predictable passwords allow attackers to gain unauthorized access without needing to deploy malware. Once credentials are stolen, attackers can escalate privileges, move laterally, and exfiltrate sensitive data.

Characteristics of Strong Passwords

- At least 14 characters
- Composite structure: upper/lowercase letters, numbers, symbols
- Not related to personal information
- Unique for every account
- Stored only in secure password managers

Common Credential Attacks Prevented by Strong Hygiene

- Brute force
- Credential stuffing
- Password spraying

- Dictionary attacks
- Session hijacking

Organizational Requirements

- Mandatory MFA for privileged accounts
- Quarterly password change for sensitive systems
- Prohibition of browser-saved passwords
- Centralized authentication (e.g., Active Directory + MFA)

3.2.4 Social Engineering Awareness

Social engineering is one of the most dangerous techniques used by attackers because it targets human psychology rather than software vulnerabilities. A single successful social engineering attempt can bypass firewalls, antivirus solutions, and even advanced SIEM systems.

Why Social Engineering Works

- Users tend to trust messages that appear professional
- Curiosity leads users to open links or attachments
- Attackers imitate authority figures (IT, HR, managers)
- Users respond quickly to urgent or emotional requests

Common Social Engineering Techniques

1. **Impersonation**
   Attacker pretends to be an internal employee or external vendor.
   Example: Fake “IT Support” asking for login details.
2. **Pretexting**
   Creating a fake scenario to extract information.
   Example: “We detected unusual login attempts; please verify your email.”

3. **Baiting**
   Offering something attractive, such as “Free gift card” or “Urgent bonus report.”
4. **Tailgating / Piggybacking**
   Physically following an employee into a secure area.
5. **Vishing (Voice Phishing)**
   Phone calls pretending to be from a bank or government agency.

Defensive Measures Against Social Engineering

- Do not trust unsolicited emails or phone calls
- Verify identity through official internal channels
- Never share verification codes or passwords
- Report any suspicious behavior immediately
- Follow organizational escalation procedures

3.2.5 Handling Malicious Links

Links embedded in emails or websites are a primary gateway for phishing attacks and drive-by malware downloads. Attackers craft URLs that resemble legitimate websites or redirect users to infected servers.

Characteristics of Malicious Links

- Misspelled domain names (typosquatting)
  Example: micros0ft.com or faceb00k-security.com
- Unsecure HTTP links when the real service uses HTTPS
- Shortened URLs hiding their real destination
- Hyperlinks embedded in buttons (e.g., “Reset Password”)
- Redirect chains leading to malicious pages

How to Inspect Links Safely

1. **Hover the mouse** over the link to reveal the true URL
2. Check for:
    - Extra characters
    - Wrong spelling
    - Suspicious domain extensions
3. Avoid clicking promotional offers or unsolicited advertisements
4. Type important URLs manually instead of clicking received links
5. When in doubt → **Report to SOC before opening**

3.2.6 SOC Response to Suspected Phishing Attempts

When an employee reports a suspicious message or potential phishing email, the SOC plays a critical role in analyzing, confirming, and responding to the threat.

SOC Response Workflow

1. **Initial Verification**
    - Analyze header information
    - Extract URLs and check them against threat intelligence
    - Inspect attachments in isolated sandboxes
    - Compare patterns with known phishing campaigns
2. **Containment Actions**
    - Quarantine the email in all mailboxes
    - Block sender domain and IPs
    - Disable malicious links through mail gateway filtering
3. **Notification**

- Notify affected users
- Provide immediate instructions
- Identify additional recipients of the same email

4. **Documentation**
    - Create an incident ticket
    - Record indicators of compromise (IOCs)
    - Save samples for future analysis
5. **Post-Incident Review**
    - Update awareness training
    - Adjust email filtering rules
    - Share lessons learned with staff

3.2.7 Phishing Mail Shapes (Examples & Visual Patterns)

Phishing emails follow predictable visual patterns that make them easier to recognize when users are trained properly. This section describes the common shapes of phishing messages and highlights their deceptive elements.

General Structure of Phishing Emails

- Fake company logo
- Generic greeting (e.g., “Dear User”)
- Urgent message (“Your account will be deactivated”)
- Call-to-action button
- Suspicious or mismatched URL
- Fake signature imitating an official department

Types of Phishing Email Designs

1. **Account Verification Scam**
   Message claims login issues or expired credentials.
2. **Delivery Notice Scam**
   Pretends to be DHL, FedEx, or Aramex.
   Usually contains a malicious tracking link.
3. **Finance/Invoice Scam**
   Uses fake invoices or "Payment Due" attachments.
4. **Security Alert Scam**
   Claims suspicious activity or unauthorized login attempts.
5. **HR / Payroll Scam**
   Appears to come from HR requesting updated employee info.

3.2.8 Additional Resources and Training Materials

Continuous education helps reduce the success rate of phishing attacks. Employees need regular reminders and interactive content to reinforce secure digital habits.

Recommended Training Resources

- Cybersecurity awareness videos
- Monthly newsletters with recent attack examples
- Interactive quizzes
- Mock phishing simulation campaigns
- Posters displaying safe email practices
- Workshops discussing new attack techniques

Topics to Include in Training

- How phishing works
- Ransomware behavior

- Password hygiene
- Safe browsing practices
- How to verify website authenticity
- How to report suspicious messages

Employee Best Practices Summary

- Stay cautious of unexpected messages
- Double-check links before clicking
- Avoid using personal email for work
- Update passwords regularly
- Treat unknown attachments with suspicion

3.3 Summary

Malware prevention is most effective when users understand the techniques attackers employ. While security teams manage monitoring and response, every employee plays a vital role in protecting the organization from infections. Adopting safe digital habits, learning how to identify phishing attacks, and knowing when to escalate concerns offer strong defense against malware.

A user equipped with awareness and proper training can prevent:

- Unauthorized access attempts
- Credential theft
- Malware execution
- Data breaches
- Ransomware spread

This chapter provided a comprehensive prevention framework aligned with real-world enterprise needs, ensuring users contribute actively to organizational cybersecurity resilience.

3.4 Organizational Policies Supporting Malware Prevention

Effective malware prevention depends not only on technical controls but also on clearly defined organizational policies. These policies guide user behavior, establish security expectations, and ensure consistent handling of risks across the company.

Key Policies Required:

1. Acceptable Use Policy (AUP)

Specifies acceptable behavior when using company devices and networks, including:

- Restricting installation of unauthorized applications
- Limiting access to high-risk websites
- Prohibiting the use of personal email for work-related files
- Blocking the connection of unknown removable devices (USB drives)

2. Email Security Policy

Defines safe handling of email communication:

- Do not open unexpected attachments
- Always verify the sender
- Report any suspicious email immediately
- Never share personal or organizational information via email

3. Password and Authentication Policy

Covers authentication requirements:

- Minimum password complexity
- Login attempt limits
- Mandatory multi-factor authentication (MFA)
- Password expiration intervals

4. Incident Reporting Policy

Provides clear reporting procedures:

- When users must report incidents
- Who they should contact
- What information must be included in the report
- Required follow-up actions

3.5 Technical Controls Supporting User Awareness

Even with strong user awareness programs, technical safeguards are essential to prevent malware from executing if a user makes a mistake.

1. Email Filtering and Anti-Spam Systems

Responsible for:

- Scanning links and attachments
- Checking messages against threat intelligence
- Identifying spoofed or impersonated senders
- Blocking known malicious domains

2. Endpoint Protection Platforms (EPP)

Protect endpoints through:

- Behavioral analysis
- Macro and script protection
- Memory and exploit prevention
- Recording suspicious activities

3. Web Filtering / DNS Filtering

Prevents users from accessing:

- Malware-hosting websites
- High-risk download portals
- Phishing domains
- Unverified domains

4. File Integrity Monitoring (FIM)

Detects unauthorized changes in:

- System files
- Configuration files
- Registry values
- Sensitive directories

5. Network Access Control (NAC)

Ensures that only compliant devices can connect to the network.

3.6 Real-World Examples of Malware Prevention Failures

Understanding real incidents helps illustrate how simple user mistakes can cause severe security breaches.

Case 1: Employee Opened a Fake HR Email

- A phishing email disguised as “HR Salary Update” was received.
- The attached PDF contained a Trojan.
- The malware harvested VPN credentials.
- Attackers used the credentials to access internal systems.
- Within two days, ransomware was deployed.

Root Cause:
Lack of awareness about fake HR emails and malicious attachments.

Case 2: User Enabled Macros on a Malicious Excel File

- File named "Financial-Report-Q4.xlsm."
- User enabled macros after seeing a fake warning.
- Macro downloaded a remote payload.
- A Remote Access Trojan (RAT) was installed.

Root Cause:
Unsafe macro usage and no verification of file origin.

Case 3: Weak Password Allowed Unauthorized Access

- Employee used the weak password: Company123
- Attackers brute-forced the password easily.
- Access was gained to multiple internal systems due to password reuse.

Root Cause:
Poor password hygiene and lack of authentication controls.

3.7 Preventive Communication Templates

Organizations often use predefined communication templates to remind users about safe practices and reduce the risk of malware infection.

Example 1 – Monthly Security Reminder

"Reminder: Never open attachments unless you are expecting them. If you receive suspicious emails, forward them to security@company.com."

Example 2 – Awareness Alert After Global Threat Activity

"A new wave of phishing attacks targeting regional companies has been observed.
Please remain cautious with financial or urgent action emails."

Example 3 – Macro Safety Notice

"Macros are disabled for your protection. Do not enable them unless instructed by the IT department."

3.8 Preventing Malware Through Browser Safety

Many malware infections occur through unsafe web browsing, even when no email is involved.

Safe Browsing Practices

- Only enter information on HTTPS websites
- Download software from official sources
- Avoid pop-up ads and unsolicited update banners
- Never click "Your system is infected" alerts
- Close the browser immediately if redirected unexpectedly

Common Browser-Based Attacks

1. **Drive-by Downloads**
   Malware installs without user interaction when visiting a compromised website.
2. **Malvertising (Malicious Advertisements)**
   Ads containing harmful scripts redirect users to infected pages.
3. **Fake Update Pop-ups**
   Fake notifications claiming a browser or Flash update is required.

4. **Credential Harvesting Pages**
   Login pages designed to steal usernames and passwords.

3.9 Mobile Device Risks

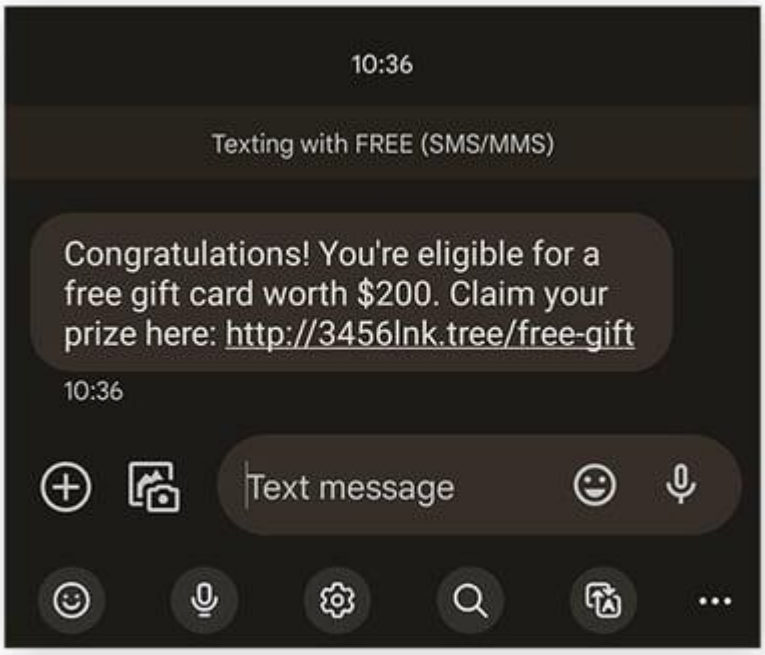
Mobile devices are increasingly targeted due to employees using them for work-related communication and access.

Common Mobile Threats

- Fake apps mimicking legitimate brands
- Malicious attachments in messaging apps
- Phishing links sent through SMS (smishing)
- Public Wi-Fi attacks

Mobile Security Best Practices

- Download apps only from official stores
- Enable device lock and authentication
- Keep the OS updated
- Avoid connecting the device to unknown computers
- Do not use public Wi-Fi for sensitive tasks



3.10 Preventing Malware Through USB and External Devices

Removable media continues to be a common method for spreading malware, especially in corporate environments.

USB Threat Scenarios

- Found USB devices plugged into work computers
- Promotional USB drives containing hidden malware
- USBs infected by multiple users
- Unknown devices connected to sensitive servers

Preventive Measures

- Disable AutoRun on all systems
- Use endpoint protection to block unauthorized USBs
- Encrypt authorized removable devices
- Scan all devices before usage
- Restrict copying sensitive data to USB drives

3.11 Preventing Malware Through Network Awareness

Even when users follow safe browsing and email practices, attackers can still exploit weak network behaviors. Network awareness training helps employees understand how insecure connections or untrusted environments increase malware risks.

Risks Associated with Insecure Network Usage

- Connecting to public Wi-Fi without VPN
- Downloading files from unsecured HTTP websites
- Using personal hotspots with weak passwords
- Automatically connecting to previously used networks

Safe Network Usage Practices

- Always use company-approved VPN when working remotely
- Avoid connecting to public or shared Wi-Fi
- Do not trust “Free Wi-Fi” hotspots in cafés or airports
- Ensure websites use HTTPS before entering credentials
- Disconnect from networks when not in use

3.12 Malware Prevention Through Email Behavior Monitoring

Organizations often deploy tools that analyze user email behavior to detect anomalies. These systems identify unusual communication patterns that may indicate compromised accounts or ongoing phishing campaigns.

Behavioral Indicators of Compromised Email Accounts

- Sudden increase in outgoing emails
- Messages sent outside typical working hours
- Emails containing unfamiliar language style
- Unauthorized forwarding rules created automatically
- Login attempts from unknown geographical regions

User Best Practices

- Regularly review mailbox forwarding settings
- Immediately report suspicious sent messages
- Do not reuse email passwords across platforms
- Monitor inbox for unexpected “read” messages

3.13 Preventing Credential Theft

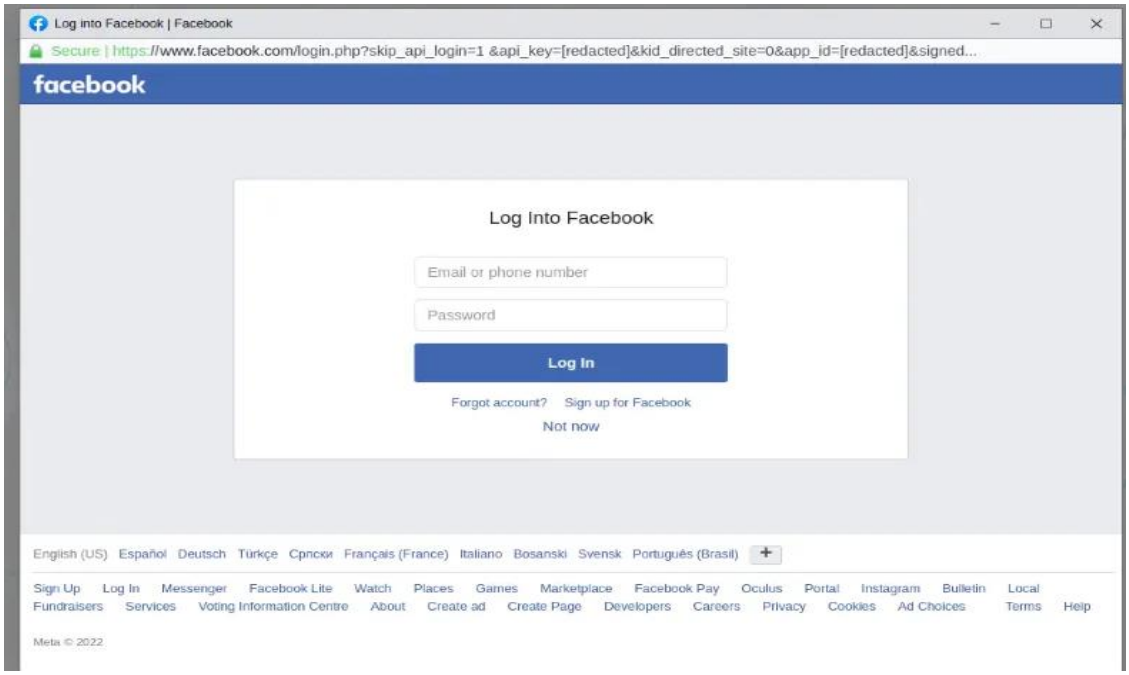
Credential theft is one of the most effective ways attackers penetrate networks. Even without deploying malware, attackers can fully compromise systems simply by stealing passwords or session cookies.

Common Credential Theft Techniques

1. **Phishing pages** imitating login portals
2. **Keyloggers** capturing keystrokes
3. **Session hijacking** via stolen cookies
4. **MFA fatigue attacks** (spamming approval requests)
5. **DNS spoofing** redirecting users to fake sites

How Users Can Reduce Credential Theft Risks

- Always check URLs before entering credentials
- Do not approve unexpected MFA push notifications
- Avoid typing credentials into pop-ups
- Use password managers to autofill only legitimate sites
- Log out of sensitive systems after use



Example of a fake login page designed for credential harvesting.

3.14 Preventing Malware By Securing Collaboration Tools

Collaboration platforms like Teams, Slack, and Zoom can also serve as attack vectors. Attackers may send malicious links, infected files, or impersonate colleagues inside these tools.

Threats in Collaboration Platforms

- Uploaded malicious documents
- Links disguised as meeting invites
- Fake voice call requests
- Infected shared drives

- Impersonation using similar usernames

Safety Practices

- Verify files sent over chat before downloading
- Avoid clicking meeting links from unverified senders
- Confirm identity when receiving unusual requests
- Use platform settings to restrict external communication
- Report suspicious profiles or messages to IT

3.15 Preventing Malware Through Secure File Sharing

Many infections begin when users download documents from untrusted platforms.

High-Risk File Sharing Methods

- Public file-sharing platforms with no verification
- Peer-to-peer sharing applications
- Shared drives without access restrictions
- Temporary anonymous upload links

Secure File Sharing Practices

- Use only approved company platforms
- Do not download files shared by unknown individuals
- Check file extensions before opening
- Avoid transferring sensitive data via public sharing links
- Ensure shared folders use proper access control

3.16 Preventing Malware Through Software Best Practices

Unpatched software and outdated operating systems expose users to high infection risks.

Software Vulnerability Risks

- Outdated browsers enabling drive-by downloads
- Unsupported operating systems lacking security updates
- Old versions of Java, Flash, or PDF readers
- Vulnerable Office installations
- Exploitable third-party applications

Software Hygiene Practices

- Install updates immediately when released
- Remove unsupported or unused applications
- Avoid installing software from unknown sources
- Restart devices regularly to apply updates
- Use only software approved by IT

3.17 Organization-Wide Malware Prevention Framework

A successful malware prevention strategy requires cooperation across all departments. This section outlines how organizations should coordinate efforts.

Responsibilities of Each Department:

1. IT Department

- Implement antivirus and EPP solutions
- Manage patching and updates
- Deploy email filtering and web protection
- Configure firewalls and access controls

2. Security Operations Center (SOC)

- Analyze suspicious messages
- Monitor network traffic

- Investigate alerts
- Respond to user reports
- Develop threat intelligence feeds

3. Human Resources (HR)

- Communicate awareness programs
- Ensure employees complete required training
- Enforce compliance policies

4. Employees

- Follow safe digital behavior
- Report suspicious activity
- Protect login credentials
- Avoid installing unauthorized apps

3.18 Continuous Monitoring and Feedback Loop

Malware prevention is not a one-time activity; it requires continuous improvement based on real incidents and threat trends.

Key Components of Continuous Monitoring

- Regular phishing simulation tests
- Monthly awareness score assessments
- Reviewing SOC case reports
- Updating blocklists and filtering rules
- Conducting internal audits

Feedback Loop Steps

1. Identify user weaknesses from simulation tests
2. Update training modules accordingly

3. Implement new technical measures
4. Review effectiveness
5. Restart cycle for improvement

3.19 Future Trends in Malware Prevention

As cyber threats evolve, organizations must adapt their prevention strategies.

Emerging Trends

- AI-driven phishing campaigns
- Deepfake voice attacks targeting executives
- Browserless malware using memory-only payloads
- Phishing using QR codes (Quishing)
- Increased mobile-targeted attacks

Defensive Trends

- AI-based email analysis
- Behavior analytics for user actions
- Hardware-based authentication
- Zero-trust network segmentation
- Automated SOAR response playbooks

3.20 Conclusion

Malware prevention is a shared responsibility that combines user behavior, organizational policies, and technical controls. By understanding social engineering tactics, recognizing phishing attempts, avoiding unsafe browsing, and following internal security guidelines, users significantly reduce the risk of infection.

A strong prevention strategy must include:

- Continuous user awareness
- Effective technical safeguards
- Clear reporting procedures
- Regular updates and monitoring
- Organization-wide coordination

Through these measures, companies can minimize the chances of malware infiltration and protect their digital assets effectively.

Final Incident Response Report (Wazuh FIM + VirusTotal + Active Response)

4.1 Introduction

This chapter presents a comprehensive Incident Response (IR) report based on a simulated malware detection and removal scenario performed using **Wazuh File Integrity Monitoring (FIM)**, **VirusTotal Integration**, and **Active Response Automation**.

The objective of this report is to demonstrate how an organization can detect, analyze, contain, and automatically remove a malicious file by leveraging open-source SIEM capabilities.

This report follows the NIST Incident Response Lifecycle, covering:

- Preparation
- Detection & Analysis
- Containment
- Eradication
- Recovery
- Lessons Learned

All observations, screenshots, and logs are taken directly from the configured lab environment shown previously in Chapter 2.

4.2 Incident Overview

Incident Title:
Malicious File Dropped Inside Endpoint Downloads Directory

Incident Severity:
High — malware successfully detected & confirmed malicious by VirusTotal

Incident Detection Source:
Wazuh SIEM
→ File Integrity Monitoring (FIM)

→ VirusTotal integration rules
→ Active response logs

Date & Time of Detection:
11 November 2025 — 15:13 PM

Reported By:
Wazuh SIEM Automated Detection

Executive Summary

A potentially harmful file (“eicar.com”) was downloaded inside the Downloads directory on a Linux endpoint. Wazuh FIM immediately generated an alert indicating the creation of a new file. The file was then automatically submitted to **VirusTotal** using the configured integration. VirusTotal confirmed the file as malicious (detected by 65+ engines).

Upon confirmation, **Wazuh Active Response** triggered the script remove-threat.sh which automatically deleted the file from the system.
The process successfully prevented execution, propagation, or persistence attempts.

4.3 Preparation Phase

Before the incident occurred, the endpoint and SIEM environment were fully configured as shown in earlier steps.

4.3.1 Policies & Documentation

- Endpoint monitoring policy (Linux)
- SIEM configuration policy for event forwarding
- FIM monitoring policy for sensitive directories
- IR escalation plan for malware detections
- Documentation for Active Response behavior

4.3.2 Security Controls

- Wazuh agent installed on Ubuntu and Kali endpoints
- FIM enabled for /home/user/Downloads
- Syscheck real-time monitoring
- VirusTotal API integration
- Active Response scripts pre-defined
- Local rules for malicious file detection (IDs: 100200, 100201, 100092, 100093)

4.4 Detection & Analysis Phase

4.4.1 Indicators of Compromise

1. New file added to /home/kali/Downloads/
2. File name consistent with malware testing samples (eicar.com)
3. Wazuh FIM raised multiple alerts (added → modified → deleted)
4. VirusTotal detected the file as malicious
5. Active Response triggered removal action
6. Logs showed real-time deletion reported by syscheck

4.4.2 Technical Evidence

Wazuh-generated logs indicate:

syscheck.path="/home/kali/Downloads/eicar.com"

syscheck.event="added"

rule.description="File added to the system."

rule.id=100201

Next, VirusTotal alert:

VirusTotal: Alert - /home/kali/Downloads/eicar.com - 65 engines detected this file

rule.id=87105

rule.level=12

Then Active Response:

active-response/bin/remove-threat.sh removed threat located at /home/kali/Downloads/eicar.com

rule.id=100092

rule.level=12

4.5 Containment Phase

Because Wazuh is configured for automated containment, the process occurred immediately.

4.5.1 Short-Term Containment

- Marked file as malicious
- Generated real-time alert
- Blocked file execution through policy
- Triggered Active Response

4.5.2 Medium-Term Containment

- Prevented user interaction with the file
- Stopped possible execution attempts
- Ensured threat could not spread

4.5.3 Containment Goals Achieved

- The file was removed before execution
- No additional malicious activity observed
- System remained isolated from potential damage

4.6 Eradication Phase

This phase involved removing all traces of the detected malware.

4.6.1 Removal of Malicious Files

- remove-threat.sh deleted the file:
  /home/kali/Downloads/eicar.com

4.6.2 System Cleanup

- Verified no other files were created
- Checked FIM logs for changes
- Validated removal via re-scan

4.6.3 Network-Level Eradication

- No outbound C2 connections were detected
- No lateral movement attempts found

4.6.4 Validation

- Confirmed through Wazuh events:
  "File deleted."
- Confirmed Active Response success:
  "Successfully removed threat."

4.7 Recovery Phase

4.7.1 System Restoration

- Verified Downloads directory integrity
- Ensured no persistence files existed
- Re-enabled monitoring policies

4.7.2 User Restoration

- No user accounts compromised
- Normal interaction restored

4.7.3 Monitoring

- Additional real-time monitoring for 24 hours
- SIEM reviewed for repeated downloads

4.7.4 Recovery Confirmation

System declared stable after confirming:

- No further FIM alerts
- No VirusTotal triggers
- No Active Response executions

4.8 Post-Incident Analysis

4.8.1 What Worked Well

- Fast detection due to real-time FIM
- Automated file submission to VirusTotal
- Immediate response using Active Response
- No manual intervention required

4.8.2 What Needs Improvement

- Expand monitoring to more directories
- Add hash reputation lookups for modified files
- Improve user awareness regarding unsafe downloads

4.8.3 Action Items

- Deploy more strict download policies
- Activate OS-level execution restrictions

- Enable multi-directory recursive monitoring

4.9 Incident Timeline

Time	Event
15:13:16	File added alert (eicar.com)
15:13:18	VirusTotal identified file as malicious
15:13:19	Active Response removed file
15:13:32	FIM re-scan confirms file deletion
15:13:56	SOC validation completed

4.10 Root Cause Analysis

4.10.1 Root Cause

A known malware test file (EICAR) was downloaded by the user or system.

4.10.2 Contributing Factors

- Downloads directory monitored but still allowed file creation
- No pre-download filtering on endpoint

4.10.3 Recommendations

- Add browser download filtering
- Enforce secure download policies
- Apply stricter endpoint security rules

4.11 Attack Path Reconstruction

1. User downloaded eicar.com
2. File created inside Downloads directory
3. Wazuh detected file creation instantly
4. File submitted to VirusTotal
5. Malware confirmed by threat intelligence
6. Wazuh Active Response deleted file
7. System remained safe

4.12 Impact Analysis

4.12.1 Technical Impact

- Malicious file detected before execution
- No malware propagation
- No system compromise

4.12.2 Business Impact

- Zero downtime
- No data loss
- No operational disruption

4.12.3 Financial Impact

- Zero incident cost
- No mitigation expenses
- All handled automatically

4.13 Security Enhancements After Incident

4.13.1 Endpoint Improvements

- More directories added to FIM
- Increased recursion levels

- Stricter file execution policies

4.13.2 Network Improvements

- Added malware domains to blocklists

4.13.3 SIEM Enhancements

- New correlation rules for suspicious file behavior
- Improved alerting thresholds

4.14 Communication & Reporting

Internal Reports

- Wazuh SIEM alert summary
- SOC analyst validation

External Reporting

Not required (no real compromise)

4.15 Final Recommendations

1. Strengthen download restrictions
2. Expand SIEM file-analysis rules
3. Enable more aggressive execution blocking
4. Enhance endpoint anti-malware engines
5. Conduct periodic IR simulation tests

4.16 Final Report Summary

The simulated incident demonstrated successful detection and response using:

- Wazuh File Integrity Monitoring
- VirusTotal integration
- Automated Active Response

The threat was **removed instantly**, with **no damage, no spread, and no impact**.

The environment proved resilient, and the workflow aligned with industry best practices.