Wazuh – Software Installation and Uninstallation Monitoring

Lab Created By: MUHAMMAD MOIZ UD DIN RAFAY

Follow Me: linkedin.com/in/moizuddinrafay

Windows Software Installation and Uninstallation Detection in Wazuh: A Security Perspective

In a security context, monitoring software installations and uninstallation activities on Windows systems is crucial for maintaining the integrity and security of an IT environment. Unauthorized or unapproved software installations and removals can pose significant security risks, including the introduction of malware, unauthorized tools, or the removal of critical security software.

Wazuh, as an open-source security monitoring platform, plays a vital role in detecting such events on Windows systems by monitoring Windows Event Logs and generating alerts based on predefined rules. Below is a detailed overview of how Wazuh can be used to monitor software installation and uninstallation activities from a security perspective.

Importance of Monitoring Software Installation and Uninstallation

Software installation and uninstallation events are key indicators of system changes that can impact security. These activities may involve:

Malicious software installation: Attackers often install malicious programs, remote access tools, or backdoors on compromised systems to maintain persistence or exfiltrate data.

Unauthorized applications: Employees or users installing unauthorized or unapproved software might introduce security vulnerabilities into the system.

Removal of security software: Uninstallation of antivirus programs, firewalls, or other security tools can significantly weaken the system's defense against cyber threats.

Software updates and patch management: Ensuring that software is properly installed, updated, and patched can help mitigate vulnerabilities associated with outdated or unpatched software versions.

Detecting Software Installations and Uninstallations Using Windows Event Logs

Windows logs software installations and uninstallations in Application Event Logs and Windows Installer Logs. Some of the key event IDs that help in detecting these activities are:

Event ID 11707 (MsiInstaller): This event is logged during a successful software installation using Windows Installer. It indicates that a software installation operation has completed successfully.

Event ID 11724 (MsiInstaller): This event is logged when a software uninstallation operation is successfully completed. It marks the successful removal of a software package.

By monitoring these events, Wazuh can generate alerts whenever a software installation or uninstallation event occurs.

Generating Alerts in Wazuh

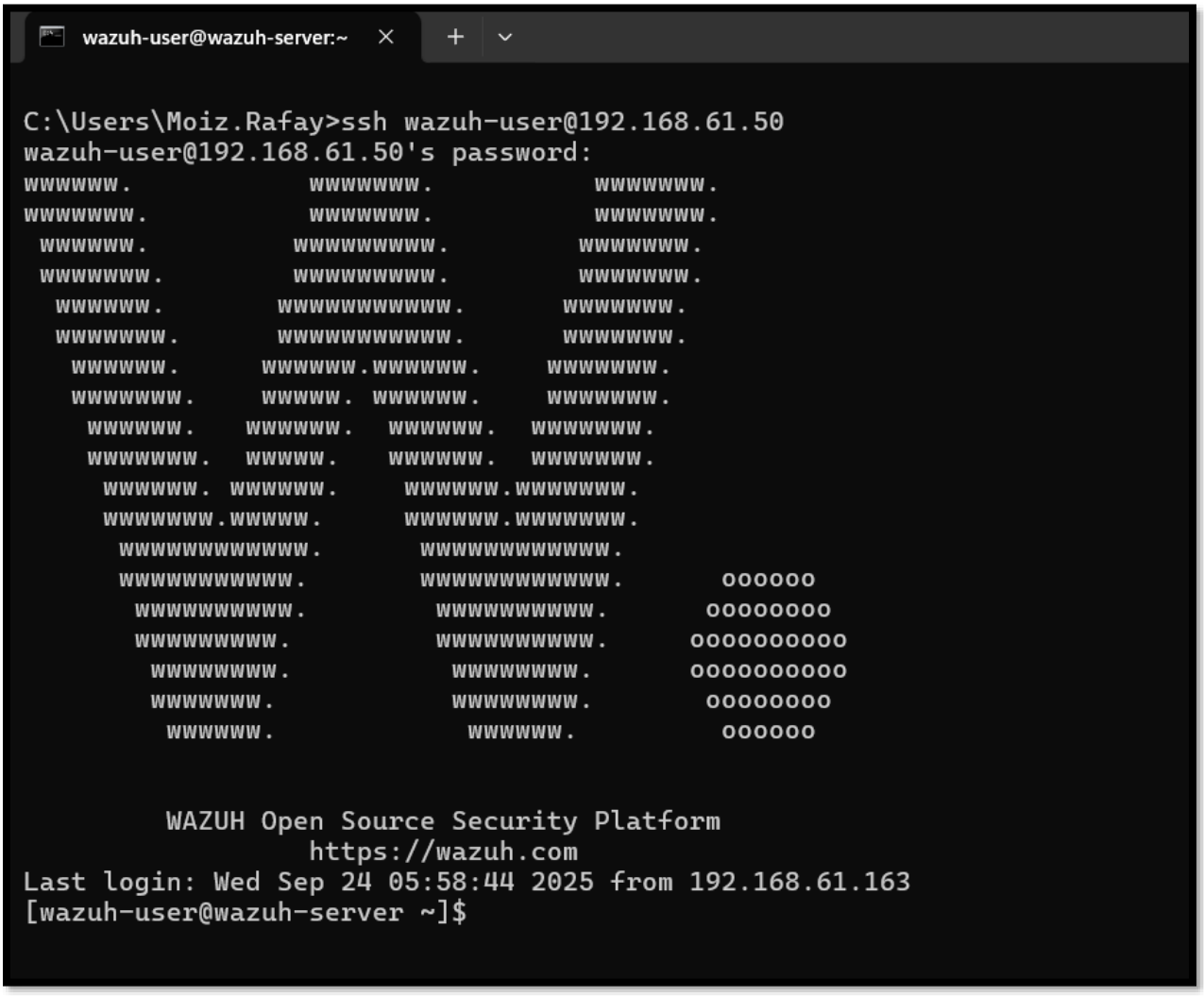
Once the configuration is complete, Wazuh will begin generating alerts whenever the specified event IDs are triggered. The following types of alerts can be generated:

Software Installation Alerts: Alerts will be generated when Event ID 11707 is detected, indicating that a software package has been installed successfully on the system.

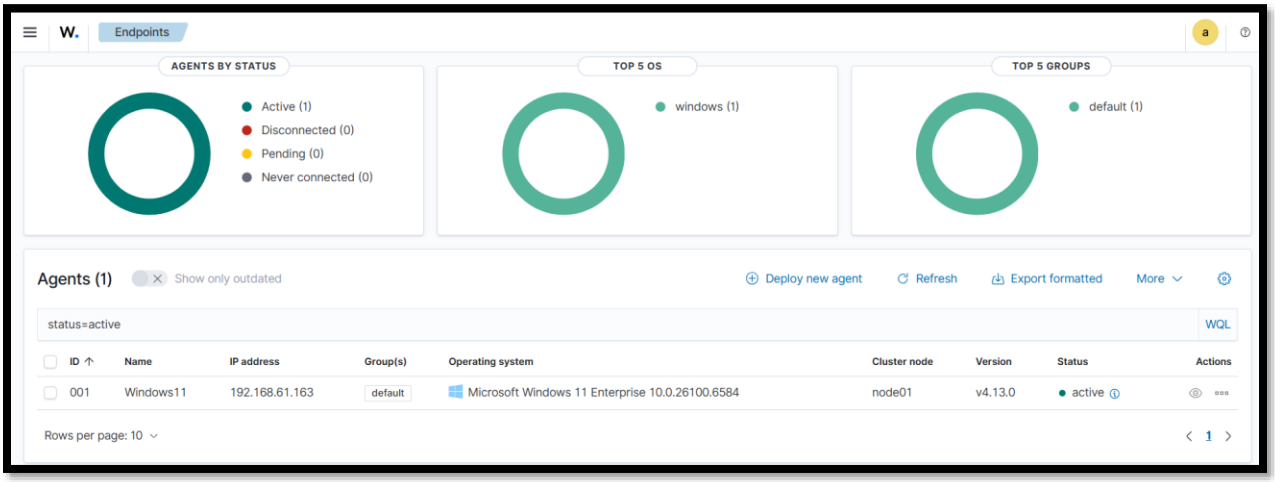
Software Uninstallation Alerts: Alerts will be generated when Event ID 11724 is detected, indicating that a software package has been uninstalled from the system.

These alerts can be viewed in the Wazuh dashboard or integrated with other SIEM solutions for further analysis and response.

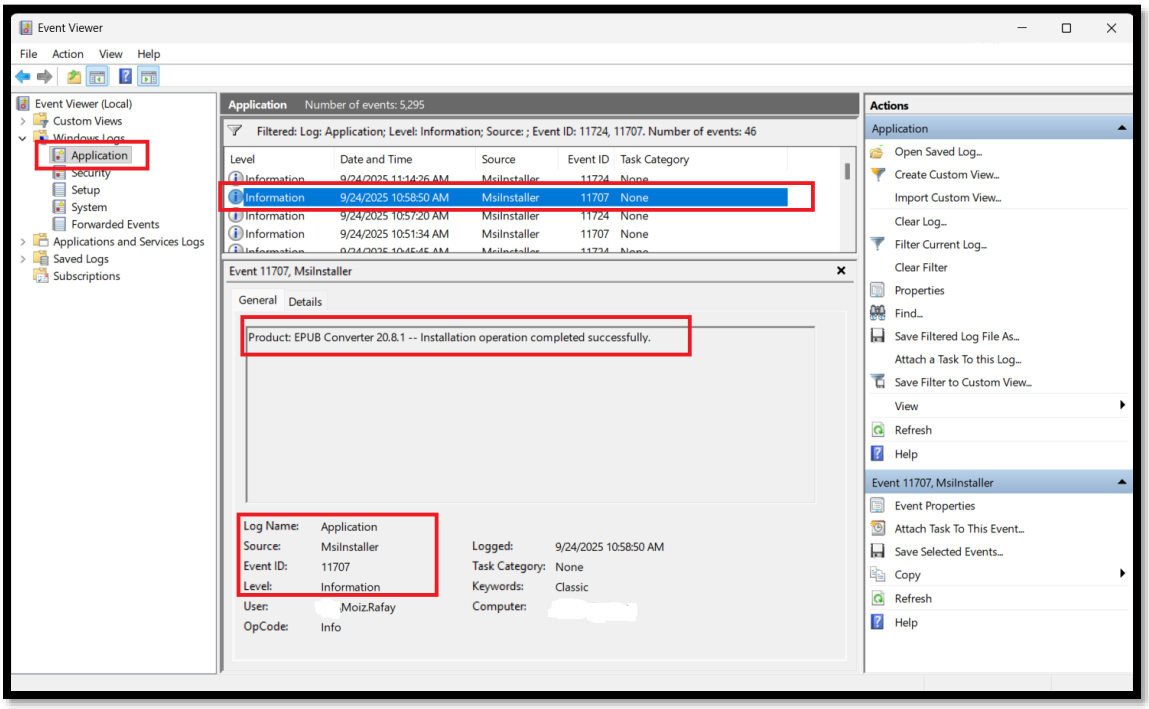
Here is my Wazuh test VM is running on VirtualBox.



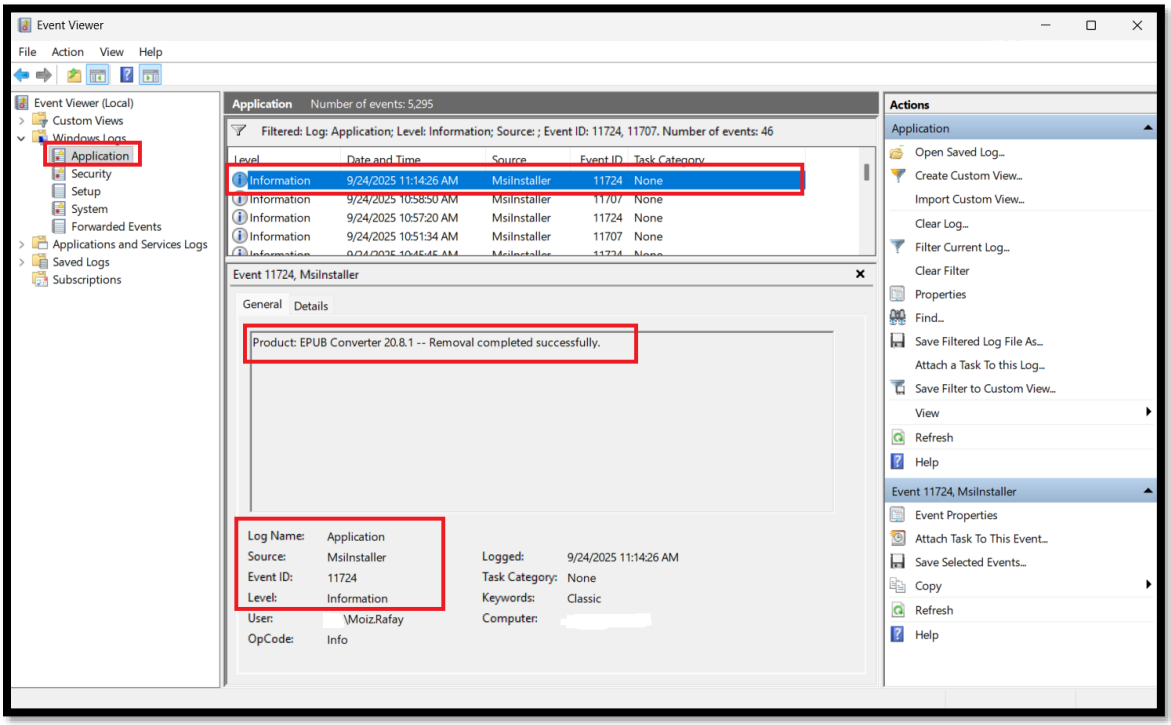
In the Wazuh dashboard, I have configured Windows11 agent.



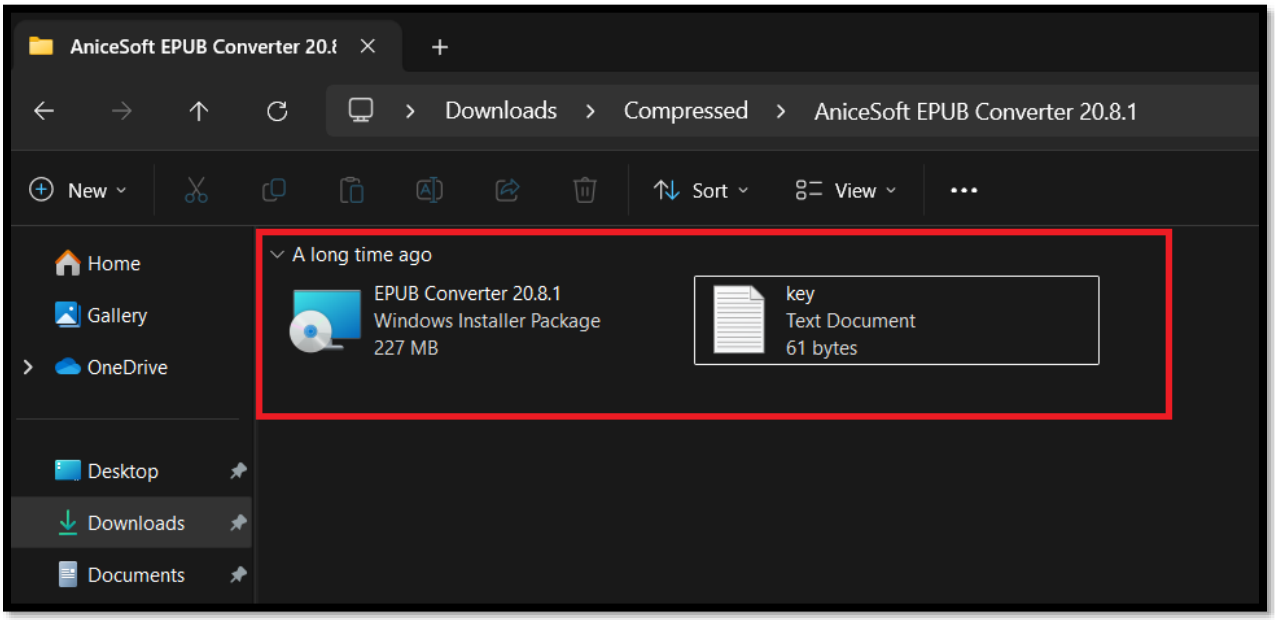
In the Windows go to “Event Viewer” then “Application Logs” and search for the Event ID’s “11707 and 11724”.



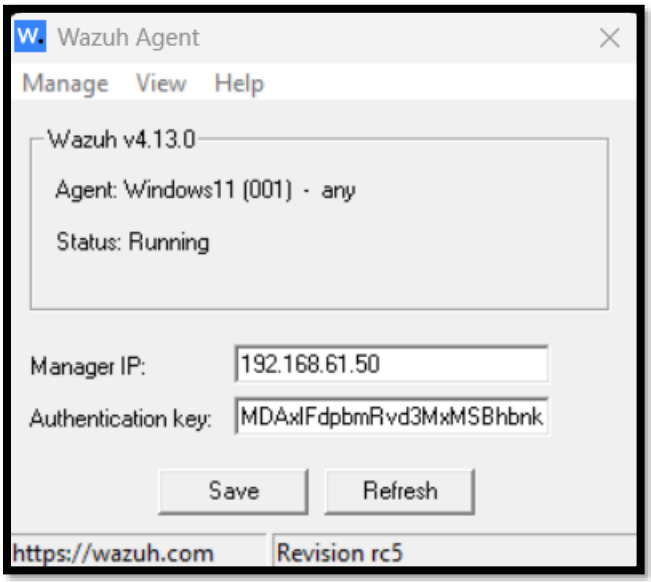
The highlighted fields show the relevant in information “Level, Date Time, Source, EventID”, about the software installation and uninstallation.



Here is the software application “EPUB Converter” we use to generate installation and uninstallation events.



Now go to Wazuh Agent and edit “ossec.conf” file.



Here is the “ossec.conf” file. Search for the “localfile”.

ossec.conf

File Edit View

```
<localfile>
  <location>Security</location>
  <log_format>eventchannel</log_format>
  <query>Event/System[EventID != 5145 and EventID != 5156 and EventID != 5447 and
    EventID != 4656 and EventID != 4658 and EventID != 4663 and EventID != 4660 and
    EventID != 4670 and EventID != 4690 and EventID != 4703 and EventID != 4907 and
    EventID != 5152 and EventID != 5157]</query>
</localfile>

<localfile>
  <location>System</location>
  <log_format>eventchannel</log_format>
</localfile>

<localfile>
  <location>active-response\active-responses.log</location>
  <log_format>syslog</log_format>
</localfile>

<!-- Policy monitoring -->
<rootcheck>
  <disabled>no</disabled>
  <windows_apps>./shared/win_applications_rcl.txt</windows_apps>
  <windows_malware>./shared/win_malware_rcl.txt</windows_malware>
</rootcheck>

<!-- Security Configuration Assessment -->
<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>
```

Ln 53, Col 15 9,783 characters Plain text

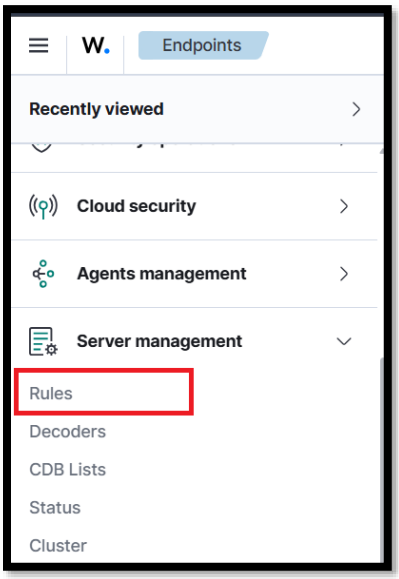
We have to add configuration files here.

```
<localfile>
  <location>Application</location>
  <log_format>eventlog</log_format>
</localfile>

<localfile>
  <location>Application</location>
  <log_format>eventlog</log_format>
  <query>Event/Application[EventID=11707 or EventID=11724]</query>
</localfile>
```

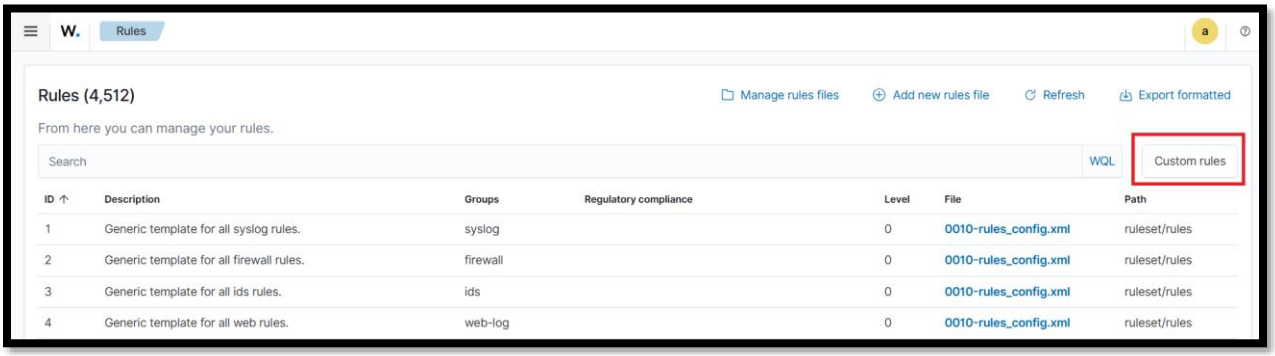
```
ossec.conf
File  Edit  View
  <localfile>
    <location>Security</location>
    <log_format>eventchannel</log_format>
    <query>Event/System[EventID != 5145 and EventID != 5156 and EventID != 5447 and
      EventID != 4656 and EventID != 4658 and EventID != 4663 and EventID != 4660 and
      EventID != 4670 and EventID != 4690 and EventID != 4703 and EventID != 4907 and
      EventID != 5152 and EventID != 5157]</query>
  </localfile>

  <localfile>
    <location>System</location>
    <log_format>eventchannel</log_format>
  </localfile>

<!-- Application Installation and Uninstallation Monitoring -->
  <localfile>
    <location>Application</location>
    <log_format>eventlog</log_format>
  </localfile>

  <localfile>
    <location>Application</location>
    <log_format>eventlog</log_format>
    <query>Event/Application[EventID=11707 or EventID=11724]</query>
  </localfile>

  <localfile>
    <location>active-response\active-responses.log</location>
    <log_format>syslog</log_format>
  </localfile>
```

After add this configuration in “ossec.conf” file we have to save this and close.

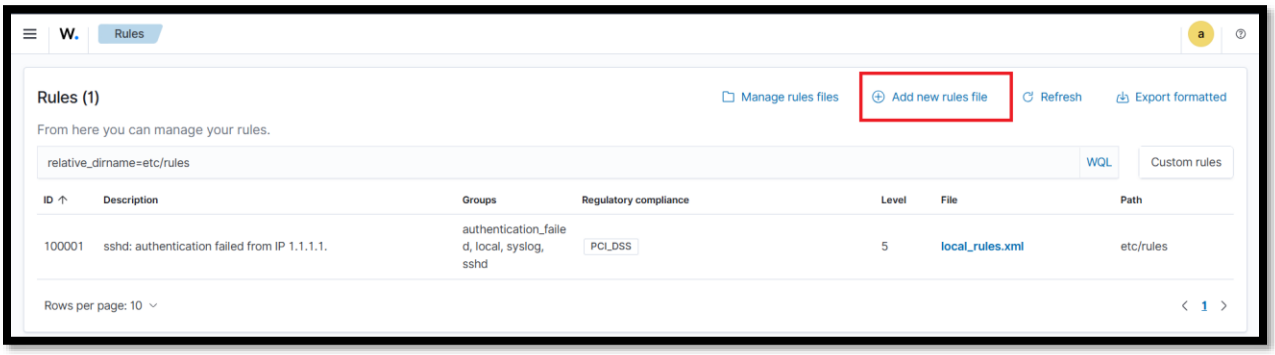
Next, we have to configure rules.



Click on “Custom Rules”.



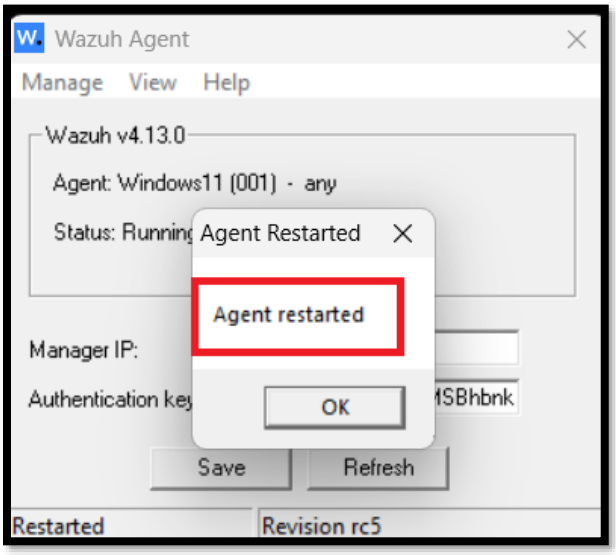
Add new rules file



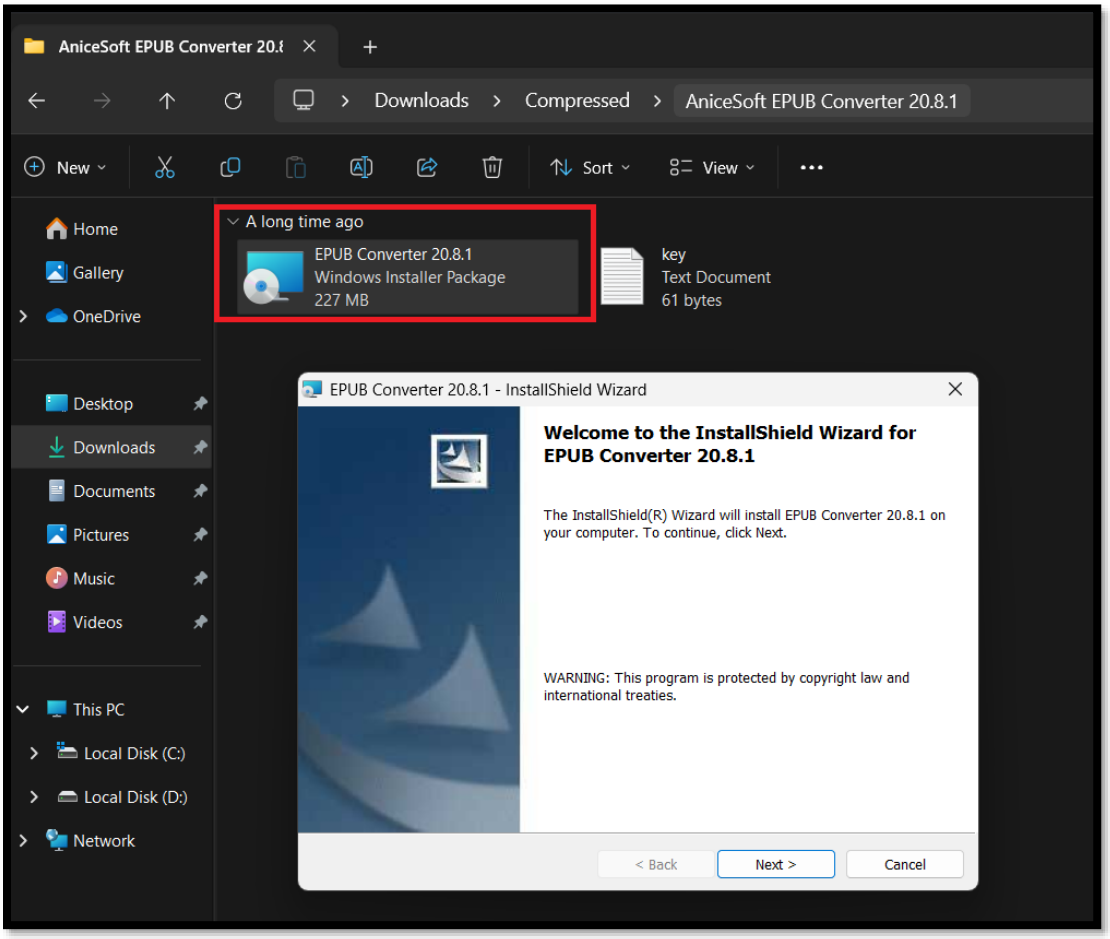
Adding custom rules “win-application_rules.xml” file name.

```
<rule id="60611" level="3">
  <if_sid>60609</if_sid>
  <field name="win.system.eventID">^11724$|^1034$</field>
  <options>alert_by_email</options>
  <description>Application Uninstalled $(win.eventdata.data)</description>
  <options>no_full_log</options>
 </rule>

 <rule id="60612" level="3">
  <if_sid>60609</if_sid>
  <field name="win.system.eventID">^11707$|^1033$</field>
  <options>alert_by_email</options>
  <description>Application Installed $(win.eventdata.data)</description>
  <options>no_full_log</options>
 </rule>
```

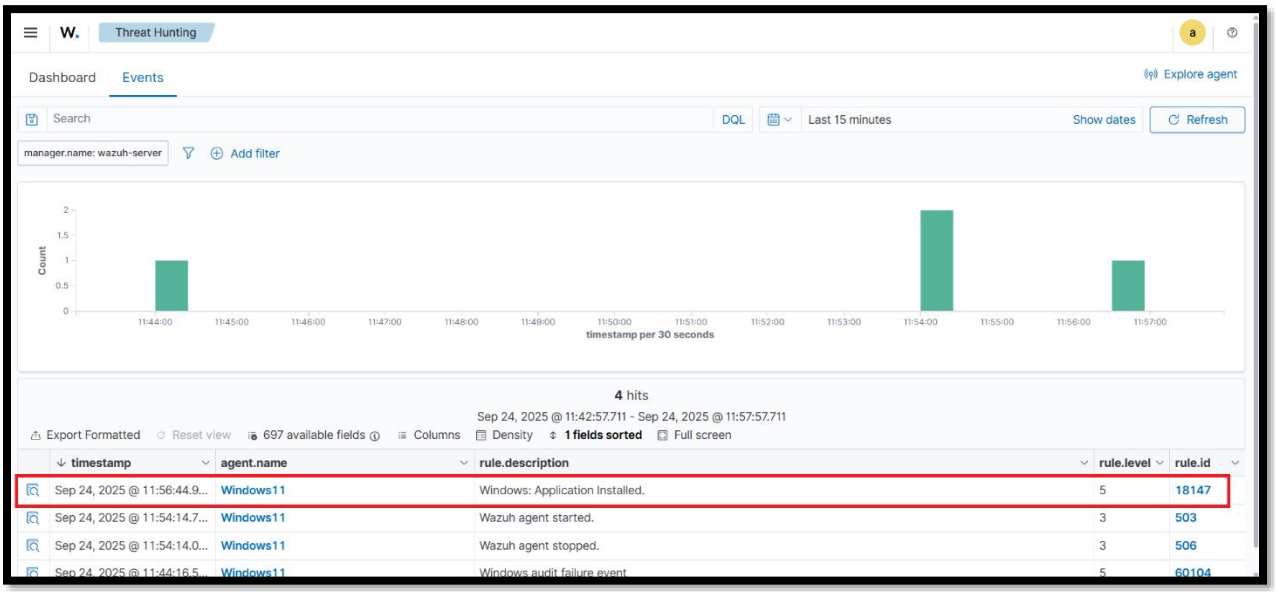
Save these rules and restart Wazuh-manager and then restart Wazuh-agent.



Now install the application software.



When the installation completed we get alerts in Wazuh “Threat Hunting”.

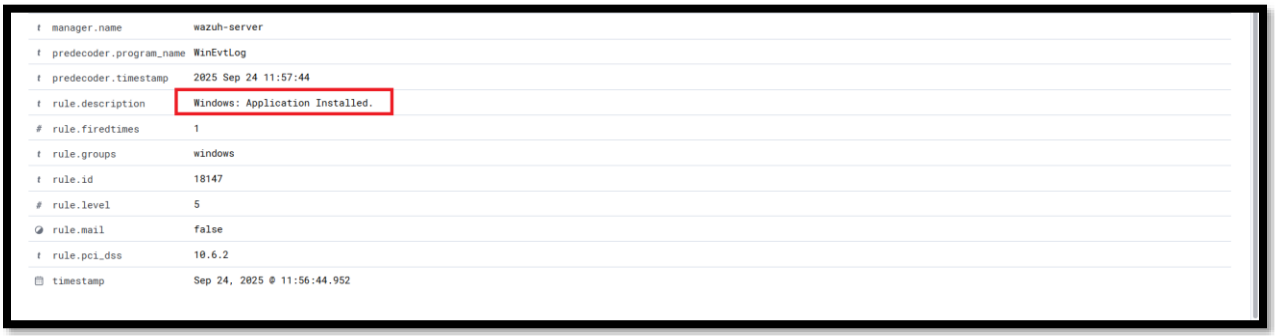


Now check the full logs “Windows: Application Installed.”

Discover | wazuh-alerts-4.x-2025.09.24#0k6CepkBdOQ7GoX3uziB

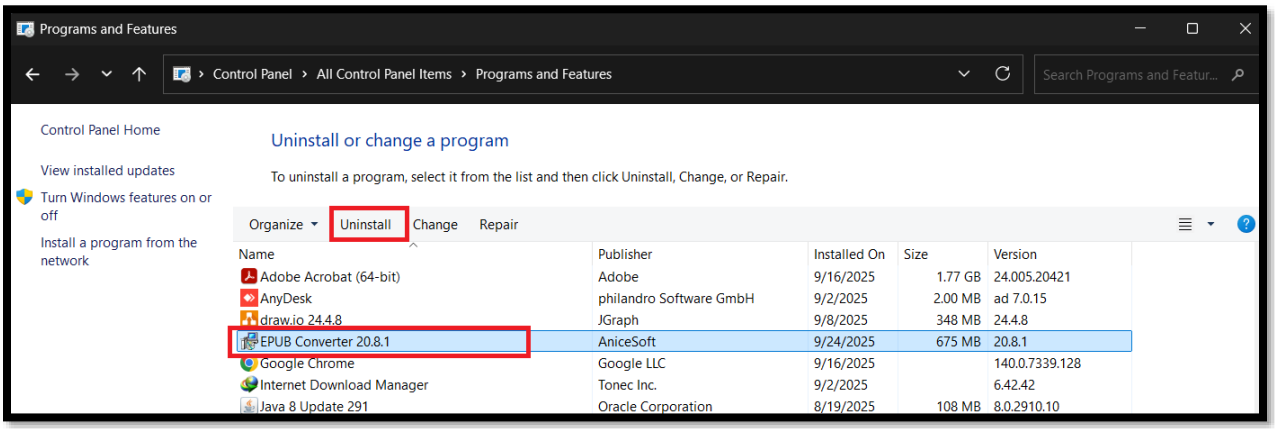
Table JSON

@timestamp	Sep 24, 2025 @ 11:56:44.952
_index	wazuh-alerts-4.x-2025.09.24
agent.id	001
agent.ip	192.168.61.163
agent.name	Windows11
data.dstuser	Moiz.Rafay
data.extra_data	MsiInstaller
data.id	11707
data.status	INFORMATION
data.system_name	
data.type	Application
decoder.name	windows
decoder.parent	windows
full_log	2025 Sep 24 11:57:44 WinEvtLog: Application: INFORMATION(11707): MsiInstaller: Moiz.Rafay: : Product: EPUB Converter 20.8.1 -- Installation operation completed successfully. (NULL) (NULL) (NULL) (NULL) (NULL)
id	1758697004.2406251
input.type	log
location	WinEvtLog
manager.name	wazuh-server



manager.name	wazuh-server
predecoder.program_name	WinEvtLog
predecoder.timestamp	2025 Sep 24 11:57:44
rule.description	Windows: Application Installed.
rule.firedtimes	1
rule.groups	windows
rule.id	18147
rule.level	5
rule.mail	false
rule.pci_dss	10.6.2
timestamp	Sep 24, 2025 @ 11:56:44.952

Next, we have to Uninstall the application.



Programs and Features

Control Panel › All Control Panel Items › Programs and Features

Control Panel Home

View installed updates

Turn Windows features on or off

Install a program from the network

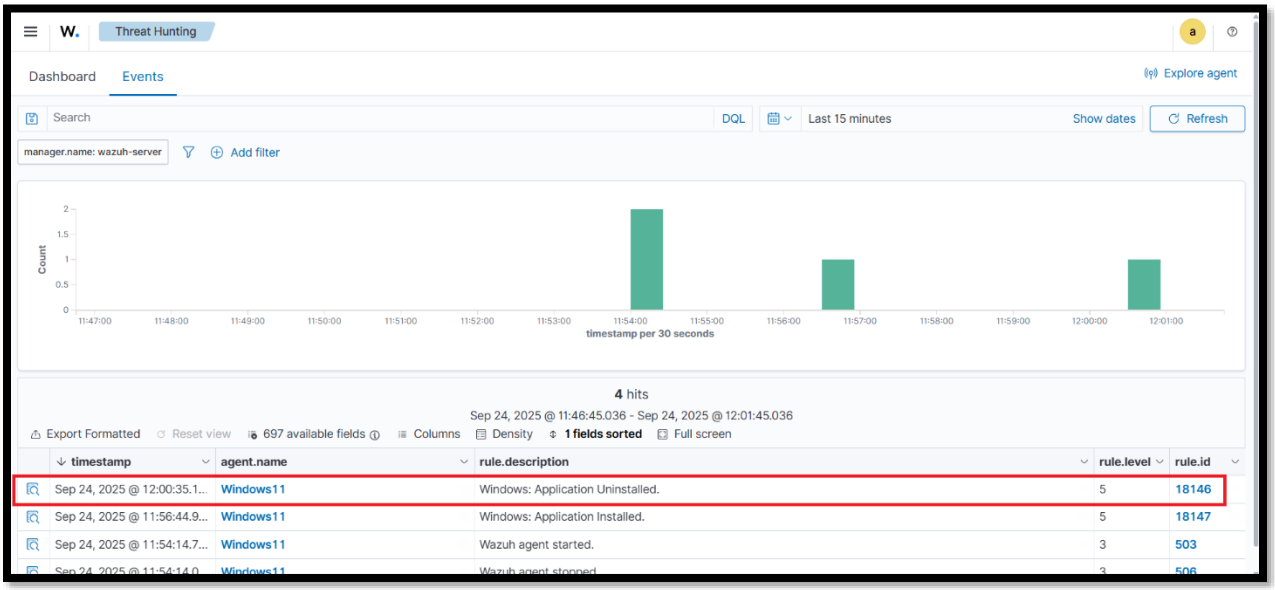
Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

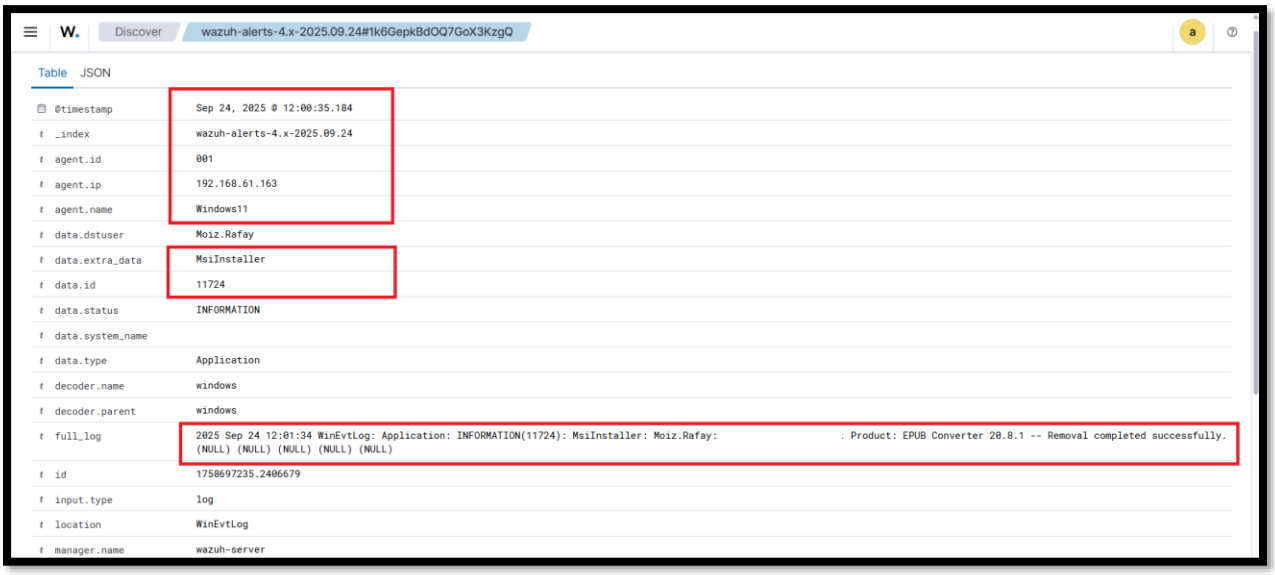
Organize | Uninstall | Change | Repair

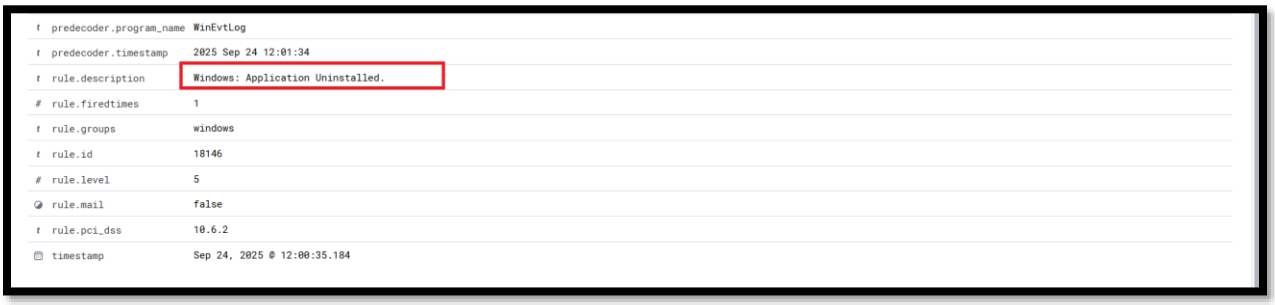
Name	Publisher	Installed On	Size	Version
Adobe Acrobat (64-bit)	Adobe	9/16/2025	1.77 GB	24.005.20421
AnyDesk	philandro Software GmbH	9/2/2025	2.00 MB	ad 7.0.15
draw.io 24.4.8	JGraph	9/8/2025	348 MB	24.4.8
EPUB Converter 20.8.1	AniceSoft	9/24/2025	675 MB	20.8.1
Google Chrome	Google LLC	9/16/2025		140.0.7339.128
Internet Download Manager	Tonec Inc.	9/2/2025		6.42.42
Java 8 Update 291	Oracle Corporation	8/19/2025	108 MB	8.0.2910.10

When the uninstallation completed we get alerts in Wazuh “Threat Hunting”.



Now check the full logs “Windows: Application Uninstalled.”





Security Considerations and Best Practices

While Wazuh can effectively monitor software installations and uninstallations, it's important to follow these security best practices to enhance the detection process:

User Privileges: Limit user privileges on Windows systems to prevent unauthorized users from installing or uninstalling software without proper approval.

Use Group Policies: Enforce Group Policies to restrict the installation of unauthorized applications. Group Policies can be configured to only allow software installations from trusted sources.

System Hardening: Regularly apply software patches and updates to mitigate the risk of exploitation from outdated or vulnerable software.

Software Whitelisting: Implement application whitelisting to only allow approved software to be installed, reducing the likelihood of unauthorized or malicious software installations.

Conclusion

Incorporating software installation and uninstallation monitoring into your security operations is an essential step in detecting malicious activities and ensuring that only approved software is installed on Windows systems. Wazuh, with its ability to monitor Windows Event Logs, plays a key role in identifying unauthorized or suspicious software activities. By configuring Wazuh to track specific event IDs such as 11707 (installation) and 11724 (uninstallation), organizations can gain valuable insight into their IT environment and quickly respond to potential security threats. Additionally, integrating Sysmon provides more in-depth monitoring and strengthens the overall security posture.