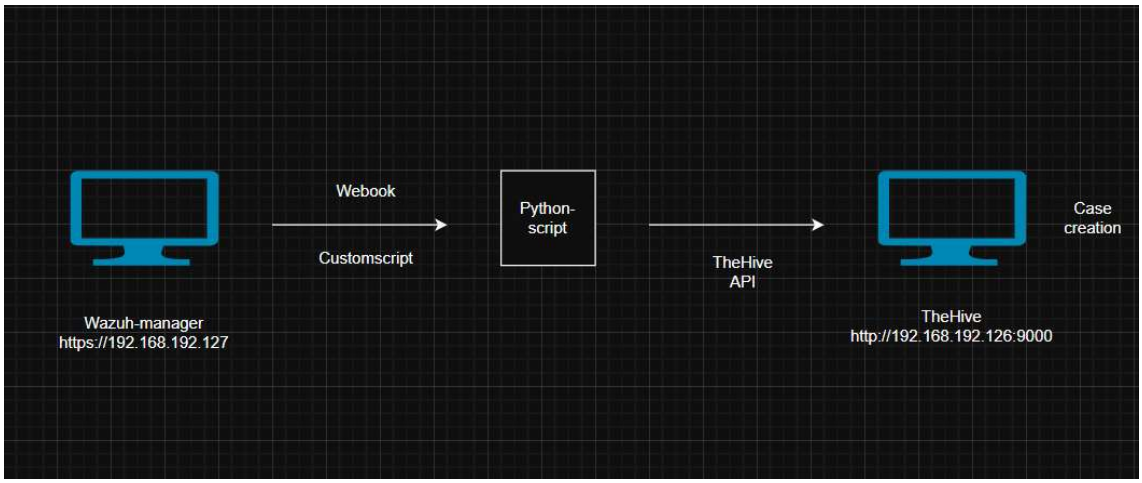
– Description :

- This document demonstrates the integration of **Wazuh Manager** with **TheHive SOAR** platform for automated incident creation.
- The Wazuh Manager is responsible for event collection, rule-based alert generation, and integration execution, while TheHive serves as the incident response platform that receives alerts as cases.
- Before configuring the integration, it is essential to verify the versions of the operating system, Wazuh, TheHive, and Python to ensure compatibility.
- The purpose is to ensure that when a security alert is detected by Wazuh (e.g., an ICMP flood or SSH brute force), the alert is automatically sent to TheHive via an integration script, where a corresponding **case (ticket)** is created for analyst investigation.

– Network Architecture :



- **Wazuh Manager** detects security alerts and triggers a custom Python script to forward incident details.

- The script communicates with **TheHive API** to automatically create and populate new cases for analysis.

- **Analysts** review, investigate, and enrich the cases using TheHive and integrated Cortex analyzers.

– Environment Setup :

- This section describes the environment used for deploying and testing the Wazuh–TheHive integration.Both tools were installed on same machine.
- The Wazuh Manager is responsible for event collection, rule-based alert generation, and integration execution, while TheHive serves as the incident response platform that receives alerts as cases.
- Before configuring the integration, it is essential to verify the versions of the operating system, Wazuh, TheHive, and Python to ensure compatibility.
- This also helps future maintenance and replication of the same environment.

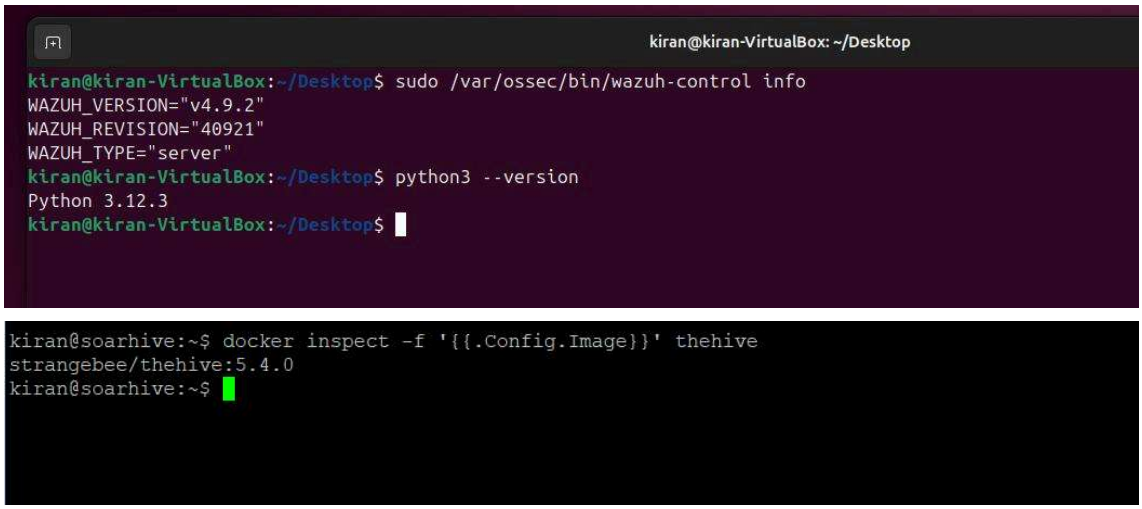


Figure 1 — Environment verification.

- *This screenshot confirms the versions of Ubuntu, Wazuh Manager, TheHive, and Python used in the deployment.*
- *This ensures compatibility between components before configuring integration.*

– TheHive Installation & Verification :

- TheHive was installed and configured as the Security Operations and Incident Response platform.
- After installation, the service was verified to ensure it runs on the default port 9000.
- TheHive provides an intuitive web interface for analysts to manage security incidents as cases, link observables, and correlate events with MITRE ATT&CK techniques.
- Once logged into the web interface, the administrator can generate an **API key** (under Profile → API keys).
- This API key is crucial because it allows the Wazuh Manager integration script to authenticate securely with TheHive API for automated case creation.

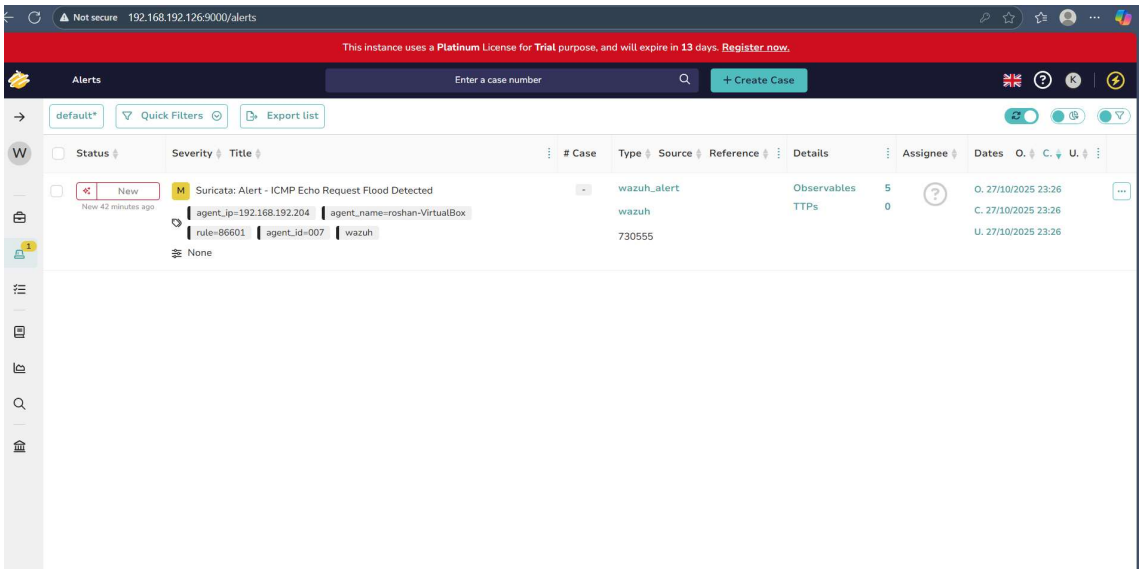


Figure 2 — TheHive successfully deployed.

- *The web interface indicates that TheHive is running correctly on port 9000 and ready to receive API requests from Wazuh.*
- *The administrator user can generate an **API key** under Profile → API keys, which will later be used in the Wazuh integration script. After the organization is created logon with that organization admin credentials.*

– Wazuh Integration Configuration :

- In this phase, Wazuh is configured to automatically forward alerts to TheHive using a **custom integration script**.
- The integration is placed under /var/ossec/integrations/ where Wazuh looks for executable scripts matching names defined in the configuration file.
- The integration consists of two parts :
    1) A **Bash wrapper** (custom-w2thive) that Wazuh executes directly.
    2) A **Python script** (custom-w2thive.py) that uses thehive4py to send alerts to TheHive's API endpoint.
- And at last, Proper permissions (chmod 755) and ownership (root:wazuh or root:ossec) are assigned to ensure Wazuh can execute the files without permission errors.

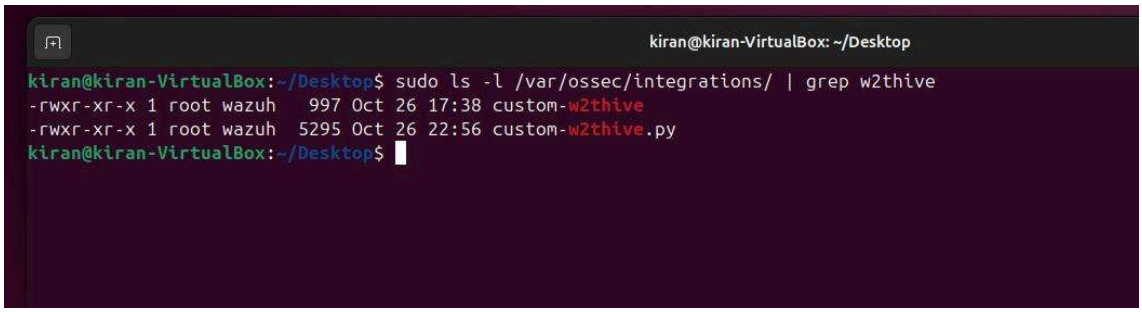


Figure 3 — Custom integration scripts in Wazuh.

- *The files custom-w2thive (Bash wrapper) and custom-w2thive.py (Python script) are present and executable under /var/ossec/integrations/. These handle the communication between Wazuh and TheHive using the thehive4py API library.*

Figure 4 — Wazuh Manager integration configuration.

- *The integration passes the alert JSON to the custom script for forwarding to TheHive.*

Figure 5 —thresold level of Wazuh

- *This snippet instructs Wazuh to trigger the custom-w2thive integration whenever an alert of level ≥ 3 is generated*

– Alert Generation and Forwarding :

- After the integration setup, the next step is to verify the flow of alerts from Wazuh to TheHive.
- When a detection rule is triggered — for example, during an ICMP flood or SSH brute-force test — Wazuh generates an alert in JSON format and passes it to the configured integration.
- The Bash wrapper captures this alert and calls the Python script, which then sends it to TheHive via REST API.
- The alert JSON contains all important details such as rule.id, rule.description, srcip, and agent.name, which are used to create a meaningful case title and description in TheHive

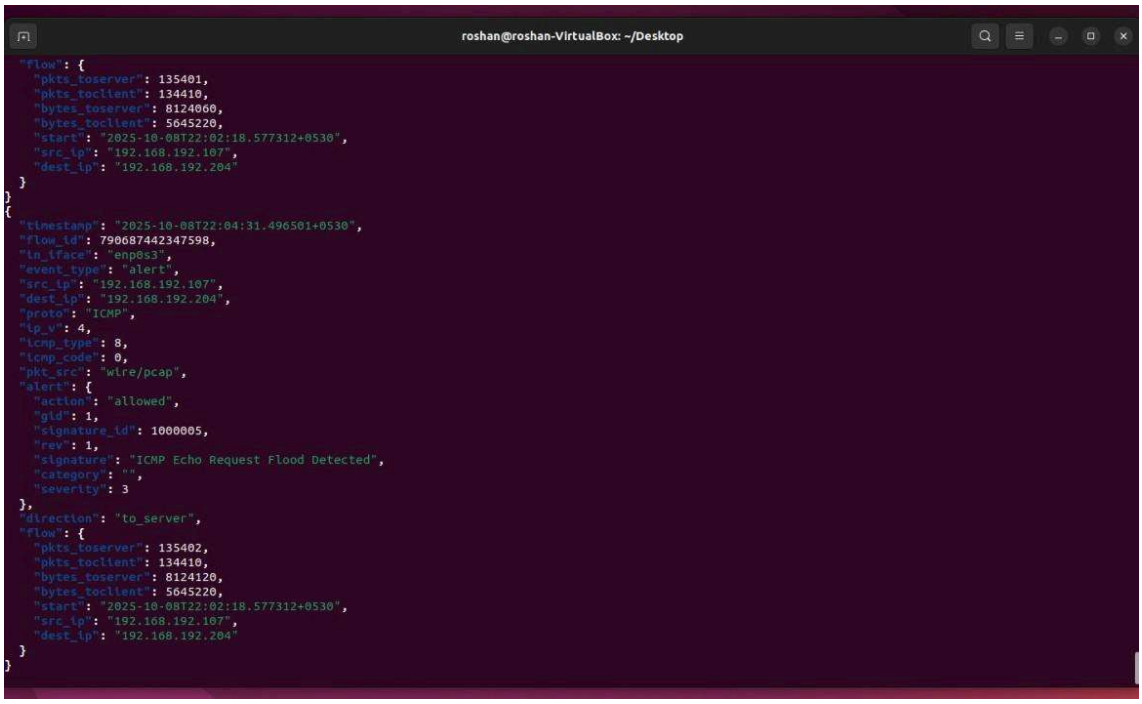


Figure 6 — Example Wazuh alert triggered by test attack.

- *This shows the ICMP flood detection alert generated by Wazuh. The alert JSON includes key fields such as rule.id, rule.description, agent.name, and srcip, which are forwarded to TheHive through the integration script.*

– TheHive Case Creation Result :

- When TheHive receives the alert through its API, it automatically creates a new case based on the data provided by the integration script.
- Each case in TheHive includes :
  > A **title** (derived from the Wazuh alert description),
  > A **description** (the complete alert JSON),
  > **Tags** for identification (e.g., wazuh, icmp_flood, ssh_bruteforce),
  > And optionally, **observables** extracted from the alert (such as source IP or file hash).
- This automated case creation streamlines the workflow — security analysts can directly open TheHive, review alert details, add comments, and take response actions without manually checking Wazuh logs.

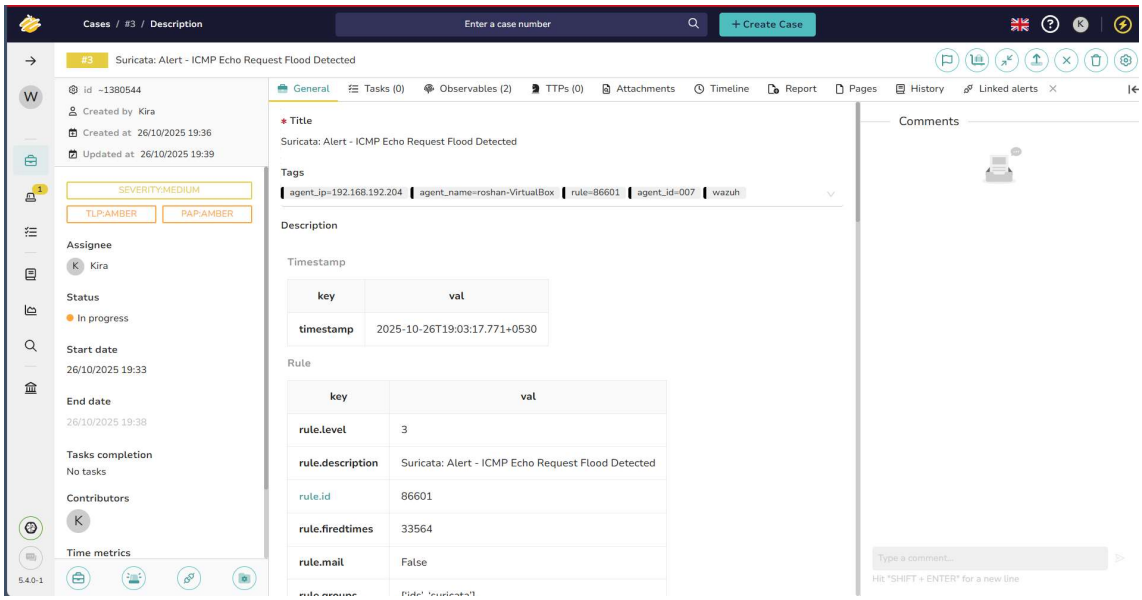
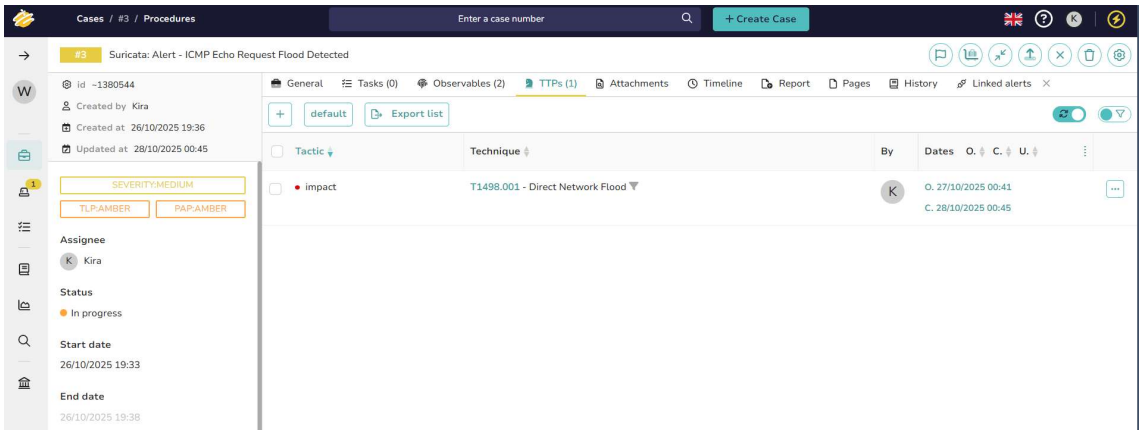


Figure 6 — Case automatically created in TheHive.

- *This confirms successful end-to-end integration: Wazuh detected the ICMP flood attack, forwarded the alert, and TheHive created a case with the full alert JSON in its description.*
- *Analysts can now triage, assign, and enrich the case directly within TheHive*

– MITRE ATT&CK Mapping :

- TheHive allows analysts to enrich and categorize cases according to the **MITRE ATT&CK framework**.
- By linking alerts or cases to specific ATT&CK techniques (e.g., T1498 – *Network Denial of Service*), SOC teams can quickly assess the adversary tactics involved and prioritize response efforts.
- During the integration process, MITRE technique identifiers can be added either automatically by the Python script (via a mapping file of rule IDs to techniques) or manually by analysts within TheHive UI.



– Conclusion :

- The successful integration of **Wazuh Manager** with **TheHive SOAR** platform demonstrates an effective open-source solution for automating the security incident lifecycle — from detection to case creation and response.
- Through this setup, Wazuh continuously monitors system and network events, identifies potential security threats such as ICMP floods or SSH brute-force attempts, and automatically forwards these alerts to TheHive using a custom integration script. TheHive, in turn, transforms these alerts into actionable cases that can be triaged, analyzed, and enriched by security analysts.
- This integrated ecosystem enhances situational awareness, reduces manual workload, and significantly improves the organization's response time to emerging security threats.
- After a case is created in TheHive, analysts triage, investigate, enrich with threat intelligence, respond to contain the threat, and document their findings for closure.