# w

# 

# 

# Step-by-step Guide: “Wazuh-osTicket” Integration

# Create by: Syed Jawad Ali Shah

# Wazuh-osTicket Integration Guide

## Table of Contents

1. [Introduction](#Xe3d0fc0bea9a42ce7605565d0964033d7f6ee47)
   1. [Overview of osTicket](#X150802f5825e677416346b4a452b68c60717087)
   2. [Benefits of osTicket](#X8d2c817a5ddd6f682ed738da9455c4f79641343)
   3. [Why Integrate with Wazuh?](#Xe86c1b43c07d1de3f712e1bdd6c5a57e5877979)
2. [Installing osTicket](#X4700525ec05206e263e9a93cf2c15f3cb84404e)
   1. [Prerequisites](#X54acc0ec3bcdd33503f81646d20f35c1d1e5008)
   2. [Installation Steps](#X0dd50e1286b7562d5883e1a0b0ec5496293c1ad)
3. [Integrating Wazuh with osTicket](#X7e0b4e6d9487ab48d7ea968205947b2addd1436)
   1. [Prerequisites](#Xbd96d25617be560fe07e4630b9966763cfa7f31)
   2. [Step 1: Configure osTicket API](#Xbaae09dd7fb3aa892bb8ca65a33a81d99afbf86)
   3. [Step 2: Create Integration Scripts](#X501894d6ce4e585599556c7564fad78880c388e)
   4. [Step 3: Configure Wazuh](#Xace5db339e86633865c87da55a1d37beb68b07a)
   5. [Step 4: Test the Integration](#Xc5402d9c6779682e55f403a0d31742a5a91a13d)
   6. [Step 5: Troubleshooting](#X14f2a899f67e7f295048558bc1bd09c442f6b05)
4. [Conclusion](#X50c7748aa6d421f569a5daf2425c9e7043a6cad)
5. [References](#Xa559bd5247f9128e74e6edcc17de37648e8cde6)

## 1. Introduction

### 1.1 Overview of osTicket

osTicket is a robust, open-source helpdesk and ticketing system designed to streamline the management of support requests and internal issues. Its intuitive web-based interface enables efficient ticket submission and management, making it ideal for IT support, customer service, and incident response.

### 1.2 Benefits of osTicket

* **Cost-Effective**: Free, open-source platform with no licensing fees.
* **Customizable**: Supports custom fields, forms, and workflows for tailored solutions.
* **User-Friendly**: Provides intuitive portals for end-users and staff.
* **Automated**: Features auto-responses, ticket filters, and escalations to optimize processes.
* **API-Enabled**: Offers a REST API for seamless integration with external systems.
* **Scalable**: Handles high ticket volumes for organizations of all sizes.
* **Community-Driven**: Backed by comprehensive documentation and an active community ([osTicket Documentation](https://docs.osticket.com/), [osTicket Forums](https://forum.osticket.com/)).

### 1.3 Why Integrate with Wazuh?

Wazuh is an open-source security monitoring platform excelling in intrusion detection, log analysis, and threat detection. Integrating Wazuh with osTicket automates the creation of support tickets for security alerts, enhancing incident response. Key benefits include:

* **Automation**: Converts Wazuh alerts (e.g., failed logins, malware detection) into osTicket tickets.
* **Rapid Response**: Enables real-time ticketing for swift action.
* **Centralized Workflow**: Combines security monitoring with ticketing for streamlined management.
* **Accountability**: Tracks incident handling and resolution via tickets.
* **Flexible Workflows**: Routes and escalates alerts based on severity.

This integration is critical for organizations like DIAMATIX, a Wazuh Premium Partner, to strengthen security operations.

## 2. Installing osTicket

This section provides a step-by-step guide to installing osTicket on an Ubuntu 22.04/24.04 server at http://10.10.11.195. Adjust commands for other distributions as needed.

### 2.1 Prerequisites

| **Requirement** | **Details** |
| --- | --- |
| Server | Ubuntu 22.04/24.04, 2GB RAM, 2 CPUs, 20GB disk space |
| Network | Reachable at 10.10.11.195:80 (your IP here) |
| Root Access | Required for installation |
| Dependencies | Apache/Nginx, PHP, MySQL/MariaDB |

### 2.2 Installation Steps

1. **Update the System**:

* sudo apt update && sudo apt upgrade -y

1. **Install Dependencies**:

* sudo apt install -y apache2 mariadb-server php php-cli php-mysql php-gd php-imap php-intl php-json php-curl php-mbstring php-xml php-zip unzip

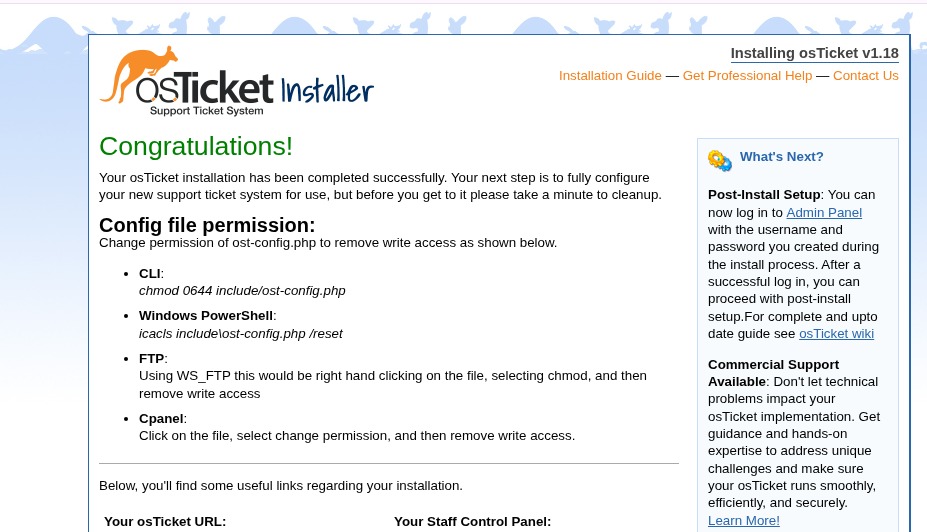
1. **Configure MySQL/MariaDB**:
   * Secure the installation:
   * sudo mysql\_secure\_installation
   * Follow prompts to set a root password and secure settings.
   * Create a database and user:
   * sudo mysql -u root -p
   * CREATE DATABASE osticket\_db;  
     CREATE USER 'osticket\_user'@'localhost' IDENTIFIED BY 'StrongPassword123!';  
     GRANT ALL PRIVILEGES ON osticket\_db.\* TO 'osticket\_user'@'localhost';  
     FLUSH PRIVILEGES;  
     EXIT;
   * Replace StrongPassword123! with a secure password.
2. **Download osTicket**:

* cd /tmp  
  wget https://github.com/osTicket/osTicket/releases/download/v1.18/osTicket-v1.18.zip  
  unzip osTicket-v1.18.zip  
  sudo mv upload /var/www/html/osticket

1. **Set Permissions**:

* sudo chown -R www-data:www-data /var/www/html/osticket  
  sudo chmod -R 755 /var/www/html/osticket

1. **Configure Apache**:
   * Create a virtual host:
   * sudo nano /etc/apache2/sites-available/osticket.conf
   * Add:
   * <VirtualHost \*:80>  
      ServerName 10.10.11.195  
      DocumentRoot /var/www/html/osticket  
      <Directory /var/www/html/osticket>  
      Options -Indexes +FollowSymLinks  
      AllowOverride All  
      Require all granted  
      </Directory>  
      ErrorLog ${APACHE\_LOG\_DIR}/osticket\_error.log  
      CustomLog ${APACHE\_LOG\_DIR}/osticket\_access.log combined  
     </VirtualHost>
   * Enable the site:
   * sudo a2ensite osticket.conf  
     sudo a2enmod rewrite  
     sudo systemctl restart apache2
2. **Run the osTicket Web Installer**:
   * Navigate to http://10.10.11.195/setup in a browser.
   * Follow the wizard:
     + **System Settings**: Set helpdesk name and URL.
     + **Database Settings**:
       - Database: osticket\_db
       - Username: osticket\_user
       - Password: StrongPassword123!
     + Complete the installation.



*osTicket Setup Wizard: System and Database Settings at http://10.10.11.195/setup.*

1. **Secure Configuration File**:

* sudo mv /var/www/html/osticket/include/ost-config.php /var/www/html/osticket/include/ost-config.php.bak  
  sudo chmod 644 /var/www/html/osticket/include/ost-config.php.bak

1. **Access osTicket**:
   * Admin Panel: http://10.10.11.195/scp
   * User Portal: http://10.10.11.195
   * Use admin credentials set during installation.

  
*osTicket Admin Panel (http://10.10.11.195/scp) and User Portal (http://10.10.11.195).*

1. **Troubleshooting**:
   * Check Apache logs:
   * tail -f /var/log/apache2/osticket\_error.log
   * Verify MySQL:
   * sudo systemctl status mariadb

## 3. Integrating Wazuh with osTicket

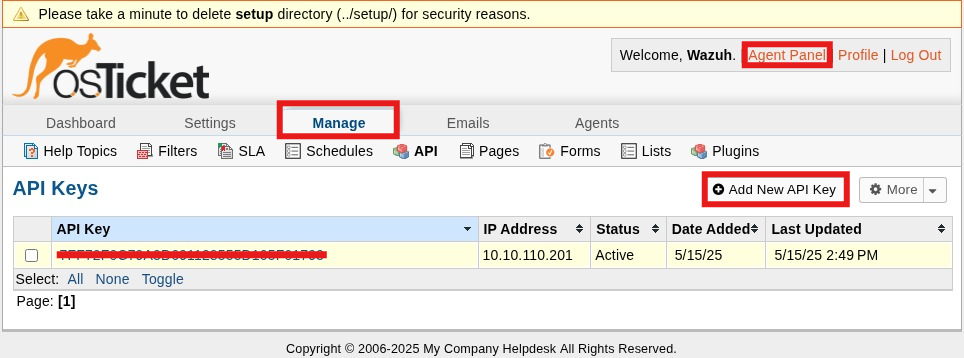
This section details the integration tested on May 15, 2025, with osTicket at http://10.10.11.195 and Wazuh Manager at 10.10.11.100. It uses active response to trigger custom-osticket.sh for rules 5715 (SSH authentication success) and 5716 (SSH authentication failure).

### 3.1 Prerequisites

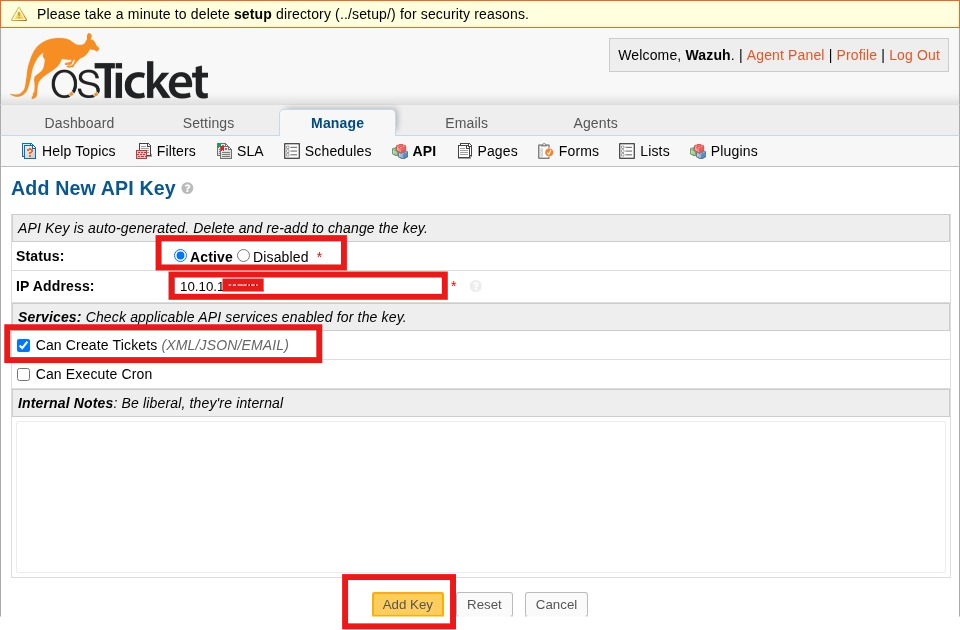
| **Requirement** | **Details** |
| --- | --- |
| osTicket | Installed at http://10.10.11.195 with API key 7FF72F0C79A3D691128555D105F61793 |
| Wazuh | Running manager at 10.10.110.100(your Wazuh server IP Here) |
| Network | Wazuh Manager must reach http://10.10.11.195:80 |
| Permissions | Root or wazuh group access on Wazuh server |

### 3.2 Step 1: Configure osTicket API

1. **Log in to osTicket Admin Panel**:
   * Access http://10.10.110.195/scp.
   * Navigate to **Admin Panel → Manage → API Keys**.



1. **Verify/Create API Key**:
   * Ensure API key 7FF72F0C79A3D691128555D105F61793 is active.
   * Set allowed IP to 10.10.110.100.
   * Enable **Can Create Tickets (XML/JSON/EMAIL)**.

*****Configuring API Key in osTicket Admin Panel for Wazuh Integration.*

1. **Notes**:
   * Include header X-API-Key: 7FF72F0C79A05F61793 in requests.
   * Configure X-Forwarded-For for 10.10.110.100 if behind a reverse proxy.
2. **Test API**:

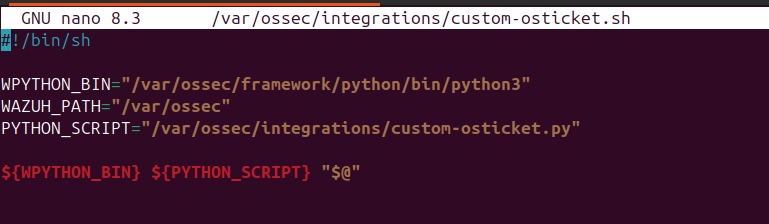
* curl -X POST http://10.10.110.195/api/tickets.json \  
  -H "X-API-Key: Your API KEY HERE" \  
  -H "Content-Type: application/json" \  
  -d '{"alert": true, "autorespond": false, "source": "API", "name": "Test User", "email": "test@wazuh.com", "subject": "Test Ticket", "message": "This is a test ticket from Wazuh", "ip": "10.10.110.100"}'
* Expected: 201 Created with a ticket ID (e.g., 137661).

### 3.3 Step 2: Create Integration Scripts

#### 3.3.1 Shell Script: /var/ossec/integrations/custom-osticket.sh

1. Create the file:

* nano /var/ossec/integrations/custom-osticket.sh
* Add:
* #!/bin/sh  
    
  WPYTHON\_BIN="/var/ossec/framework/python/bin/python3"  
  WAZUH\_PATH="/var/ossec"  
  PYTHON\_SCRIPT="/var/ossec/integrations/custom-osticket.py"  
    
  ${WPYTHON\_BIN} ${PYTHON\_SCRIPT} "$@"



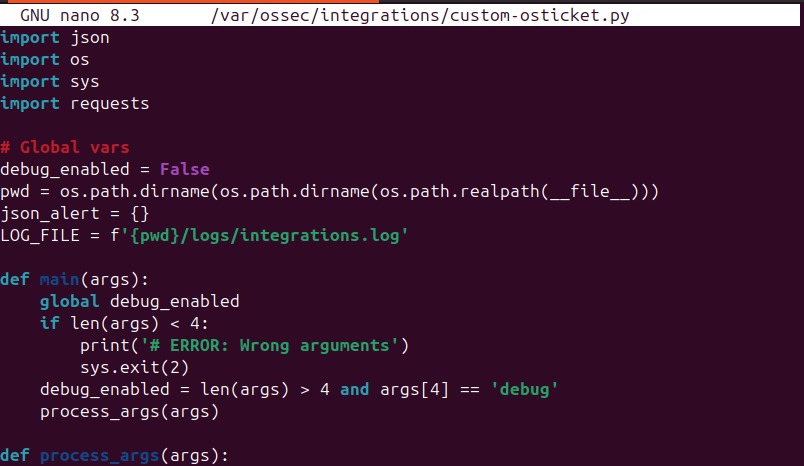
1. Set permissions:

* chmod 750 /var/ossec/integrations/custom-osticket.sh  
  chown root:wazuh /var/ossec/integrations/custom-osticket.sh

#### 3.3.2 Python Script: /var/ossec/integrations/custom-osticket.py

1. Create the file:

* nano /var/ossec/integrations/custom-osticket.py
* Add:
* import json  
  import os  
  import sys  
  import requests  
    
  # Global vars  
  debug\_enabled = False  
  pwd = os.path.dirname(os.path.dirname(os.path.realpath(\_\_file\_\_)))  
  json\_alert = {}  
  LOG\_FILE = f'{pwd}/logs/integrations.log'  
    
  def main(args):  
   global debug\_enabled  
   if len(args) < 4:  
   print('# ERROR: Wrong arguments')  
   sys.exit(2)  
   debug\_enabled = len(args) > 4 and args[4] == 'debug'  
   process\_args(args)  
    
  def process\_args(args):  
   alert\_file\_location, api\_key, hook\_url = args[1], args[2], args[3]  
   if not is\_valid\_url(hook\_url):  
   debug(f"# ERROR: Invalid Hook URL: {hook\_url}")  
   sys.exit(5)  
   json\_alert = load\_alert(alert\_file\_location)  
   ticket\_data = generate\_ticket\_data(json\_alert)  
   send\_ticket(ticket\_data, api\_key, hook\_url)  
    
  def generate\_ticket\_data(alert):  
   subject = f"Alert: {alert['rule'].get('description', 'No description')}"  
   message = alert.get('full\_log', 'No log available')  
   ticket\_data = {  
   "alert": True,  
   "autorespond": False,  
   "source": "API",  
   "name": "Wazuh Integration",  
   "email": "alert@wazuh.com",  
   "phone": "123456789",  
   "subject": subject,  
   "message": f"data:text/html,MESSAGE <b>{message}</b>",  
   "ip": "Your Wazuh Manager IP"  
   }  
   return ticket\_data  
    
  def send\_ticket(ticket\_data, api\_key, hook\_url):  
   headers = {  
   'Content-Type': 'application/json',  
   'X-API-Key': api\_key  
   }  
   response = requests.post(hook\_url, data=json.dumps(ticket\_data), headers=headers)  
   if response.status\_code != 201:  
   debug(f"# ERROR: Failed to create ticket. Status Code: {response.status\_code}")  
   raise Exception(f"Failed to create ticket, status code: {response.status\_code}")  
   debug(f"# Ticket created successfully with response: {response.text}")  
    
  def load\_alert(file\_path):  
   try:  
   with open(file\_path) as alert\_file:  
   return json.load(alert\_file)  
   except FileNotFoundError:  
   debug(f"# ERROR: Alert file {file\_path} doesn't exist")  
   sys.exit(6)  
   except json.decoder.JSONDecodeError as e:  
   debug(f"Failed to parse JSON alert: {e}")  
   sys.exit(7)  
    
  def is\_valid\_url(url):  
   from urllib.parse import urlparse  
   parsed\_url = urlparse(url)  
   return all([parsed\_url.scheme, parsed\_url.netloc])  
    
  def debug(msg):  
   if debug\_enabled:  
   print(msg)  
   with open(LOG\_FILE, "a") as f:  
   f.write(msg + '\n')  
    
  if \_\_name\_\_ == "\_\_main\_\_":  
   main(sys.argv)



1. Set permissions:

* chmod 750 /var/ossec/integrations/custom-osticket.py  
  chown root:wazuh /var/ossec/integrations/custom-osticket.py

### 3.4 Step 3: Add Cron job

1. **Create Cron Job**:

* nano /etc/cron.d/wazuh-osticket

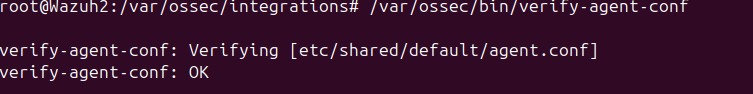
  
*\* \* \* \* \* wazuh /var/ossec/integrations/custom-osticket.sh /var/ossec/logs/alerts/alerts.json 7FF72F0C79A3D691128555D105F61793 http://10.10.110.195/api/tickets.json debug >> /var/ossec/logs/integrations.log 2>&1*

1. **Verify local\_rules.xml**:
   * Ensure the existing rule is unchanged:
   * cat /var/ossec/etc/rules/local\_rules.xml
   * Should match:
   * <!-- Local rules -->  
     <!-- Modify it at your will. -->  
     <!-- Copyright (C) 2015, Wazuh Inc. -->  
       
     <group name="local,syslog,sshd,">  
      <rule id="100001" level="5">  
      <if\_sid>5716</if\_sid>  
      <srcip>1.1.1.1</srcip>  
      <description>sshd: authentication failed from IP 1.1.1.1.</description>  
      <group>authentication\_failed,pci\_dss\_10.2.4,pci\_dss\_10.2.5,</group>  
      </rule>  
     </group>
   * If modified, restore it:
   * nano /var/ossec/etc/rules/local\_rules.xml
   * Add the above content.
   * Set permissions:
   * chmod 640 /var/ossec/etc/rules/local\_rules.xml  
     chown root:wazuh /var/ossec/etc/rules/local\_rules.xml

*Verifying Existing Rule 100001 in /var/ossec/etc/rules/local\_rules.xml.*

1. **Validate Configuration**:

* /var/ossec/bin/verify-agent-conf



1. **Restart Wazuh Manager**:

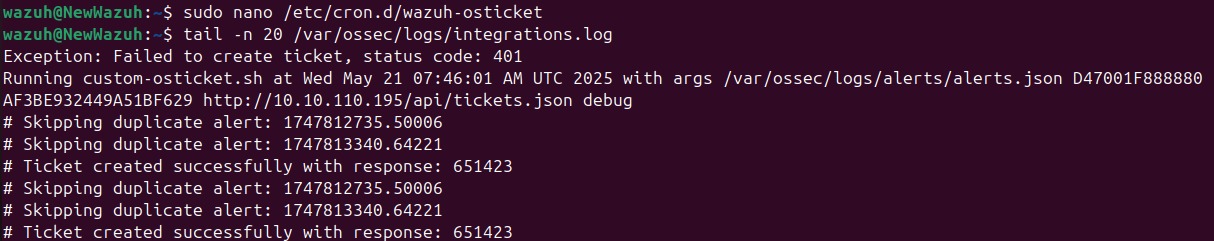
* systemctl restart wazuh-manager
* Verify:
* systemctl status wazuh-manager

### 3.5 Step 4: Test with Real Alerts:

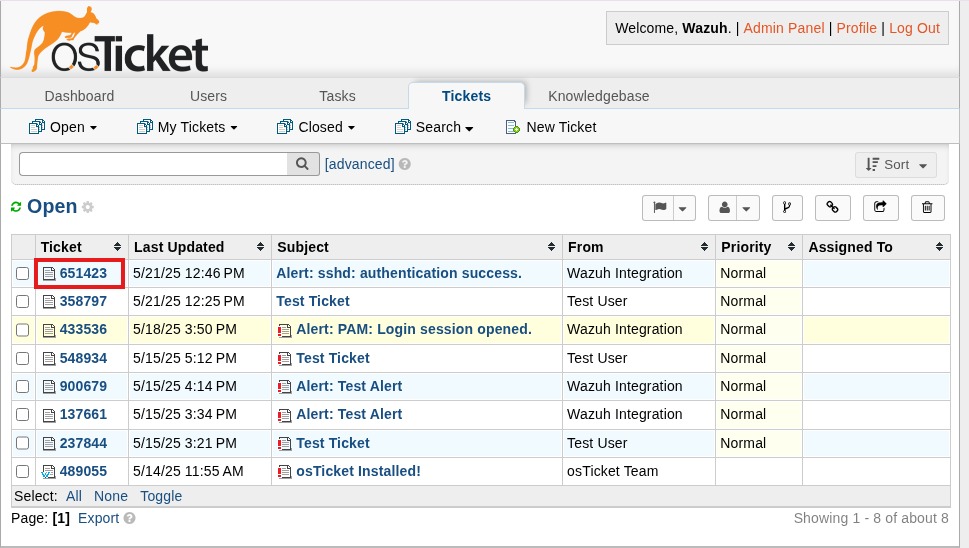
1. **Verify Previous Test**:

Trigger SSH authentication success (rule 5715):

* + tail -f /var/ossec/logs/integrations.



* + Check ticket ID 651423 at <http://10.10.110.195/scp>.



*Manual Test Output and Ticket 651423 in osTicket Admin Panel.*

1. **Test with Real Alerts**:
   * Trigger SSH authentication success (5715):
   * ssh Wazuh@10.10.11.100
   * Trigger SSH authentication failure (5716):
   * ssh invaliduser@10.10.110.100
   * Check alerts:
   * tail -f /var/ossec/logs/alerts/alerts.log | grep -E "5715|5716"
   * Verify tickets at http://10.10.11.195/scp.
2. **Monitor Logs**:

* tail -f /var/ossec/logs/integrations.log  
  tail -f /var/ossec/logs/ossec.log

### 3.6 Step 5: Troubleshooting

* **Wazuh Manager Fails**:
  + Check:
  + journalctl -xeu wazuh-manager.service
  + **[Insert Screenshot: Wazuh Manager Logs]**  
    *Checking Wazuh Manager Errors with journalctl -xeu wazuh-manager.service.*
* **No Tickets Generated**:
  + Verify wazuh-execd:
  + tail -f /var/ossec/logs/ossec.log | grep execd
  + Test as wazuh user:
  + sudo -u wazuh /var/ossec/integrations/custom-osticket.sh /var/ossec/logs/alerts/alerts.json 7FF72F0C79A3D691128555D105F61793 http://10.10.11.195/api/tickets.json debug
  + Check integration logs:
  + tail -f /var/ossec/logs/integrations.log

*Monitoring Integration Logs for Ticket Creation Issues.*

* + Use logtest:
  + /var/ossec/bin/wazuh-logtest
  + Input: May 15 12:25:07 Wazuh2 sshd-session[76661]: Accepted password for Wazuh from 10.10.110.156 port 63080 ssh2
* **ossec.conf Error**:
  + Share the <ossec\_config> section if XML errors persist:
  + cat /var/ossec/etc/ossec.conf | grep -A 10 -B 10 "<ossec\_config>"

**Note**: If automatic ticket generation fails, ensure wazuh-execd is running and the API key is correctly configured. Contact DIAMATIX support for persistent issues.

## 4. Conclusion

This guide enables seamless integration of Wazuh with osTicket, security incident ticketing as demonstrated by ticket. By following the osTicket installation, active response configuration for rules 5715 and 5716, and troubleshooting steps, organizations can enhance incident response efficiency.

