**MỤC LỤC**

[PHẦN 1: CÀI ĐẶT 2](#_Toc194584564)

[1️. Dừng từng thành phần của Wazuh 4](#_Toc194584565)

[1.1 Dừng Wazuh Manager 4](#_Toc194584566)

[1.2 Dừng Wazuh Indexer (dựa trên OpenSearch) 4](#_Toc194584567)

[1.3 Dừng Wazuh Dashboard (giao diện web) 4](#_Toc194584568)

[1.4 Dừng Filebeat (thu thập logs, nếu có cài đặt) 5](#_Toc194584569)

[2. Mở lại Wazuh 5](#_Toc194584570)

[3. Mở website Wazuh 5](#_Toc194584571)

[PHẦN 2: CÁC KỊCH BẢN 5](#_Toc194584572)

[1. Wazuh Vulnerability Detector 5](#_Toc194584573)

[2. Phát hiện các nỗ lực đăng nhập trên máy chủ Linux - Linux authentication monitoring 7](#_Toc194584574)

[3. Phát hiện đăng nhập RDP trái phép vào Windows với Wazuh - RDP monitoring with Wazuh!!! 10](#_Toc194584575)

[4. Cấu hình tính năng FIM (File integrity monitoring) theo dõi tính toàn vẹn của File. 13](#_Toc194584576)

[5. Phát hiện và loại bỏ phần mềm độc hại bằng cách tích hợp VirusTotal vào Wazuh Manager: 18](#_Toc194584577)

[5.1. Agent Window 18](#_Toc194584578)

[5.2. Agent Linux Ubuntu 31](#_Toc194584579)

[6. Detecting an SQL injection attack 39](#_Toc194584580)

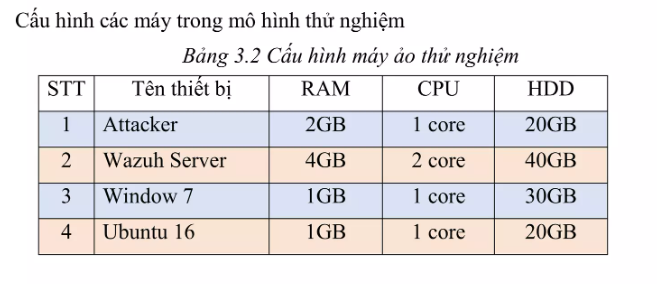
[7. Detecting malware using YARA integration 43](#_Toc194584581)

[7.1. Cấu hình trên Ubuntu Agent 43](#_Toc194584582)

[7.2. Cấu hình trên Window Agent 54](#_Toc194584583)

**TRIỂN KHAI HỆ THỐNG MÃ NGUỒN MỞ SIEM: WAZUH**





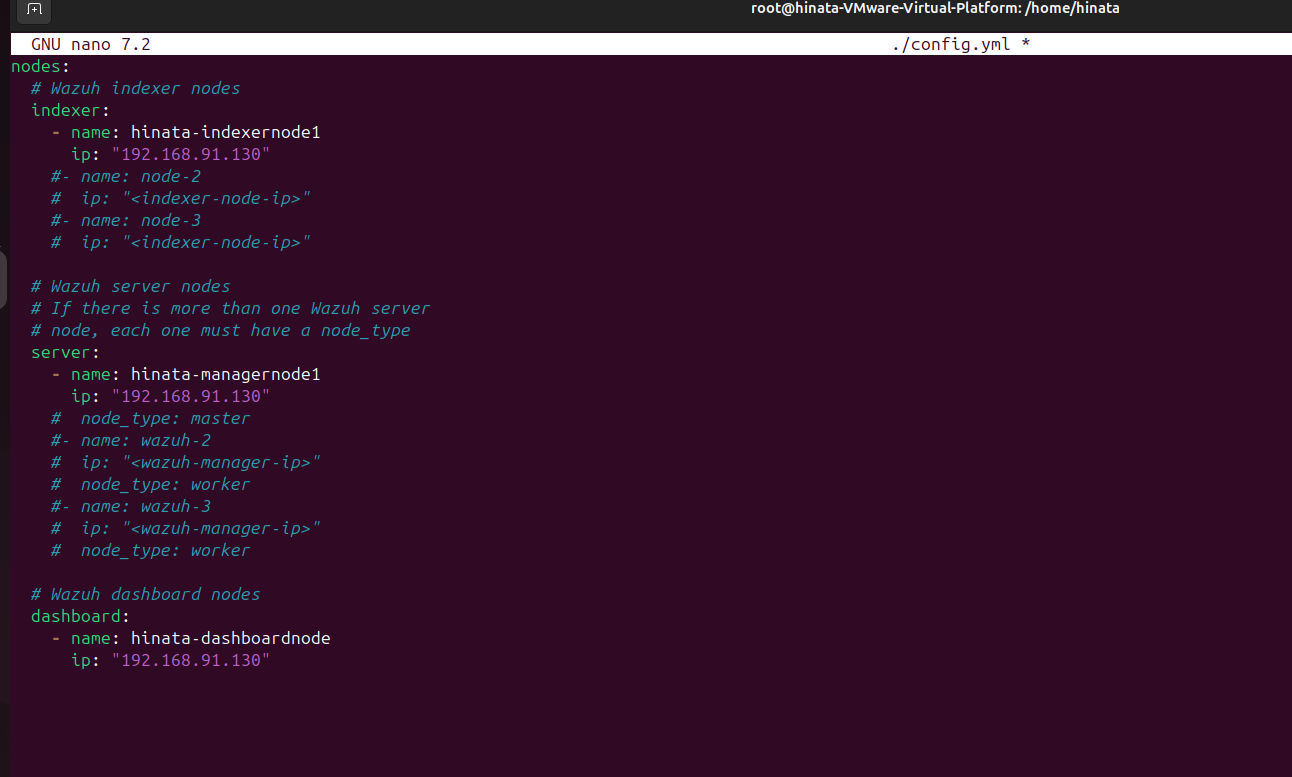
# PHẦN 1: CÀI ĐẶT

Các bước cài

**curl -sO** [**https://packages.wazuh.com/4.10/wazuh-install.sh**](https://packages.wazuh.com/4.10/wazuh-install.sh)

**curl -sO** [**https://packages.wazuh.com/4.10/config.yml**](https://packages.wazuh.com/4.10/config.yml)

**nano config.yml**

****

**nodes:**

**# Wazuh indexer nodes**

**indexer:**

**- name: hinata-indexernode1**

**ip: "192.168.91.135"**

**#- name: node-2**

**# ip: "<indexer-node-ip>"**

**#- name: node-3**

**# ip: "<indexer-node-ip>"**

**# Wazuh server nodes**

**# If there is more than one Wazuh server**

**# node, each one must have a node\_type**

**server:**

**- name: hinata-managernode1**

**ip: "192.168.91.135"**

**# node\_type: master**

**#- name: wazuh-2**

**# ip: "<wazuh-manager-ip>"**

**# node\_type: worker**

**#- name: wazuh-3**

**# ip: "<wazuh-manager-ip>"**

**# node\_type: worker**

**# Wazuh dashboard nodes**

**dashboard:**

**- name: hinata-dashboardnode**

**ip: "192.168.91.135"**

**bash wazuh-install.sh --generate-config-files**

**bash wazuh-install.sh --wazuh-indexer hinata-indexernode1**

**bash wazuh-install.sh --start-cluster**

**tar -xf wazuh-install-files.tar**

**root@hinata-VMware-Virtual-Platform:/home/hinata/wazuh-install-files# cat wazuh-passwords.txt**

indexer\_username: 'admin'

indexer\_password: ' gsGmbKJqyoH725ZGi8mrKY\*N+a+Yew+?'

***curl -k -u admin:Sqbxdo6v8Z08Z+CM.M403onehVcyORyM*** [***http://192.168.91.135:9200***](http://192.168.91.135:9200)

***bash wazuh-install.sh --wazuh-server hinata-managernode1***

***bash wazuh-install.sh --wazuh-dashboard hinata-dashboardnode***

***sudo lsof -i :9200***

***sudo lsof -i :9300***

***sudo kill -9 <PID>***

**Cách tắt đúng**

## **1️. Dừng từng thành phần của Wazuh**

sudo systemctl stop wazuh-manager

sudo systemctl stop wazuh-indexer

sudo systemctl stop wazuh-dashboard

sudo systemctl stop filebeat

sudo systemctl status wazuh-manager wazuh-indexer wazuh-dashboard filebeat

### **1.1 Dừng Wazuh Manager**

**sudo systemctl stop wazuh-manager**

### **1.2 Dừng Wazuh Indexer (dựa trên OpenSearch)**

**sudo systemctl stop wazuh-indexer**

### **1.3 Dừng Wazuh Dashboard (giao diện web)**

**sudo systemctl stop wazuh-dashboard**

### **1.4 Dừng Filebeat (thu thập logs, nếu có cài đặt)**

**sudo systemctl stop filebeat**

**sudo systemctl status wazuh-manager wazuh-indexer wazuh-dashboard filebeat**

**Tắt nhanh:**

**sudo systemctl stop wazuh-manager wazuh-indexer wazuh-dashboard filebeat**

## **2. Mở lại Wazuh**

**sudo systemctl start wazuh-manager wazuh-indexer wazuh-dashboard filebeat**

## **3. Mở website Wazuh**

[**https://192.168.91.135/**](https://192.168.91.135/)

indexer\_username: 'admin'

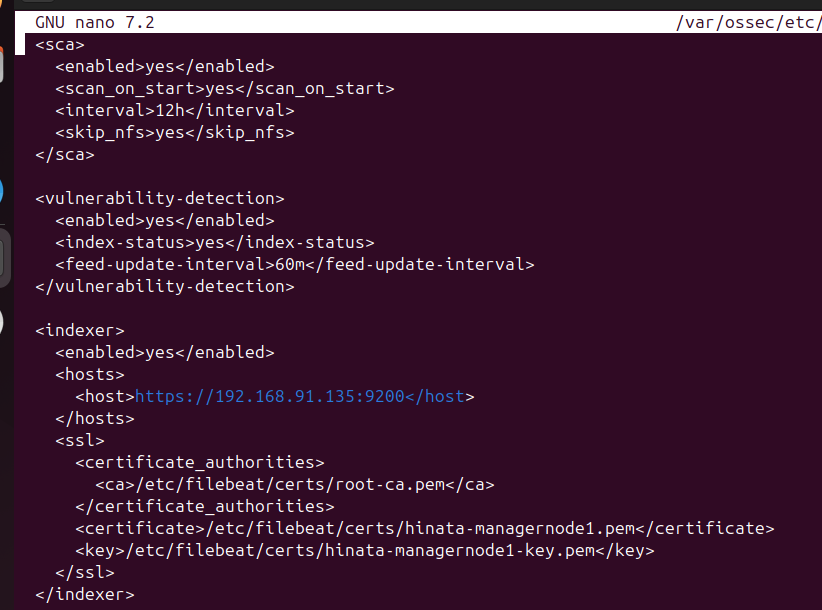
indexer\_password: ' gsGmbKJqyoH725ZGi8mrKY\*N+a+Yew+?'

# PHẦN 2: CÁC KỊCH BẢN

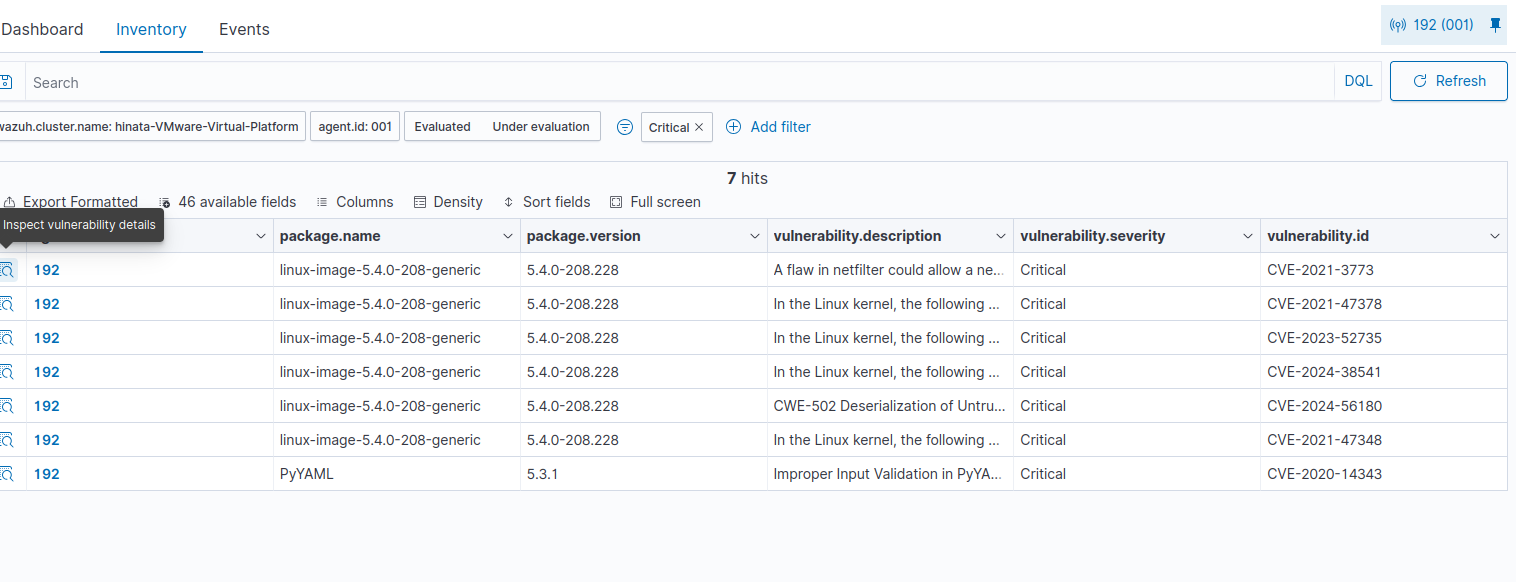
**Kịch bản:**

1. **Wazuh Vulnerability Detector**

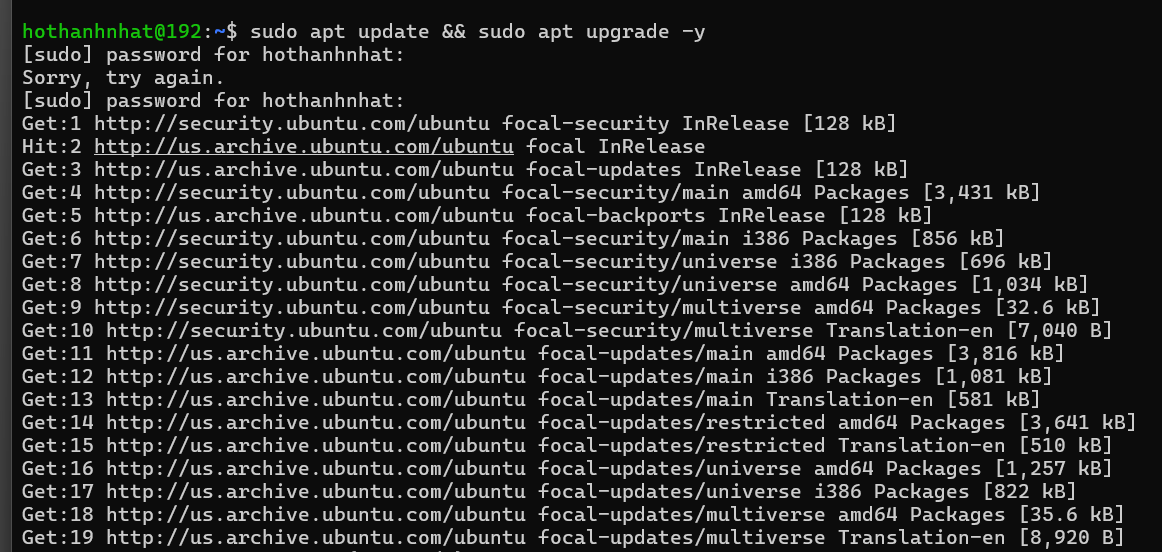
Mở sudo nano /var/ossec/etc/ossec.conf.



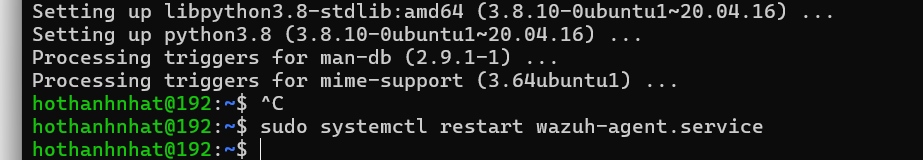
Agent Ubuntu:



Vào agent ubuntu vá lỗ hổng:

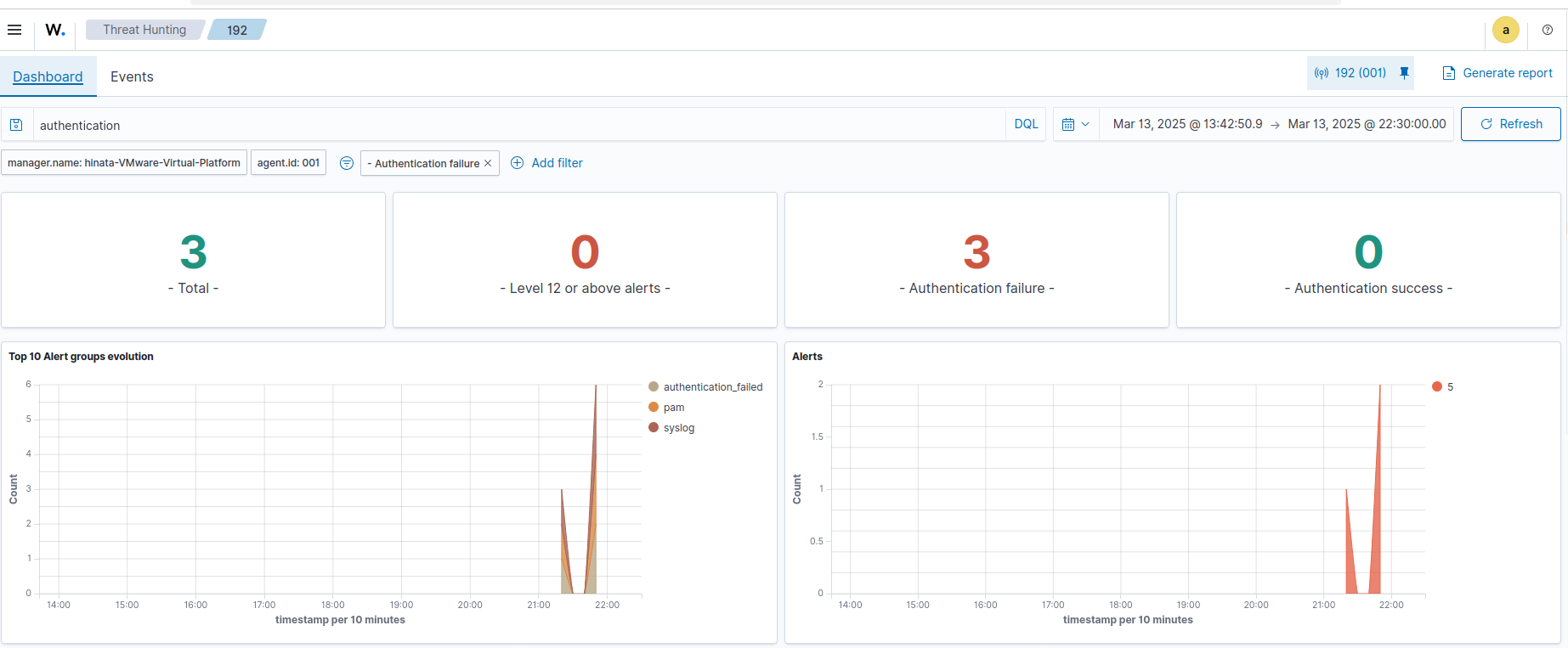


Restart lại dịch vụ wazuh-agent.



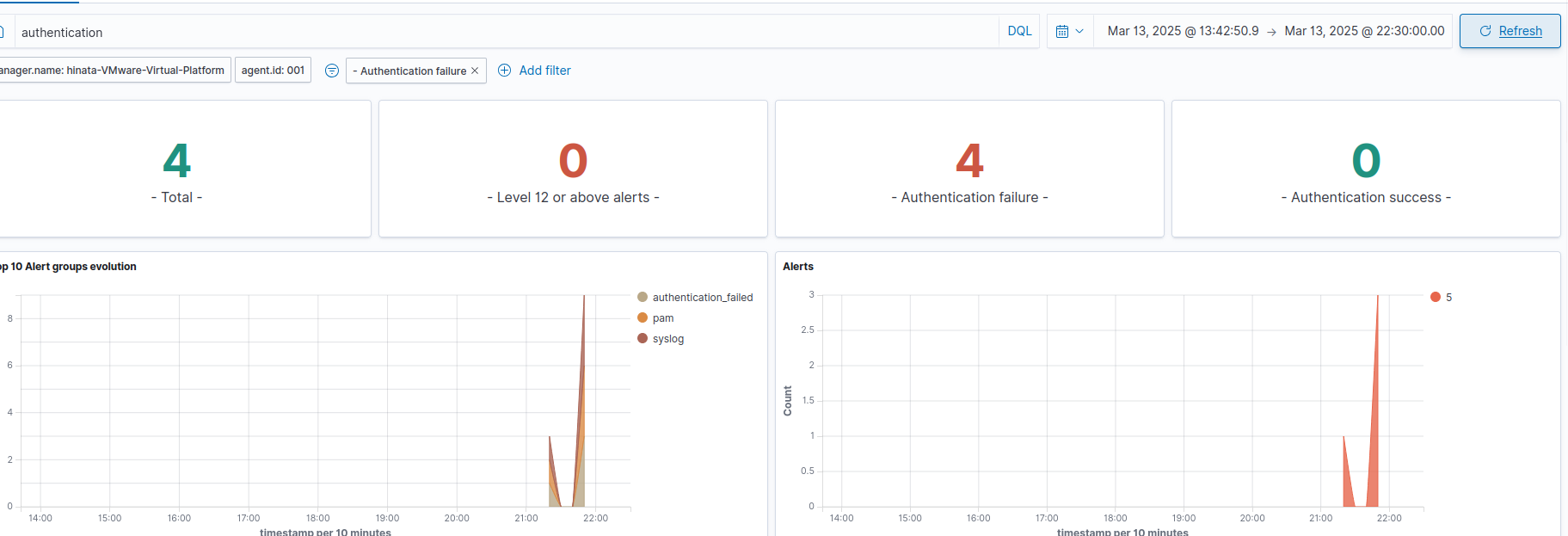
1. **Phát hiện các nỗ lực đăng nhập trên máy chủ Linux - Linux authentication monitoring**

Vào Endpoint (agent ubuntu) trên wazuh 🡪 Threat Hunting 🡪 Chọn **Event module: "authentication"** hoặc **"security"** 🡪 Đặt **khoảng thời gian** rộng hơn nếu không thấy kết quả.

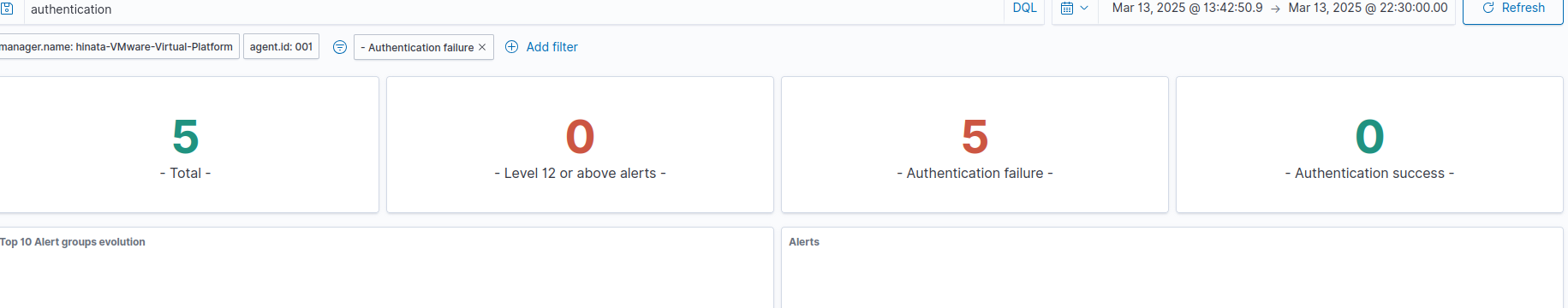


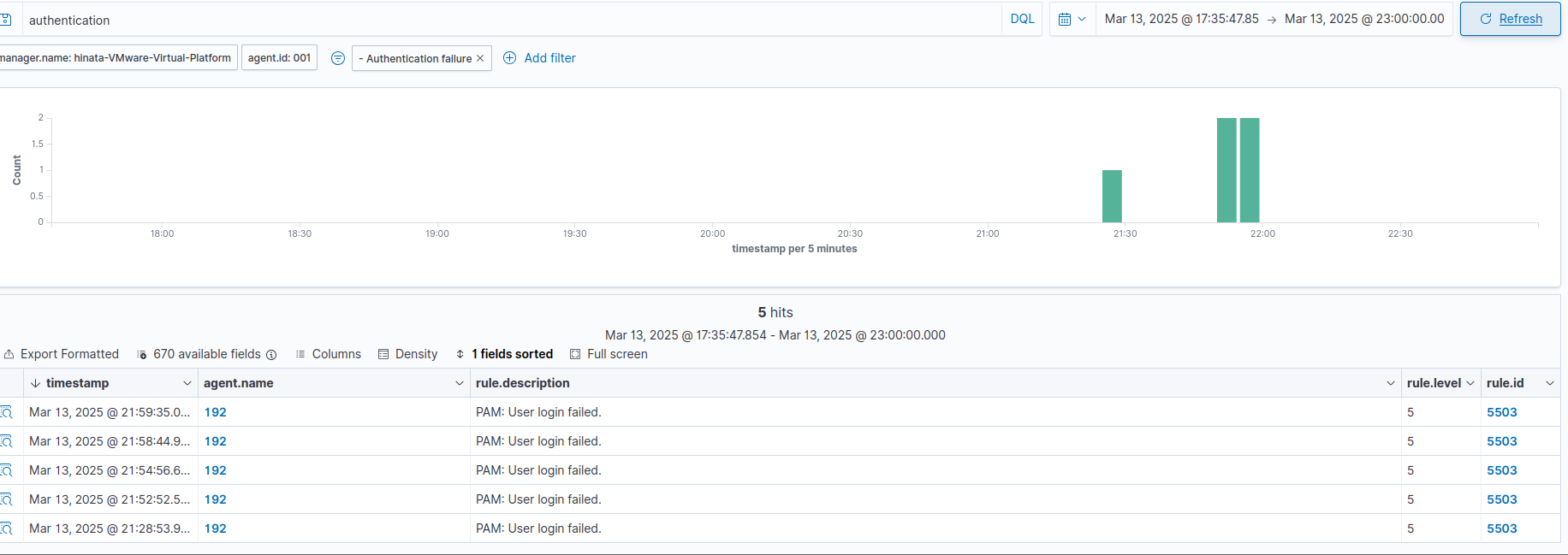
Thử ssh thất bại vào agent và xem kết quả:





Hoặc sai tên:



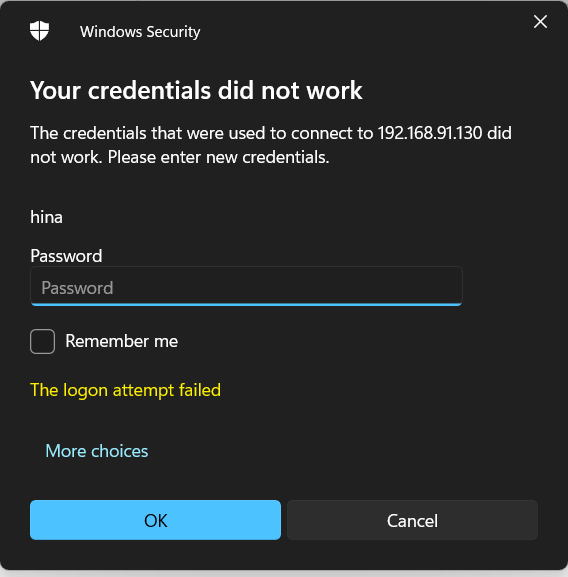




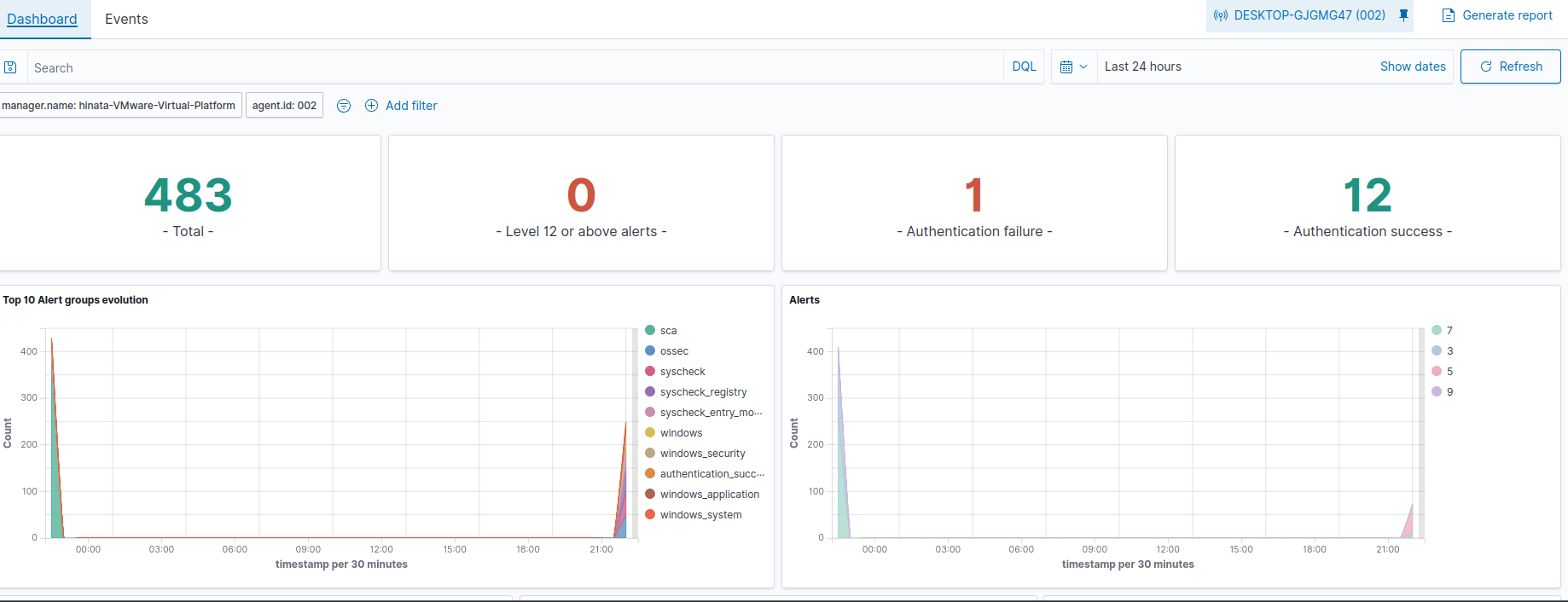


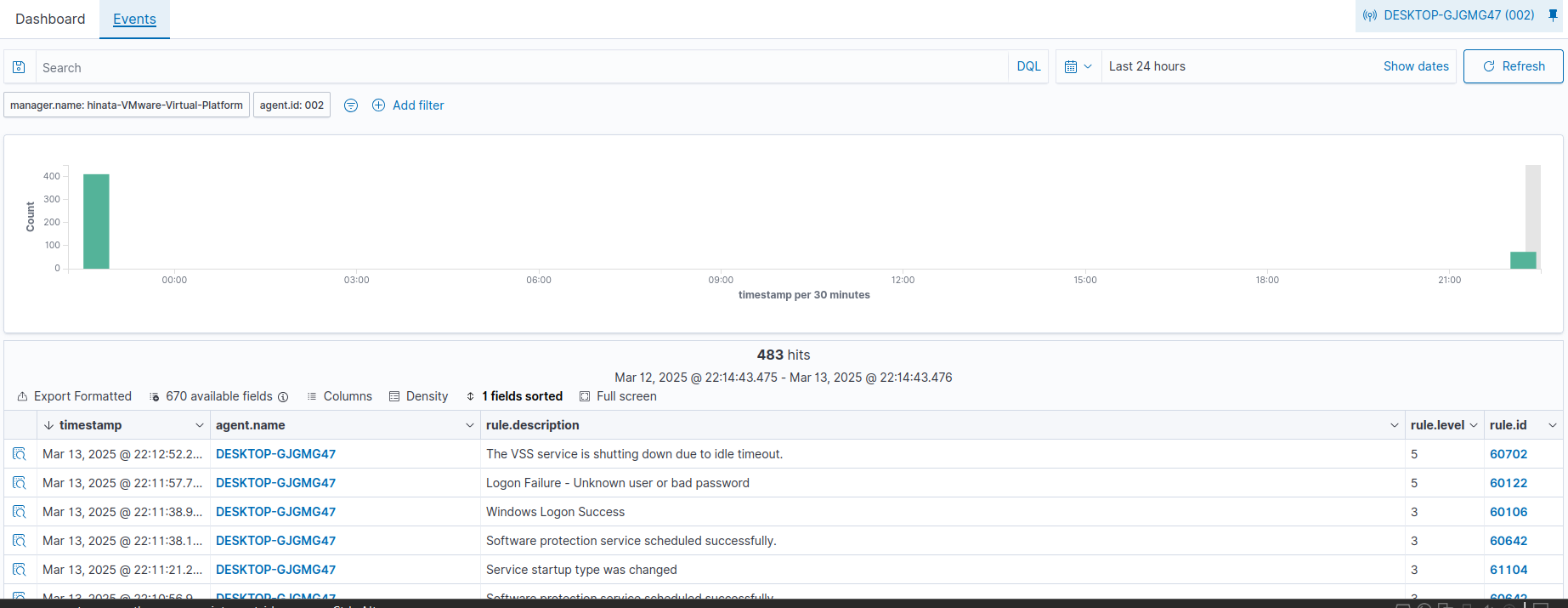
1. **Phát hiện đăng nhập RDP trái phép vào Windows với Wazuh - RDP monitoring with Wazuh!!!**

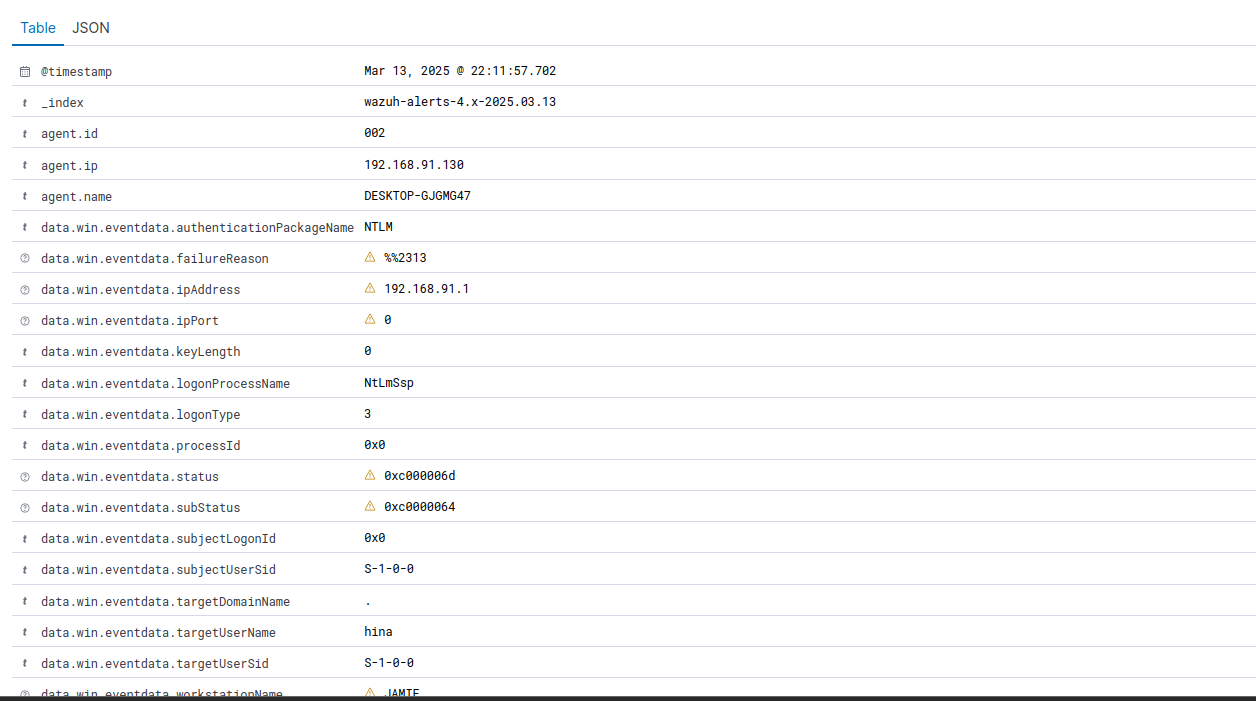
Thử RDP với tên và mật khẩu sai vào agent win 10.



Kết quả:



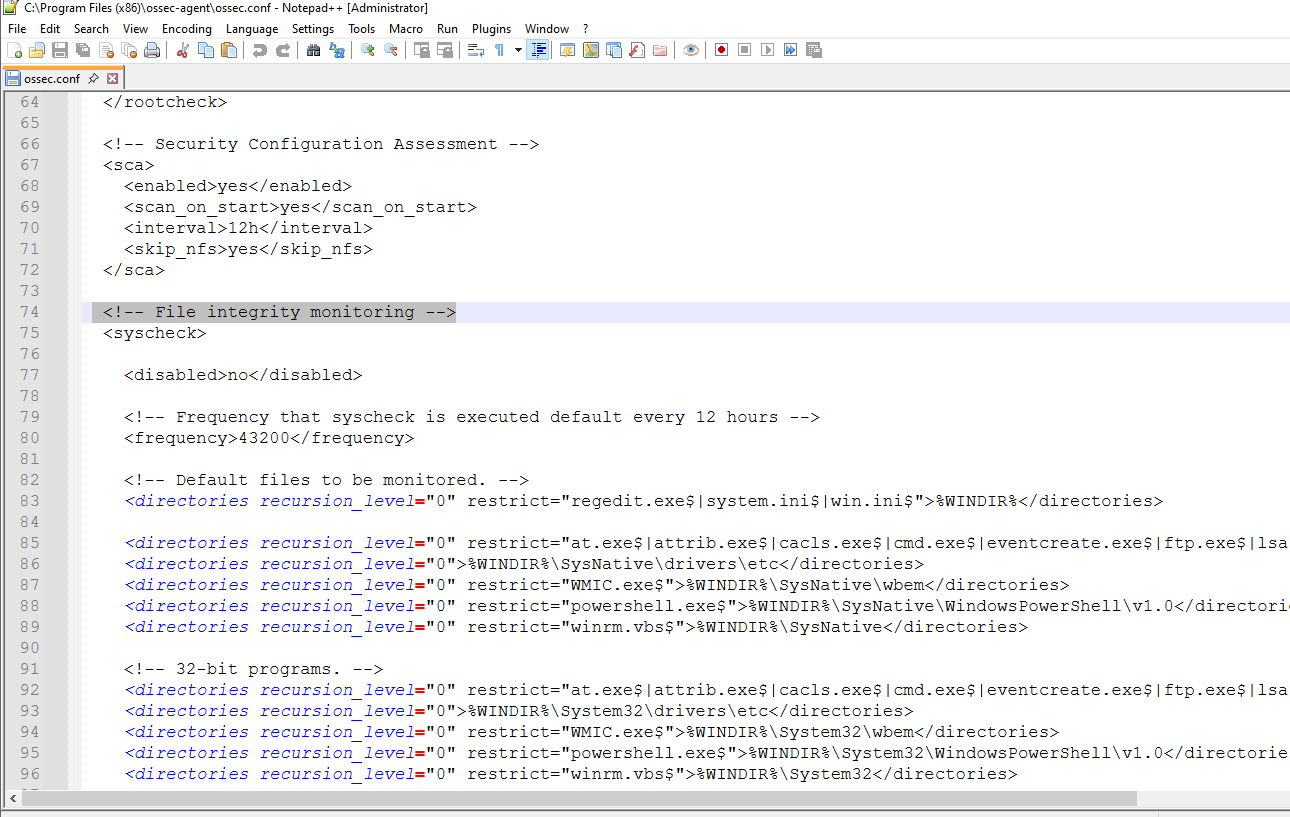




1. **Cấu hình tính năng FIM (File integrity monitoring) theo  
   dõi tính toàn vẹn của File.**

File Integrity Monitoring (FIM) trong Wazuh hoạt động bằng cách  
liên tục theo dõi và ghi nhận những thay đổi xảy ra đối với các tệp tin và  
thư mục quan trọng trong hệ thống. Khi một tệp tin được tạo mới, chỉnh  
sửa, hoặc xóa, FIM ghi lại các thông tin liên quan như thời gian thay đổi,  
người thực hiện, và loại thay đổi. Nó sử dụng các thuật toán để so sánh  
trạng thái hiện tại của tệp tin với các dấu hiệu (hash) đã được lưu trước đó  
để xác định xem có thay đổi bất thường hay không.

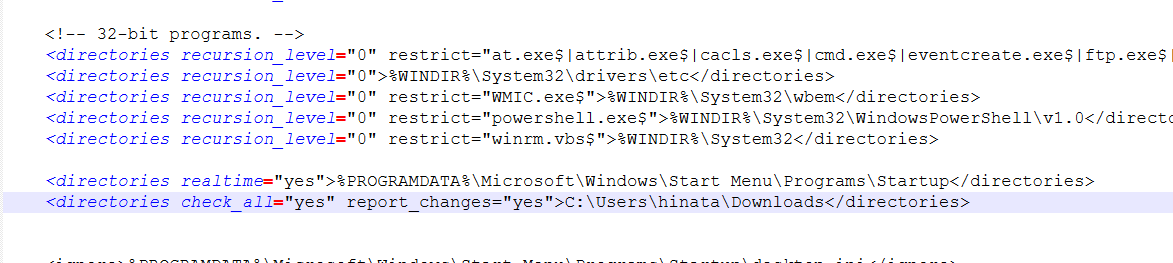
Ở trong file *“ossec.conf”* của Windows 10 Agent chúng ta sẽ kéo  
xuống và tìm mục *“File integrity monitoring”* và chúng ta sẽ chèn vào phía  
dưới dòng *<syscheck>*



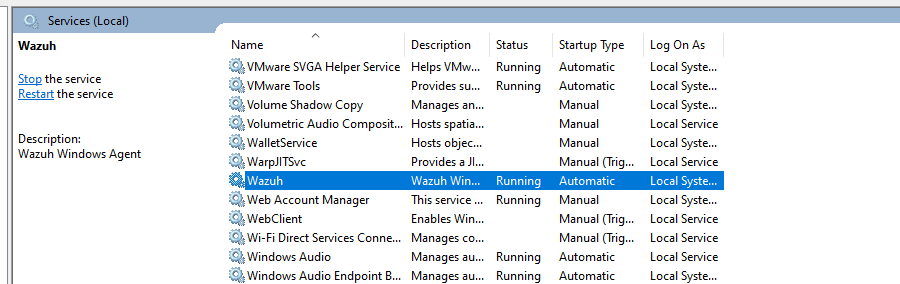
Và sau đó chèn dòng lệnh này vào giữa :

*<directories check\_all="yes"  
report\_changes="yes"><FILEPATH\_OF\_MONITORED\_FILE></directori es>*

*Thực hiện giám sát ở thư mục Downloads*



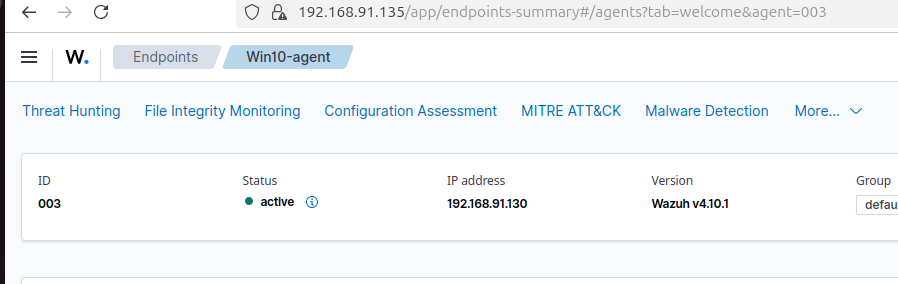
Thực hiện “Restart” lại Wazuh.



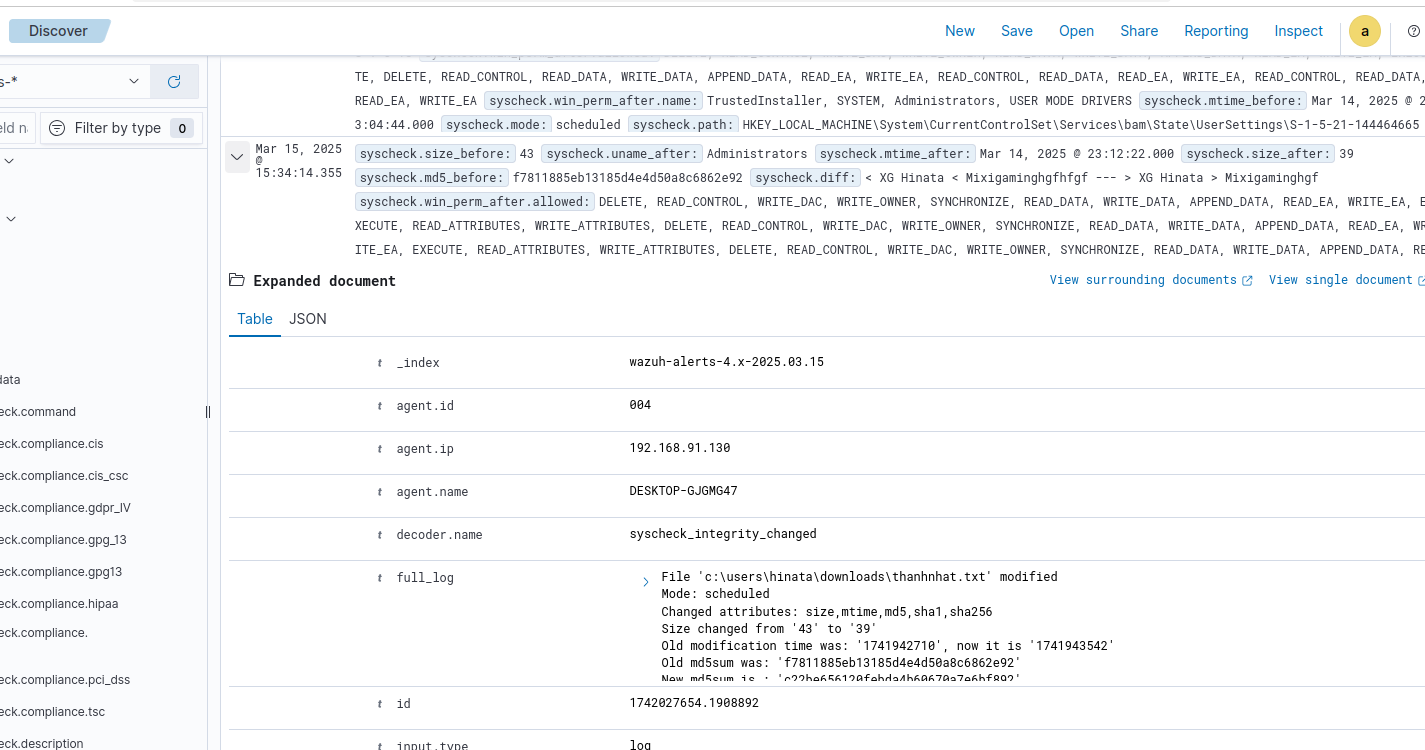
Thử tạo một file txt trong DownLoad và đợi:

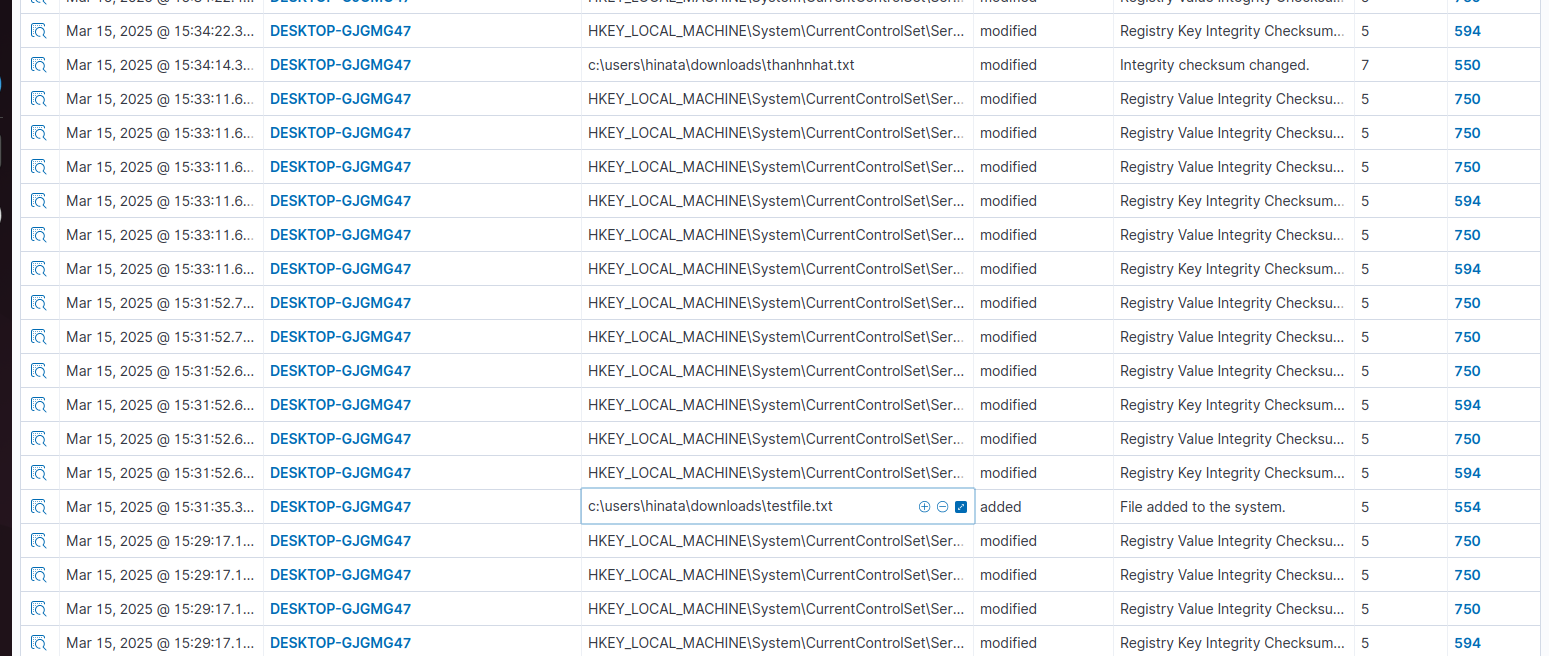


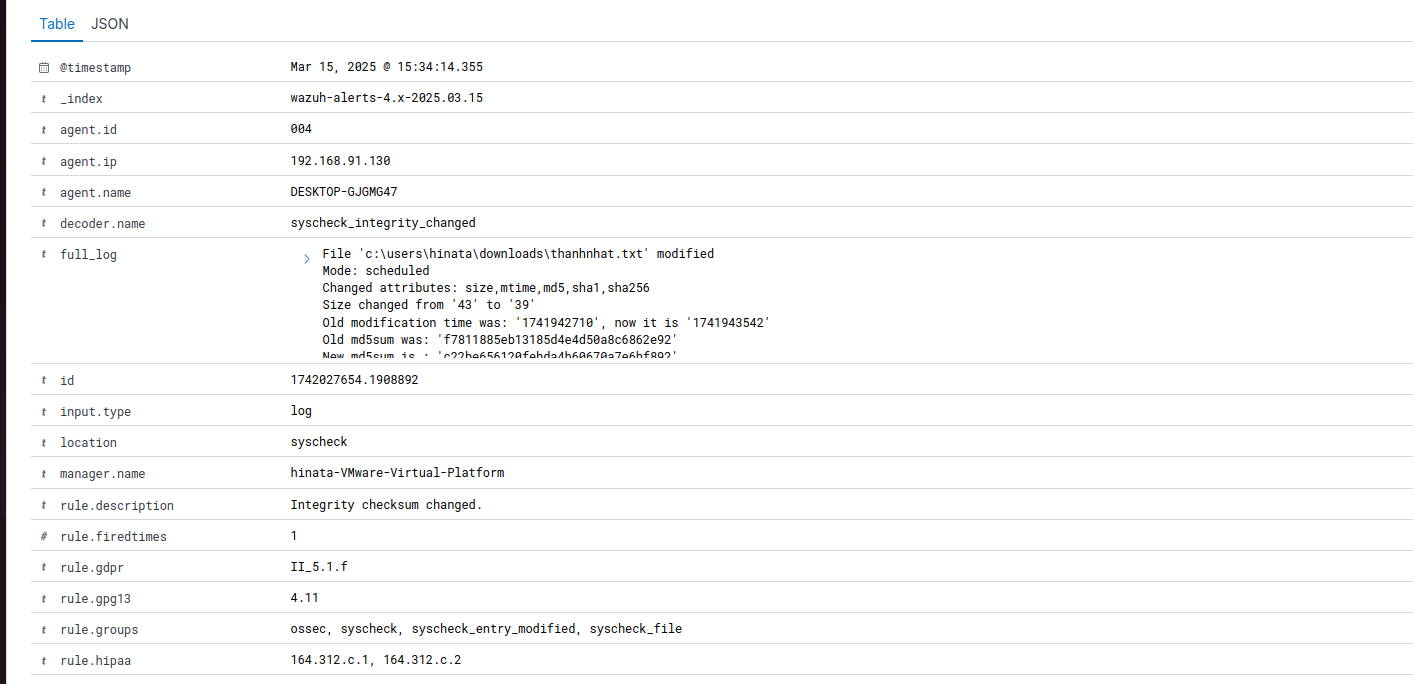
Quá trình này sẽ gửi log về cho Wazuh Sever , chúng ta có thể đăng nhập  
vào Wazuh Server mở Dashboard và vào mục Integrity Monitoring để  
check.

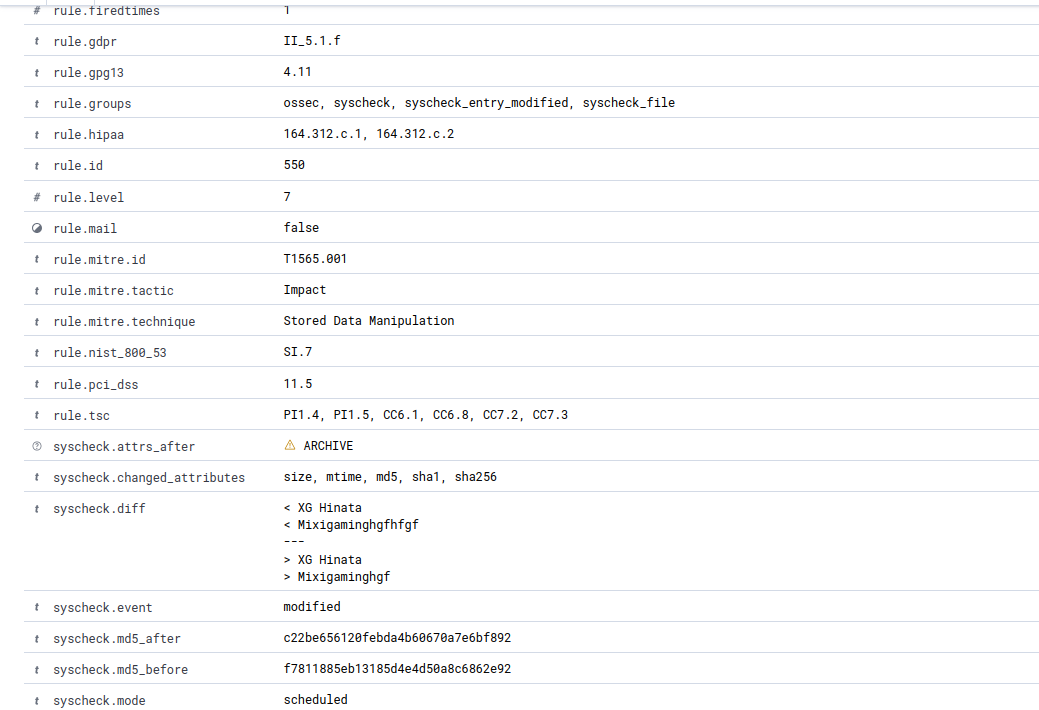


Kết quả:







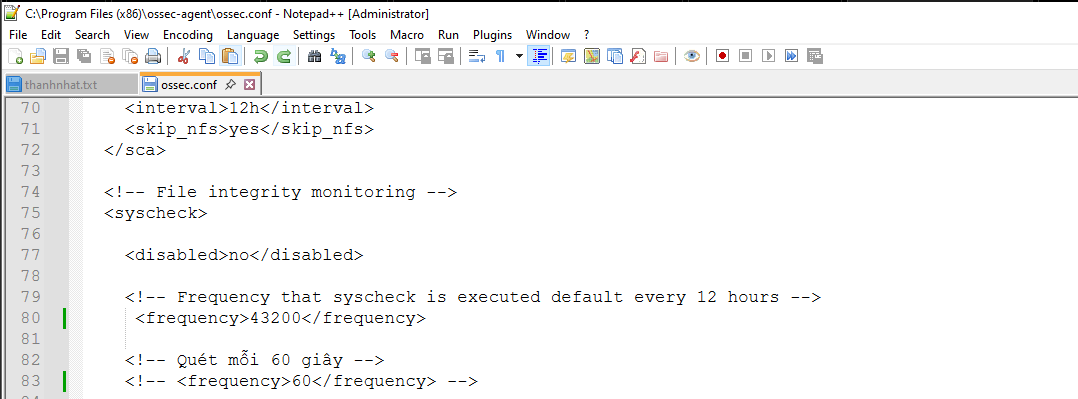




1. **Phát hiện và loại bỏ phần mềm độc hại bằng cách tích hợp VirusTotal vào Wazuh Manager:**
   1. **Agent Window**

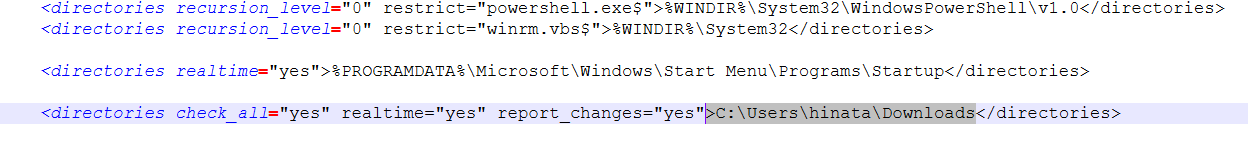
* Trên Endpoint Window

Tìm kiếm <syscheck> khối trong đại lý Wazuh tập tin. Hãy chắc chắn rằng được đặt thành . Điều này cho phép mô-đun Wazuh FIM giám sát các thay đổi thư mục. **C:\Program Files (x86)\ossec-agent\ossec.conf<disabled>no**

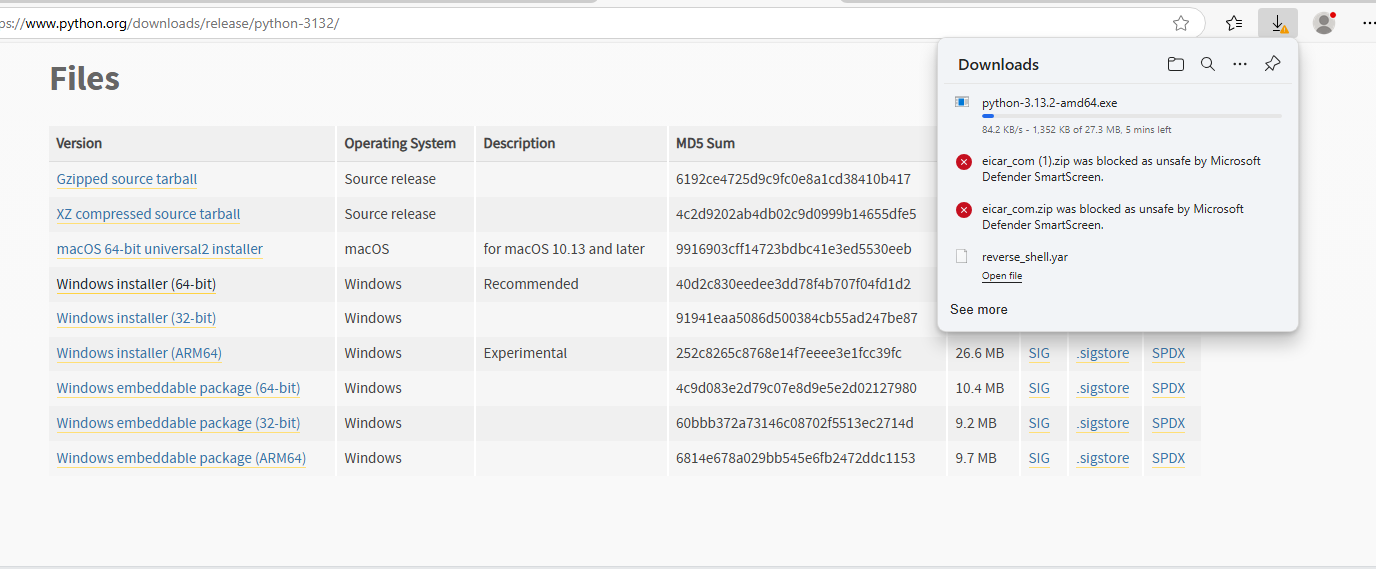


Thêm một mục trong <syscheck> chặn để cấu hình một thư mục sẽ được theo dõi trong thời gian gần. Trong trường hợp sử dụng này, bạn định cấu hình Wazuh để theo dõi C:\Users\<USER\_NAME>\Downloads thư mục. Thay thế <USER\_NAME> biến với tên người dùng thích hợp:

**<directories realtime="yes">C:\Users\<USER\_NAME>\Downloads</directories>**

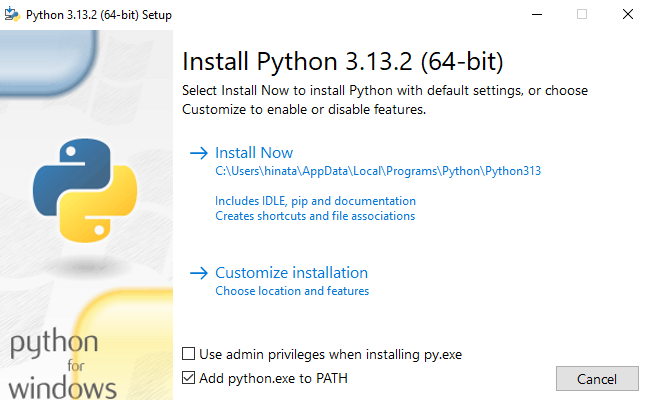
****

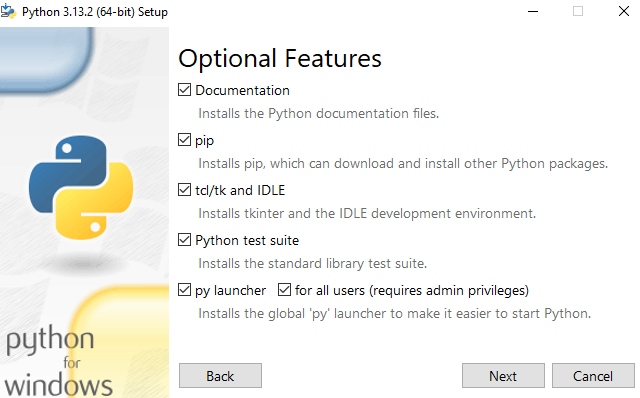
Tải xuống trình cài đặt thực thi Python từ [trang web Python chính thức](https://www.python.org/downloads/windows/).



Chạy trình cài đặt Python sau khi tải xuống. Đảm bảo kiểm tra các hộp sau:

* + Install launcher for all users
  + Add Python 3.X to PATH(Điều này đặt trình thông dịch trong đường dẫn thực hiện)



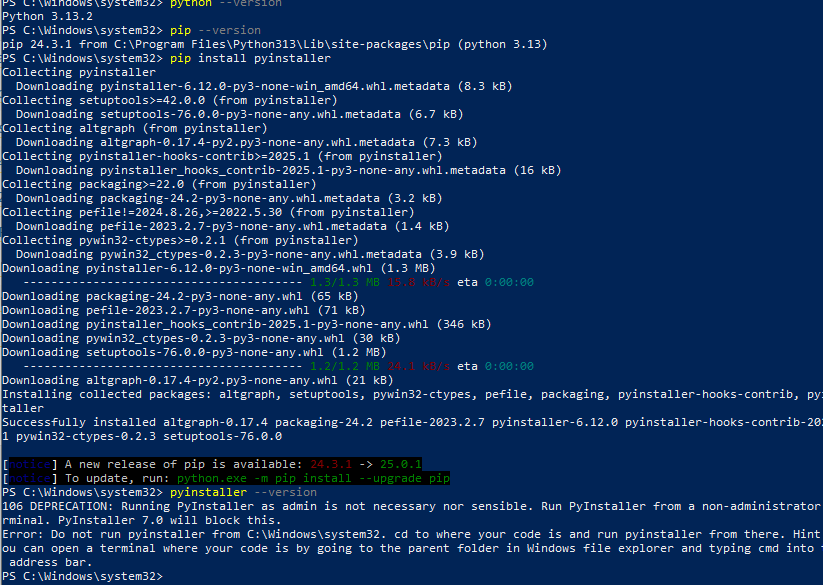


Khi Python hoàn tất quá trình cài đặt, hãy mở thiết bị đầu cuối PowerShell của quản trị viên và sử dụng pip để cài đặt PyInstaller:

**pip install pyinstaller**

**pyinstaller –version**

*Sử dụng Pyinstaller ở đây để chuyển đổi tập lệnh Python phản hồi hoạt động thành một ứng dụng thực thi có thể chạy trên điểm cuối Windows.*

**

Tạo tập lệnh phản hồi hoạt động remove-threat.py để xóa một tệp khỏi điểm cuối Windows:

#!/usr/bin/python3

# Copyright (C) 2015-2022, Wazuh Inc.

# All rights reserved.

import os

import sys

import json

import datetime

if os.name == 'nt':

    LOG\_FILE = "C:\\Program Files (x86)\\ossec-agent\\active-response\\active-responses.log"

else:

    LOG\_FILE = "/var/ossec/logs/active-responses.log"

ADD\_COMMAND = 0

DELETE\_COMMAND = 1

CONTINUE\_COMMAND = 2

ABORT\_COMMAND = 3

OS\_SUCCESS = 0

OS\_INVALID = -1

class message:

    def \_\_init\_\_(self):

        self.alert = ""

        self.command = 0

def write\_debug\_file(ar\_name, msg):

    with open(LOG\_FILE, mode="a") as log\_file:

        log\_file.write(str(datetime.datetime.now().strftime('%Y/%m/%d %H:%M:%S')) + " " + ar\_name + ": " + msg +"\n")

def setup\_and\_check\_message(argv):

    # get alert from stdin

    input\_str = ""

    for line in sys.stdin:

        input\_str = line

        break

    try:

        data = json.loads(input\_str)

    except ValueError:

        write\_debug\_file(argv[0], 'Decoding JSON has failed, invalid input format')

        message.command = OS\_INVALID

        return message

    message.alert = data

    command = data.get("command")

    if command == "add":

        message.command = ADD\_COMMAND

    elif command == "delete":

        message.command = DELETE\_COMMAND

    else:

        message.command = OS\_INVALID

        write\_debug\_file(argv[0], 'Not valid command: ' + command)

    return message

def send\_keys\_and\_check\_message(argv, keys):

    # build and send message with keys

    keys\_msg = json.dumps({"version": 1,"origin":{"name": argv[0],"module":"active-response"},"command":"check\_keys","parameters":{"keys":keys}})

    write\_debug\_file(argv[0], keys\_msg)

    print(keys\_msg)

    sys.stdout.flush()

    # read the response of previous message

    input\_str = ""

    while True:

        line = sys.stdin.readline()

        if line:

            input\_str = line

            break

    # write\_debug\_file(argv[0], input\_str)

    try:

        data = json.loads(input\_str)

    except ValueError:

        write\_debug\_file(argv[0], 'Decoding JSON has failed, invalid input format')

        return message

    action = data.get("command")

    if "continue" == action:

        ret = CONTINUE\_COMMAND

    elif "abort" == action:

        ret = ABORT\_COMMAND

    else:

        ret = OS\_INVALID

        write\_debug\_file(argv[0], "Invalid value of 'command'")

    return ret

def main(argv):

    write\_debug\_file(argv[0], "Started")

    # validate json and get command

    msg = setup\_and\_check\_message(argv)

    if msg.command < 0:

        sys.exit(OS\_INVALID)

    if msg.command == ADD\_COMMAND:

        alert = msg.alert["parameters"]["alert"]

        keys = [alert["rule"]["id"]]

        action = send\_keys\_and\_check\_message(argv, keys)

        # if necessary, abort execution

        if action != CONTINUE\_COMMAND:

            if action == ABORT\_COMMAND:

                write\_debug\_file(argv[0], "Aborted")

                sys.exit(OS\_SUCCESS)

            else:

                write\_debug\_file(argv[0], "Invalid command")

                sys.exit(OS\_INVALID)

        try:

            file\_path = msg.alert["parameters"]["alert"]["data"]["virustotal"]["source"]["file"]

            if os.path.exists(file\_path):

                os.remove(file\_path)

            write\_debug\_file(argv[0], json.dumps(msg.alert) + " Successfully removed threat")

        except OSError as error:

            write\_debug\_file(argv[0], json.dumps(msg.alert) + "Error removing threat")

    else:

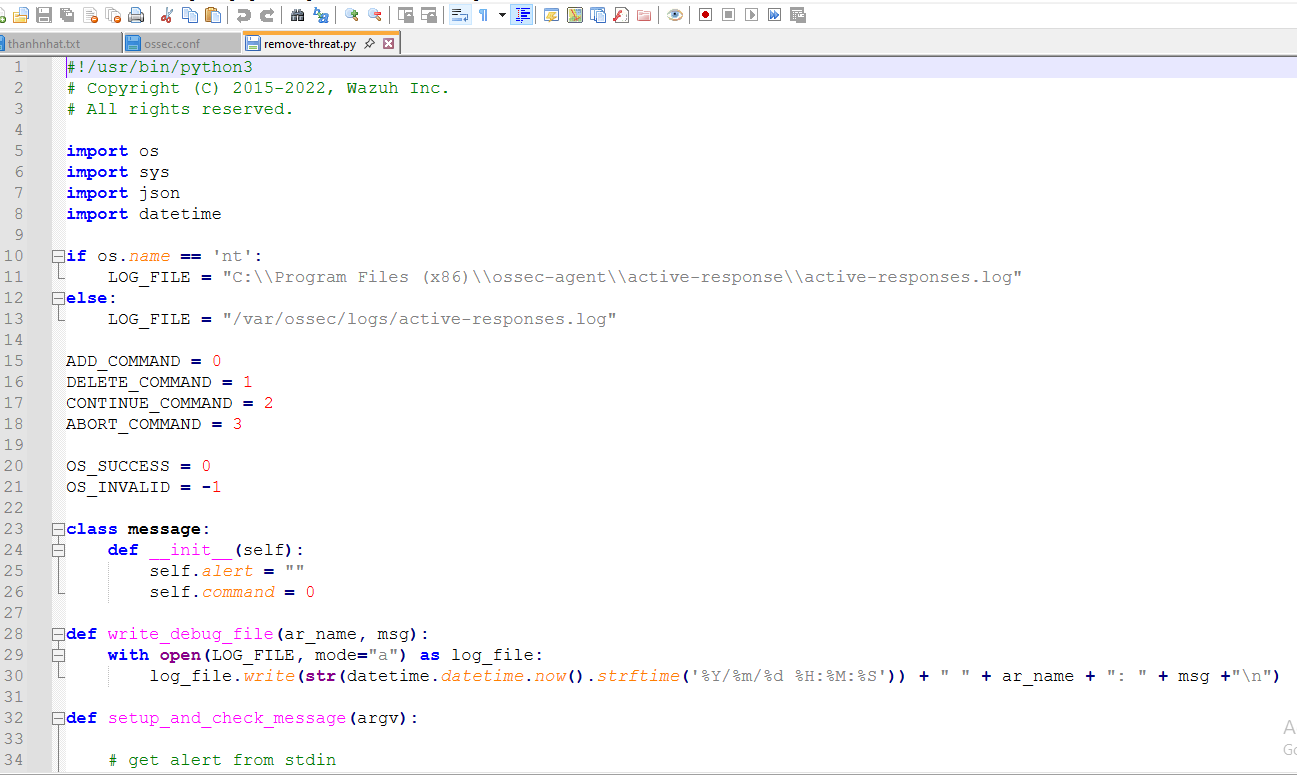
        write\_debug\_file(argv[0], "Invalid command")

    write\_debug\_file(argv[0], "Ended")

    sys.exit(OS\_SUCCESS)

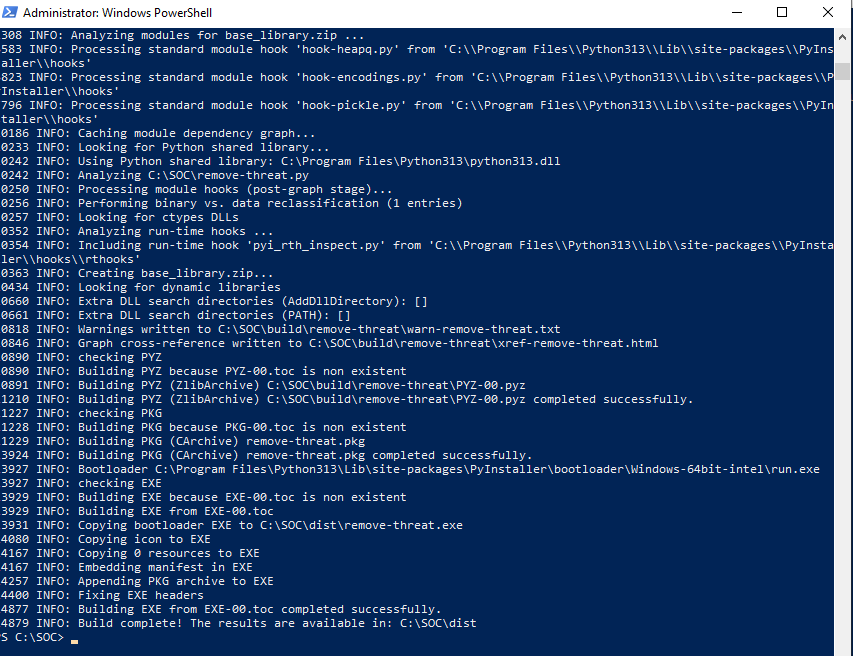
if \_\_name\_\_ == "\_\_main\_\_":

    main(sys.argv)



Chuyển đổi tập lệnh Python phản hồi hoạt động remove-threat.py đến một ứng dụng thực thi Windows. Chạy lệnh PowerShell sau đây với tư cách quản trị viên để tạo tệp thực thi:

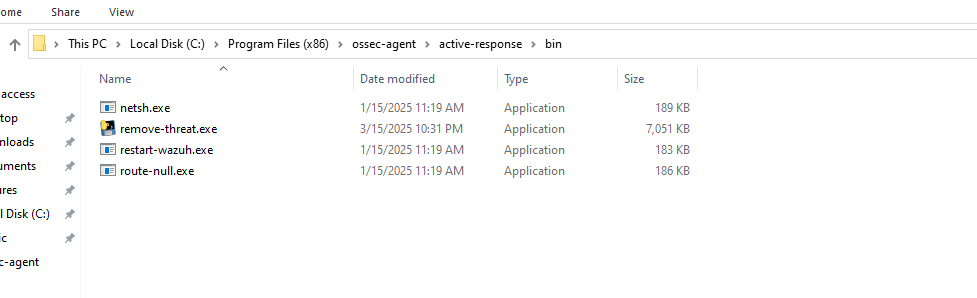
**pyinstaller -F remove-threat.py**

****

Hãy lưu ý về con đường nơi pyinstaller tạo ra remove-threat.exe.

Build complete! The results are available in: **C:\SOC\dist**

Di chuyển tập tin thực thi remove-threat.exe đến thư mục. **C:\Program Files (x86)\ossec-agent\active-response\bin**

****

Khởi động lại tác nhân Wazuh để áp dụng các thay đổi. Chạy lệnh PowerShell sau đây với tư cách quản trị viên:

**Restart-Service -Name wazuh**

* Trên Wazuh Server:

Thêm cấu hình sau vào /var/ossec/etc/ossec.conf tệp trên máy chủ Wazuh để kích hoạt tích hợp VirusTotal. Thay thế <YOUR\_VIRUS\_TOTAL\_API\_KEY> với bạn [Khóa API VirusTotal](https://developers.virustotal.com/reference). Điều này cho phép kích hoạt truy vấn VirusTotal bất cứ khi nào có bất kỳ quy tắc nào trong FIM syscheck nhóm được kích hoạt:

<ossec\_config>

  <integration>

    <name>virustotal</name>

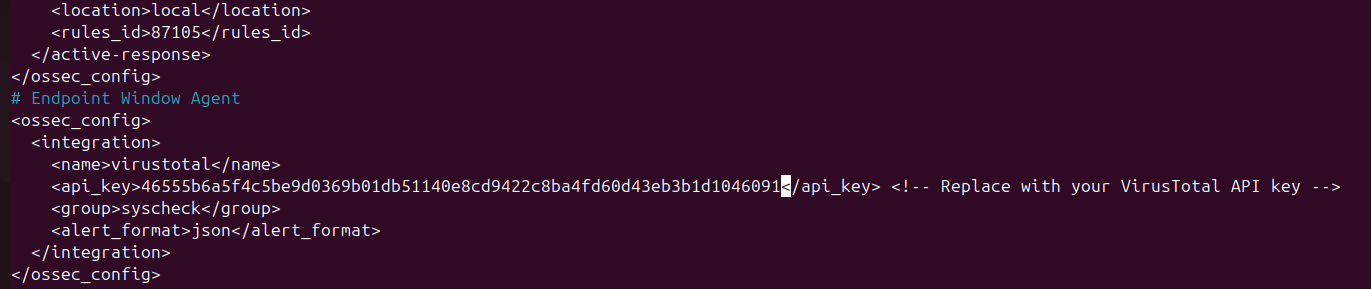
    <api\_key><YOUR\_VIRUS\_TOTAL\_API\_KEY></api\_key> <!-- Replace with your VirusTotal API key -->

    <group>syscheck</group>

    <alert\_format>json</alert\_format>

  </integration>

</ossec\_config>



Nối các khối sau vào máy chủ Wazuh /var/ossec/etc/ossec.conf tập tin. Điều này cho phép Phản hồi chủ động và kích hoạt remove-threat.exe thực thi khi truy vấn VirusTotal trả về các kết quả khớp dương cho các mối đe dọa:

<ossec\_config>

  <command>

    <name>remove-threat</name>

    <executable>remove-threat.exe</executable>

    <timeout\_allowed>no</timeout\_allowed>

  </command>

  <active-response>

    <disabled>no</disabled>

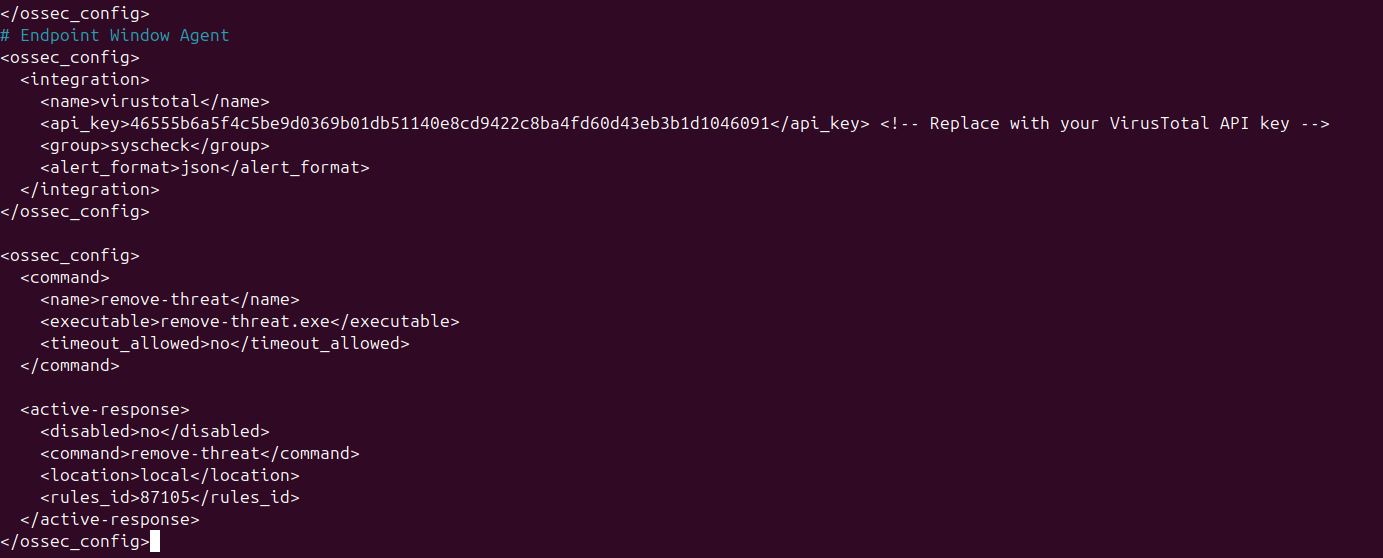
    <command>remove-threat</command>

    <location>local</location>

    <rules\_id>87105</rules\_id>

  </active-response>

</ossec\_config>



Thêm các quy tắc sau vào máy chủ Wazuh /var/ossec/etc/rules/local\_rules.xml tập tin để cảnh báo về kết quả Phản hồi chủ động.

<group name="virustotal,">

  <rule id="100092" level="12">

      <if\_sid>657</if\_sid>

      <match>Successfully removed threat</match>

      <description>$(parameters.program) removed threat located at $(parameters.alert.data.virustotal.source.file)</description>

  </rule>

  <rule id="100093" level="12">

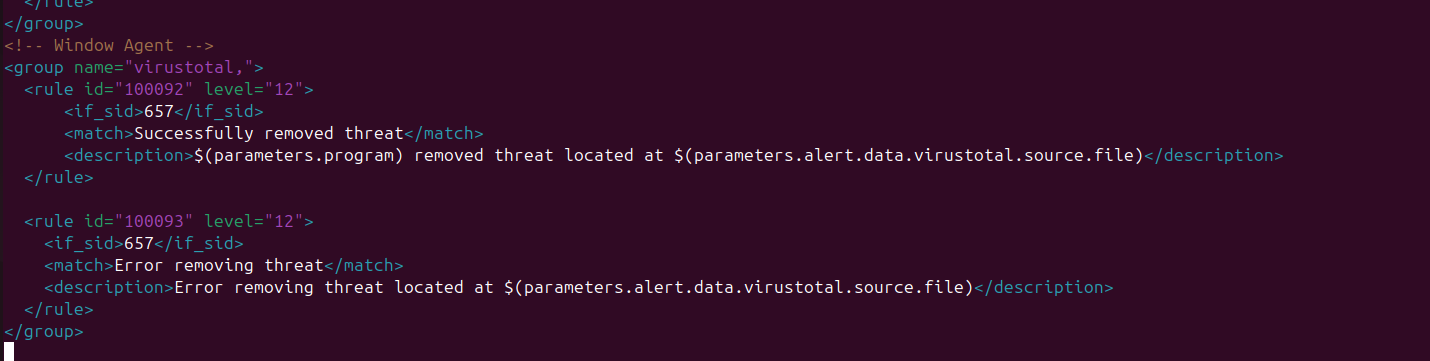
    <if\_sid>657</if\_sid>

    <match>Error removing threat</match>

    <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>

  </rule>

</group>



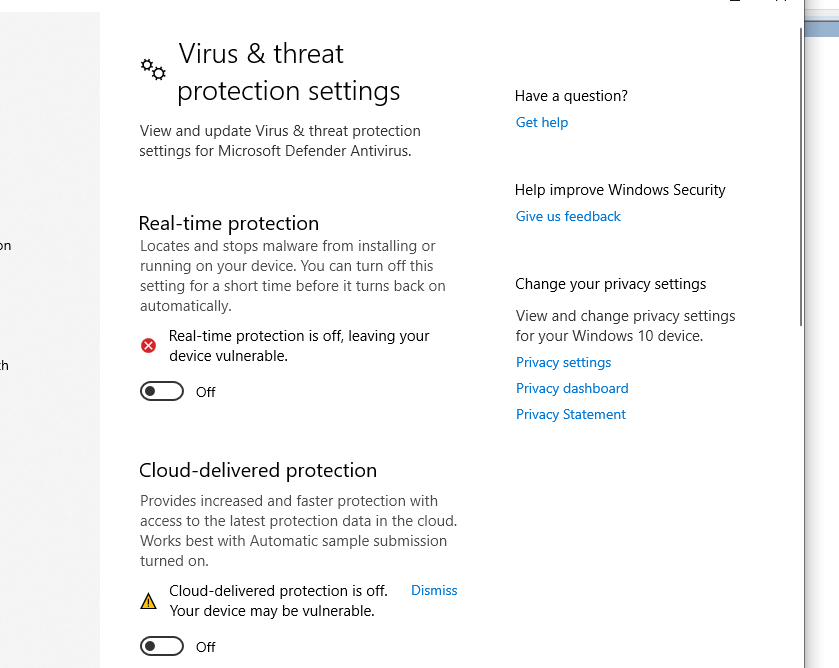
Khởi động lại trình quản lý Wazuh để áp dụng các thay đổi cấu hình:

**sudo systemctl restart wazuh-manager**

* Thực thi:

***Thực hiện theo các bước tiếp theo để tạm thời tắt bảo vệ chống vi-rút Microsoft Defender thời gian thực trong Windows Security:***

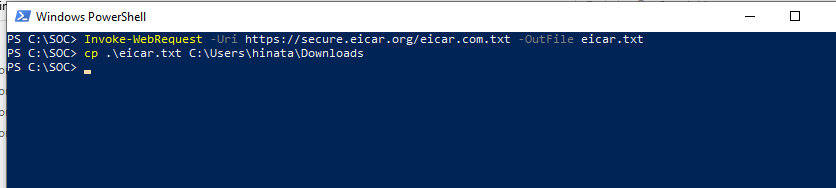
* ***Nhấp vào Bắt đầu thực đơn và loại để tìm kiếm ứng dụng đó. Windows Security***
* ***Chọn Ứng dụng Windows Security từ kết quả, đi đến Bảo vệ virus và đe dọavà dưới Cài đặt bảo vệ virus và mối đe dọa chọn Quản lý cài đặt.***
* ***Công tắc Bảo vệ thời gian thực để Tắt.***



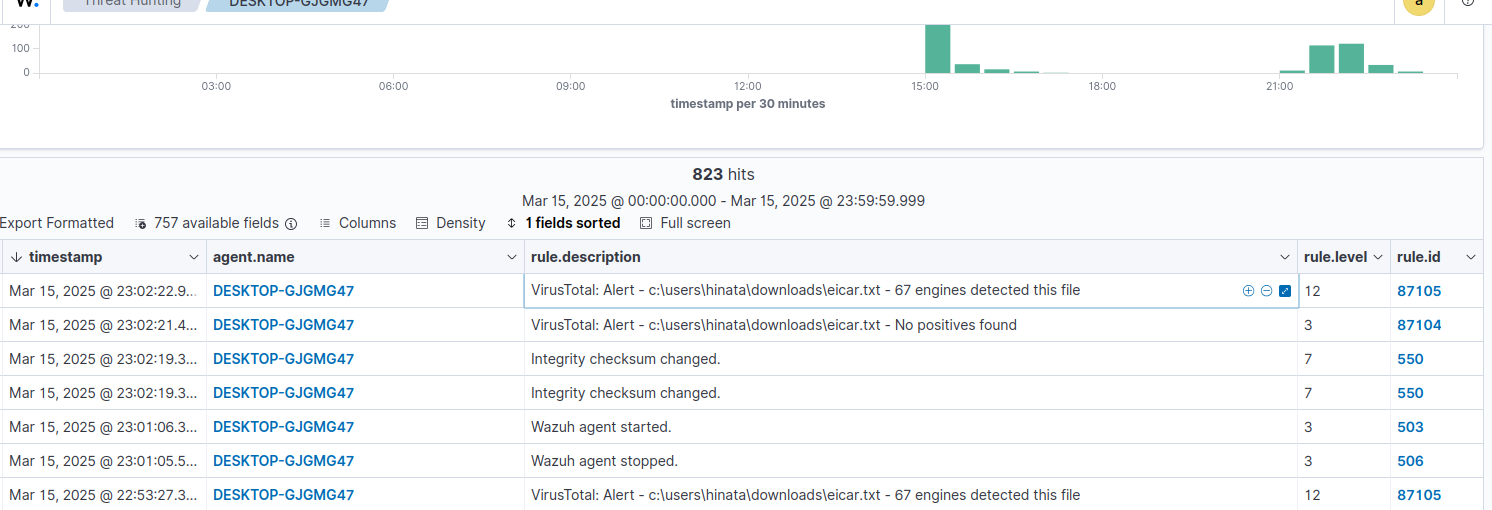
Tải xuống [Kiểm tra EICAR](https://secure.eicar.org/eicar.com.txt) nộp vào C:\Users\hinata\Downloads thư mục trên điểm cuối Windows.

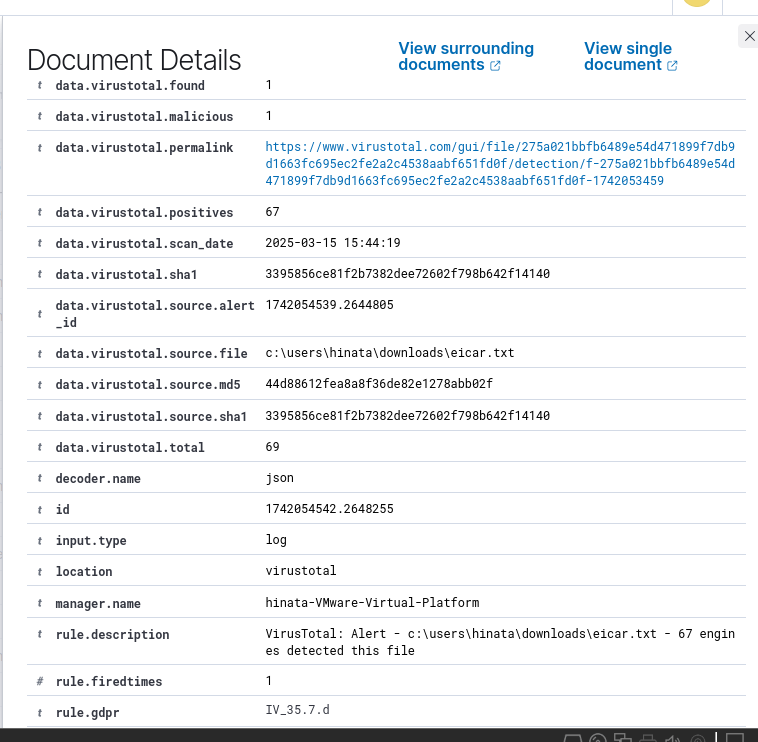
**Invoke-WebRequest -Uri https://secure.eicar.org/eicar.com.txt -OutFile eicar.txt**

**cp .\eicar.txt C:\Users\hinata\Downloads**

****

* Kết quả trên Wazuh:

****

****

* 1. **Agent Linux Ubuntu**
* Trên agent:

Mở file */var/ossec/etc/ossec.conf.*

Thêm một mục trong <syscheck> chặn để cấu hình một thư mục sẽ được theo dõi trong thời gian gần. Trong trường hợp này, bạn đang theo dõi /root thư mục:

**<directories realtime="yes">/root</directories>**



Cài đặt jq, một tiện ích xử lý đầu vào JSON từ tập lệnh phản hồi hoạt động.

**sudo apt update**

**sudo apt -y install jq**

Tạo /var/ossec/active-response/bin/remove-threat.sh tập lệnh phản hồi hoạt động để xóa các tệp độc hại khỏi điểm cuối:

#!/bin/bash

LOCAL=`dirname $0`;

cd $LOCAL

cd ../

PWD=`pwd`

read INPUT\_JSON

FILENAME=$(echo $INPUT\_JSON | jq -r .parameters.alert.data.virustotal.source.file)

COMMAND=$(echo $INPUT\_JSON | jq -r .command)

LOG\_FILE="${PWD}/../logs/active-responses.log"

#------------------------ Analyze command -------------------------#

if [ ${COMMAND} = "add" ]

then

 # Send control message to execd

 printf '{"version":1,"origin":{"name":"remove-threat","module":"active-response"},"command":"check\_keys", "parameters":{"keys":[]}}\n'

 read RESPONSE

 COMMAND2=$(echo $RESPONSE | jq -r .command)

 if [ ${COMMAND2} != "continue" ]

 then

  echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT\_JSON Remove threat active response aborted" >> ${LOG\_FILE}

  exit 0;

 fi

fi

# Removing file

rm -f $FILENAME

if [ $? -eq 0 ]; then

 echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT\_JSON Successfully removed threat" >> ${LOG\_FILE}

else

 echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT\_JSON Error removing threat" >> ${LOG\_FILE}

fi

exit 0;

Thay đổi /var/ossec/active-response/bin/remove-threat.sh quyền sở hữu tập tin và quyền:

**sudo chmod 750 /var/ossec/active-response/bin/remove-threat.sh**

**sudo chown root:wazuh /var/ossec/active-response/bin/remove-threat.sh**

Khởi động lại tác nhân Wazuh để áp dụng các thay đổi:

**sudo systemctl restart wazuh-agent**

* Trên wazuh server:

Thêm các quy tắc sau vào /var/ossec/etc/rules/local\_rules.xml tập tin trên máy chủ Wazuh. Các quy tắc này cảnh báo về những thay đổi trong /root thư mục được phát hiện bởi quét FIM:

<group name="syscheck,pci\_dss\_11.5,nist\_800\_53\_SI.7,">

    <!-- Rules for Linux systems -->

    <rule id="100200" level="7">

        <if\_sid>550</if\_sid>

        <field name="file">/root</field>

        <description>File modified in /root directory.</description>

    </rule>

    <rule id="100201" level="7">

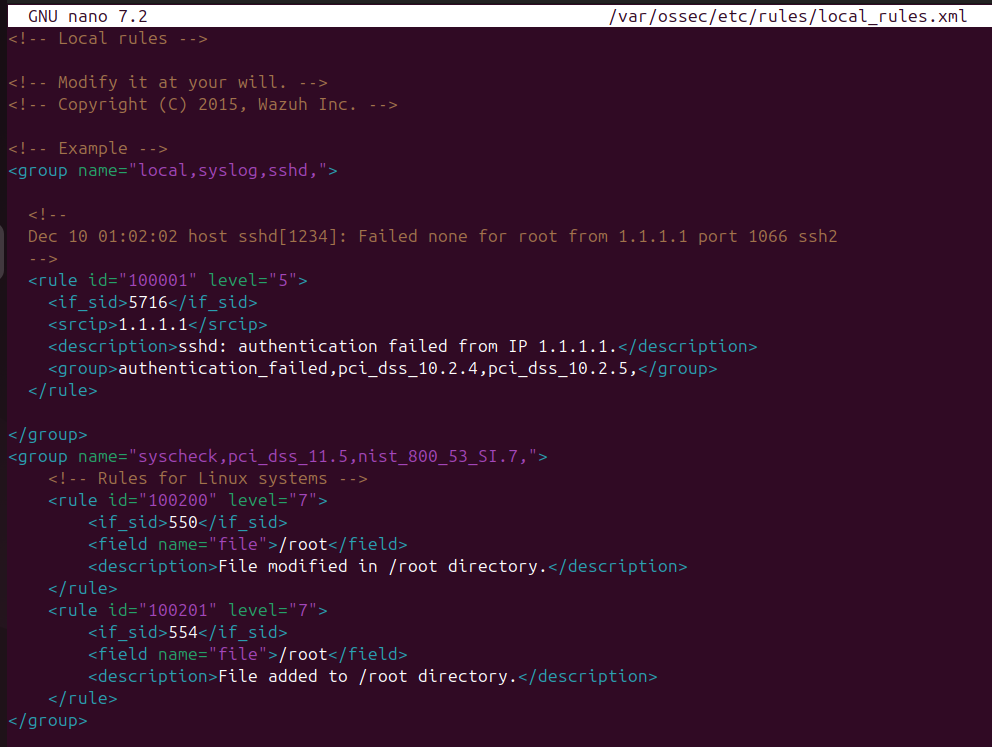
        <if\_sid>554</if\_sid>

        <field name="file">/root</field>

        <description>File added to /root directory.</description>

    </rule>

</group>



Thêm cấu hình sau vào máy chủ Wazuh /var/ossec/etc/ossec.conf tập tin để kích hoạt tích hợp Virustotal. Thay thế <YOUR\_VIRUS\_TOTAL\_API\_KEY> với bạn [Khóa API VirusTotal](https://developers.virustotal.com/reference). Điều này cho phép kích hoạt truy vấn VirusTotal bất cứ khi nào có quy tắc 100200 và 100201 được kích hoạt: 46555b6a5f4c5be9d0369b01db51140e8cd9422c8ba4fd60d43eb3b1d1046091

<ossec\_config>

  <integration>

    <name>virustotal</name>

    <api\_key><YOUR\_VIRUS\_TOTAL\_API\_KEY></api\_key> <!-- Replace with your VirusTotal API key -->

    <rule\_id>100200,100201</rule\_id>

    <alert\_format>json</alert\_format>

  </integration>

</ossec\_config>

Nối các khối sau vào máy chủ Wazuh /var/ossec/etc/ossec.conf tập tin. Điều này cho phép Phản hồi chủ động và kích hoạt remove-threat.sh tập lệnh khi VirusTotal gắn cờ một tệp là độc hại:

<ossec\_config>

  <command>

    <name>remove-threat</name>

    <executable>remove-threat.sh</executable>

    <timeout\_allowed>no</timeout\_allowed>

  </command>

  <active-response>

    <disabled>no</disabled>

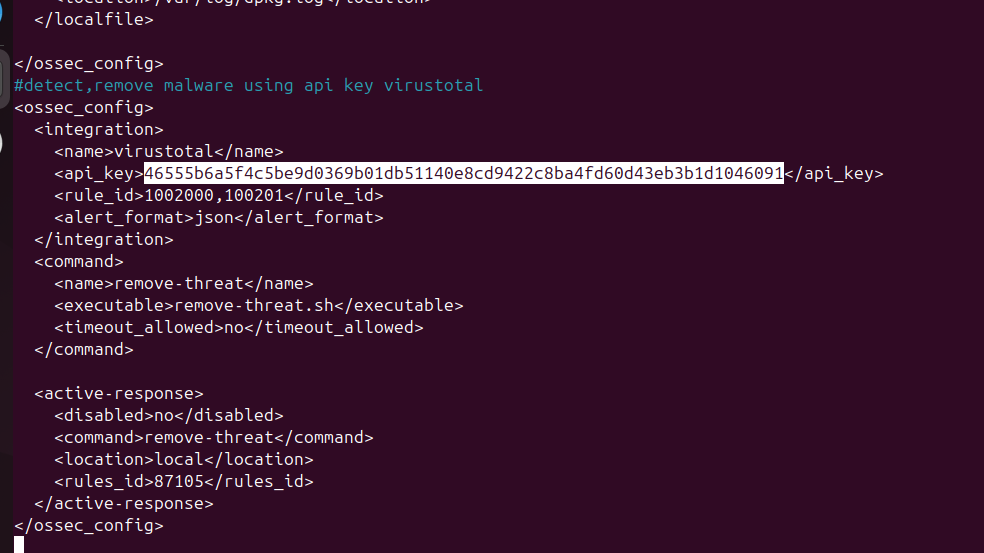
    <command>remove-threat</command>

    <location>local</location>

    <rules\_id>87105</rules\_id>

  </active-response>

</ossec\_config>



Thêm các quy tắc sau vào máy chủ Wazuh /var/ossec/etc/rules/local\_rules.xml tệp để cảnh báo về kết quả Phản hồi chủ động:

<group name="virustotal,">

  <rule id="100092" level="12">

    <if\_sid>657</if\_sid>

    <match>Successfully removed threat</match>

    <description>$(parameters.program) removed threat located at $(parameters.alert.data.virustotal.source.file)</description>

  </rule>

  <rule id="100093" level="12">

    <if\_sid>657</if\_sid>

    <match>Error removing threat</match>

    <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>

  </rule>

</group>

Khởi động lại trình quản lý Wazuh để áp dụng các thay đổi cấu hình:

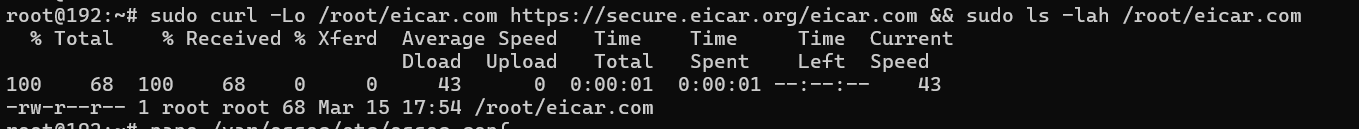
**sudo systemctl restart wazuh-manager**

****

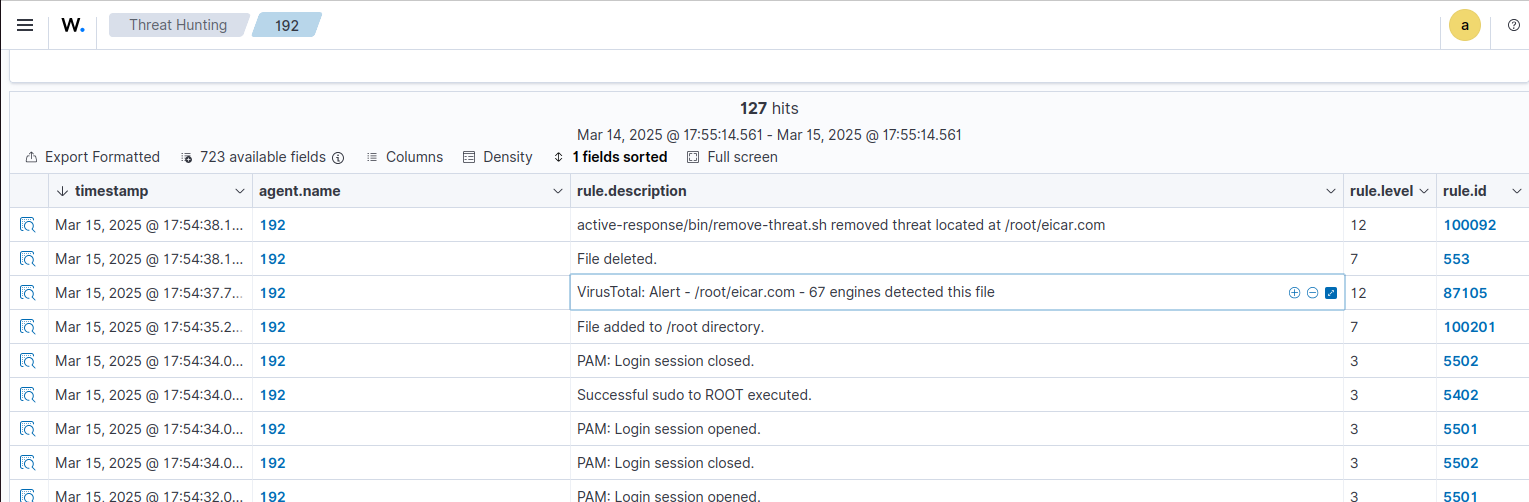
* Thực thi trên agent ubuntu:

**Tải xuống tệp kiểm tra EICAR vào /root thư mục trên điểm cuối Ubuntu:**

sudo curl -Lo /root/eicar.com https://secure.eicar.org/eicar.com && sudo ls -lah /root/eicar.com

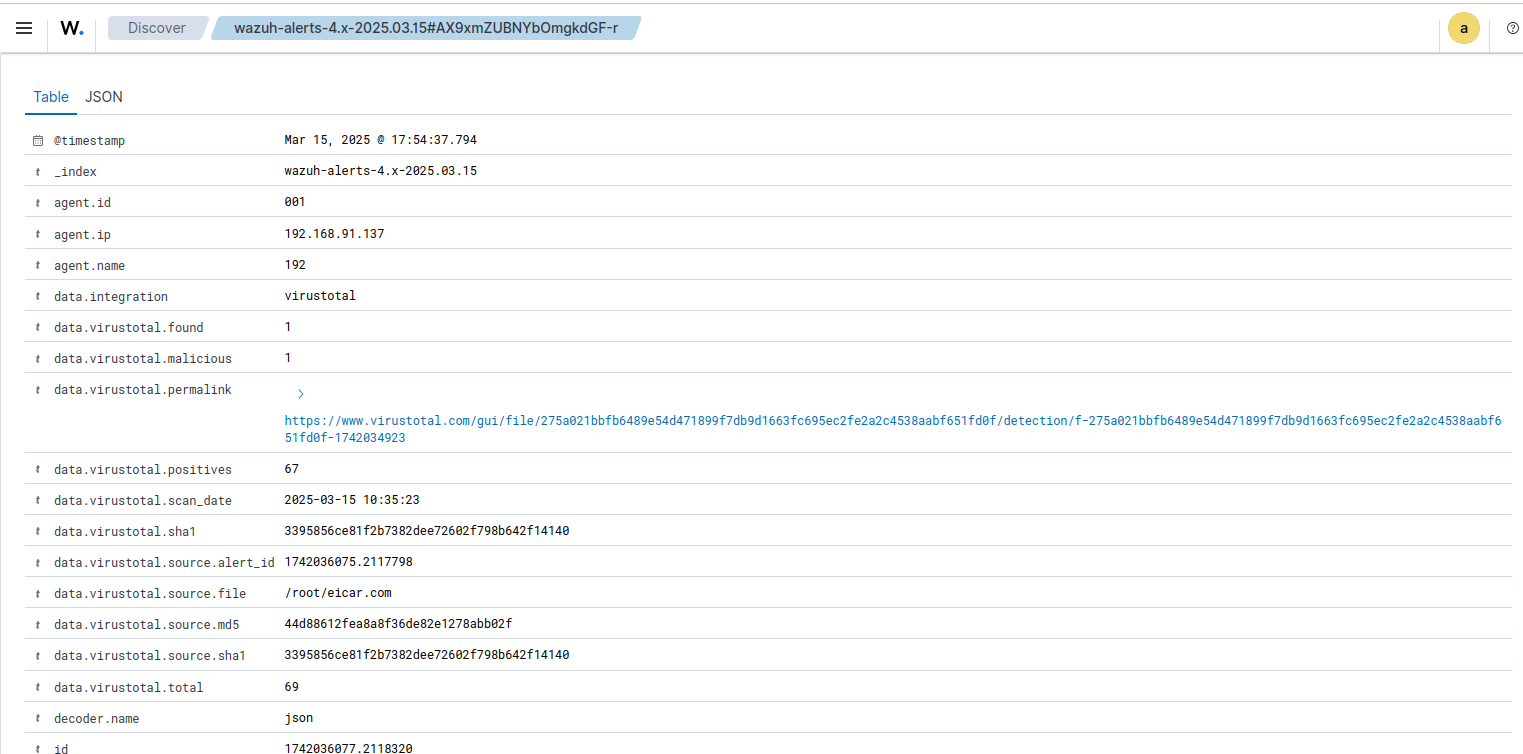
****

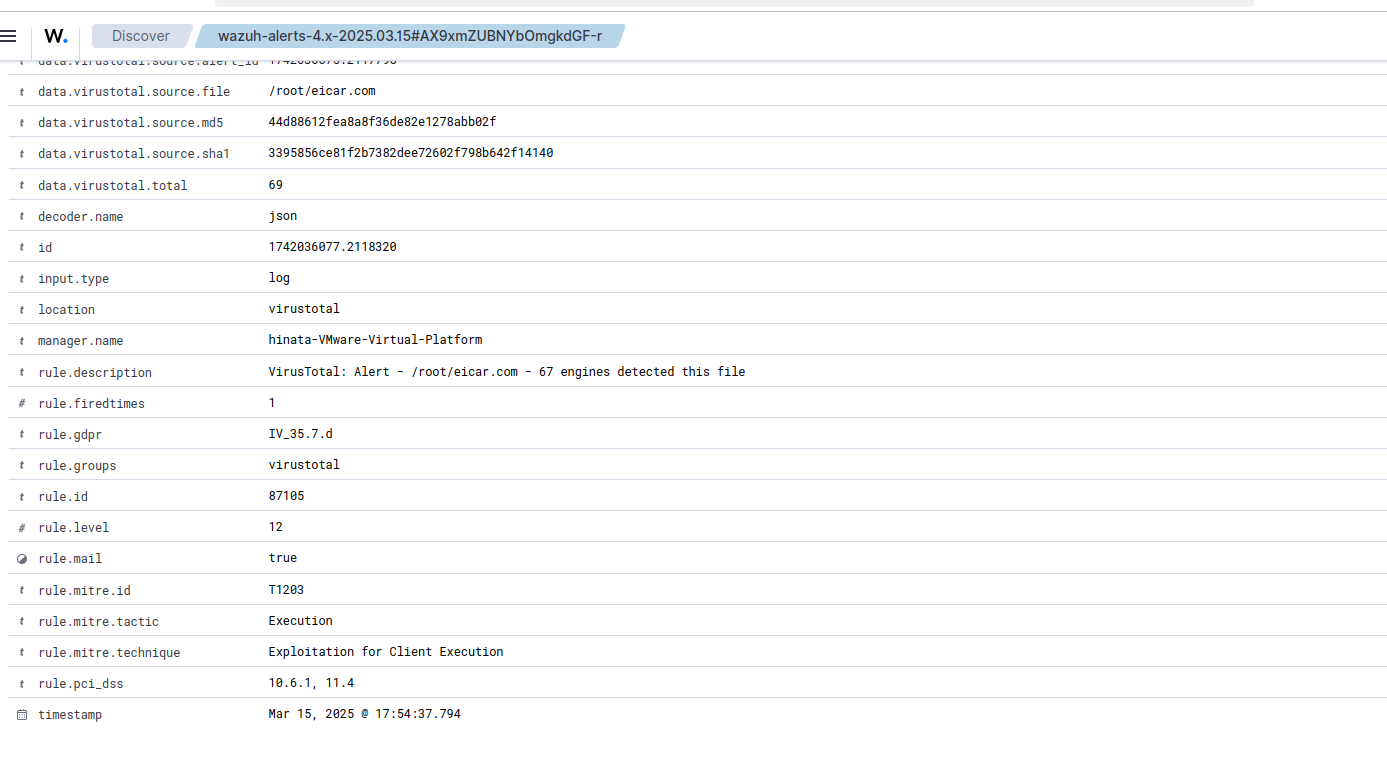
* Kết quả trên wazuh:

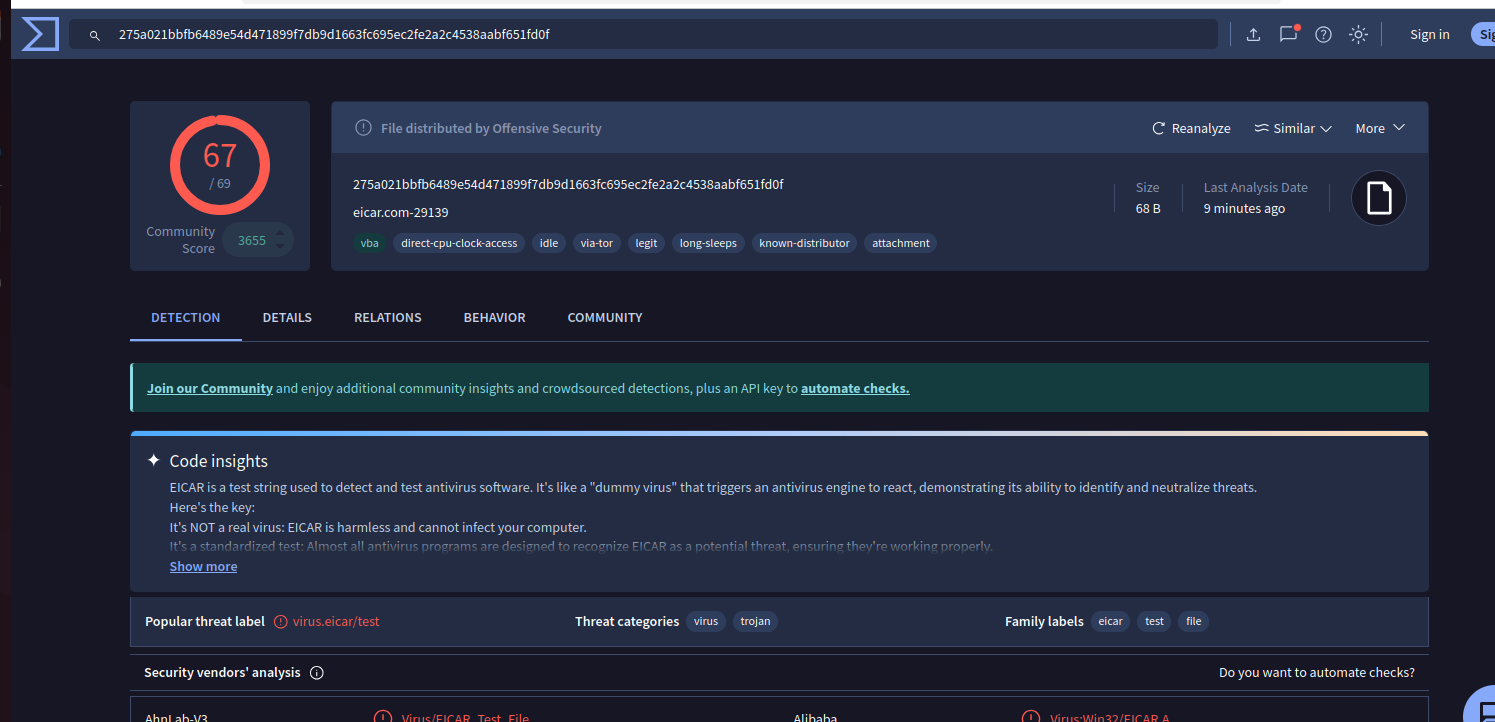


**Chi tiết:**

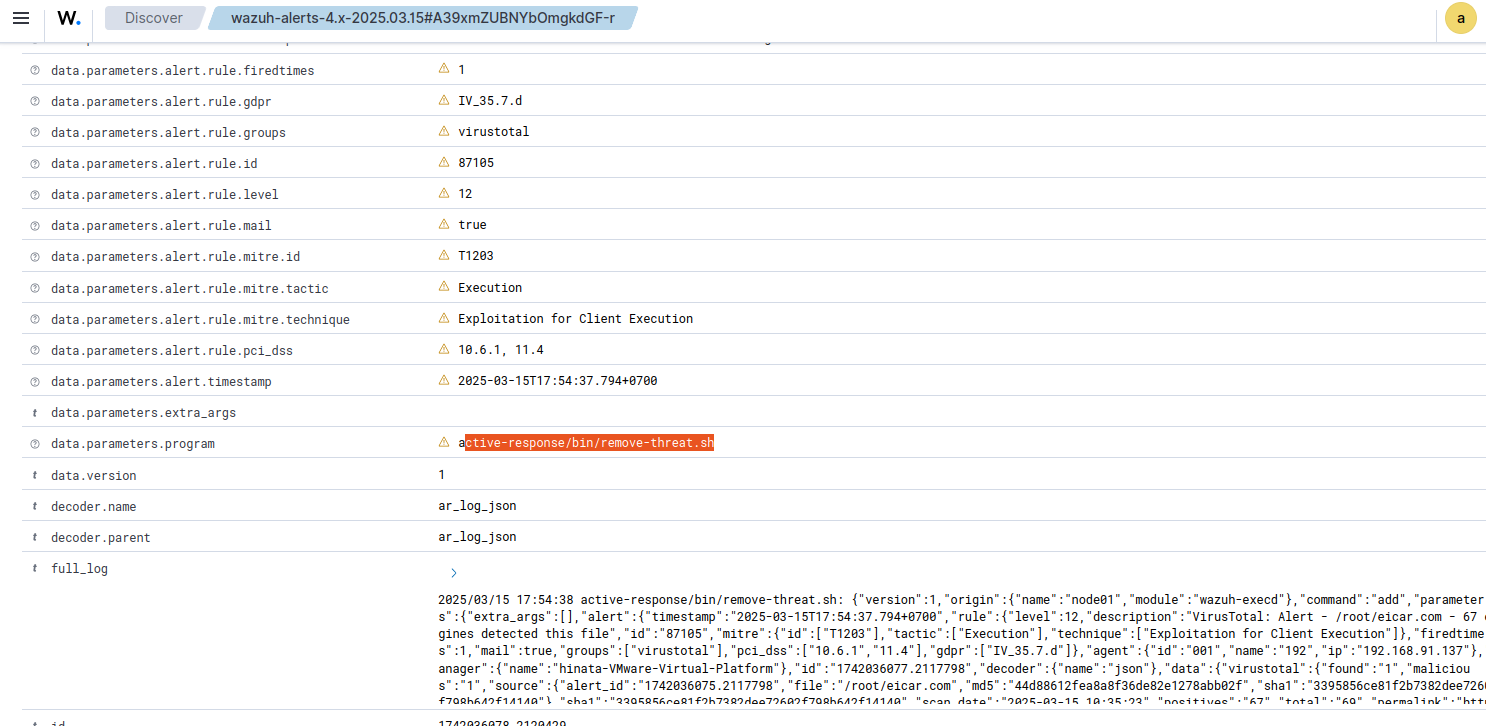
**Virustotal phát hiện:**



****

****

**File mã độc bị xóa:**

****

1. **Detecting an SQL injection attack**

* Trên Ubuntu Agent

Cập nhật các gói cục bộ và cài đặt máy chủ web Apache:

**sudo apt update**

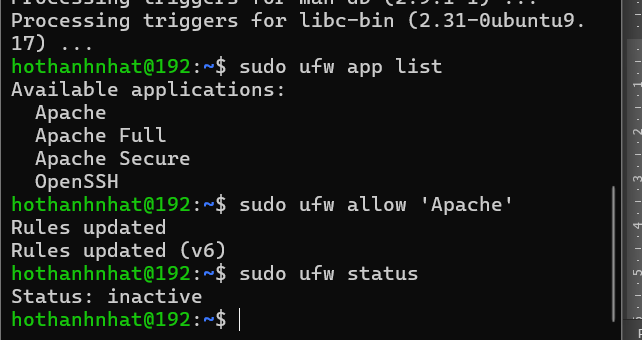
**sudo apt install apache2**

Nếu tường lửa được bật, hãy sửa đổi nó để cho phép truy cập bên ngoài vào các cổng web. Bỏ qua bước này nếu tường lửa bị tắt.

**sudo ufw app list**

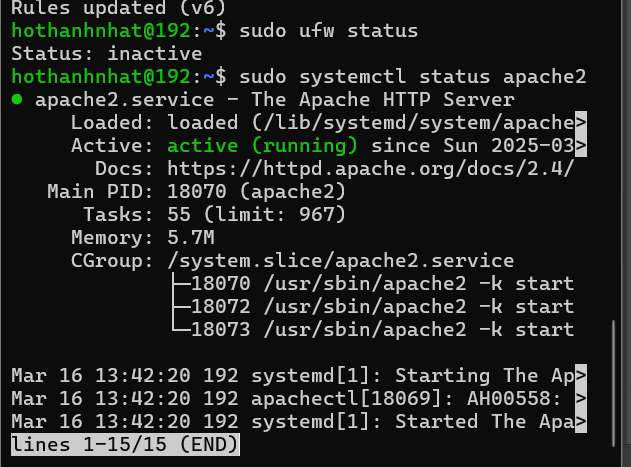
**sudo ufw allow 'Apache'**

**sudo ufw status**



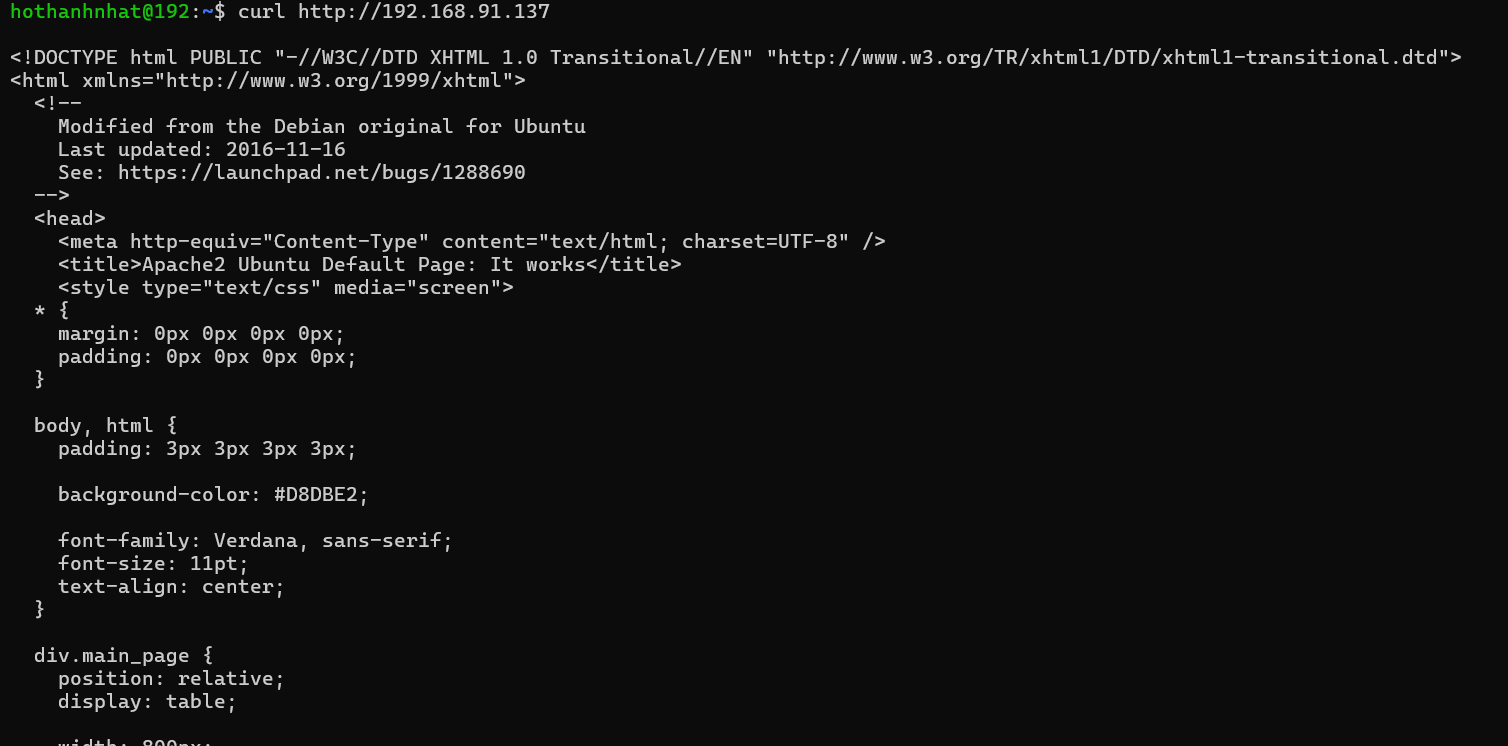
Kiểm tra trạng thái của dịch vụ Apache để xác minh rằng máy chủ web đang chạy:

**sudo systemctl status apache2**



Sử dụng curl lệnh hoặc mở http://<UBUNTU\_IP> trong trình duyệt để xem trang đích Apache và xác minh cài đặt:

**curl http://<UBUNTU\_IP> (192.168.91.137)**



Thêm các dòng sau vào tác nhân Wazuh /var/ossec/etc/ossec.conf tập tin. Điều này cho phép tác nhân Wazuh giám sát nhật ký truy cập của máy chủ Apache của bạn:

<ossec\_config>

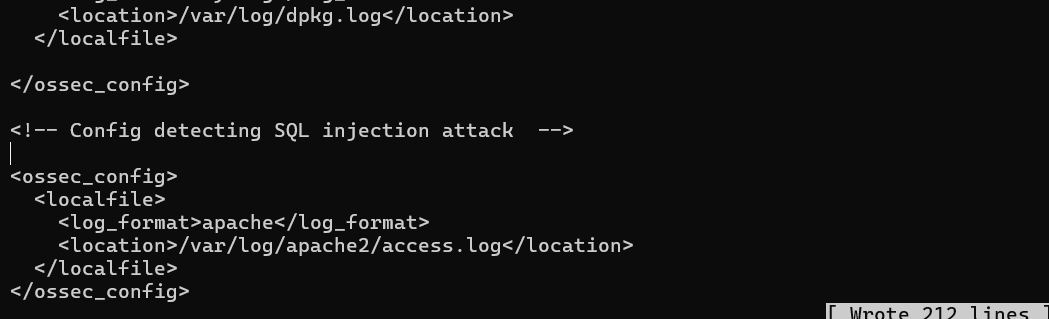
  <localfile>

    <log\_format>apache</log\_format>

    <location>/var/log/apache2/access.log</location>

  </localfile>

</ossec\_config>



Khởi động lại tác nhân Wazuh để áp dụng các thay đổi cấu hình:

**sudo systemctl restart wazuh-agent**

* Trên Attacker

Thay thế <UBUNTU\_IP> với địa chỉ IP thích hợp và thực hiện lệnh sau từ điểm cuối kẻ tấn công:

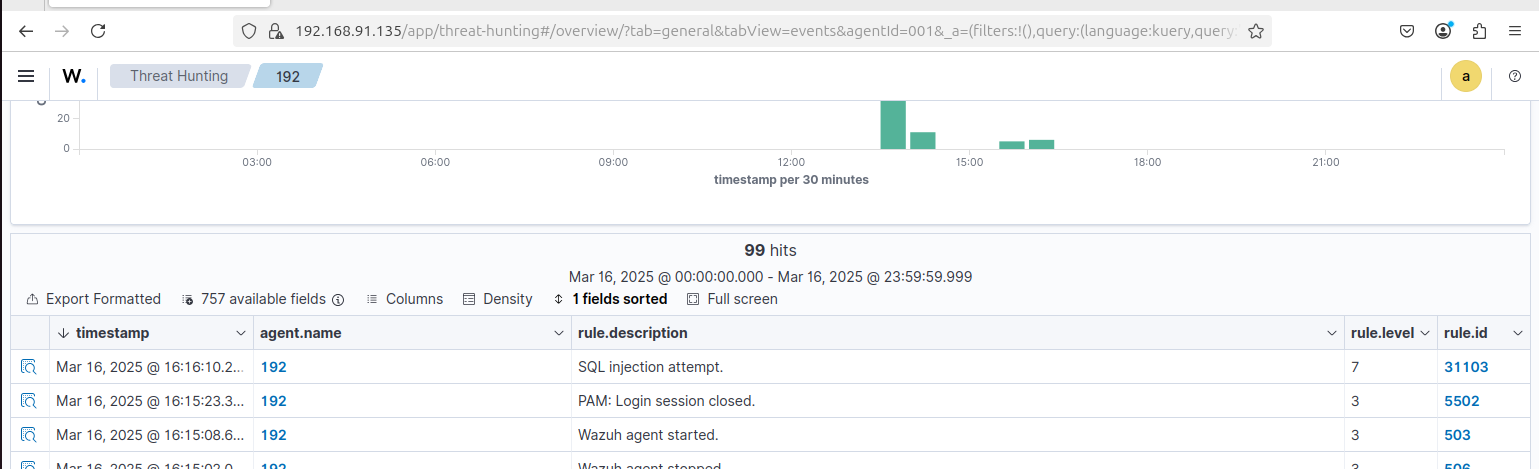
**curl -XGET "http://<UBUNTU\_IP>/users/?id=SELECT+\*+FROM+users";**

**🡪 curl -XGET "http://192.168.91.137/users/?id=SELECT+\*+FROM+users";**

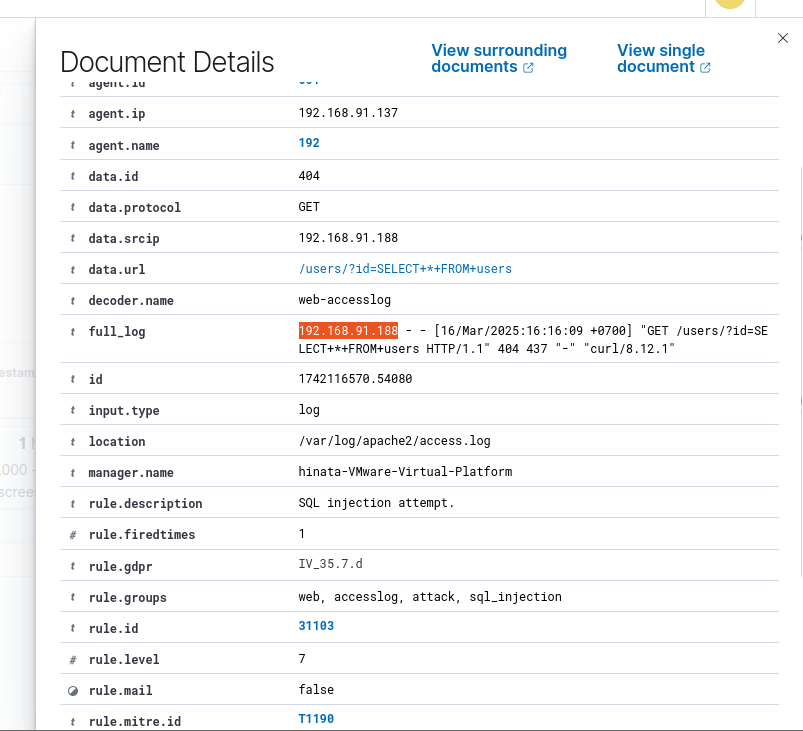
*Kết quả mong đợi ở đây là một cảnh báo với quy tắc ID 31103 nhưng một nỗ lực tiêm SQL thành công tạo ra một cảnh báo với quy tắc ID 31106.*

****

* Kết quả:



rule.id:31103:



1. **Detecting malware using YARA integration**
   1. **Cấu hình trên Ubuntu Agent**

* Trên Ubuntu Agent

Tải xuống, biên dịch và cài đặt YARA:

sudo apt update

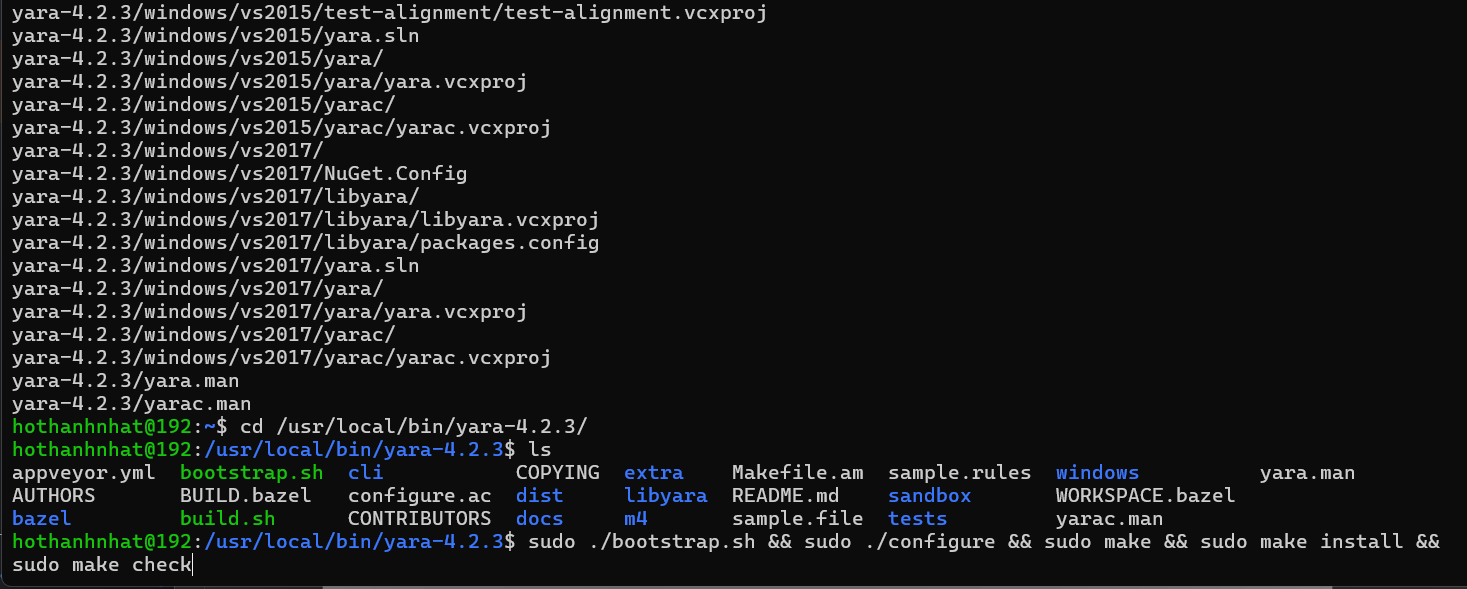
sudo apt install -y make gcc autoconf libtool libssl-dev pkg-config jq

sudo curl -LO https://github.com/VirusTotal/yara/archive/v4.2.3.tar.gz

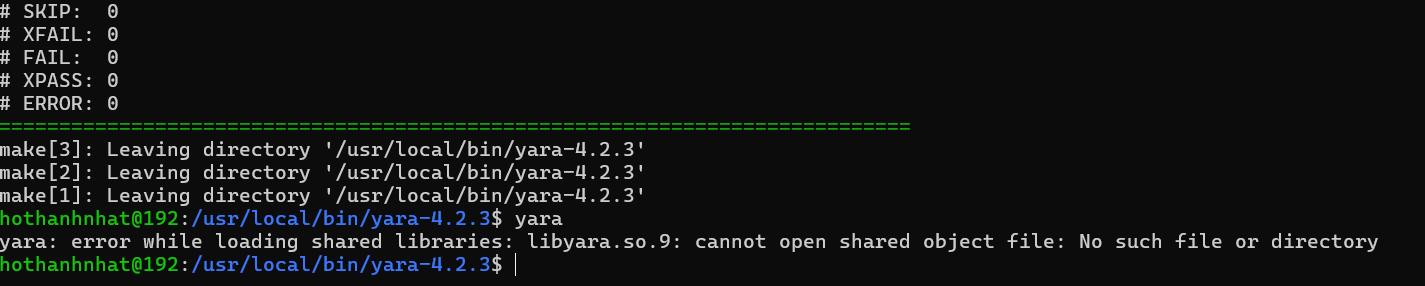
sudo tar -xvzf v4.2.3.tar.gz -C /usr/local/bin/ && rm -f v4.2.3.tar.gz

cd /usr/local/bin/yara-4.2.3/

sudo ./bootstrap.sh && sudo ./configure && sudo make && sudo make install && sudo make check



Kiểm tra xem YARA có chạy đúng không:



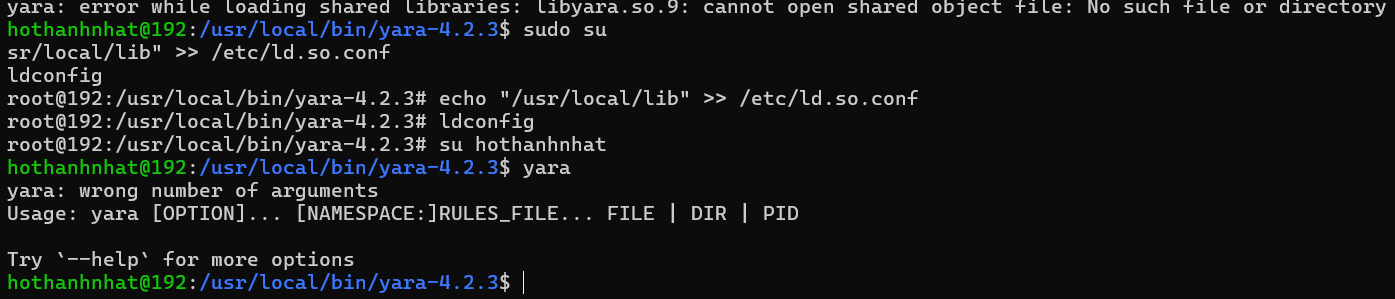
Nếu thông báo lỗi dưới đây được hiển thị:

Điều này có nghĩa là trình tải không tìm thấy libyara thư viện thường nằm trong /usr/local/lib. Thêm /usr/local/lib con đường đến /etc/ld.so.conf tập tin cấu hình trình tải để giải quyết điều này:

sudo su

echo "/usr/local/lib" >> /etc/ld.so.conf

ldconfig



Tải xuống quy tắc phát hiện YARA:

sudo mkdir -p /tmp/yara/rules

sudo curl 'https://valhalla.nextron-systems.com/api/v1/get' \

-H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8' \

-H 'Accept-Language: en-US,en;q=0.5' \

--compressed \

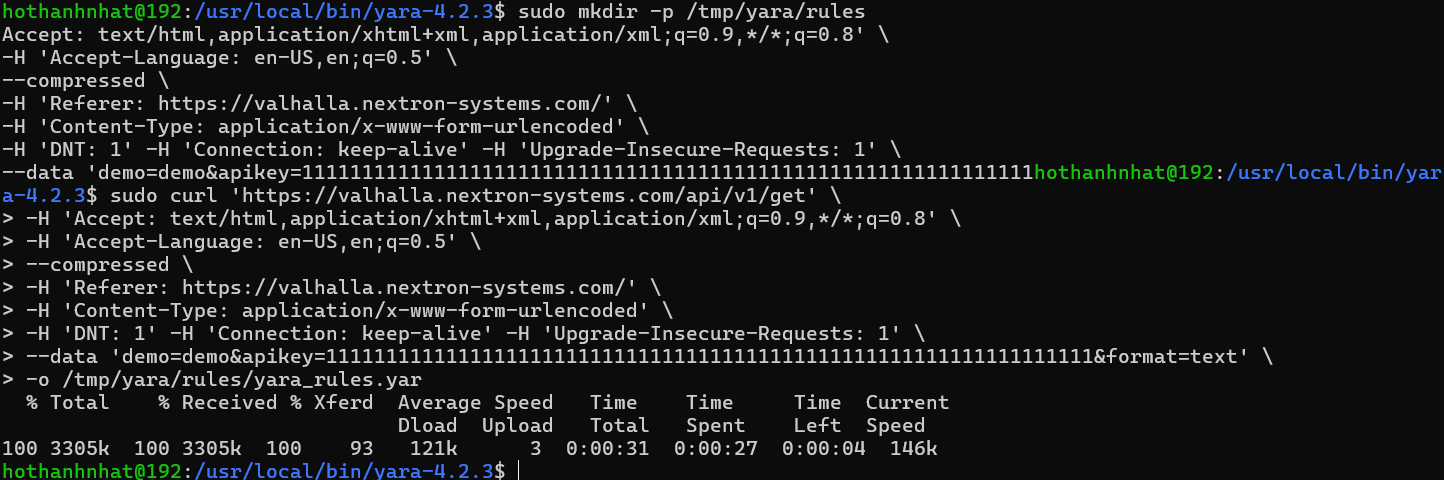
-H 'Referer: https://valhalla.nextron-systems.com/' \

-H 'Content-Type: application/x-www-form-urlencoded' \

-H 'DNT: 1' -H 'Connection: keep-alive' -H 'Upgrade-Insecure-Requests: 1' \

--data 'demo=demo&apikey=1111111111111111111111111111111111111111111111111111111111111111&format=text' \

-o /tmp/yara/rules/yara\_rules.yar



Tạo một *yara.sh* kịch bản trong /var/ossec/active-response/bin/ thư mục. Điều này là cần thiết cho việc quét Phản hồi chủ động của Wazuh-YARA:

#!/bin/bash

# Wazuh - Yara active response

# Copyright (C) 2015-2022, Wazuh Inc.

#

# This program is free software; you can redistribute it

# and/or modify it under the terms of the GNU General Public

# License (version 2) as published by the FSF - Free Software

# Foundation.

#------------------------- Gather parameters -------------------------#

# Extra arguments

read INPUT\_JSON

YARA\_PATH=$(echo $INPUT\_JSON | jq -r .parameters.extra\_args[1])

YARA\_RULES=$(echo $INPUT\_JSON | jq -r .parameters.extra\_args[3])

FILENAME=$(echo $INPUT\_JSON | jq -r .parameters.alert.syscheck.path)

# Set LOG\_FILE path

LOG\_FILE="logs/active-responses.log"

size=0

actual\_size=$(stat -c %s ${FILENAME})

while [ ${size} -ne ${actual\_size} ]; do

    sleep 1

    size=${actual\_size}

    actual\_size=$(stat -c %s ${FILENAME})

done

#----------------------- Analyze parameters -----------------------#

if [[ ! $YARA\_PATH ]] || [[ ! $YARA\_RULES ]]

then

    echo "wazuh-yara: ERROR - Yara active response error. Yara path and rules parameters are mandatory." >> ${LOG\_FILE}

    exit 1

fi

#------------------------- Main workflow --------------------------#

# Execute Yara scan on the specified filename

yara\_output="$("${YARA\_PATH}"/yara -w -r "$YARA\_RULES" "$FILENAME")"

if [[ $yara\_output != "" ]]

then

    # Iterate every detected rule and append it to the LOG\_FILE

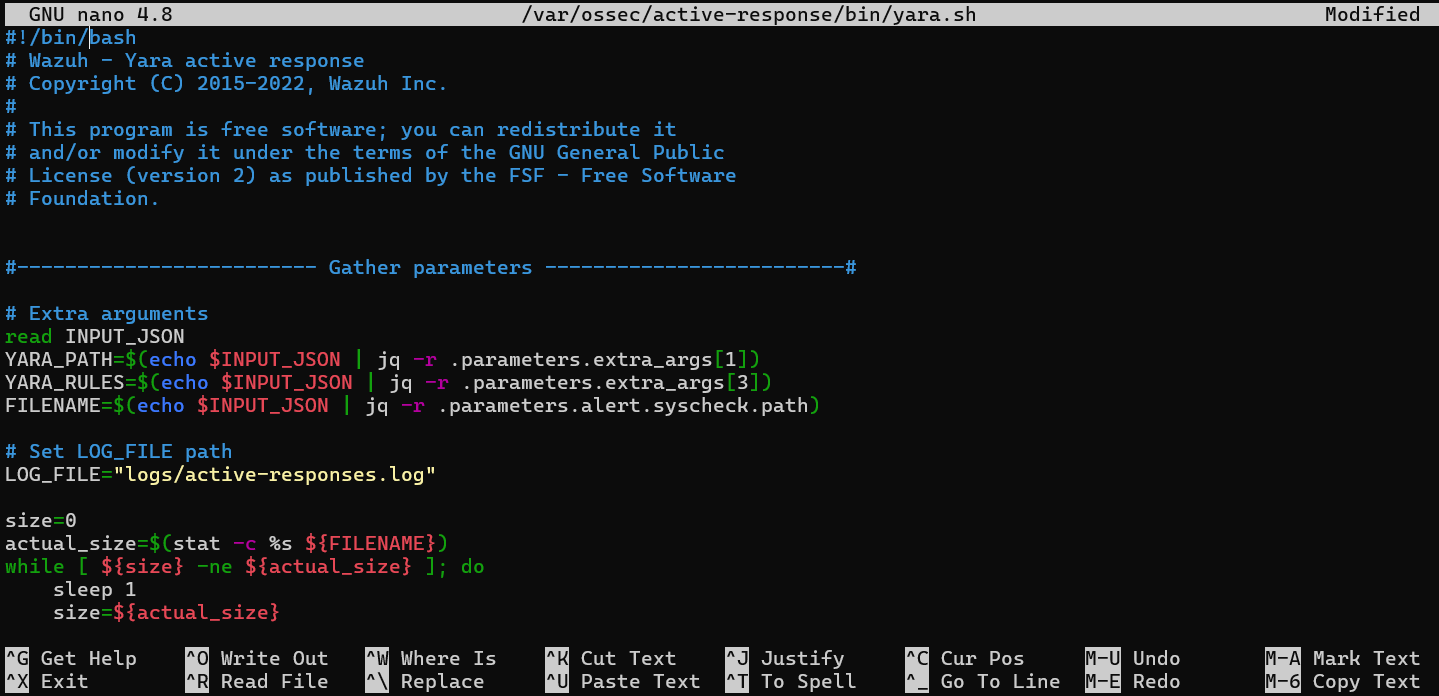
    while read -r line; do

        echo "wazuh-yara: INFO - Scan result: $line" >> ${LOG\_FILE}

    done <<< "$yara\_output"

fi

exit 0;



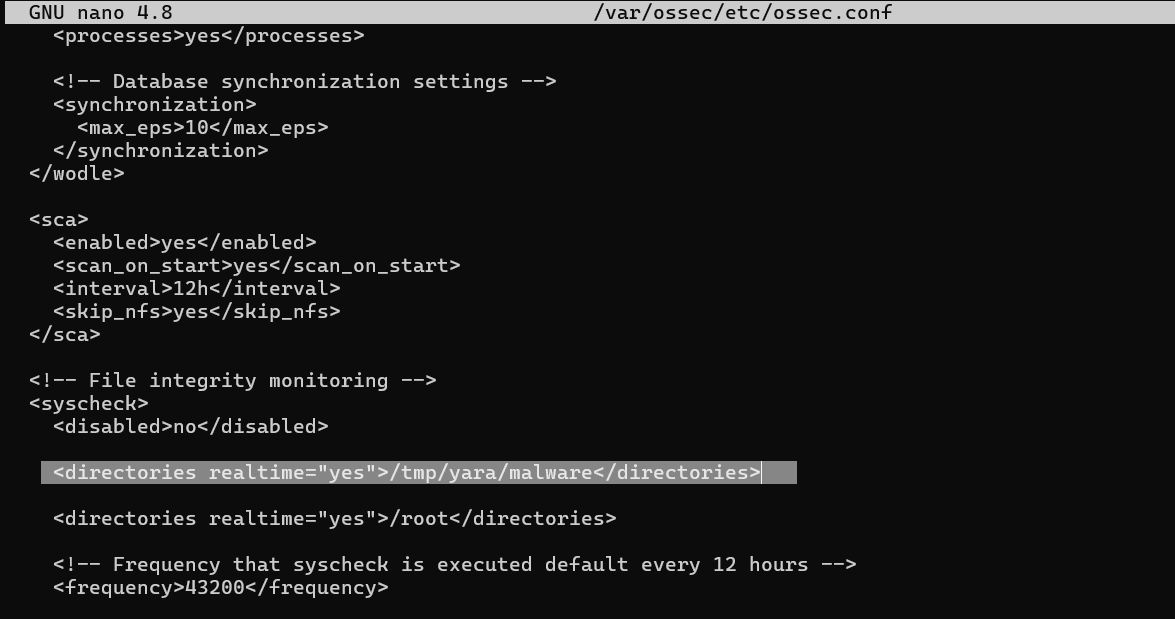
Thay đổi *yara.sh* chủ sở hữu tập tin root:wazuh và quyền tập tin 0750:

sudo chown root:wazuh /var/ossec/active-response/bin/yara.sh

sudo chmod 750 /var/ossec/active-response/bin/yara.sh

Thêm phần sau vào trong <syscheck> khối của đại lý Wazuh /var/ossec/etc/ossec.conf tập tin cấu hình để theo dõi /tmp/yara/malware thư mục:

**<directories realtime="yes">/tmp/yara/malware</directories>**

****

Khởi động lại tác nhân Wazuh để áp dụng các thay đổi cấu hình:

**sudo systemctl restart wazuh-agent**

* Trên Wazuh Server

Thêm các quy tắc sau vào /var/ossec/etc/rules/local\_rules.xml tập tin. Các quy tắc phát hiện các sự kiện FIM trong thư mục được theo dõi. Họ cũng cảnh báo khi tích hợp YARA tìm thấy phần mềm độc hại. Bạn có thể sửa đổi các quy tắc để phát hiện các sự kiện từ các thư mục khác:

<group name="syscheck,">

  <rule id="100300" level="7">

    <if\_sid>550</if\_sid>

    <field name="file">/tmp/yara/malware/</field>

    <description>File modified in /tmp/yara/malware/ directory.</description>

  </rule>

  <rule id="100301" level="7">

    <if\_sid>554</if\_sid>

    <field name="file">/tmp/yara/malware/</field>

    <description>File added to /tmp/yara/malware/ directory.</description>

  </rule>

</group>

<group name="yara,">

  <rule id="108000" level="0">

    <decoded\_as>yara\_decoder</decoded\_as>

    <description>Yara grouping rule</description>

  </rule>

  <rule id="108001" level="12">

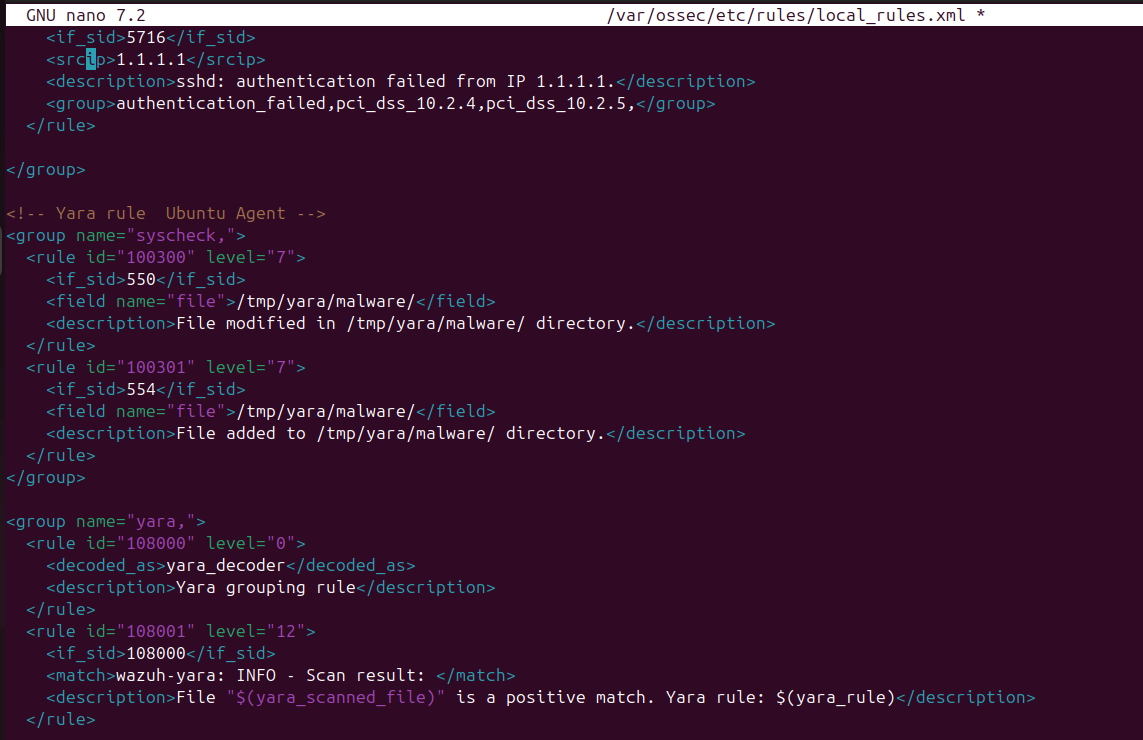
    <if\_sid>108000</if\_sid>

    <match>wazuh-yara: INFO - Scan result: </match>

    <description>File "$(yara\_scanned\_file)" is a positive match. Yara rule: $(yara\_rule)</description>

  </rule>

</group>



Thêm các bộ giải mã sau vào máy chủ Wazuh /var/ossec/etc/decoders/local\_decoder.xml tập tin. Điều này cho phép trích xuất thông tin từ kết quả quét YARA:

<decoder name="yara\_decoder">

  <prematch>wazuh-yara:</prematch>

</decoder>

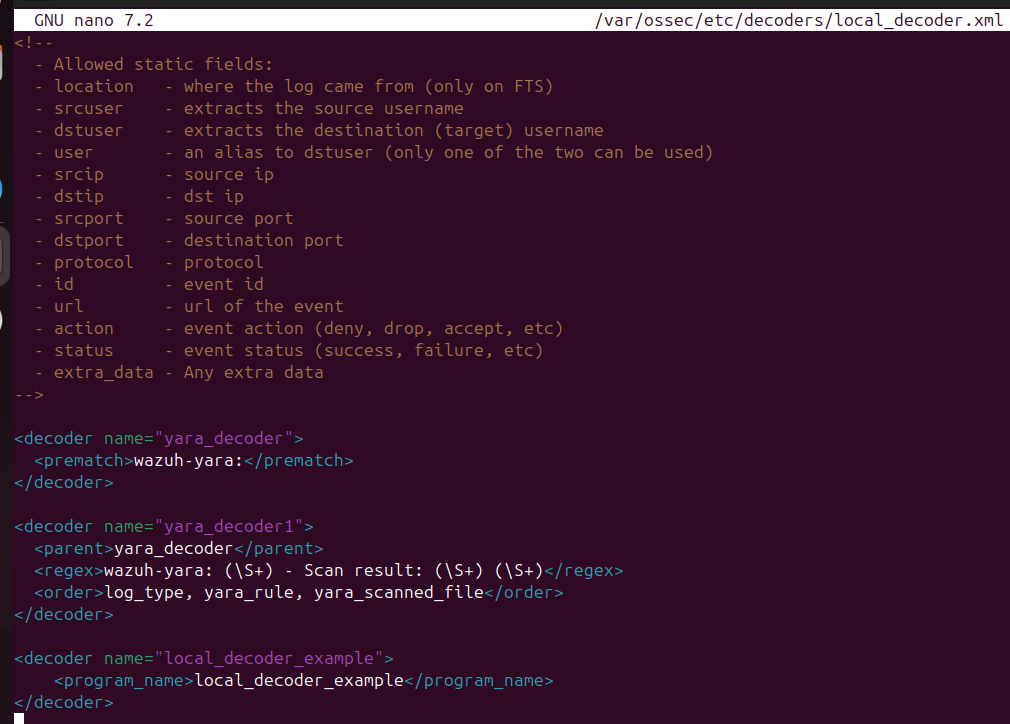
<decoder name="yara\_decoder1">

  <parent>yara\_decoder</parent>

  <regex>wazuh-yara: (\S+) - Scan result: (\S+) (\S+)</regex>

  <order>log\_type, yara\_rule, yara\_scanned\_file</order>

</decoder>



Thêm cấu hình sau vào máy chủ Wazuh /var/ossec/etc/ossec.conf tập tin cấu hình. Điều này cấu hình mô-đun Phản hồi hoạt động để kích hoạt sau khi quy tắc 100300 và 100301 được kích hoạt:

<ossec\_config>

  <command>

    <name>yara\_linux</name>

    <executable>yara.sh</executable>

    <extra\_args>-yara\_path /usr/local/bin -yara\_rules /tmp/yara/rules/yara\_rules.yar</extra\_args>

    <timeout\_allowed>no</timeout\_allowed>

  </command>

  <active-response>

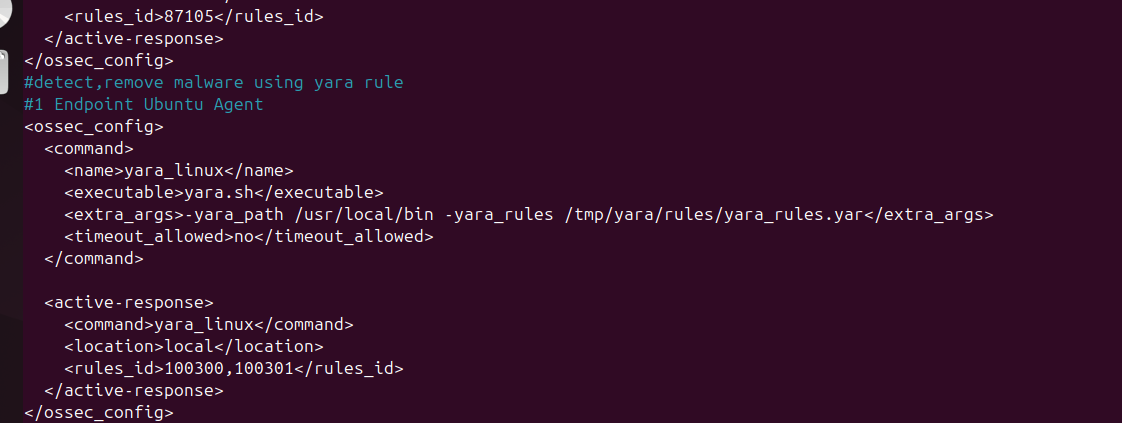
    <command>yara\_linux</command>

    <location>local</location>

    <rules\_id>100300,100301</rules\_id>

  </active-response>

</ossec\_config>

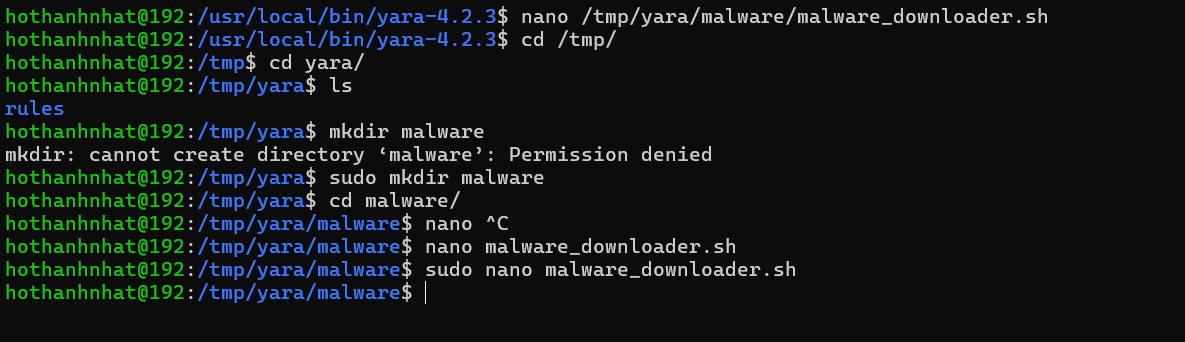


Khởi động lại trình quản lý Wazuh để áp dụng các thay đổi cấu hình:

**sudo systemctl restart wazuh-manager**

* Trên Agent Ubuntu thực hiện kịch bản:

Tạo kịch bản /tmp/yara/malware/malware\_downloader.sh trên điểm cuối được theo dõi để tải xuống các mẫu phần mềm độc hại:

****

#!/bin/bash

# Wazuh - Malware Downloader for test purposes

# Copyright (C) 2015-2022, Wazuh Inc.

#

# This program is free software; you can redistribute it

# and/or modify it under the terms of the GNU General Public

# License (version 2) as published by the FSF - Free Software

# Foundation.

function fetch\_sample(){

  curl -s -XGET "$1" -o "$2"

}

echo "WARNING: Downloading Malware samples, please use this script with  caution."

read -p "  Do you want to continue? (y/n)" -n 1 -r ANSWER

echo

if [[ $ANSWER =~ ^[Yy]$ ]]

then

    echo

    # Mirai

    echo "# Mirai: https://en.wikipedia.org/wiki/Mirai\_(malware)"

    echo "Downloading malware sample..."

    fetch\_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/mirai" "/tmp/yara/malware/mirai" && echo "Done!" || echo "Error while downloading."

    echo

    # Xbash

    echo "# Xbash: https://unit42.paloaltonetworks.com/unit42-xbash-combines-botnet-ransomware-coinmining-worm-targets-linux-windows/"

    echo "Downloading malware sample..."

    fetch\_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/xbash" "/tmp/yara/malware/xbash" && echo "Done!" || echo "Error while downloading."

    echo

    # VPNFilter

    echo "# VPNFilter: https://news.sophos.com/en-us/2018/05/24/vpnfilter-botnet-a-sophoslabs-analysis/"

    echo "Downloading malware sample..."

    fetch\_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/vpn\_filter" "/tmp/yara/malware/vpn\_filter" && echo "Done!" || echo "Error while downloading."

    echo

    # Webshell

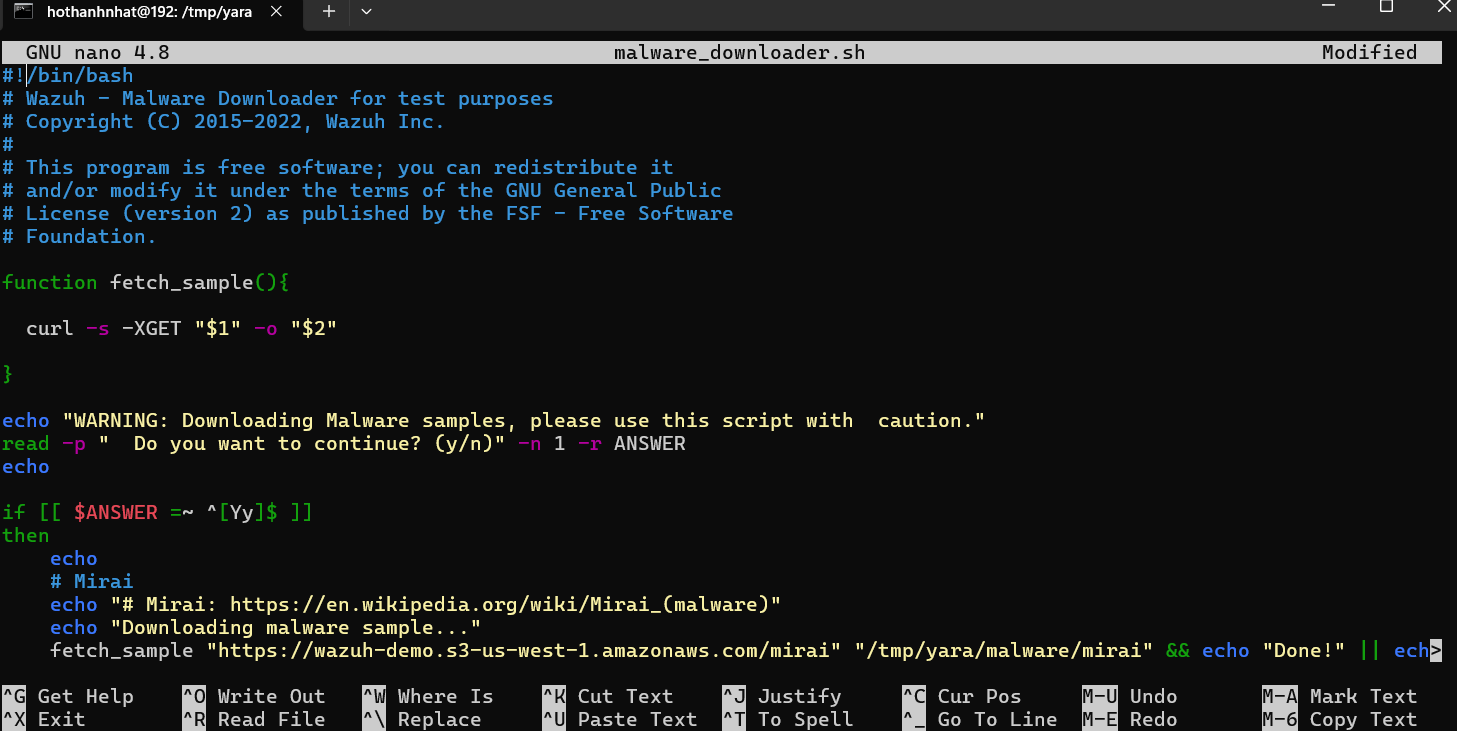
    echo "# WebShell: https://github.com/SecWiki/WebShell-2/blob/master/Php/Worse%20Linux%20Shell.php"

    echo "Downloading malware sample..."

    fetch\_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/webshell" "/tmp/yara/malware/webshell" && echo "Done!" || echo "Error while downloading."

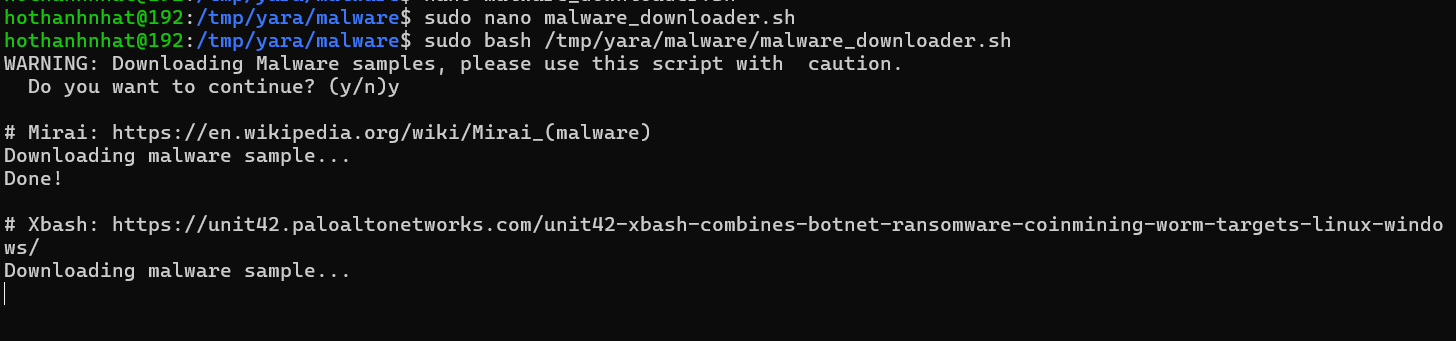
    echo

fi

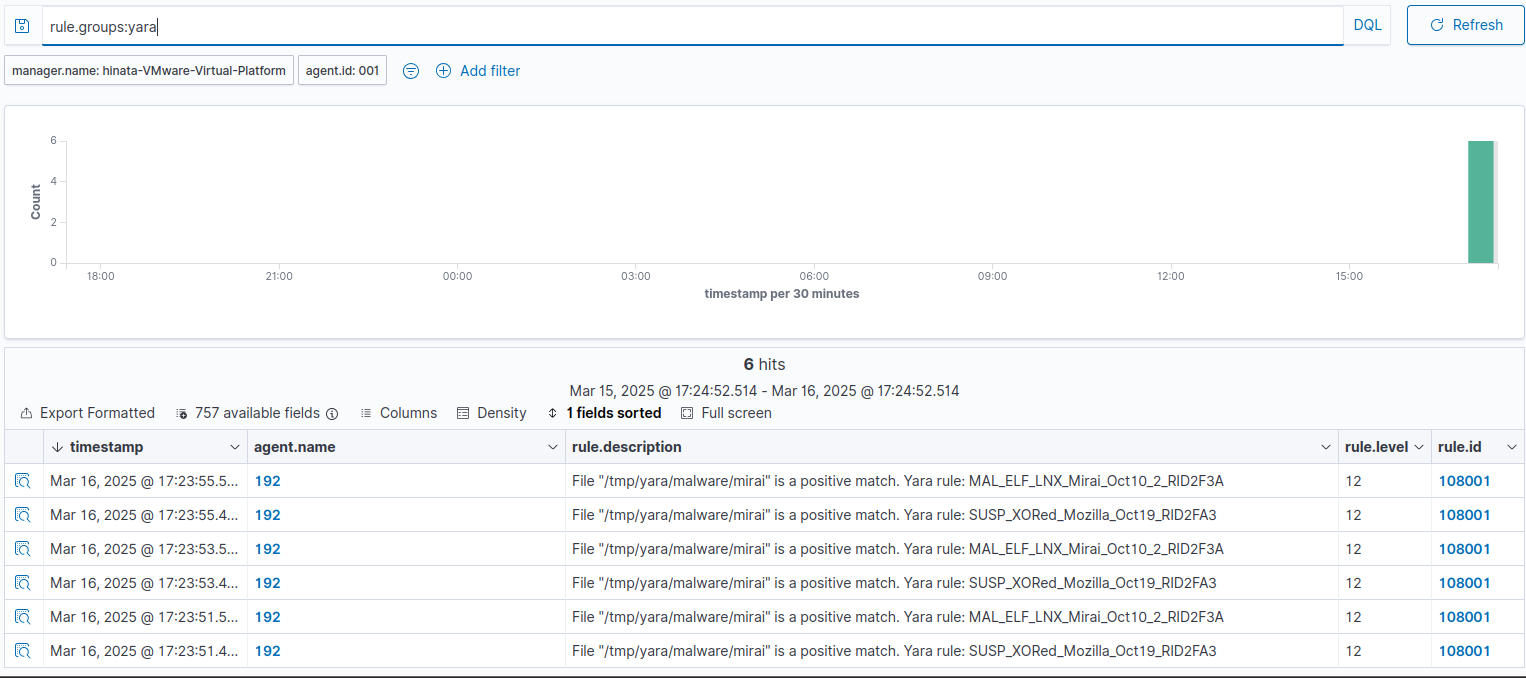
****

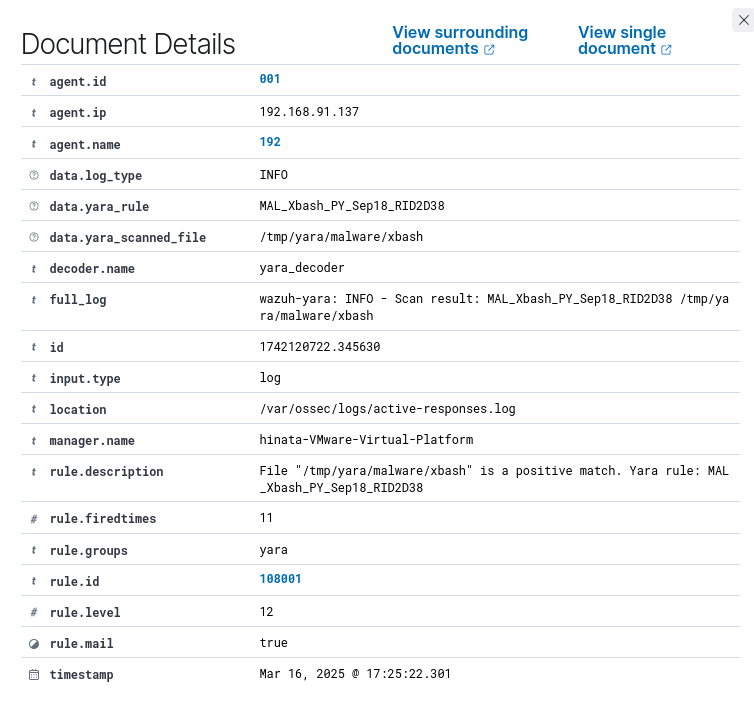
Chạy malware\_downloader.sh tập lệnh để tải mẫu phần mềm độc hại xuống /tmp/yara/malware thư mục:

**sudo bash /tmp/yara/malware/malware\_downloader.sh**

****

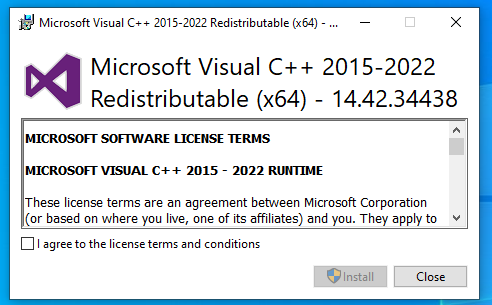
* Trên wazuh server nhận cảnh báo: **rule.groups:yara**

****

****

* 1. **Cấu hình trên Window Agent**
* Agent Window

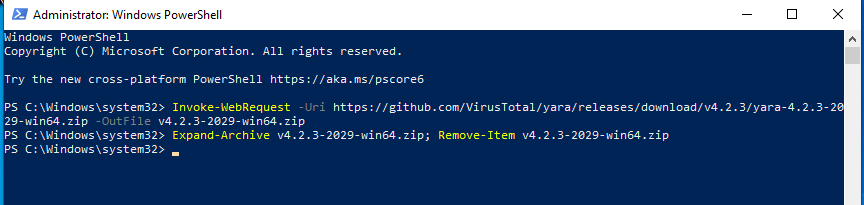
Tải xuống và cài đặt mới nhất Visual C ++ Gói phân phối lại.



Mở PowerShell với các đặc quyền của quản trị viên để tải xuống và trích xuất YARA:

Invoke-WebRequest -Uri https://github.com/VirusTotal/yara/releases/download/v4.2.3/yara-4.2.3-2029-win64.zip -OutFile v4.2.3-2029-win64.zip

Expand-Archive v4.2.3-2029-win64.zip; Remove-Item v4.2.3-2029-win64.zip

****

Tạo một thư mục được gọi và sao chép YARA thực thi vào nó: C:\Program Files (x86)\ossec-agent\active-response\bin\yara\

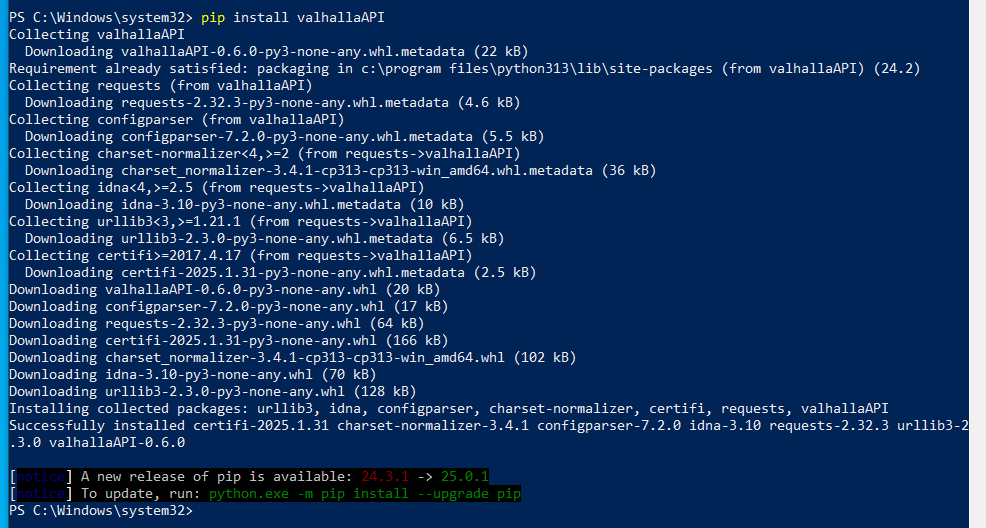
mkdir 'C:\Program Files (x86)\ossec-agent\active-response\bin\yara\'

cp .\v4.2.3-2029-win64\yara64.exe 'C:\Program Files (x86)\ossec-agent\active-response\bin\yara\'



Cài đặt valhallaAPI mô-đun:

**pip install valhallaAPI**



Sao chép tập lệnh sau và lưu nó dưới dạng download\_yara\_rules.py:

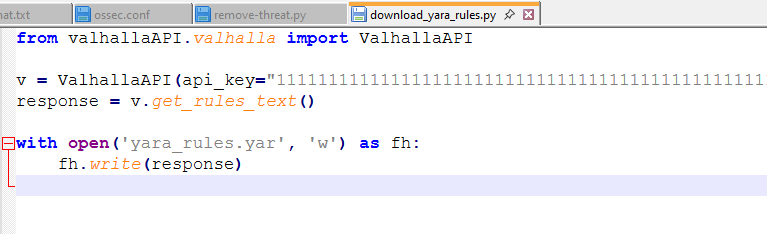
from valhallaAPI.valhalla import ValhallaAPI

v = ValhallaAPI(api\_key="1111111111111111111111111111111111111111111111111111111111111111")

response = v.get\_rules\_text()

with open('yara\_rules.yar', 'w') as fh:

    fh.write(response)

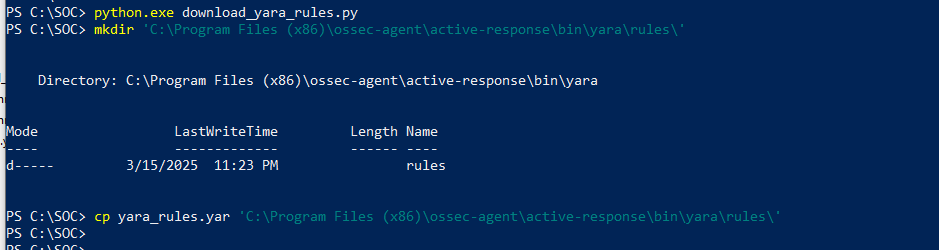


Chạy các lệnh sau để tải xuống các quy tắc và đặt chúng vào thư mục: C:\Program Files (x86)\ossec-agent\active-response\bin\yara\rules\

python.exe download\_yara\_rules.py

mkdir 'C:\Program Files (x86)\ossec-agent\active-response\bin\yara\rules\'

cp yara\_rules.yar 'C:\Program Files (x86)\ossec-agent\active-response\bin\yara\rules\'



Thực hiện các bước dưới đây để định cấu hình Wazuh FIM và tập lệnh phản hồi hoạt động để phát hiện các tệp độc hại trên điểm cuối.

Tạo yara.bat kịch bản trong thư mục. Điều này là cần thiết cho việc quét Phản hồi chủ động của Wazuh-YARA:C:\Program Files (x86)\ossec-agent\active-response\bin\

@echo off

setlocal enableDelayedExpansion

reg Query "HKLM\Hardware\Description\System\CentralProcessor\0" | find /i "x86" > NUL && SET OS=32BIT || SET OS=64BIT

if %OS%==32BIT (

    SET log\_file\_path="%programfiles%\ossec-agent\active-response\active-responses.log"

)

if %OS%==64BIT (

    SET log\_file\_path="%programfiles(x86)%\ossec-agent\active-response\active-responses.log"

)

set input=

for /f "delims=" %%a in ('PowerShell -command "$logInput = Read-Host; Write-Output $logInput"') do (

    set input=%%a

)

set json\_file\_path="C:\Program Files (x86)\ossec-agent\active-response\stdin.txt"

set syscheck\_file\_path=

echo %input% > %json\_file\_path%

for /F "tokens=\* USEBACKQ" %%F in (`Powershell -Nop -C "(Get-Content 'C:\Program Files (x86)\ossec-agent\active-response\stdin.txt'|ConvertFrom-Json).parameters.alert.syscheck.path"`) do (

set syscheck\_file\_path=%%F

)

del /f %json\_file\_path%

set yara\_exe\_path="C:\Program Files (x86)\ossec-agent\active-response\bin\yara\yara64.exe"

set yara\_rules\_path="C:\Program Files (x86)\ossec-agent\active-response\bin\yara\rules\yara\_rules.yar"

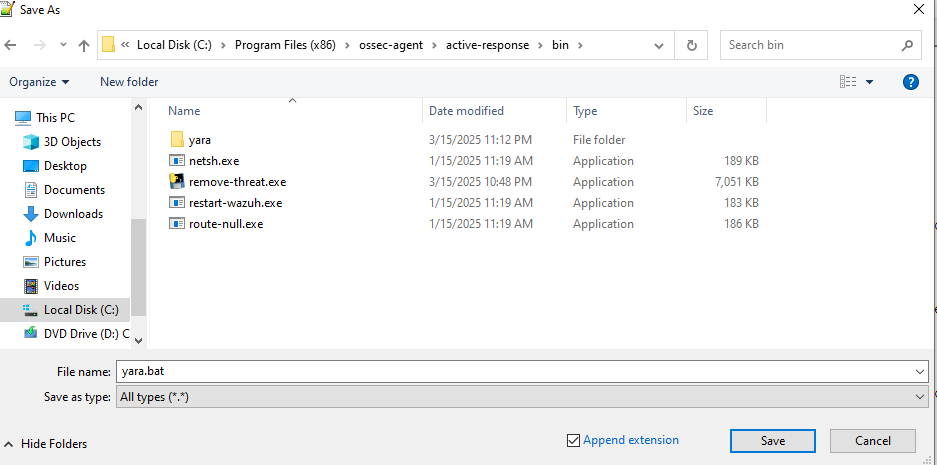
echo %syscheck\_file\_path% >> %log\_file\_path%

for /f "delims=" %%a in ('powershell -command "& \"%yara\_exe\_path%\" \"%yara\_rules\_path%\" \"%syscheck\_file\_path%\""') do (

    echo wazuh-yara: INFO - Scan result: %%a >> %log\_file\_path%

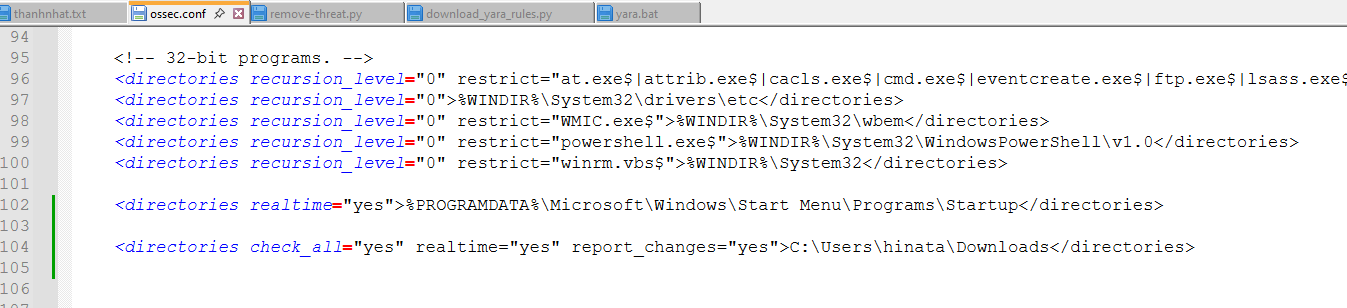
)

exit /b



Thêm C:\Users\<USER\_NAME>\Downloads thư mục để theo dõi trong <syscheck> chặn trong tệp cấu hình tác nhân Wazuh . Thay thế với tên người dùng của điểm cuối: C:\Program Files (x86)\ossec-agent\ossec.conf

<directories realtime="yes">C:\Users\<USER\_NAME>\Downloads</directories>



Khởi động lại tác nhân Wazuh để áp dụng các thay đổi cấu hình:

Restart-Service -Name wazuh

* Wazuh Server

Thêm các bộ giải mã sau vào máy chủ Wazuh /var/ossec/etc/decoders/local\_decoder.xml tập tin. Điều này cho phép trích xuất thông tin từ kết quả quét YARA:

<decoder name="yara\_decoder">

    <prematch>wazuh-yara:</prematch>

</decoder>

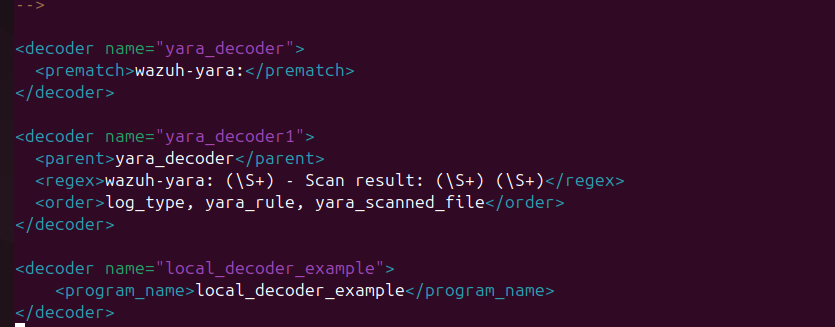
<decoder name="yara\_decoder1">

    <parent>yara\_decoder</parent>

    <regex>wazuh-yara: (\S+) - Scan result: (\S+) (\S+)</regex>

    <order>log\_type, yara\_rule, yara\_scanned\_file</order>

</decoder>



Thêm các quy tắc sau vào máy chủ Wazuh /var/ossec/etc/rules/local\_rules.xml tập tin. Các quy tắc phát hiện các sự kiện FIM trong thư mục được theo dõi. Họ cũng cảnh báo khi phần mềm độc hại được tìm thấy bởi tích hợp YARA:

<group name="syscheck,">

  <rule id="100303" level="7">

    <if\_sid>550</if\_sid>

    <field name="file">C:\\Users\\<USER\_NAME>\\Downloads</field>

    <description>File modified in C:\Users\<USER\_NAME>\Downloads directory.</description>

  </rule>

  <rule id="100304" level="7">

    <if\_sid>554</if\_sid>

    <field name="file">C:\\Users\\<USER\_NAME>\\Downloads</field>

    <description>File added to C:\Users\<USER\_NAME>\Downloads  directory.</description>

  </rule>

</group>

<group name="yara,">

  <rule id="108000" level="0">

    <decoded\_as>yara\_decoder</decoded\_as>

    <description>Yara grouping rule</description>

  </rule>

  <rule id="108001" level="12">

    <if\_sid>108000</if\_sid>

    <match>wazuh-yara: INFO - Scan result: </match>

    <description>File "$(yara\_scanned\_file)" is a positive match. Yara rule: $(yara\_rule)</description>

  </rule>

</group>



Thêm cấu hình sau vào máy chủ Wazuh /var/ossec/etc/ossec.conf tập tin:

<ossec\_config>

  <command>

    <name>yara\_windows</name>

    <executable>yara.bat</executable>

    <timeout\_allowed>no</timeout\_allowed>

  </command>

  <active-response>

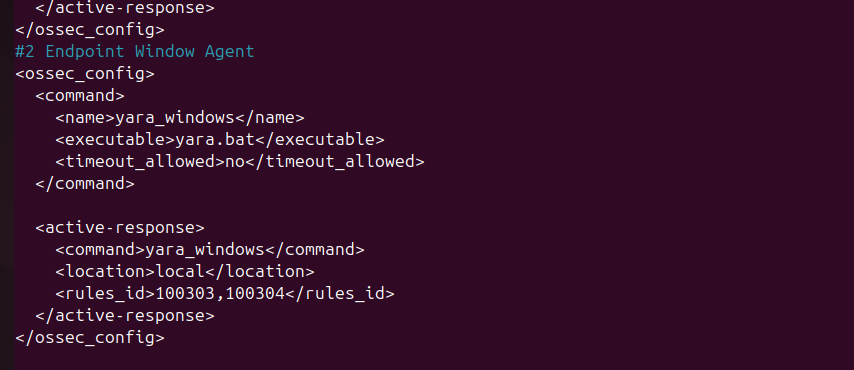
    <command>yara\_windows</command>

    <location>local</location>

    <rules\_id>100303,100304</rules\_id>

  </active-response>

</ossec\_config>



Khởi động lại trình quản lý Wazuh để áp dụng các thay đổi cấu hình:

**sudo systemctl restart wazuh-manager**

* Thực thi trên Window Agent

Tải xuống mẫu phần mềm độc hại trên điểm cuối Windows được theo dõi:

1. Tắt Microsoft Virus và bảo vệ mối đe dọa.
2. Tải xuống tệp zip EICAR:

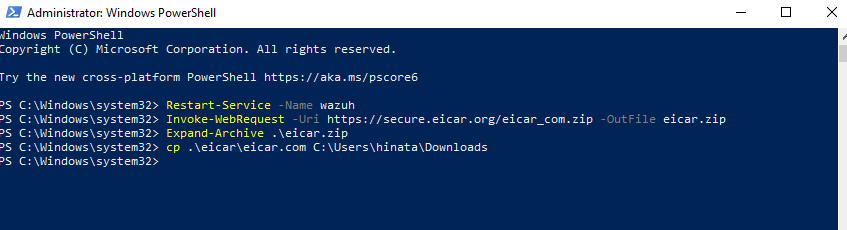
**Invoke-WebRequest -Uri https://secure.eicar.org/eicar\_com.zip -OutFile eicar.zip**

1. Giải nén nó:

**Expand-Archive .\eicar.zip**

1. Sao chép tệp EICAR vào thư mục được theo dõi:

**cp .\eicar\eicar.com C:\Users\hinata\Downloads**

****

* Các cảnh báo trên Wazuh Server: rule.groups:yara