探測使用 Yara 整合

您可以使用 Wazuh 與 YARA 整合，對端點上新增或修改的檔案進行惡意軟體掃描。 YARA 是一個用於檢測和分類惡意軟體遺留的工具。

在此使用情境中，我們將演示如何在 Linux 和 Windows 端點上配置 YARA 與 Wazuh，以檢測惡意軟體。

欲了解更多關於 Wazuh 與 YARA 整合的資訊，請參閱文件中的如何將 Wazuh 整合 YARA 一節。

基礎設施

端點

說明

Ubuntu 22.04 / RHEL 9.0

YARA 主動回應模組會在 Wazuh FIM 模組觸發警報時，掃描新檔或修改過的檔案。

Windows 11

YARA 主動回應模組會在 Wazuh FIM 模組觸發警報時，掃描新檔或修改過的檔案。

Linux 配置

Linux 端點

請按照以下步驟安裝 YARA 並配置主動回應和 FIM 模組。

下載、編譯和安裝 YARA：

Ubuntu

```
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
```

RHEL

```
sudo yum makecache
sudo yum install epel-release
sudo yum update
sudo yum install -y make automake gcc autoconf libtool openssl-devel 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
```

測試 YARA 是否正常運行：
yara
輸出
yara: wrong number of arguments
Usage: yara [OPTION]... [NAMESPACE:]RULES_FILE... FILE | DIR | PID

Try --help for more options

如果顯示以下錯誤訊息：
/usr/local/bin/yara: error while loading shared libraries: libyara.so.9: cannot open shared object file: No such file or directory.
這表示載入程式找不到 libyara 程式庫，該程式庫通常位於/usr/local/lib。請將/usr/local/lib 路徑添加到/etc/ld.so.conf 載入器配置檔中以解決此問題：

sudo su
echo "/usr/local/lib" >> /etc/ld.so.conf
ldconfig
切換回原來的使用者。

下載 YARA 偵測規則：

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

在/var/ossec/active-response/bin/目錄下創建 yara.sh 腳本。這對於 Wazuh-YARA 主動回應掃描是必要的：

#!/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;
```

將 yara.sh 檔案的所有者更改為 root:wazuh，並將檔案權限更改為0750：

```
sudo chown root:wazuh /var/ossec/active-response/bin
```

接著，修改檔案權限為0750：

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

在 Wazuh 代理的/var/ossec/etc/ossec.conf 設定檔中，將以下內容添加至<syscheck>區塊，以監控/tmp/yara/malware 目錄：

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

重新啟動 Wazuh 代理以應用配置更改：

sudo systemctl restart wazuh-agent

Wazuh 伺服器配置

按照以下步驟設定 Wazuh，使其在監控目錄中的檔案更改時發出警報。這些步驟還配置了一個主動回應腳本，以在檢測到可疑檔案時觸發。

將以下規則添加至/var/ossec/etc/rules/local_rules.xml 檔案。這些規則會在監控目錄中檢測 FIM 事件時發出警報。同時，它們還會在 YARA 整合發現惡意軟體時發出警報。您可以修改這些規則以檢測其他目錄中的事件：

```
<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>
```

將以下解碼器添加到 Wazuh 伺服器的/var/ossec/etc/decoders/local_decoder.xml 檔案。這樣可以從 YARA 掃描結果中提取資訊：

```
<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>
```

將以下配置添加到 Wazuh 伺服器的/var/ossec/etc/ossec.conf 配置檔案中。這樣配置主動回應模組，在規則100300和100301觸發後觸發：

```
<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>
```

重新啟動 Wazuh 管理員以應用配置更改：

```
sudo systemctl restart wazuh-manager
```

攻擊模擬

在受監控的端點上創建名為/tmp/yara/malware/malware_downloader.sh 的腳本，以下載惡意軟體樣本：

```
#!/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
```

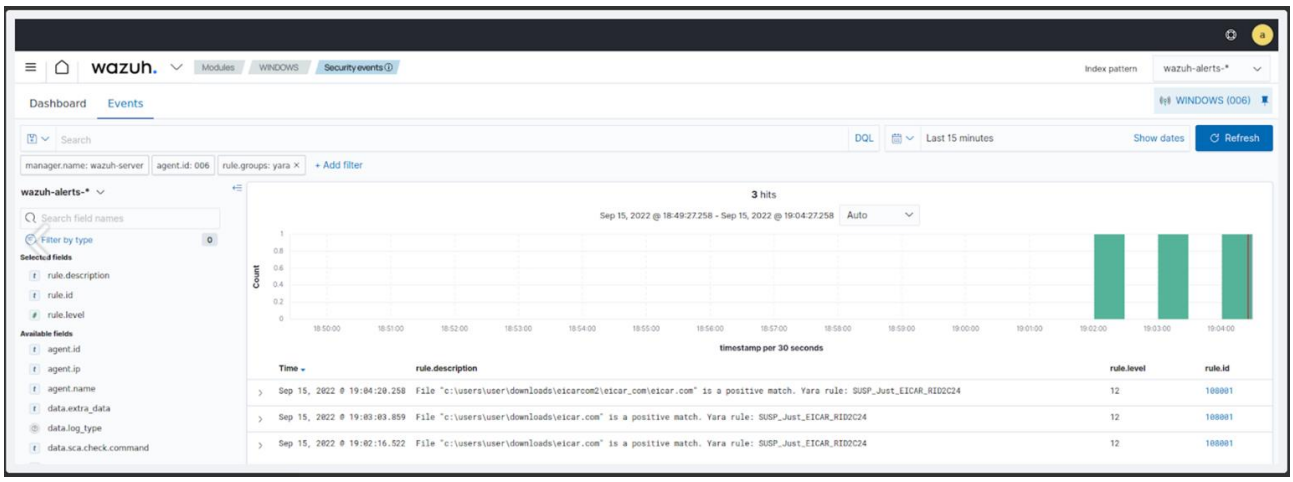
執行 malware_downloader.sh 腳本來將惡意軟體樣本下載到/tmp/yara/malware 目錄：

```
sudo bash /tmp/yara/malware/malware_downloader.sh
```

視覺化警報

您可以在 Wazuh 儀表板中視覺化警報資料。要這樣做，進入 Security events 模組，並在搜尋欄中添加過濾器以查詢警報。

```
rule.groups:yara
```


wazuh.
Modules
WINDOWS
Security events
Index pattern
wazuh-alerts-*
Dashboard
Events
WINDOWS (006)
Search
DQL
Last 15 minutes
Show dates
Refresh
manager.name: wazuh-server
agent.id: 006
rule.groups: yara
+ Add filter
wazuh-alerts-*
Search field names
Filter by type
Selected fields
rule.description
rule.id
rule.level
Available fields
agent.id
agent.ip
agent.name
data.extra_data
data.log_type
data.sca.check.command
3 hits
Sep 15, 2022 @ 18:49:27.258 - Sep 15, 2022 @ 19:04:27.258
Auto
Count
timestamp per 30 seconds
Time
rule.description
rule.level
rule.id
Sep 15, 2022 @ 19:04:20.258 File "c:\users\user\downloads\eicarcom2\eicar_com\eicar.com" is a positive match. Yara rule: SUSP_Just_EICAR_RID2C24 12 100001
Sep 15, 2022 @ 19:03:03.859 File "c:\users\user\downloads\eicar.com" is a positive match. Yara rule: SUSP_Just_EICAR_RID2C24 12 100001
Sep 15, 2022 @ 19:02:16.522 File "c:\users\user\downloads\eicar.com" is a positive match. Yara rule: SUSP_Just_EICAR_RID2C24 12 100001