Detection and Removing malware using Virus Total Integration

Background :

Wazuh uses the integrator module to connect to external APIs and alerting tools such as VirusTotal

In this use case, you use the Wazuh File Integrity Monitoring (FIM) module to monitor a directory for changes and the VirusTotal API to scan the files in the directory. Then, configure Wazuh to trigger an active response script and remove files that VirusTotal detects as malicious. We test this use case on Ubuntu and Windows endpoints.

You need a VirusTotal API key in this use case to authenticate Wazuh to the VirusTotal API.

Configuration :

On ubuntu End point

Make sure, that under File integrity section, under <syscheck> block<disabled>**no**<disabled> is set to **no**

```
<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>
```

Add this line the same section

<directories realtime="yes">Your-working-directory-that-you-want-to-monitor </directories>

```
<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>
  <directories realtime="yes">/home/project-01/Downloads</directories>
  <!-- Frequency that syscheck is executed default every 12 hours -->
  <frequency>43200</frequency>
```

Next, install **jq**, a utility that processes JSON input from the active response script.

```
project-01@project01-virtual-machine:~$ sudo apt -y install jq
[sudo] password for project-01:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Next, Create the /var/ossec/active-response/bin/remove-threat.sh active response script to remove malicious files from the linux endpoint:

```
root@project01-virtual-machine:/home/project-01/Downloads# cd /
root@project01-virtual-machine:/# nano /var/ossec/active-response/bin/remove-threat.sh
```

Now, try to add this block in that remove-threat.sh file

```
#!/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;

```
  GNU nano 6.2                    /var/ossec/active-response/bin/remove-threat.sh *
#!/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;
```

Now, Change the /var/ossec/active-response/bin/remove-threat.sh file ownership, and permissions:

Command:

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

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

```
root@project01-virtual-machine:/# sudo chmod 750 /var/ossec/active-response/bin/remove-threat.sh
sudo chown root:wazuh /var/ossec/active-response/bin/remove-threat.sh
```

After changing the file permissions, try to restart the wazuh-agent

Command:

sudo systemctl restart wazuh-agent

```
root@project01-virtual-machine:/# sudo systemctl restart wazuh-agent
```

After, restarting your wazuh agent, you check for the status of your agent

Command:

sudo systemctl status wazuh-agent

```
root@project01-virtual-machine:/# sudo systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-06-12 15:16:24 IST; 6min ago
    Process: 13676 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 32 (limit: 4554)
     Memory: 282.8M
        CPU: 47.543s
     CGroup: /system.slice/wazuh-agent.service
             ├─13699 /var/ossec/bin/wazuh-execd
             ├─13710 /var/ossec/bin/wazuh-agentd
             ├─13724 /var/ossec/bin/wazuh-syscheckd
             ├─13737 /var/ossec/bin/wazuh-logcollector
             └─13748 /var/ossec/bin/wazuh-modulesd

Jun 12 15:16:18 project01-virtual-machine systemd[1]: Starting Wazuh agent...
Jun 12 15:16:18 project01-virtual-machine env[13676]: Starting Wazuh v4.7.5...
Jun 12 15:16:19 project01-virtual-machine env[13676]: Started wazuh-execd...
Jun 12 15:16:20 project01-virtual-machine env[13676]: Started wazuh-agentd...
Jun 12 15:16:21 project01-virtual-machine env[13676]: Started wazuh-syscheckd...
Jun 12 15:16:21 project01-virtual-machine env[13676]: Started wazuh-logcollector...
Jun 12 15:16:22 project01-virtual-machine env[13676]: Started wazuh-modulesd...
Jun 12 15:16:24 project01-virtual-machine env[13676]: Completed.
Jun 12 15:16:24 project01-virtual-machine systemd[1]: Started Wazuh agent.
```

```
  <!-- Osquery integration -->
  <wodle name="osquery">
    <disabled>yes</disabled>
    <run_daemon>yes</run_daemon>
    <log_path>/var/log/osquery/osqueryd.results.log</log_path>
    <config_path>/etc/osquery/osquery.conf</config_path>
    <add_labels>yes</add_labels>
  </wodle>

  <!--- Virus Total integration ---->
  <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>
```

Now, its time for Wazuh server configuration .

Initially I have accessed my wazuh server via ssh.

```
┌──(kali㉿kali)-[~]
└─$ ssh wazuh-user@192.168.198.126
wazuh-user@192.168.198.126's password:
Permission denied, please try again.
wazuh-user@192.168.198.126's password:
Last failed login: Wed Jun 12 08:24:23 UTC 2024 from 192.168.198.131 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Wed Jun 12 08:24:15 2024
wwwwww.           wwwwwww.          wwwwwww.
wwwwwww.          wwwwwww.          wwwwwww.
 wwwwww.         wwwwwwwww.        wwwwwww.
 wwwwwww.        wwwwwwwww.        wwwwwww.
  wwwwww.       wwwwwwwwwww.      wwwwwww.
  wwwwwww.      wwwwwwwwwww.      wwwwwww.
   wwwwww.     www www.wwwwww.    wwwwwww.
   wwwwwww.    wwwwww. wwwwww.    wwwwwww.
    wwwwww.   wwwwww.  wwwwww.   wwwwwww.
    wwwwwww.  wwwwww.  wwwwww.   wwwwwww.
     wwwwww. wwwwww.    wwwwwww.wwwwwww.
     wwwwwww.wwwwww.    wwwwwww.wwwwwww.
      wwwwwwwwwwww.      wwwwwwwwwwww.
      wwwwwwwwwwww.      wwwwwwwwwwww.      000000
       wwwwwwwwwww.       wwwwwwwwwww.     00000000
       wwwwwwwwww.        wwwwwwwwww.     0000000000
        wwwwwwww.          wwwwwwww.      0000000000
        wwwwwww.           wwwwwwww.       00000000
         wwwwww.            wwwwww.         000000

        WAZUH Open Source Security Platform
                https://wazuh.com
```

Now, try to add the following rules to the /var/ossec/etc/rules/local_rules.xml file on the Wazuh server. These rules alert about changes in the /Downloads directory that are detected by FIM scans:

```
<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">Give-your-fim-configured-directory </field>
    <description>File modified in /path-to-your-directory.</description>
  </rule>
  <rule id="100201" level="7">
    <if_sid>554</if_sid>
    <field name="file"> Give-your-fim-configured-directory </field>
    <description>File added to / path-to-your-directory.</description>
  </rule>
</group>
```

In my case, I have configured Fim to my Downloads Folder

```
    <!-- Rules for Linux systems -->
    <rule id="100200" level="7">
        <if_sid>550</if_sid>
        <field name="file">/home/project-01/Downloads</field>
        <description>File modified in /Downloads directory.</description>
    </rule>
    <rule id="100201" level="7">
        <if_sid>554</if_sid>
        <field name="file">/home/project-01/Downloads</field>
        <description>File added to /Downloads directory.</description>
    </rule>
</group>

<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>
```

Now, Add the following configuration to the Wazuh server /var/ossec/etc/ossec.conf file to enable the Virustotal integration. Replace <YOUR_VIRUS_TOTAL_API_KEY> with your VirusTotal API key. This allows to trigger a VirusTotal query whenever any of the rules 100200 and 100201 are triggered:

```
<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>
```

Here, you add code at any where, but I have added under Osquery Integration section

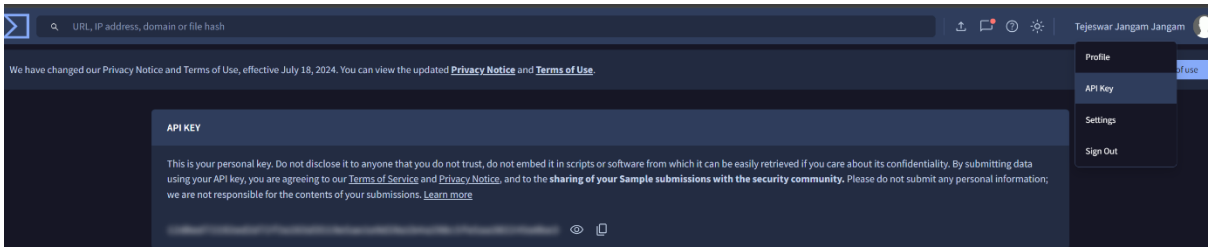
```
  <!-- Osquery integration -->
  <wodle name="osquery">
    <disabled>yes</disabled>
    <run_daemon>yes</run_daemon>
    <log_path>/var/log/osquery/osqueryd.results.log</log_path>
    <config_path>/etc/osquery/osquery.conf</config_path>
    <add_labels>yes</add_labels>
  </wodle>

<!--------VirusTotal API------>

  <integration>
    <name>virustotal</name>
    <api_key>Your_API_KEY</api_key> <!-- Replace with your VirusTotal API key -->
    <rule_id>100200,100201</rule_id>
    <alert_format>json</alert_format>
  </integration>
```

In, order to know your API key, first you need to create an account in VirusTotal, after that login to your account

Profile → API Key → copy your API key



Also, add these lines under “VirusTotal API Section”

```
<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>
```

```
<!———Remove file, when virus Total Triggers ————→

<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>
```

Now, again try to open your local_rules.xml,

nano /var/ossec/etc/rules/local_rules.xml

I my case, I have pasted the block of code under “Rules for Linux Systems”

```
GNU nano 2.9.8                                                    local_rules.xml

  <!— Rules for Linux systems —>
  <rule id="100200" level="7">
      <if_sid>550</if_sid>
      <field name="file">/home/project-01/Downloads</field>
      <description>File modified in /Downloads directory.</description>
  </rule>
  <rule id="100201" level="7">
      <if_sid>554</if_sid>
      <field name="file">/home/project-01/Downloads</field>
      <description>File added to /Downloads directory.</description>
  </rule>
</group>

<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>
```

```
<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>
```

Now, Try to restart Your wazuh-Manager

Command:

sudo systemctl restart wazuh-manager

```
[root@wazuh-server rules]# sudo systemctl restart wazuh-manager
```

Check for your wazuh-manager status

Command:

sudo systemctl status wazuh-manager

```
[root@wazuh-server rules]# sudo systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2024-06-12 10:12:57 UTC; 31s ago
  Process: 22579 ExecStop=/usr/bin/env /var/ossec/bin/wazuh-control stop (code=exited, status=0/SUCCESS)
  Process: 22729 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/wazuh-manager.service
           ├─22789 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─22790 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─22793 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─22796 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─22820 /var/ossec/bin/wazuh-integratord
           ├─22841 /var/ossec/bin/wazuh-authd
           ├─22855 /var/ossec/bin/wazuh-db
           ├─22881 /var/ossec/bin/wazuh-execd
           ├─22896 /var/ossec/bin/wazuh-analysisd
           ├─22913 /var/ossec/bin/wazuh-syscheckd
           ├─22931 /var/ossec/bin/wazuh-remoted
           ├─22965 /var/ossec/bin/wazuh-logcollector
           ├─23040 /var/ossec/bin/wazuh-monitord
           └─23086 /var/ossec/bin/wazuh-modulesd
```

Now, Attack simulation

Note, I Have downloaded a sample malware from ikarussecurity, which is of no harm.

These malwares were for testing purposes .

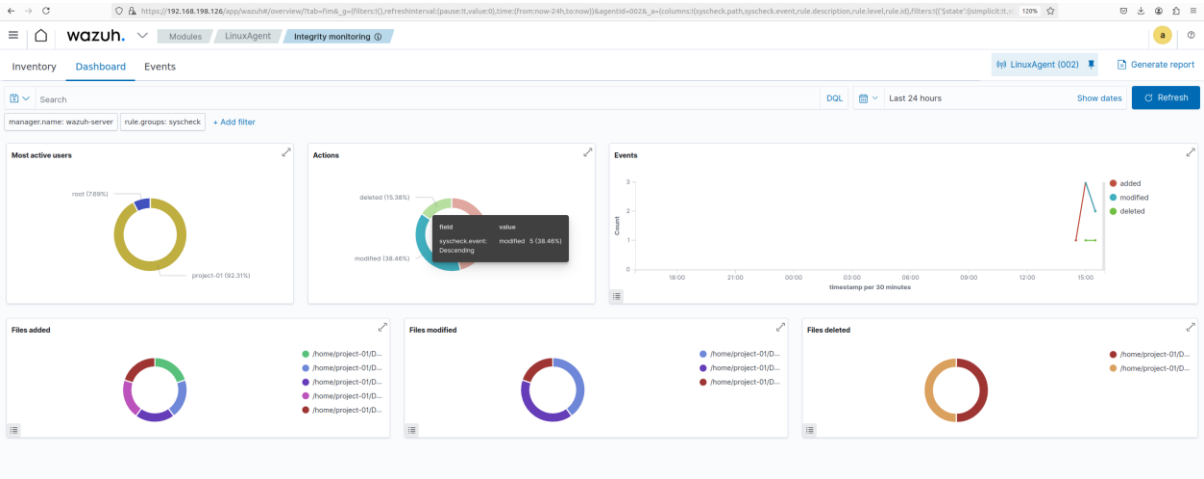
https://www.ikarussecurity.com/en/private-customers/download-test-viruses-for-free/

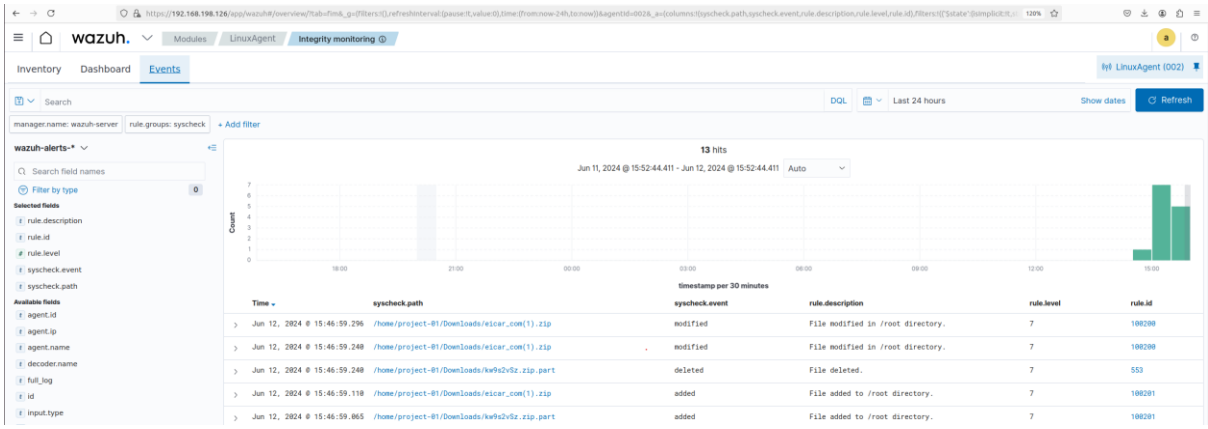
download at your own risk.

Now, I have downloaded that malware into my downloads folder.

Now, Open your wazuh dash board and you check for the event logs

Modules → select your agent → Integrity monitoring → Events





Thus, it was successful in detecting and Removing malware using VirusTotal