Detection Engineering in a Homelab — Part 4: Detecting Malware Activity on a Windows Endpoint

F Frederick Adigun 7 min read · Aug 21, 2025

The prerequisites for this section include a configured Windows endpoint with both the Wazuh agent and Sysmon installed, where we will simulate a malware infection. You also need the Wazuh server running with Wazuh set up, and the Keycloak server running in the background for SSO and IAM. If you haven't started the series, check out Part 1 of this Detection Engineering series to understand how we integrated these components.

In this guide, I will go through:

- Setting Up the Windows Endpoint for Detection
- Downloading the DeerStealer Malware
- Creating Detection Rules for DeerStealer Malware
- Validating Alerts and Observing Malware Behavior

Log in to the Wazuh dashboard URL, which will redirect you to Keycloak. Enter the credentials of the Keycloak user account you created, and you will be granted access (Fig. 1, 2).

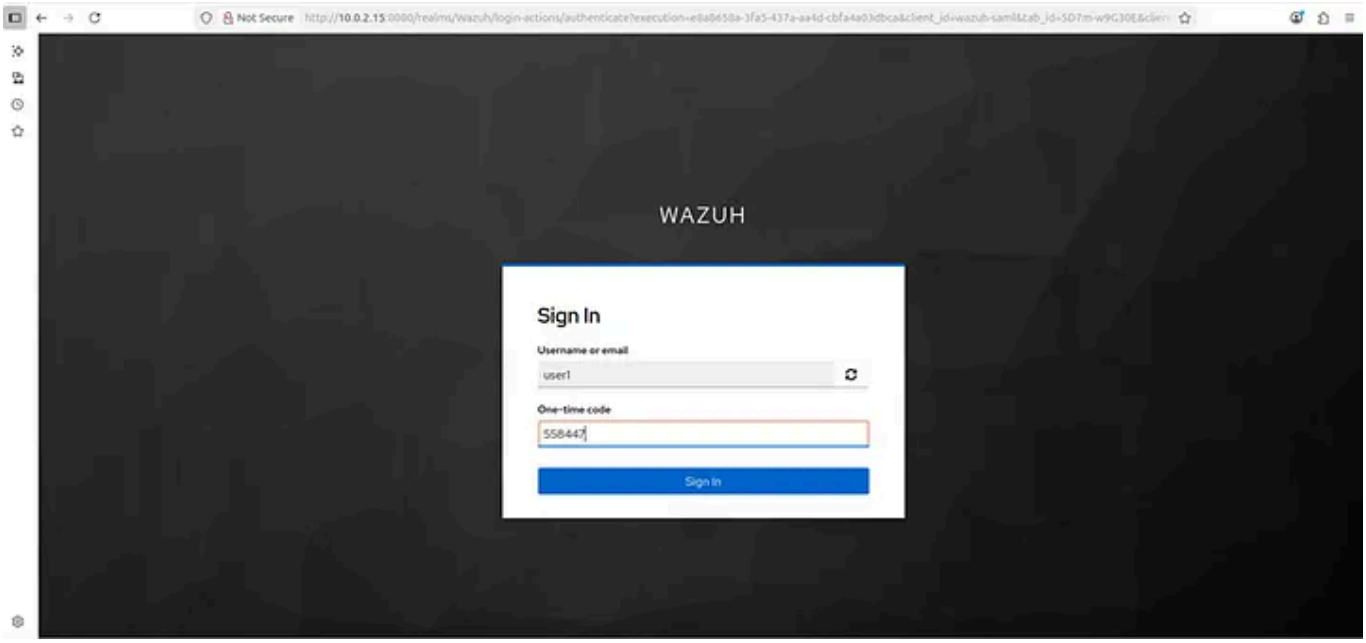

Fig. 1

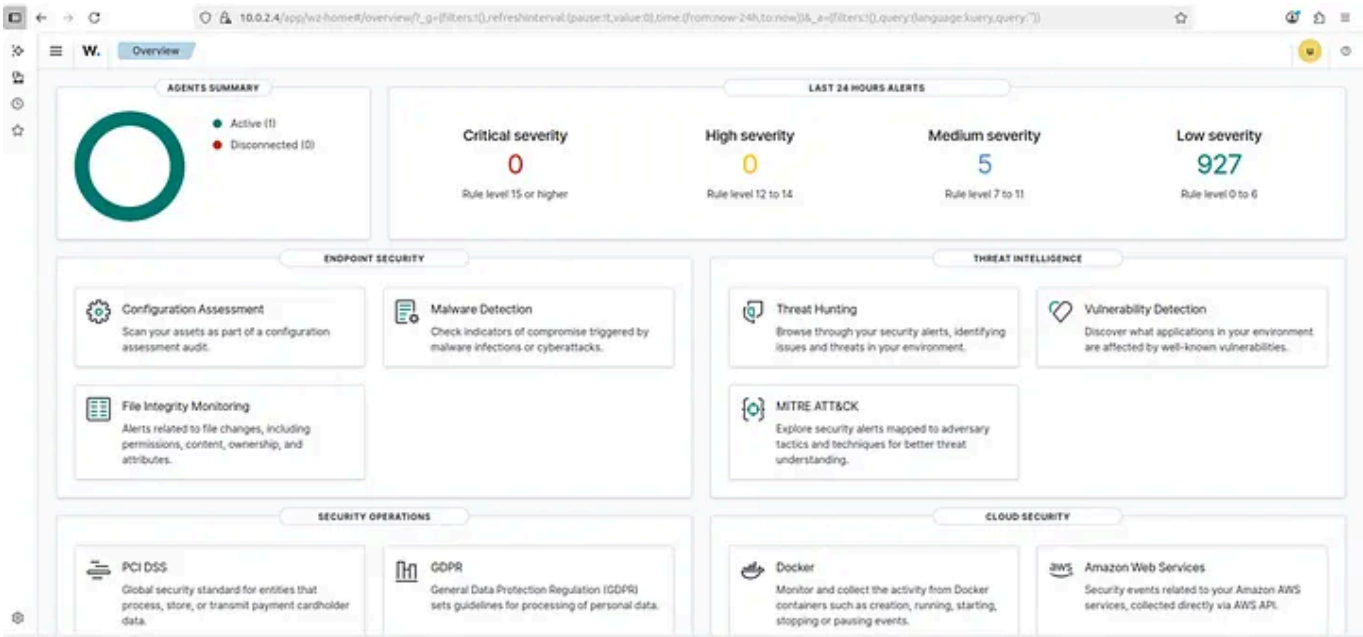

Fig. 2

Step 1: CONFIGURING THE WAZUH AGENT TO COLLECT SYSMON LOGS

You need to configure the Wazuh agent on your windows endpoint to forward these Sysmon logs to the Wazuh server. This means changing Wazuh's configuration.

The configuration file that needs to be edited is **ossec.conf**. I will use VS Code for this, opening it as administrator by right-clicking the app and selecting **Run as administrator** (Fig. 3). Once VS Code is open, navigate to C:\Program Files (x86)\ossec-agent and locate **ossec.conf** to make the necessary edits (Fig. 4).

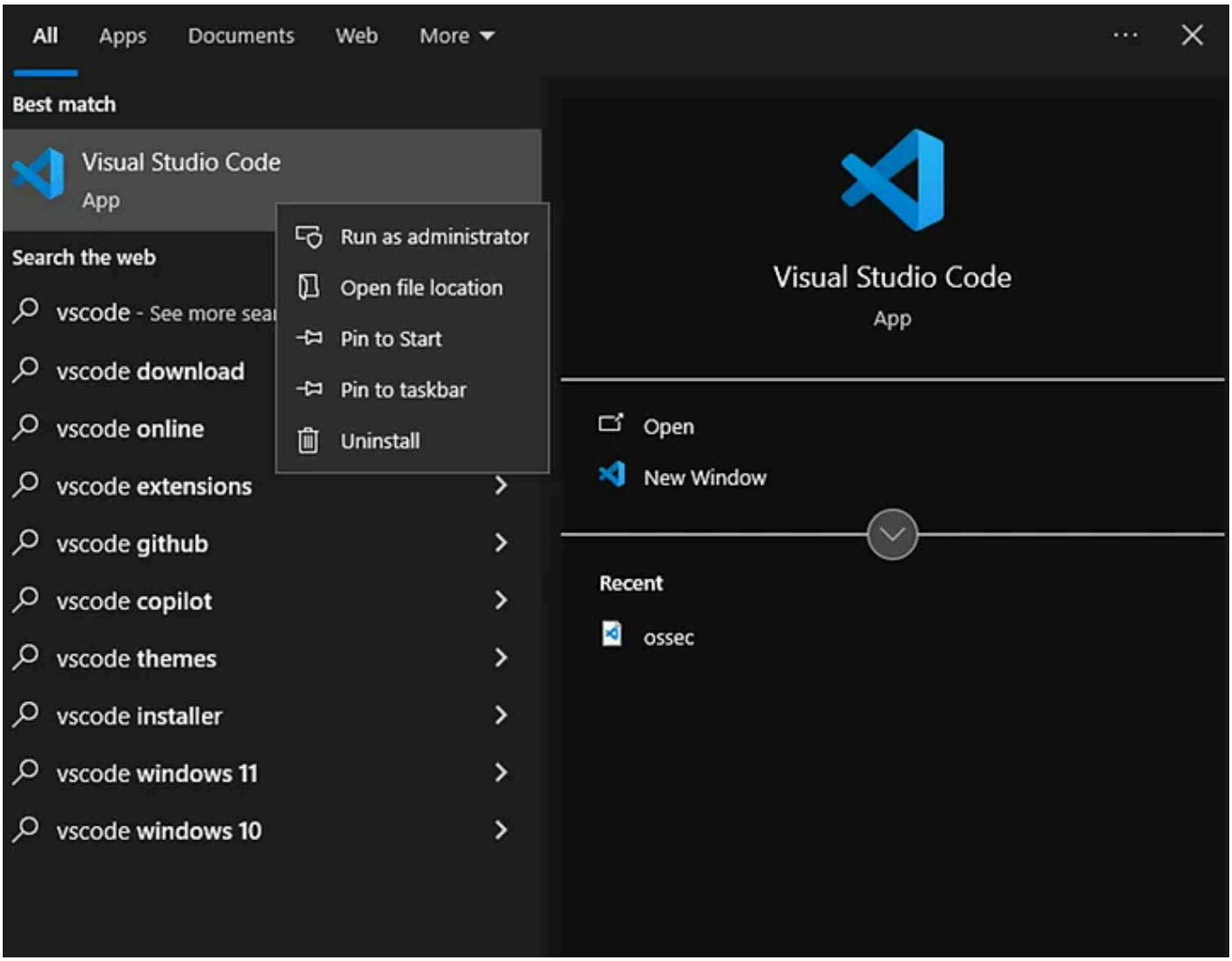

Fig. 3

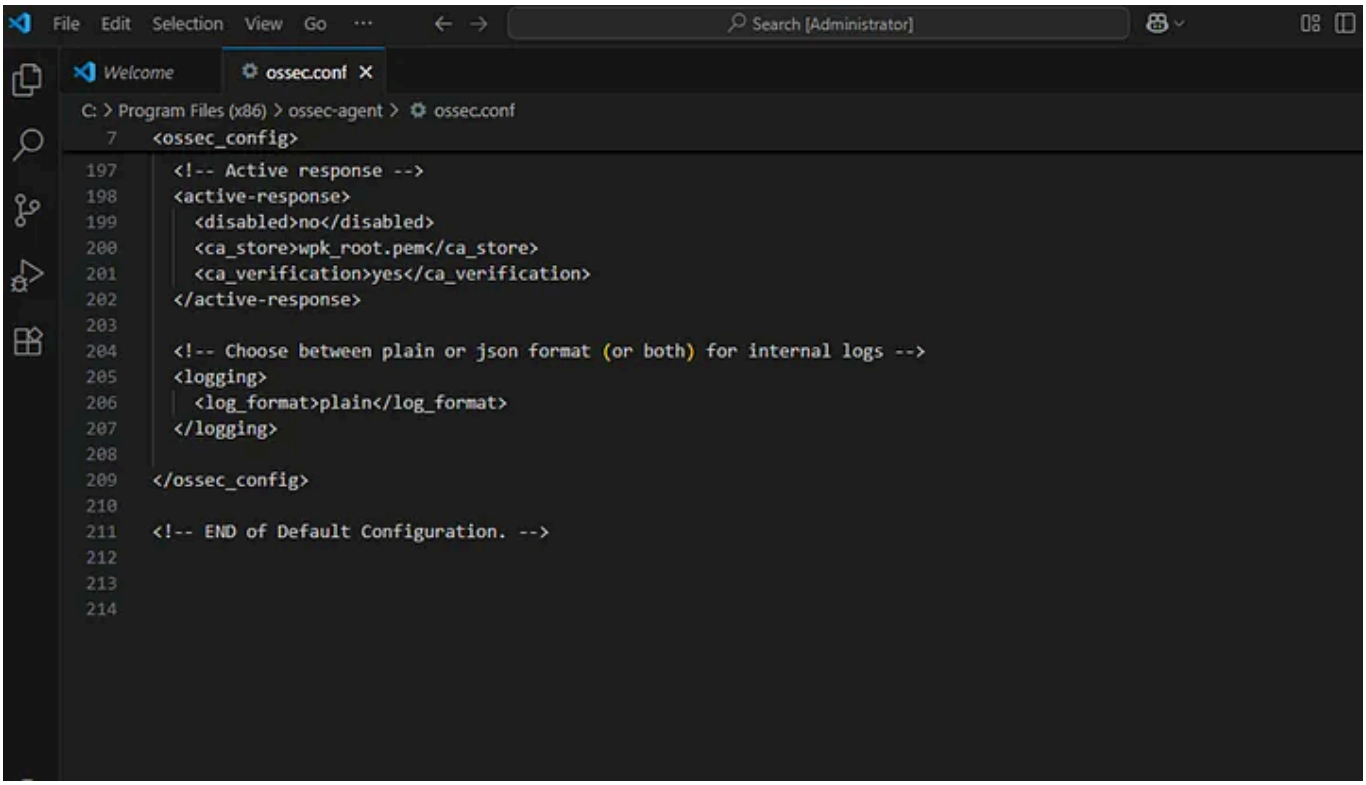

Fig. 4

The config file is in XML format. We'll add a new **<localfile>** entry inside the **<ossec_config>** tag. The **<location>** should match the Windows Event Viewer log *Microsoft-Windows-Sysmon/Operational* and the **<log_format>** should be set to *eventchannel*. You can also include a comment for clarity, e.g., *<! — Sysmon added as a log source -->*.

Append the following at the end of the file (Fig. 5):

```
<!-- Sysmon added as a log source -->
<localfile>
  <location>Microsoft-Windows-Sysmon/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>
```

```
<ossec_config>
  <!-- Active response -->
  <active-response>
    <disabled>no</disabled>
    <ca_store>wpk_root.pem</ca_store>
    <ca_verification>yes</ca_verification>
  </active-response>

  <!-- Choose between plain or json format (or both) for internal logs -->
  <logging>
    <log_format>plain</log_format>
  </logging>

  <!-- Sysmon added as a log source -->
  <localfile>
    <location>Microsoft-Windows-Sysmon/Operational</location>
    <log_format>eventchannel</log_format>
  </localfile>

</ossec_config>

<!-- END of Default Configuration. -->
```

Fig. 5

Save the file and restart the Wazuh agent service (`Restart-Service wazuhsvc` in PowerShell, Fig. 6).

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Restart-Service WazuhSvc
PS C:\Windows\system32>
```

Fig. 6

Back in the Wazuh dashboard, go to the Threat Hunting section. From the homepage dashboard, click the threat hunting card (Fig. 7).

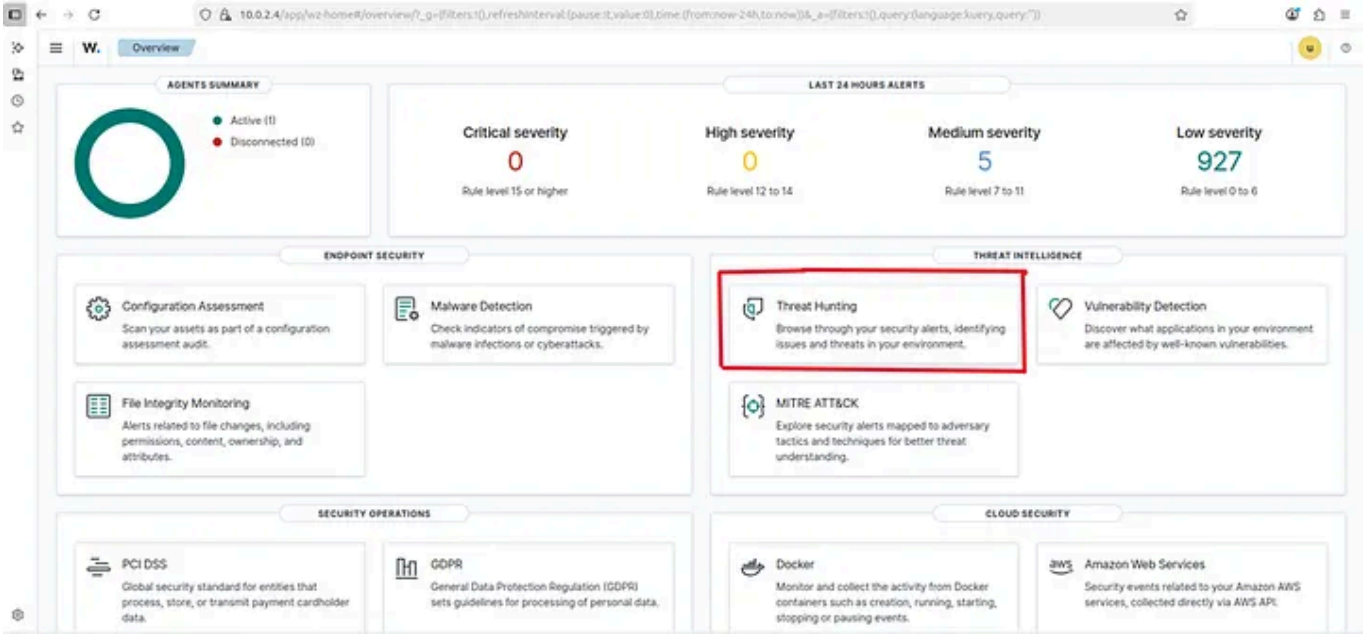

Fig. 7

This gives you an overview of alerts (Fig. 8). You can add filters, change the time range, and search logs.

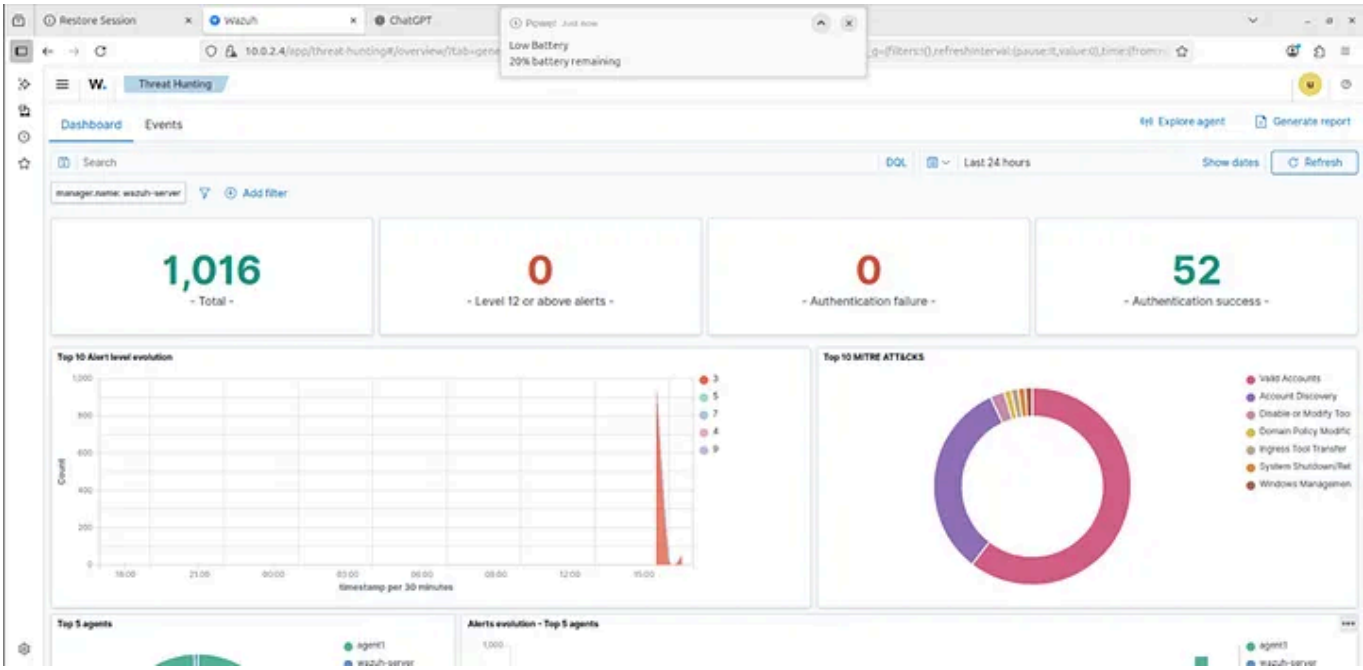

Fig. 8

You can go to the Events tab to see the raw events. These are the default detections from the Sysmon config. You'll see many events should show the

source as Sysmon (Fig. 9).

Document Details

View surrounding documents

View single document

data.win.eventdata.sourceUser	DESKTOP-1100G2N\\frederick
data.win.eventdata.targetImage	C:\\Windows\\Explorer.EXE
data.win.eventdata.targetProcessGUID	{49d79454-212e-6892-7d00-000000000e00}
data.win.eventdata.targetProcessId	4660
data.win.eventdata.targetUser	DESKTOP-1100G2N\\frederick
data.win.eventdata.utcTime	2025-08-05 15:56:16.587
data.win.system.channel	Microsoft-Windows-Sysmon/Operational
data.win.system.computer	DESKTOP-1100G2N
data.win.system.eventID	10
data.win.system.eventRecordID	10583
data.win.system.keywords	0x8000000000000000
data.win.system.level	4
data.win.system.message	"Process accessed: RuleName: technique_id=T1036,technique_name=Masquerading UtcTime: 2025-08-05 15:56:16.587 SourceProcessGUID: {49d79454-214c-6892-b500-000000000e00} SourceProcessId: 9168 SourceThreadId: 9452 SourceImage: C:\Users\frederick\AppData\Local\Microsoft\OneDrive\OneDrive.exe
data.win.system.opcode	0
data.win.system.processID	2884
data.win.system.providerGuid	{5770385f-c22a-43e0-bf4c-06f5698ffbd9}

Fig. 9

> *# At Gibraltar Airport, a runway crosses the main road into the territory, and cars must stop at traffic lights whenever a plane lands or takes off.*

Step 2: DOWNLOADING THE DEERSTEALER MALWARE

DeerStealer is a type of malware that targets sensitive data on infected machines. To generate telemetry for detection, I will download and execute

this malware on the VM running the Wazuh agent.

The malware can be downloaded from AnyRun, but you'll need to create an account to access it. You can find it here.

NB: Please this should be done in a completely virtualized environment.

Select a sample from the list under **LAST SEEN AT** (Fig. 10), click on it and click **Get sample** to download the sample (Fig. 11, 12).

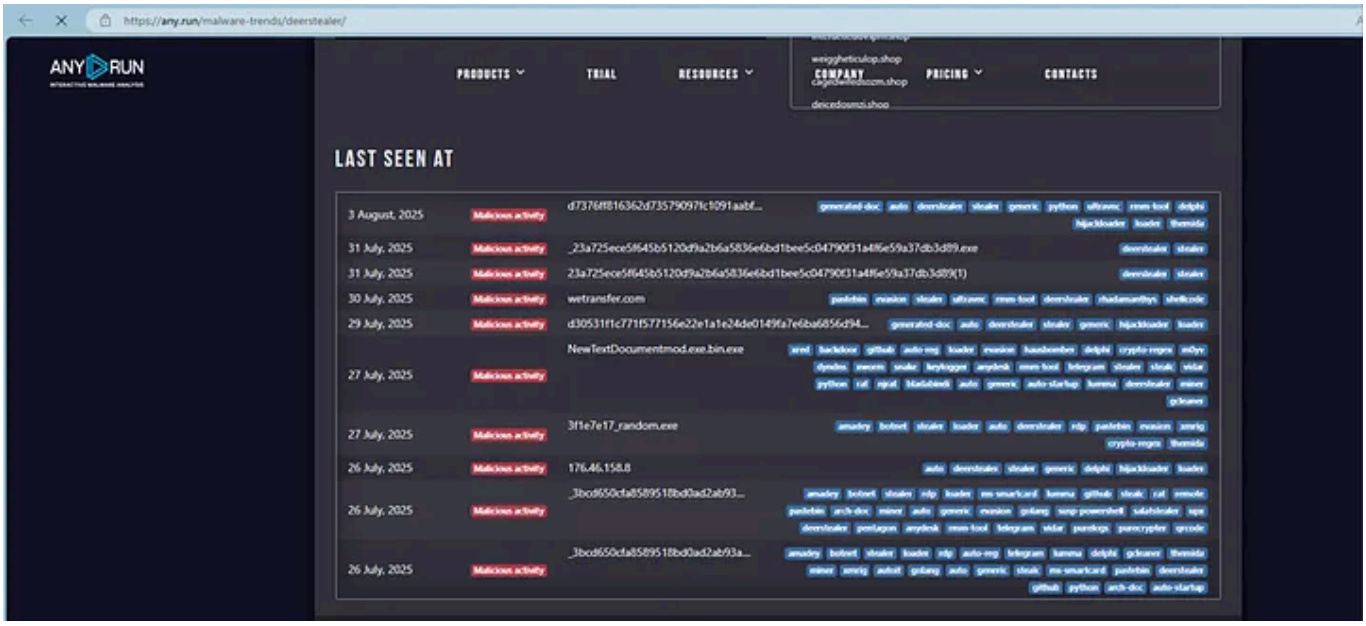

Fig. 10

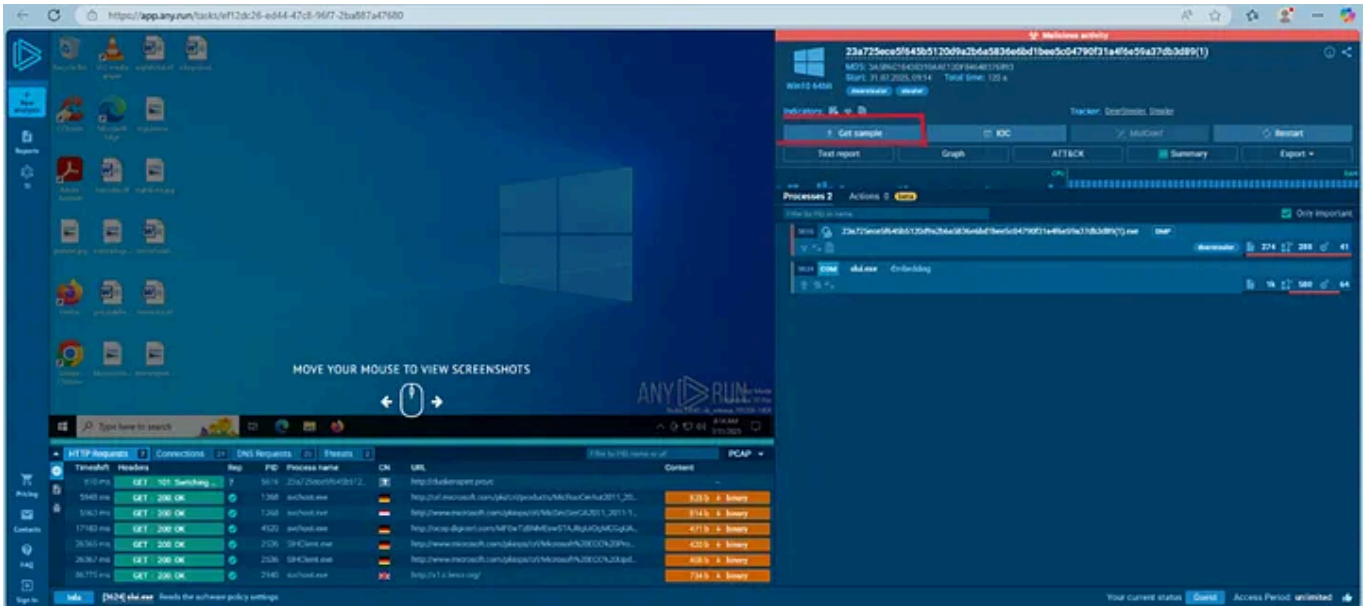

Fig. 11

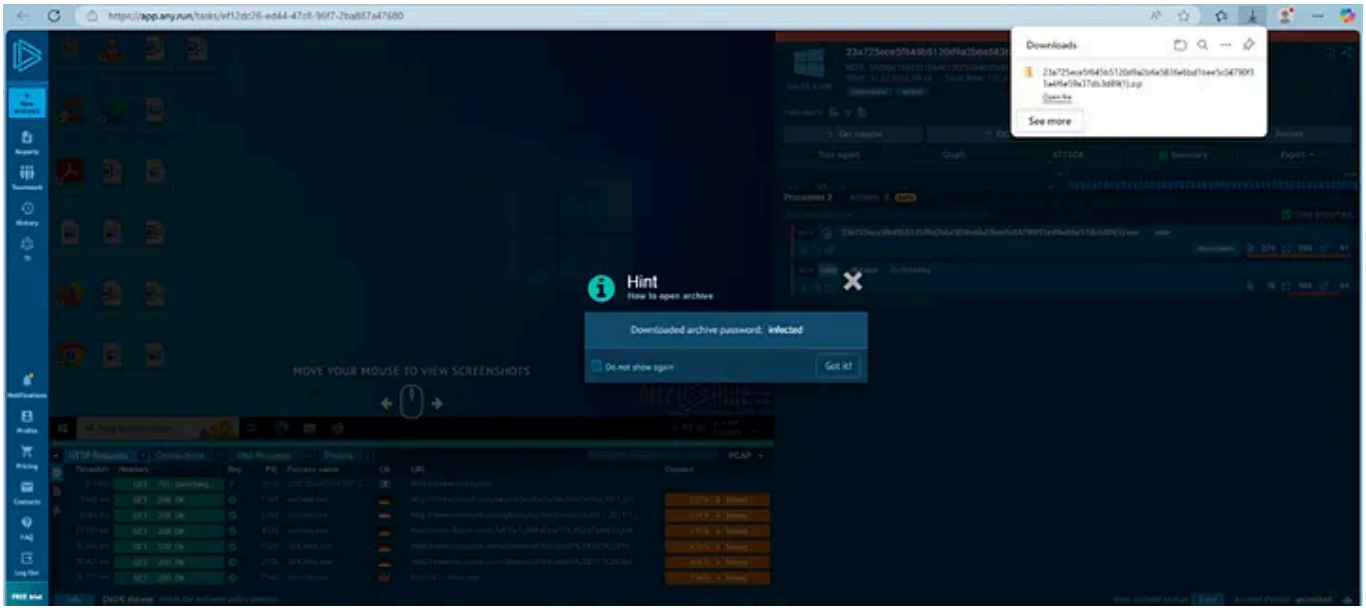

Fig. 12

After downloading, extract the file. You will be prompted for a password; use **infected**. Remove the .bin extension from the extracted file so it ends with .exe. I also renamed the long filename to deerstealer.exe to make it easier to locate (Fig. 12, 13).

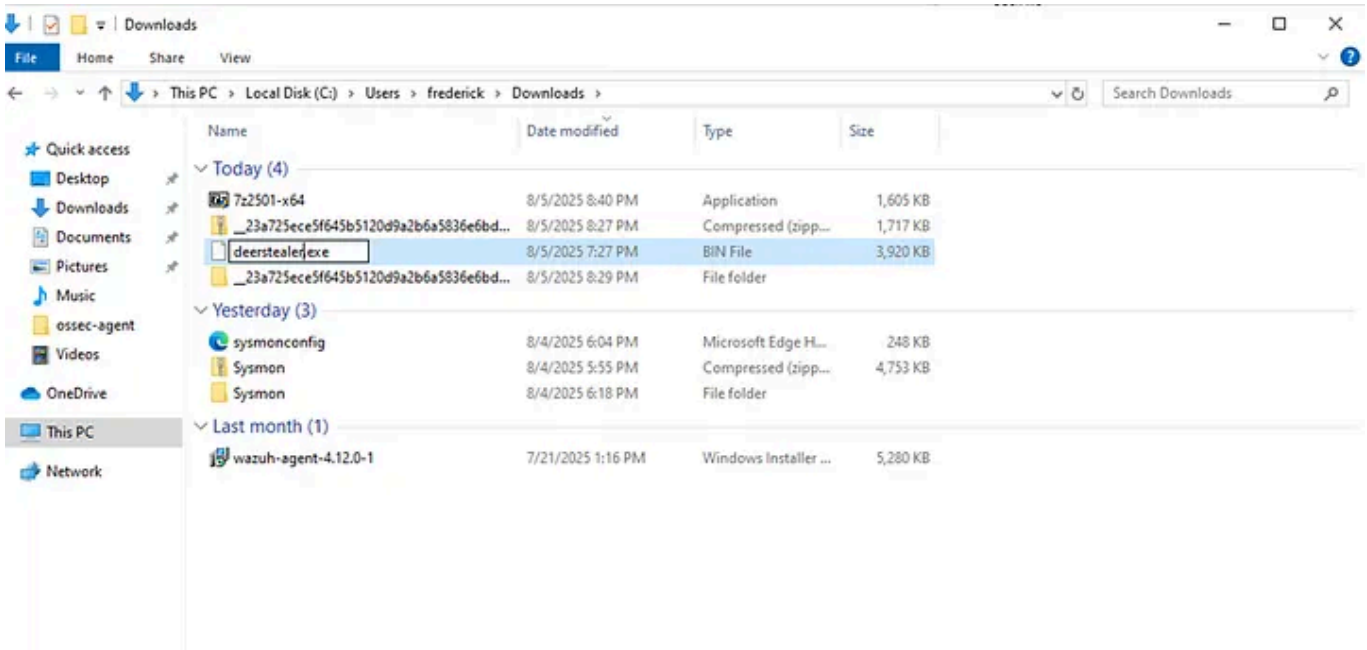

Fig. 13

Step 3: CREATING DETECTION RULES FOR DEERSTEALER MALWARE

Now, let's create a detection rule group for DeerStealer. I'll create a rule to detect DeerStealer activity on the Windows endpoint.

On the Wazuh server, navigate through the CLI as root using `sudo -i`. Wazuh rules are located in */var/ossec/etc/rules*, where you'll find local_rules.xml. You can also access these rules from the dashboard under **Server Management > Rules** (Fig. 14). Here, you'll see all preconfigured rules, including the Sysmon-related rules we added earlier (Fig. 15).

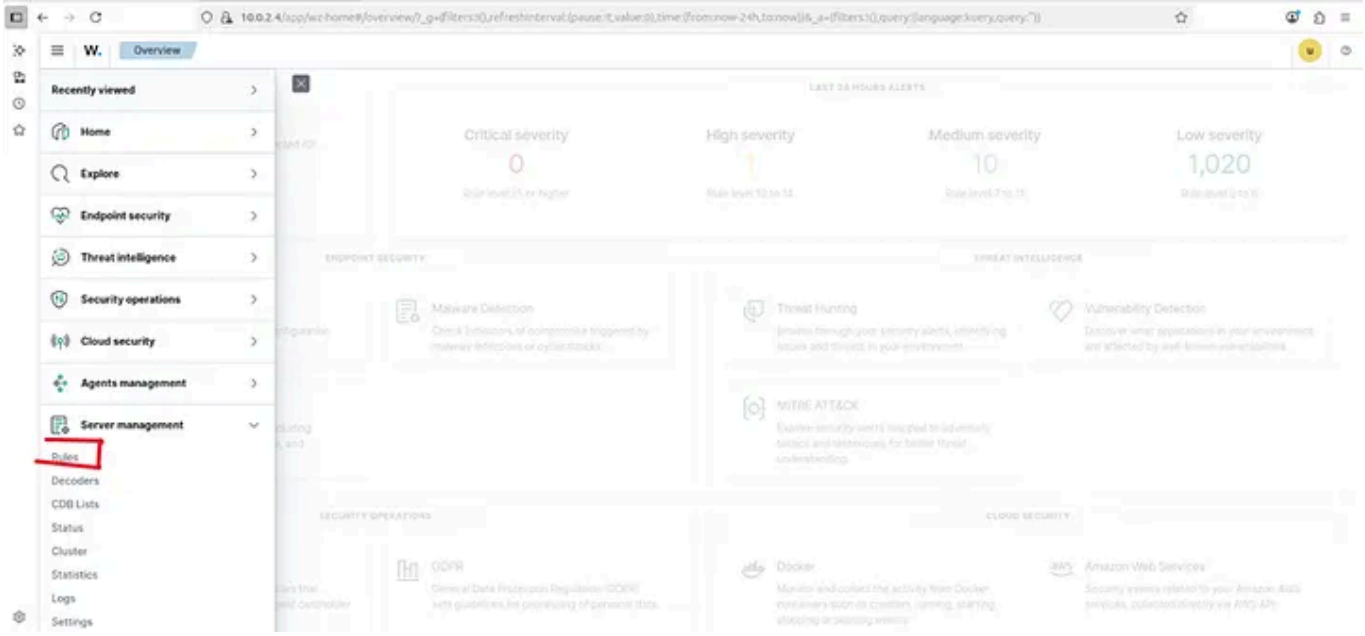

Fig. 14

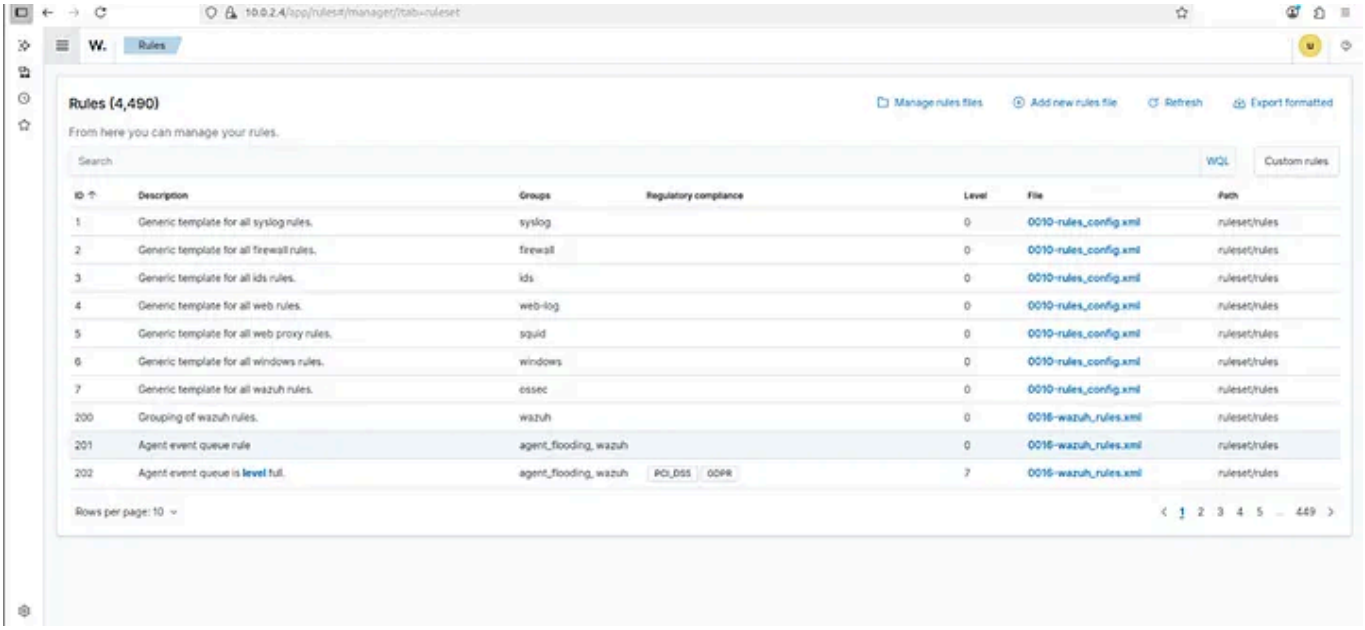

Fig. 15

We can also see the local_rules that can be located at */var/ossec/etc/rules* on the Wazuh server through the CLI (Fig. 16).

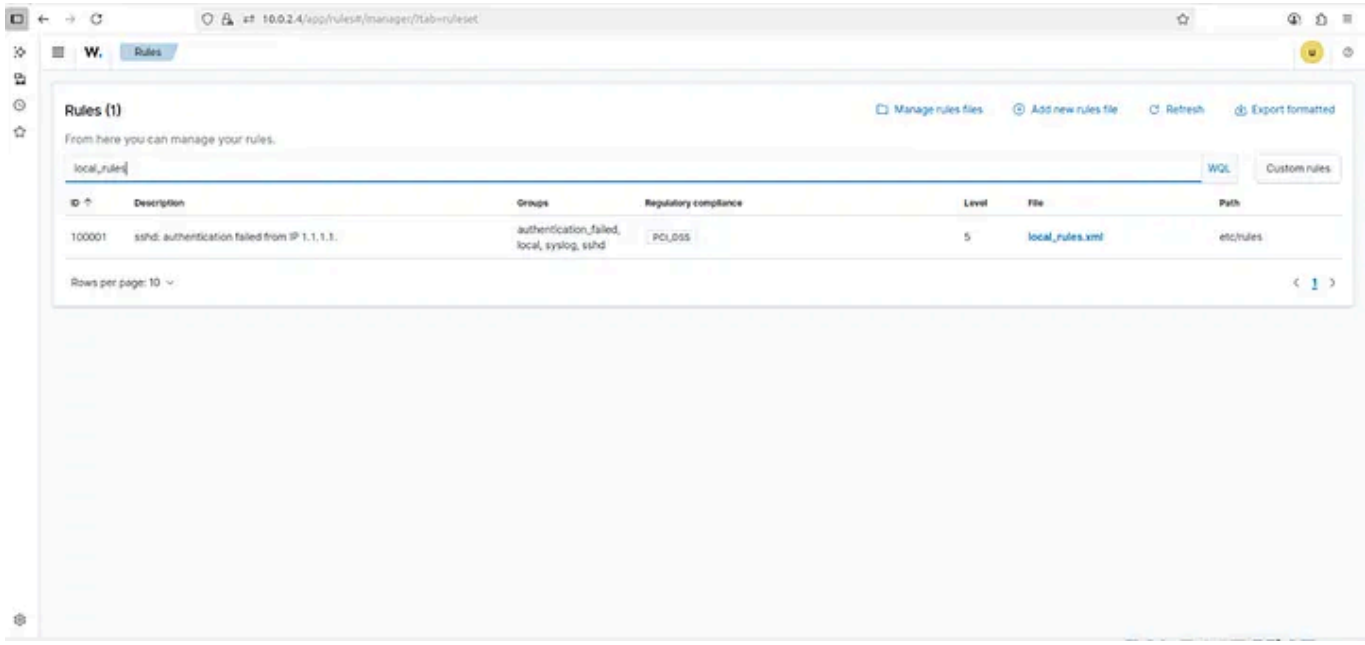

Fig. 16

To set the rule, I will use the dashboard GUI. Click on **Add new rules** file and fill in as appropriate.

Fill this in (Fig. 17):

```
<group name="deerstealer, stealer-malware,">

<!-- Persistence detection -->
<rule id="111200" level="12">
  <if_sid>61609</if_sid>
  <field name="win.eventdata.image" type="pcre2">\\\\(Windows|Users)\\\\.+\\\\(s
  <field name="win.eventdata.imageLoaded" type="pcre2">\\\\Windows\\\\SysWOW64.+
  <description>Possible DeerStealer malware detected. New scheduled task: $(win.
  <mitre>
    <id>T1053.005</id>
  </mitre>
</rule>

<!-- Malicious file creation -->
<rule id="111201" level="12">
  <if_sid>61613</if_sid>
  <field name="win.eventdata.image" type="pcre2">\\\\(Windows|Users)\\\\.+\\\\(s
  <field name="win.eventdata.targetFilename" type="pcre2">\\\\(Windows|Users)\\\
  <description>Possible DeerStealer malware activity detected. Malicious file cr
```

```
    <mitre>
      <id>T1059</id>
      <id>T1105</id>
    </mitre>
  </rule>

  <!-- Executable dropped in malicious location -->
  <rule id="111202" level="12">
    <if_sid>92213</if_sid>
    <field name="win.eventdata.image" type="pcre2">\\\\(Windows|Users)\\\\.+\\\\(s
    <field name="win.eventdata.targetFilename" type="pcre2">\\\\Users\\\\.+\\\\App
    <description>Possible DeerStealer malware activity detected. Executable file d
    <mitre>
      <id>T1105</id>
      <id>T1059</id>
    </mitre>
  </rule>

  <!-- Process creation -->
  <rule id="111203" level="12">
    <if_sid>61603</if_sid>
    <field name="win.eventdata.commandLine" type="pcre2">\\\\Users\\.+\\\\AppData\
    <description>Possible DeerStealer malware executable: $(win.eventdata.commandL
    <mitre>
      <id>T1543</id>
    </mitre>
  </rule>

  <!-- Network connection to C2 server -->
  <rule id="111204" level="12">
    <if_sid>61605</if_sid>
    <field name="win.eventdata.image" type="pcre2">\\\\Users\\\\.+\\\\AppData\\\\L
    <field name="win.system.message" type="pcre2">Network connection detected</fie
    <field name="win.eventdata.destinationPort" type="pcre2">80</field>
    <description>Possible DeerStealer network connection to C2 server: $(win.event
    <mitre>
      <id>T1105</id>
    </mitre>
  </rule>

  <!-- Registry tampering - targeting HKLM -->
  <rule id="111205" level="12">
    <if_sid>61614, 61615</if_sid>
    <field name="win.eventdata.image" type="pcre2">\\\\(Windows|Users)\\\\.+\\\\(s
    <field name="win.eventdata.eventType" type="pcre2">(CreateKey|SetValue)</field
    <field name="win.eventdata.targetObject" type="pcre2">HKLM\\\\(System|SOFTWARE
    <description>Possible DeerStealer malware executable, $(win.eventdata.image) p
    <mitre>
      <id>T1543</id>
      <id>T1053.005</id>
```

```
    </mitre>
  </rule>

  <!-- Registry tampering - targeting HKU for persistence on next logon -->
  <rule id="111206" level="12">
    <if_sid>61614, 61615, 92300</if_sid>
    <field name="win.eventdata.image" type="pcre2">\\\\(Windows|Users)\\\\.+\\\\(s
    <field name="win.eventdata.eventType" type="pcre2">(CreateKey|SetValue)</field
    <field name="win.eventdata.targetObject" type="pcre2">HKU\\\\.+\\\\Software\\\
    <description>Possible DeerStealer malware executable, $(win.eventdata.image) p
    <mitre>
      <id>T1547</id>
      <id>T1053.005</id>
    </mitre>
  </rule>

  </group>
```

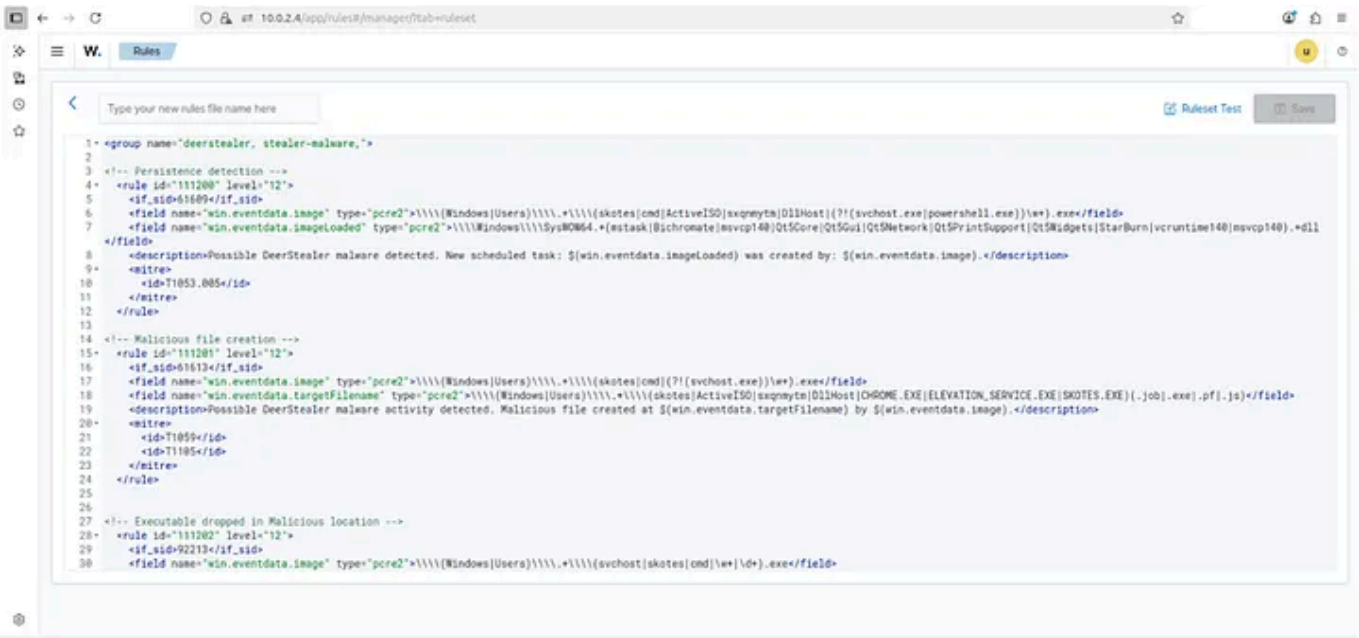

Fig. 17

This Wazuh rule group detects **DeerStealer malware** on Windows systems by monitoring its key behaviors. It flags **suspicious executables** creating files in Temp or AppData directories, loading unusual DLLs, and running processes from temporary locations. **Scheduled task creation** and **registry**

modifications in HKLM and HKU are tracked to identify persistence mechanisms. The rules also detect **network connections** to potential command-and-control servers on port 80. Covering execution, file creation, persistence, and exfiltration, this set maps to multiple **MITRE ATT&CK techniques** (T1053.005, T1059, T1105, T1543, T1547) and provides high-severity alerts for rapid detection and mitigation.

You can find extensive documentation on creating Wazuh rules here.

After creating your rule, save it with a suitable name (I saved mine as deerstealer_rules.xml) and restart the Wazuh dashboard for the rule to take effect (Fig. 18).

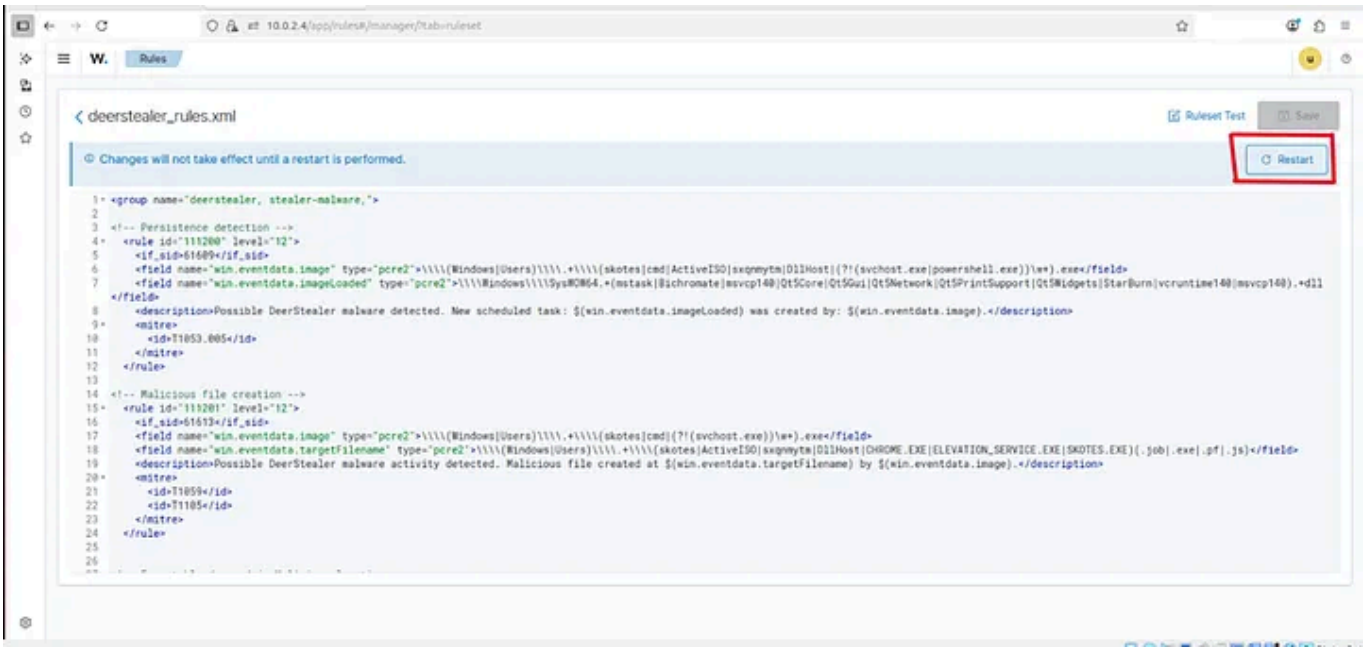

Fig. 18

Step 4: VALIDATING ALERTS AND OBSERVING MALWARE BEHAVIOR

Now, run the extracted file in the virtualized environment to generate telemetry. Double-click the executable; you might notice that nothing

appears to happen. This is normal because the malware runs quietly in the background.

Next, go to the Wazuh dashboard, navigate to **Threat Hunting** > **Events**, and you should see an alert triggered by the rule group you configured.

Here, I can see an alert with the description **"Possible DeerStealer malware executable"** (Fig. 19).

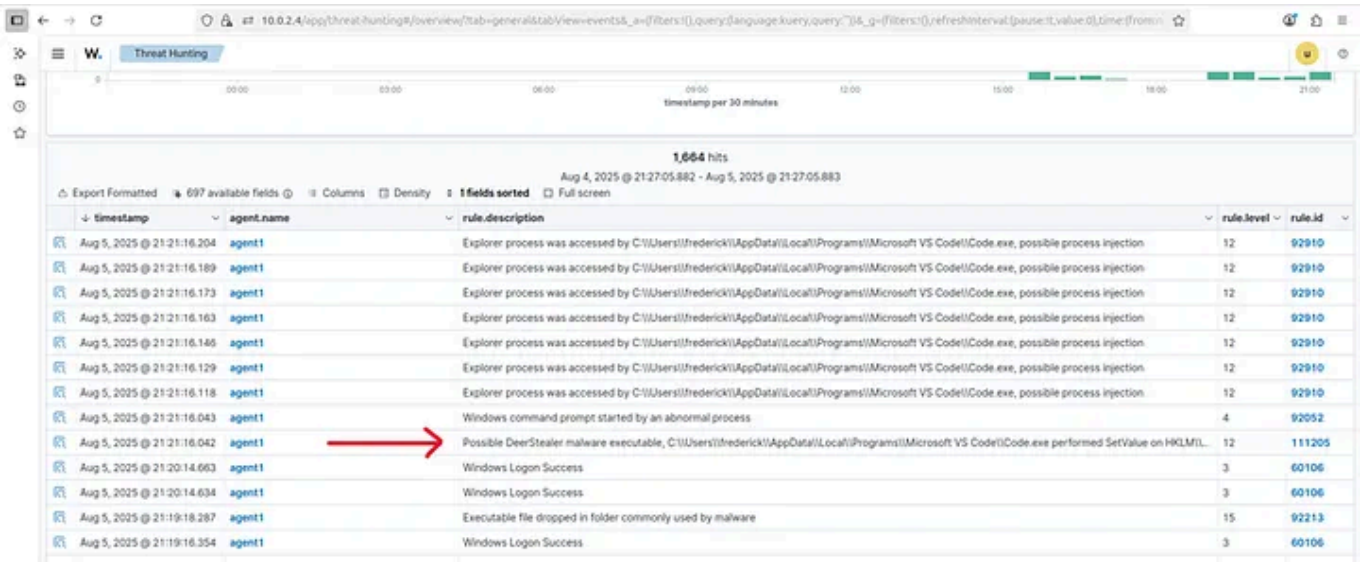

Fig .19

Clicking on the event provides more detailed information about the activity detected (Fig. 20).

Document Details

View surrounding documents

View single document

e	
data.win.system.systemTime	2025-08-05T20:21:14.7299418Z
data.win.system.task	13
data.win.system.threadID	3724
data.win.system.version	2
decoder.name	windows_eventchannel
full_log	{"win":{"system":{"providerName":"Microsoft-Windows-Sysmon","providerGuid":"{5770385f-c22a-43e0-bf4c-06f5698ffbd9}","eventID":"13","version":"2","level":"4","task":"13","opcode":"0","keywords":"0x8000000000000000","systemTime":"2025-08-05T20:21:14.7299418Z","eventRecordID":"14519","processID":"2836","threadID":"3724","channel":"Microsoft-Windows-Sysmon/Operational","computer":"DESKTOP-11O0G2N","severityValue":"INFORMATION","message":"\"Registry value set:\r\nRuleName: -\r\nEventType: SetValue\r\nUtcTi
id	1754425276.4489564
input.type	log
location	EventChannel
manager.name	wazuh-server
rule.description	Possible DeerStealer malware executable, C:\\Users\\frederick\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe performed SetValue on HKLM\\System\\CurrentControlSet\\Services\\bam\\State\\UserSettings\\S-1-5-21-964889993-316285029-1350080657-1001\\\\Device\\HarddiskVolume2\\Windows\\System32\\cmd.exe.
rule.firedtimes	1
rule.groups	deerstealer, stealer-malware
rule.id	111205
rule.level	12
rule.mail	true
rule.mitre.id	T1543 T1053.005

Fig. 20

This concludes the series on Detection Engineering. Feel free to experiment with Wazuh rules. You can simulate malicious activity, run attacks from another VM to the one with the agent, and create rules to detect them. The more you practice, the deeper your understanding will be.

I may add more rules over time, which I will also document in a Medium article.

Thank you for following along.
Hasta la vista!

Detection Engineering | Wazuh | Malware | Threat Detection | Cybersecurity

F

Written by Frederick Adigun

3 followers · 2 following

Edit profile

No responses yet

F Frederick Adigun

What are your thoughts?

More from Frederick Adigun