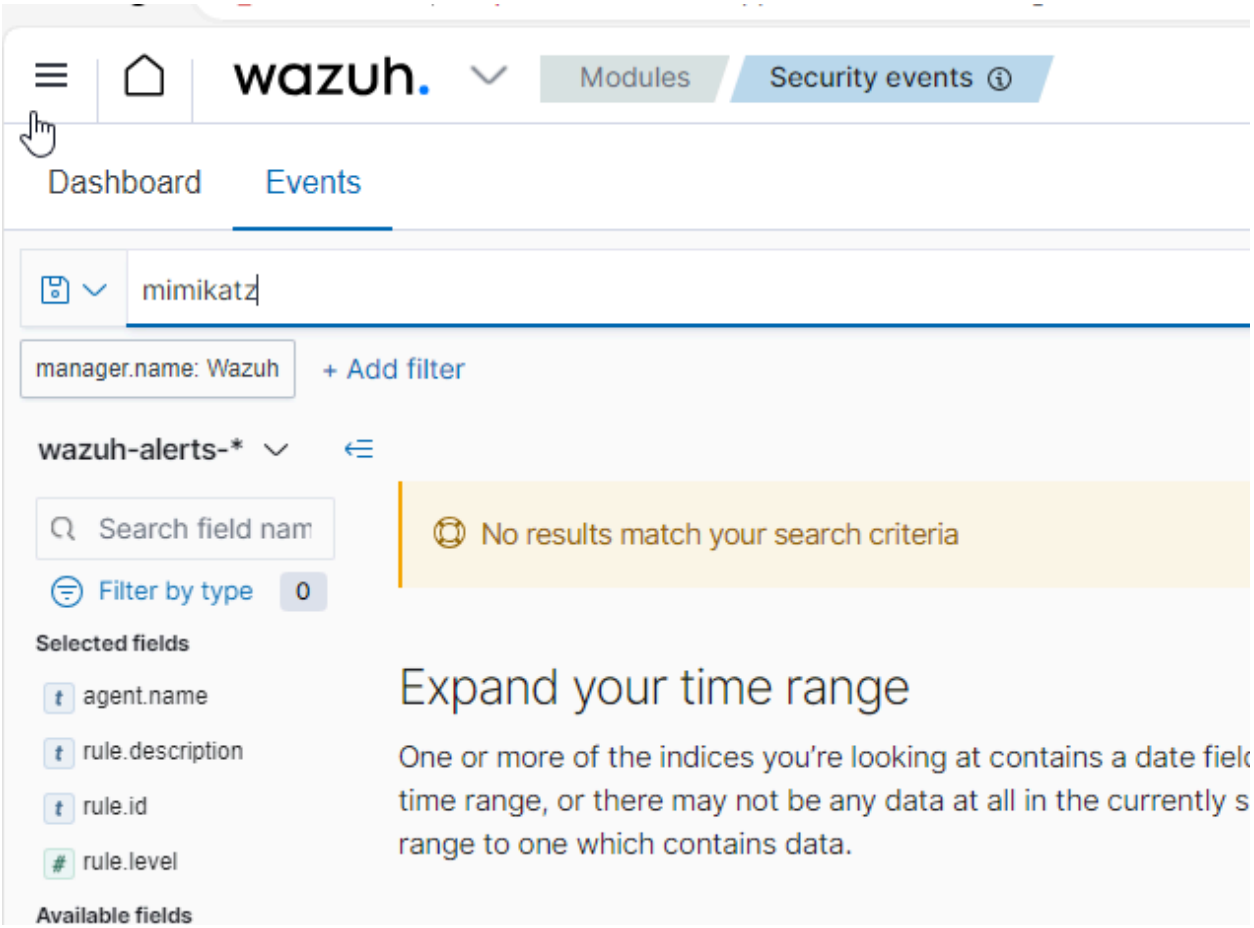
Wazuh Set up, Rule Creation, and configure Sysmon to be ingested into Wazuh.

Download and initiate Mimikatz on my windows VM
-prior to this I had to add my downloads folder to the security exclusion list
-Open powershell as admin>cd into mimikatz path and initiate .\mimikatz.exe

Go to my Wazuh under events. Type in Mimikatz. However I initially did not see events as Sysmon did not trigger any alerts or rules from Wazuh. This is because by default Wazuh does not log everything. It will only log when a rule or alert is triggered. But we can change this by going to the Wazuh manager and configuring the ossec.comf file to make it so it logs everything

or create a rule that looks at specific events.



Lets modify the ossec.conf to log everything. To do this I will need to log into the wazuh CLI After logging into the root@Wazuh we will make a copy of the ossec.conf file before we make changes and place it in my home dir.

```
*** System restart required ***
Last login: Mon Mar 11 16:43:55 2024 from 47.205.211.221
root@Wazuh:~# cp /var/ossec/etc/ossec.conf ~/ossec-backup.conf
```

```
root@Wazuh:~#   ls
ossec-backup.conf  snap  wazuh-install-files  wazuh-install-files.tar  wazuh-install.sh
root@Wazuh:~# nano /var/ossec/etc/ossec.conf
```

Logall and logall_json are set to no, we need to change these to yes

```
<!--
  Wazuh - Manager - Default configuration for ubuntu 22.04
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_t
  </global>
```

```
  <logall>yes</logall>
  <logall_json>yes</logall_json>
```

Save file and restart Wazuh manager

```
root@Wazuh:~# nano /var/ossec/etc/ossec.conf
root@Wazuh:~# systemctl restart wazuh-manager.service
```

This will force Wazuh to archive all the files in a filed called Archives

```
root@Wazuh:~# cd /var//ossec/logs/archives/
root@Wazuh:/var/ossec/logs/archives# ls
2024  archives.json  archives.log
root@Wazuh:/var/ossec/logs/archives#
```

To have Wazuh ingest the logs we need to change the config in file beat

```
2024  archives.json  archives.log
root@Wazuh:/var/ossec/logs/archives# nano /etc/filebeat/filebeat.yml
```

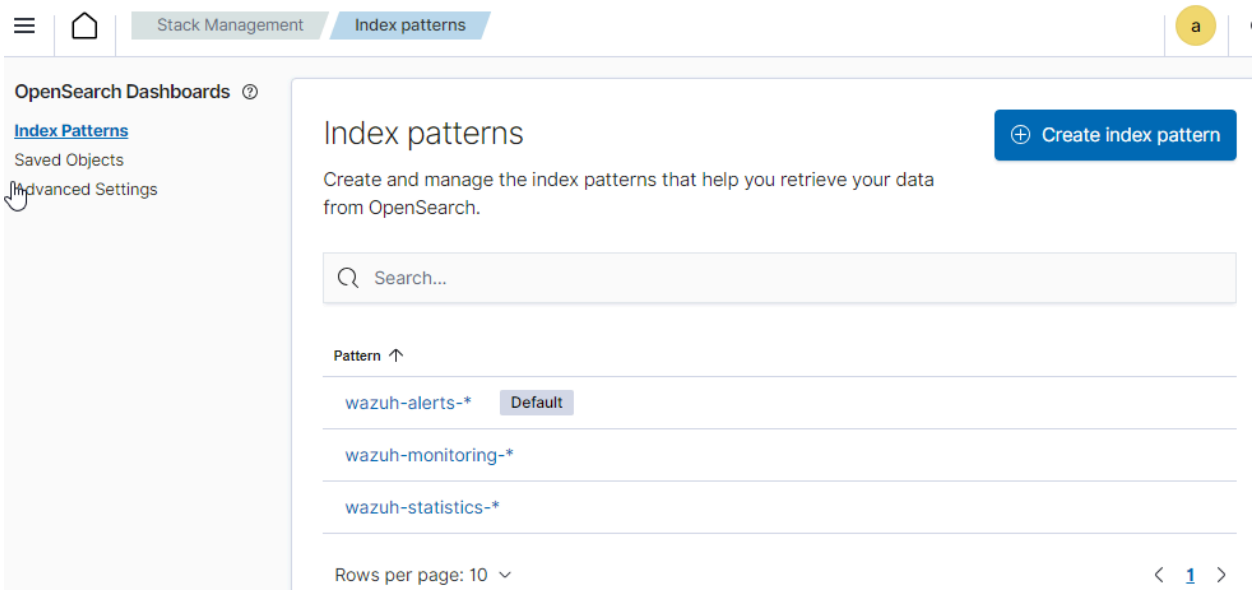
Nano into filebeat.yml and change archives enabled to ‘true’

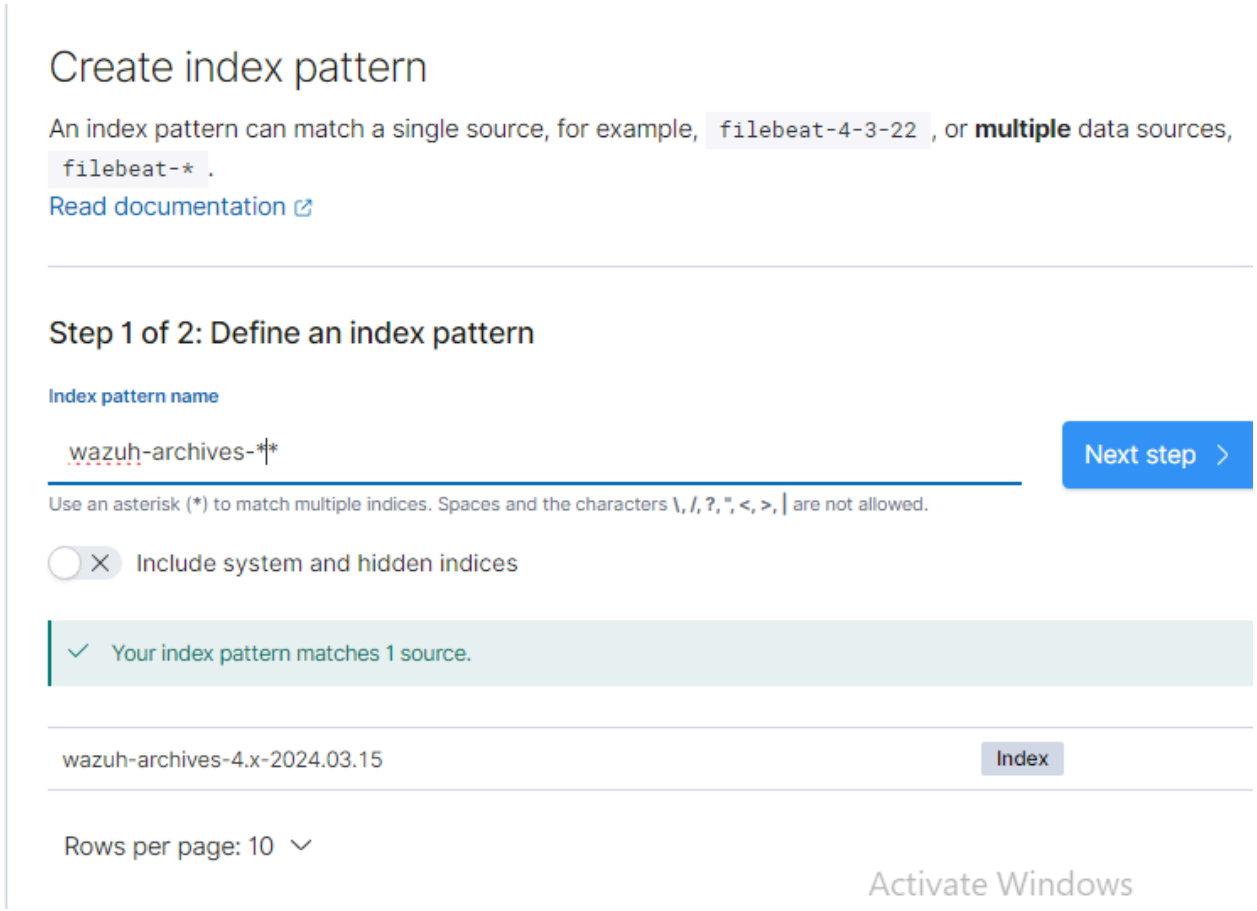
```
#        - <elasticsearch_ip_node_2>:9200
#        - <elasticsearch_ip_node_3>:9200

output.elasticsearch:
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificate_authorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/wazuh-server.pem"
  ssl.key: "/etc/filebeat/certs/wazuh-server-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: true

logging.level: info
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
```

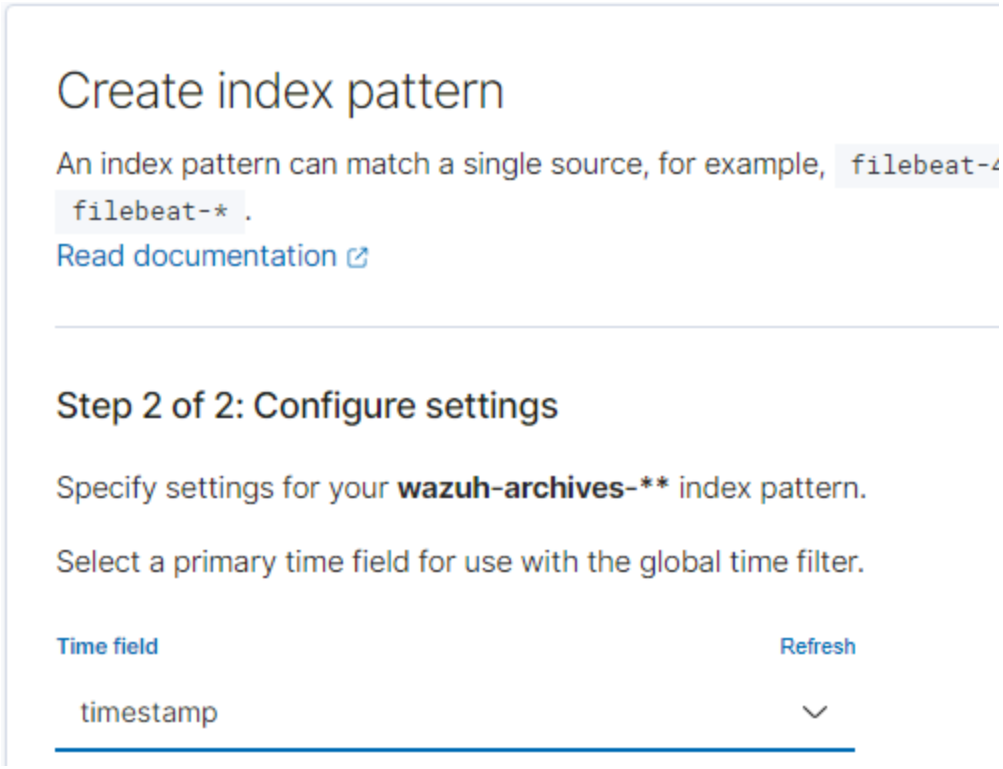
Be sure to restart filebeat as with any config changes

```
root@Wazuh:/var/ossec/logs/archives# systemctl restart filebeat
root@Wazuh:/var/ossec/logs/archives#
```

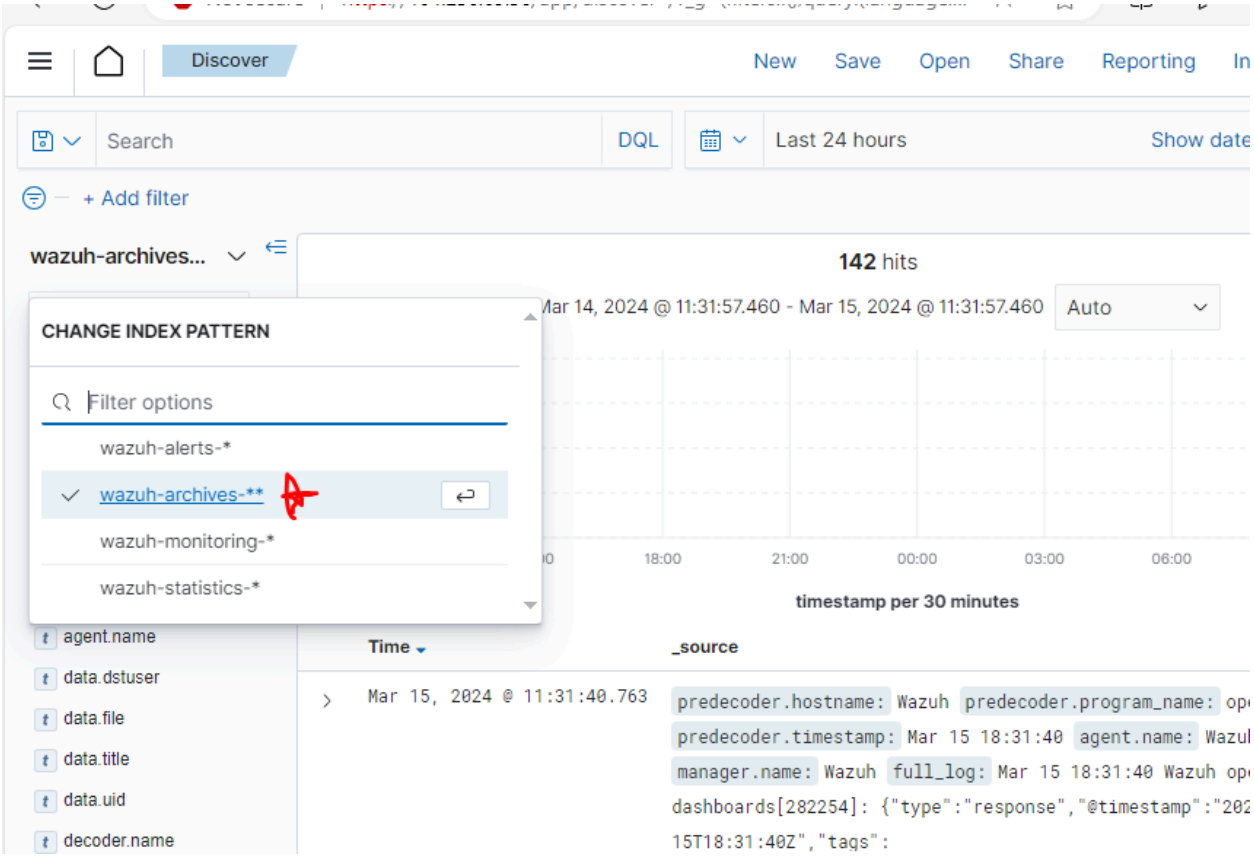
Now I will go back to wazuh dashboard>stack management> index pattern> create index
-I will create index for the archives so we can see the all the logs regardless if wazuh triggered an alert.


Stack Management
Index patterns
a
OpenSearch Dashboards
Index Patterns
Saved Objects
Advanced Settings
Index patterns
Create index pattern
Create and manage the index patterns that help you retrieve your data from OpenSearch.
Search...
Pattern
wazuh-alerts-*
Default
wazuh-monitoring-*
wazuh-statistics-*
Rows per page: 10
1


Create index pattern
An index pattern can match a single source, for example, filebeat-4-3-22 , or multiple data sources, filebeat-* .
Read documentation
Step 1 of 2: Define an index pattern
Index pattern name
wazuh-archives-**
Next step
Use an asterisk (*) to match multiple indices. Spaces and the characters \, /, ?, ", <, >, | are not allowed.
Include system and hidden indices
Your index pattern matches 1 source.
wazuh-archives-4.x-2024.03.15
Index
Rows per page: 10
Activate Windows



Create index pattern
Go back to discover and select the wazuh-archive index we just made.

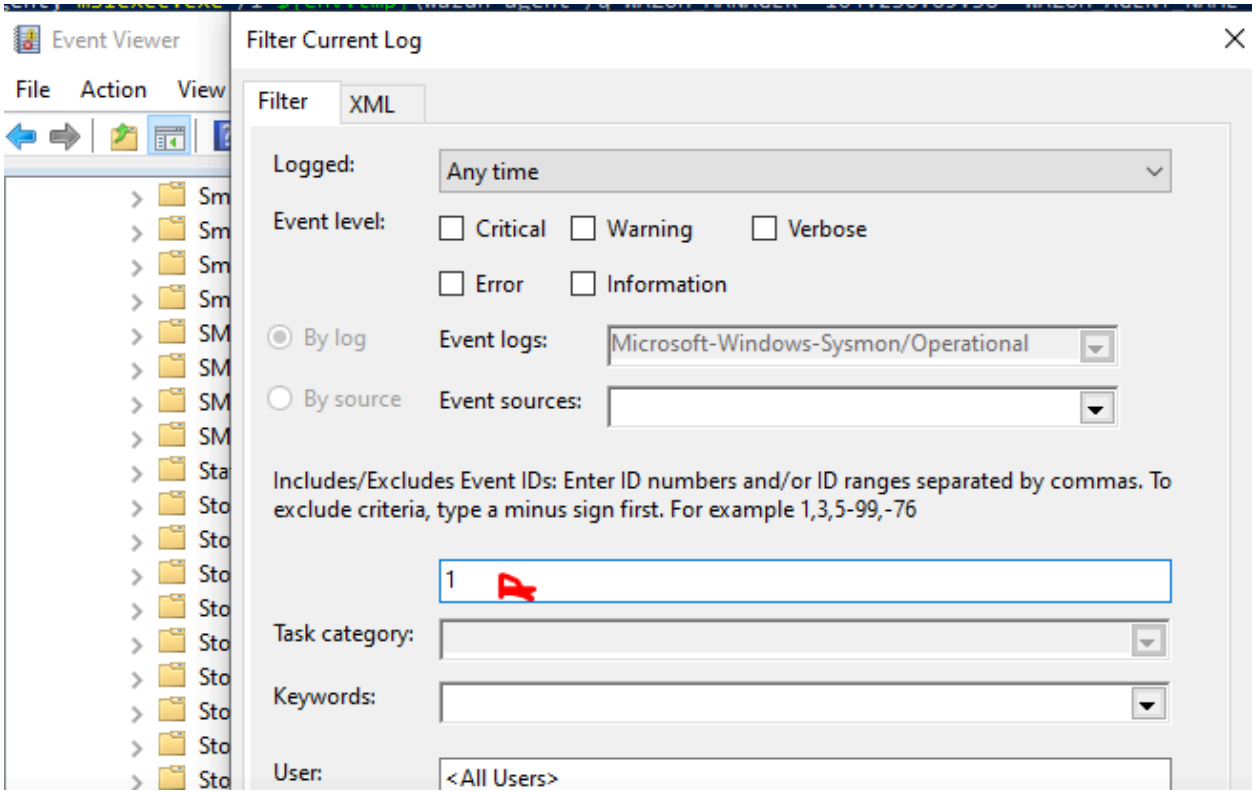


This is important as Wazuh by default will not log anything. We enabled wazuh to archive everything for our test regardless if it triggers or not.

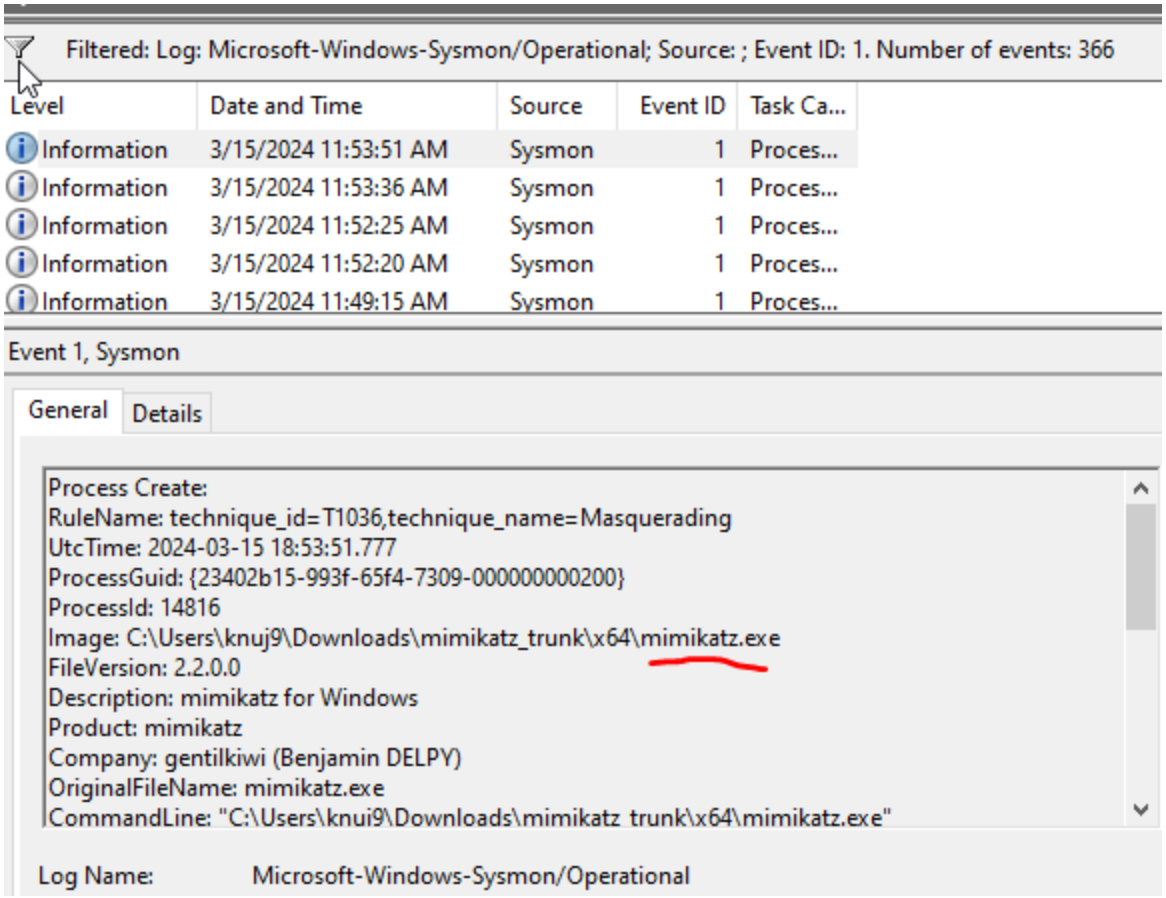
Lets restart mimikatz

```
mimikatz # exit
Bye!
PS C:\Users\knuj9\Downloads\mimikatz_trunk\x64> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # _
```

And check event view to ensure Sysmon is capturing mimikatz



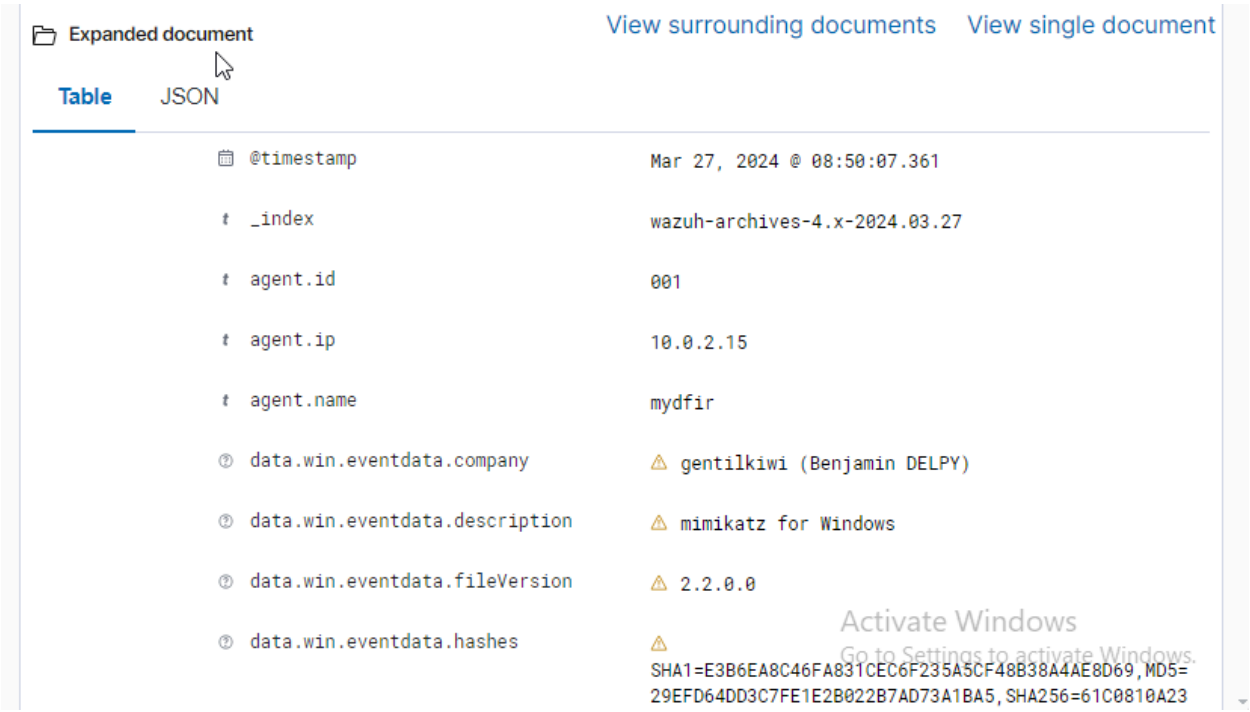
We will filter with Event ID 1 as this is for process creations.



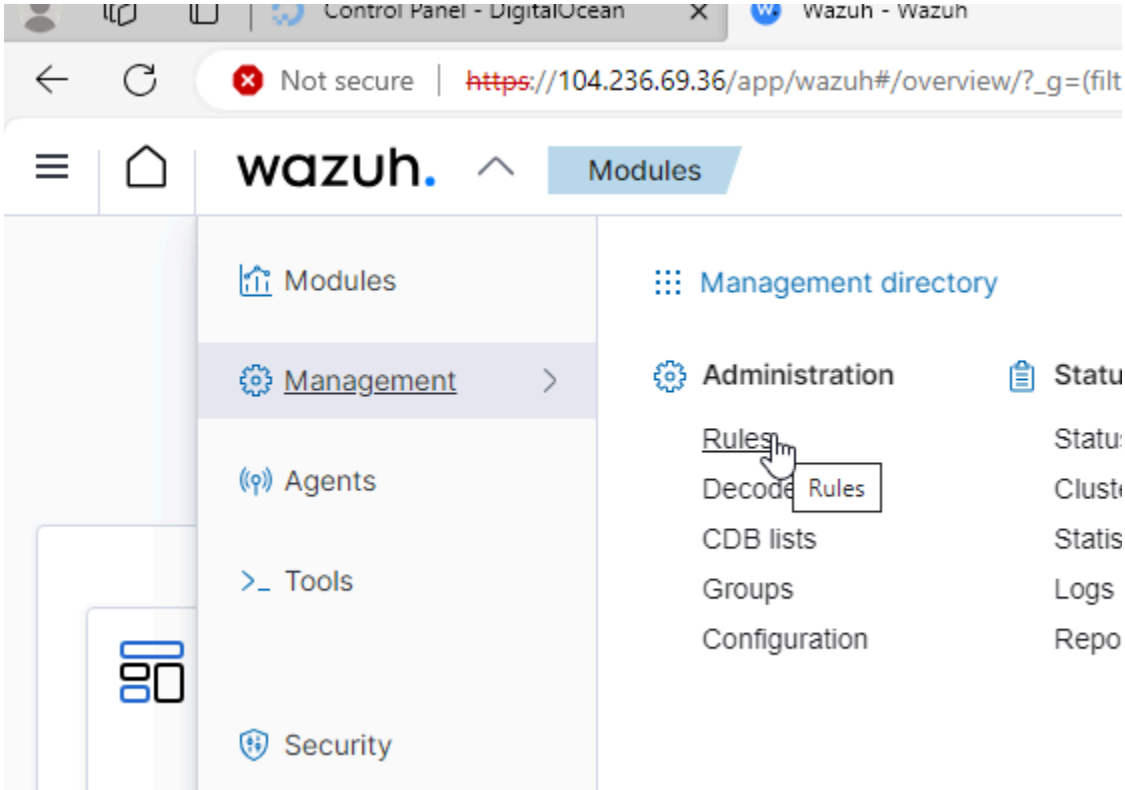
Sysmon is generating on my windows machine and I configured ossec.conf to push this data to wazuh. I will check Wazuh manager again and grep for Mimikat

```
root@Wazuh:/var/ossec/logs/archives# cat archives.json | grep -i mimikatz
```

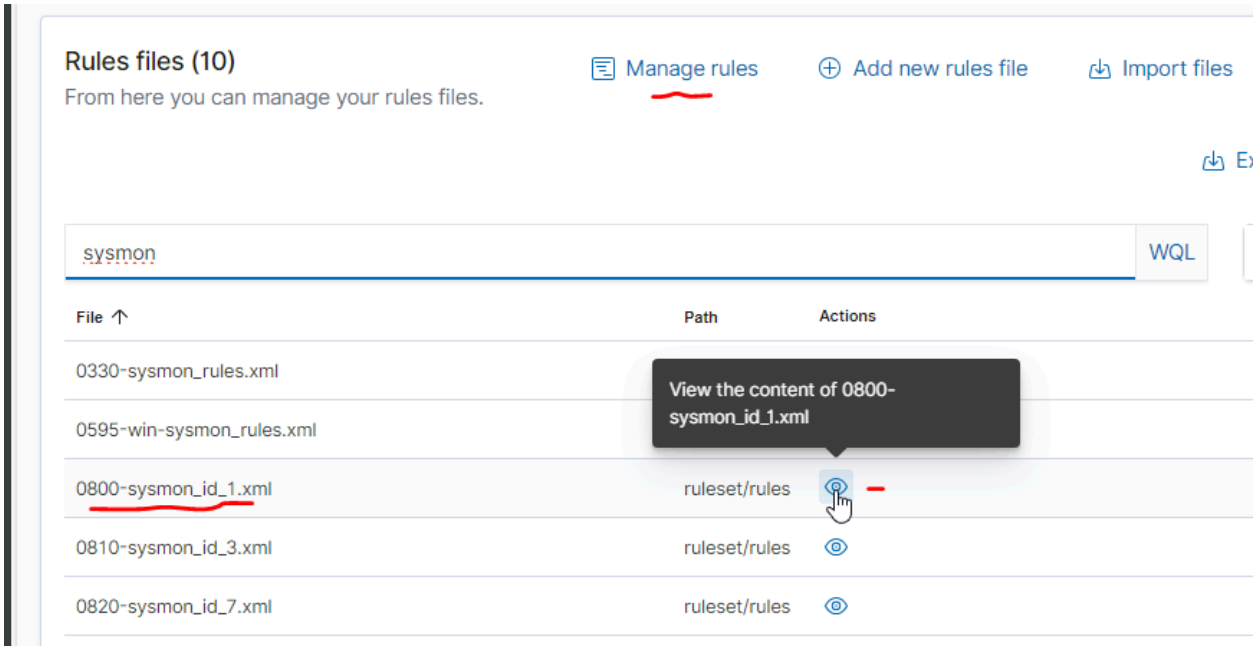
Going back to our dashboard in wazuh and using the wazuh archives search we can see it has now logged mimikatz there as well.



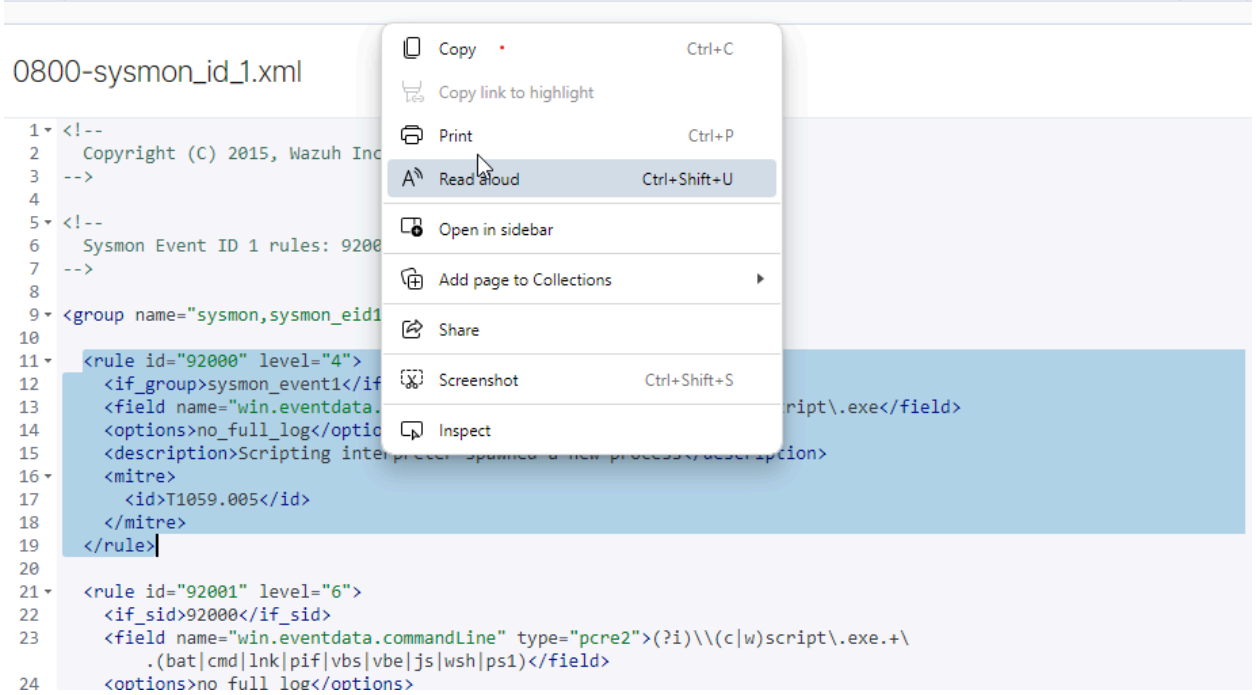
Now to set some rules we can go back to the wazuh home to utilize the dashboard



We are specifically interested in the event ID 1 for sysmon. Select manage rule files and type in sysmon



I will copy one of these rules to use in my custom rule to detect mimikatz



I will go to make a custom rule in the local rules and paste this into that file. Be aware of the indentation as they use spaces and the rule IDs start at 100000. You can see the rule above this one is 100001 so i cannot use that.

```
<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->

<!-- Example -->
<group name="local,syslog,sshd,">

  <!--
  Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
  -->
  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
  </rule>

  <rule id="92000" level="4">
    <if_group>sysmon_event1</if_group>
    <field name="win.eventdata.parentImage" type="pcre2">(?i)\\(c|w)script\.exe</field>
    <options>no_full_log</options>
    <description>Scripting interpreter spawned a new process</description>
    <mitre>
      <id>T1059.005</id>
    </mitre>
  </rule>
```

We will used 100002, and change the level to 15, this is the max level and the higher the level, the more important. Also will change the “parentImage” to “originalFileName”. The field name is capitalized so we must match it or else the rule will never be used.

```
</rule>

<rule id="100002" level="15">
  <if_group>sysmon_event1</if_group>
  <field name="win.eventdata.originalFileName" type="pcre2">(?
  <options>no_full_log</options>
  <description>Scripting interpreter spawned a new process</de
  <mitre>
```

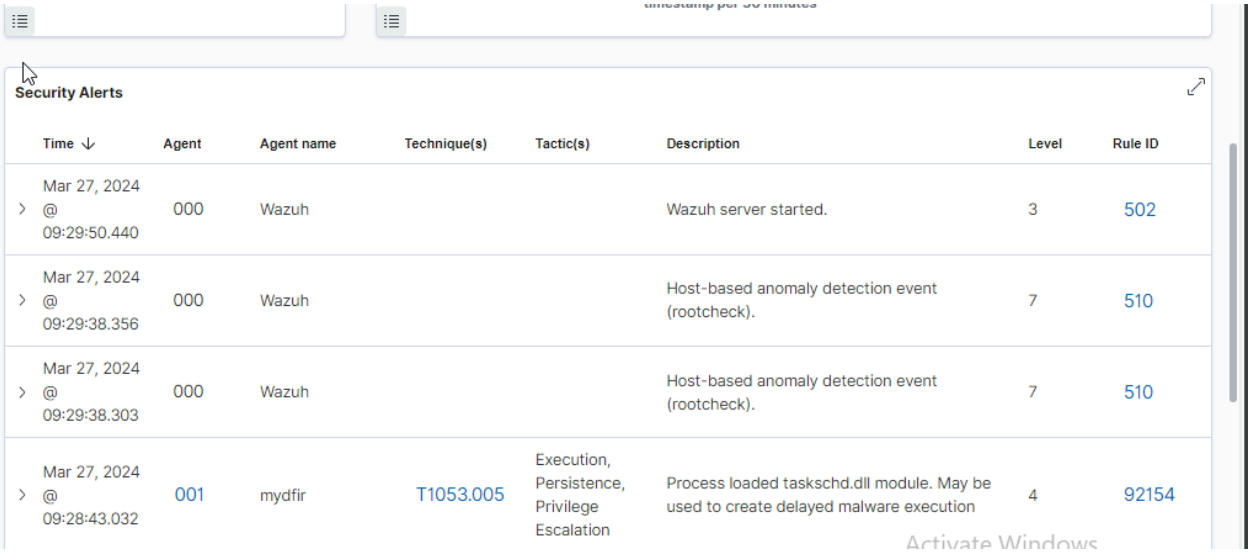
The pcre2 is basically regex. Set to ignore case sensitivity. We will edit this to search for mimikatz.

```
<rule id="100002" level="15">
  <if_group>sysmon_event1</if_group>
  <field name="win.eventdata.originalFileName" type="pcre2">(?i)mimikatz\.exe</field>
  <options>no_full_log</options>
  <description>Scripting interpreter spawned a new process</description>
  <mitre>
    <id>T1059.005</id>
  </mitre>
</rule>
```

Next:
Remove option for no full logs because we want the full logs.
Change description to “ Mimikatz Usage Detected”
And mitre id to T1003 which is credentialed dumping

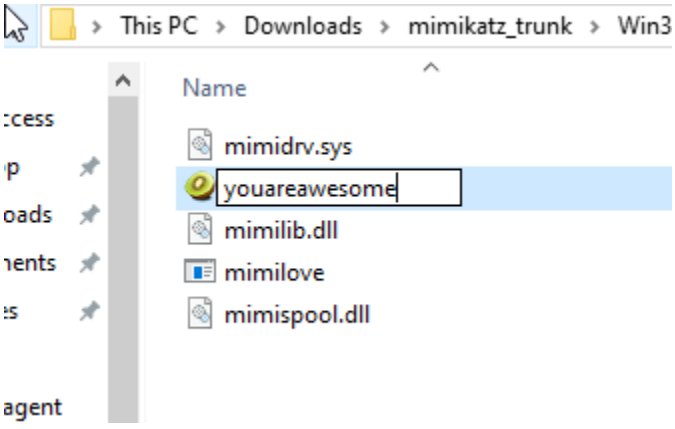
```
<rule id="100002" level="15">
  <if_group>sysmon_event1</if_group>
  <field name="win.eventdata.originalFileName" type="pcre2">(?i)mimikatz\.exe</field>
  <description>Mimikatz Usage Detected</description>
  <mitre>
    <id>T1003</id>
  </mitre>
</rule>
```

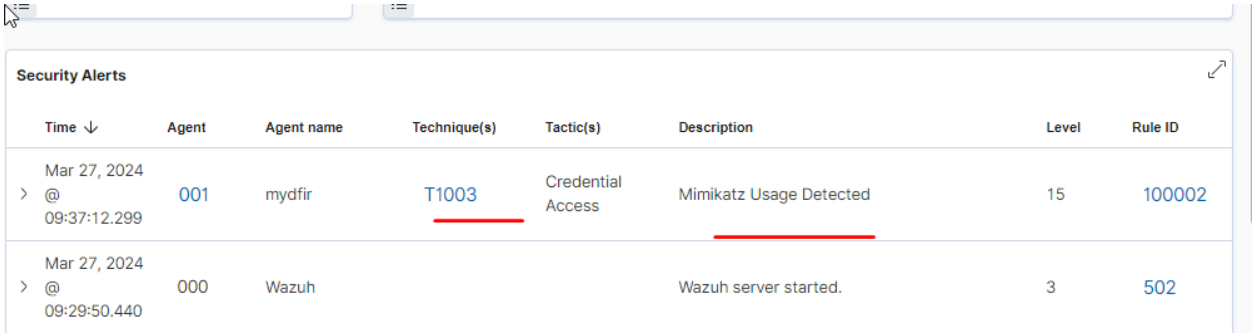
Save and restart manager. Go back to dashboard.
To test this we can see nothing is running at the moment. We will then change the mimikatx.exe file to “youareawesome” and re run it on our command prompt and see if wazuh will pick it up with the new rule.



Security Alerts

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Mar 27, 2024 @ 09:29:50.440	000	Wazuh			Wazuh server started.	3	502
Mar 27, 2024 @ 09:29:38.356	000	Wazuh			Host-based anomaly detection event (rootcheck).	7	510
Mar 27, 2024 @ 09:29:38.303	000	Wazuh			Host-based anomaly detection event (rootcheck).	7	510
Mar 27, 2024 @ 09:28:43.032	001	mydfir	T1053.005	Execution, Persistence, Privilege Escalation	Process loaded taskschd.dll module. May be used to create delayed malware execution	4	92154





Boom! Because we changed the rule to look at the original File Name and not the image regardless of the name change.