

---

# Deployment and Integration of Wazuh SIEM

---

This assignment is submitted to the *Bytewise Cybersecurity fellowship*. The purpose of this assignment is to gain hands-on experience with deploying Wazuh SIEM, integrating it with Windows 7 machines, and creating custom security rules within Wazuh. Additionally, to utilize regular expressions for extracting properties from log payloads.

Wazuh is a threat prevention, detection, and response platform that is free and open source. Wazuh is a security data collection, aggregation, indexing, and analysis tool that aids businesses in detecting intrusions, threats, and suspicious behavior.

---

**Environment:** Wazuh SEIM is deployed in Kali Linux Virtual Machine. It is integrated with Windows 7 Virtual Machine. Due to insufficient resources only one host is integrated.

# Contents

<b>Contents</b>	<b>1</b>
<b>1 Deploying Wazuh SIEM:</b>	<b>3</b>
1.1 Installing the Wazuh Indexer: . . . . .	3
1.2 Installing the Wazuh Server: . . . . .	5
1.3 Installing the Wazuh DashBoard: . . . . .	7
<b>2 Integrate Windows 7:</b>	<b>9</b>
2.1 Verification of Integration: . . . . .	10
<b>3 Creating Custom Rules:</b>	<b>11</b>
3.1 Rule . . . . .	12
<b>4 Property Extraction:</b>	<b>14</b>
<b>5 Conclusion</b>	<b>16</b>

# Chapter 1

## Deploying Wazuh SIEM:

Officially, the Wazuh Installation guide is provided at [documentation.wazuh.com](https://documentation.wazuh.com). [\[documentation.wazuh.com\]](https://documentation.wazuh.com).

Wazuh 4.8 (Latest) SEIM is deployed in our system as a single-node cluster in the following three phases:

### 1.1 Installing the Wazuh Indexer:

The Wazuh indexer is a real-time, full-text search and analytics engine for security data. Log data ingested into the Wazuh server is analyzed and forwarded to the indexer for indexing and storage. These events are then queried on the Wazuh dashboard. The Wazuh indexer stores data as JSON documents.

(Make sure to run these commands as the admin user.)

1. Download the Certification and Configuration file

```
$> curl -sO https://packages.wazuh.com/4.8/wazuh-certs-tool.sh
$> curl -sO https://packages.wazuh.com/4.8/config.yml
```

2. Edit *config.yml* and add the IP address of the server machine at three IP fields.

3. Run *./wazuh-certs-tool.sh*:

```
$> bash ./wazuh-certs-tool.sh -A
```

4. Now compress the file as it would be required later.

```
$> tar -cvf ./wazuh-certificates.tar -C ./wazuh-certificates /
$> rm -rf ./wazuh-certificates
```

5. Install the following packages:

```
$> apt-get install debconf adduser procs
$> apt-get install gnupg apt-transport-https
$> apt-get install debhelper tar curl libcap2-bin
```

6. Add the Wazuh Repository:

```
$> curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH |
gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg
--import && chmod 644 /usr/share/keyrings/wazuh.gpg
$> echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/stable/main"
| tee -a /etc/apt/sources.list.d/wazuh.list
$> apt-get update
```

7. Install the wazuh indexer package:

```
$> apt-get -y install wazuh-indexer
```

8. Edit the `/etc/wazuh-indexer/opensearch.yml`. Enter the server machine IP address into the `network.host`.

9. Deploy Certificates:

```
$> NODE_NAME=node-1
$> mkdir /etc/wazuh-indexer/certs
$> tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/
./$NODE_NAME.pem $> ./$NODE_NAME-key.pem ./admin.pem ./admin-key.pem
./root-ca.pem
$> mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem
/etc/wazuh-indexer/certs/indexer.pem
$> mv -n /etc/wazuh-indexer/certs/$NODE_NAME-key.pem
/etc/wazuh-indexer/certs/indexer-key.pem
$> chmod 500 /etc/wazuh-indexer/certs
$> chmod 400 /etc/wazuh-indexer/certs/*
$> chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
```

10. Start the service:

```
$> systemctl daemon-reload
$> systemctl enable wazuh-indexer
$> systemctl start wazuh-indexer
```

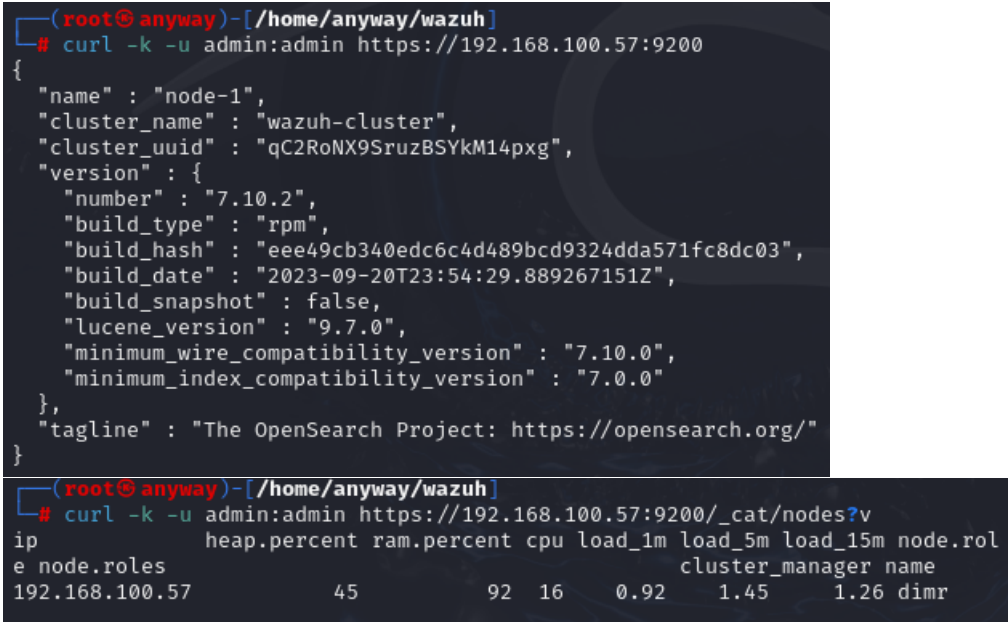
11. Load new Certificate information:

```
$> /usr/share/wazuh-indexer/bin/indexer-security-init.sh
```

12. Test the cluster:

```
$> curl -k -u admin:admin https://<WAZUH_INDEXER_IP_ADDRESS>:9200
```

```
$> curl -k -u admin:admin https://<WAZUH_INDEXER_IP_ADDRESS>:9200/_cat/nodes?
```



The terminal screenshot shows two commands and their outputs. The first command is a curl request to the Wazuh Indexer API, which returns a JSON object describing the cluster. The second command is a curl request to the Wazuh Indexer API, which returns a table of node details.

```
(root@anyway)-[/home/anyway/wazuh]
# curl -k -u admin:admin https://192.168.100.57:9200
{
  "name" : "node-1",
  "cluster_name" : "wazuh-cluster",
  "cluster_uuid" : "qC2RoNX9SruzBSYkM14pxg",
  "version" : {
    "number" : "7.10.2",
    "build_type" : "rpm",
    "build_hash" : "eee49cb340edc6c4d489bcd9324dda571fc8dc03",
    "build_date" : "2023-09-20T23:54:29.889267151Z",
    "build_snapshot" : false,
    "lucene_version" : "9.7.0",
    "minimum_wire_compatibility_version" : "7.10.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}

(root@anyway)-[/home/anyway/wazuh]
# curl -k -u admin:admin https://192.168.100.57:9200/_cat/nodes?v
ip heap.percent ram.percent cpu load_1m load_5m load_15m node.role
e node.roles cluster_manager name
192.168.100.57 45 92 16 0.92 1.45 1.26 dimr
```

## 1.2 Installing the Wazuh Server:

The Wazuh server is a key component of the Wazuh security platform, which is a free and open-source solution focused on threat detection, incident response, and security monitoring. The server analyzes data gathered from agents installed on endpoints and triggers alerts when potential threats are identified.

1. Install Wazuh manager:

```
$> apt-get -y install wazuh-manager
```

2. Install and configure Filebeat:

```
$> apt-get -y install filebeat
```

```
$> curl -so /etc/filebeat/filebeat.yml
```

```
https://packages.wazuh.com/4.8/tpl/wazuh/filebeat/filebeat.yml
```

```
r
```

3. Add the IP address of the server machine in the host field.

```

$> filebeat keystore create
$> echo admin | filebeat keystore add username --stdin --force
$> echo admin | filebeat keystore add password --stdin --force
$> curl -so /etc/filebeat/wazuh-template.json
https://raw.githubusercontent.com/wazuh/wazuh/v4.8.2/extensions/elasticsearch
/7.x/wazuh-template.json
$> chmod go+r /etc/filebeat/wazuh-template.json
$> curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.4.tar.gz
| tar -xvz -C /usr/share/filebeat/module

```

#### 4. Deploy the certificate:

```

$> NODE_NAME=wazuh
$> mkdir /etc/filebeat/certs
$> tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/ ./${NODE_NAME}.pem
./${NODE_NAME}-key.pem ./root-ca.pem
$> mv -n /etc/filebeat/certs/${NODE_NAME}.pem /etc/filebeat/certs/filebeat.pem
$> mv -n /etc/filebeat/certs/${NODE_NAME}-key.pem
/etc/filebeat/certs/filebeat-key.pem
$> chmod 500 /etc/filebeat/certs
$> chmod 400 /etc/filebeat/certs/*
$> chown -R root:root /etc/filebeat/certs

```

#### 5. Configure Wazuh indexer connection with server:

```

$> /var/ossec/bin/wazuh-keystore -f indexer -k username -v admin
$> /var/ossec/bin/wazuh-keystore -f indexer -k password -v admin

```

#### 6. Edit */var/ossec/etc/ossec.conf*, add Server machine IP address in *host* section.

#### 7. Start the services:

```

$> systemctl daemon-reload
$> systemctl enable filebeat
$> systemctl start filebeat
$> filebeat test output

```

```

(root@anyway)-[/home/anyway/wazuh]
# filebeat test output
elasticsearch: https://192.168.100.57:9200...
parse url ... OK
connection ...
parse host ... OK
dns lookup ... OK
addresses: 192.168.100.57
dial up ... OK
TLS ...
security: server's certificate chain verification is enabled
handshake ... OK
TLS version: TLSv1.3
dial up ... OK
talk to server... OK
version: 7.10.2

```

### 1.3 Installing the Wazuh DashBoard:

The Wazuh dashboard is a powerful web user interface designed for data visualization and analysis of security events and alerts from the Wazuh server. It allows users to mine and visualize Wazuh alerts and archived events, providing out-of-the-box dashboards for easy monitoring and regulatory compliance. To fully utilize the dashboard, users typically need root user privileges.

1. Install wazuh dashboard:

```
$> apt-get -y install wazuh-dashboard
```

2. Edit the `/etc/wazuh-dashboard/opensearchdashboards.yml` and enter the server machine IP address in `inserver.host` field.

3. Deploy Certificate:

```

$> NODE_NAME=dashboard
$> mkdir /etc/wazuh-dashboard/certs
$> tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs /
./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem
$> mv -n /etc/wazuh-dashboard/certs/$NODE_NAME.pem
/etc/wazuh-dashboard/certs/dashboard.pem
$> mv -n /etc/wazuh-dashboard/certs/$NODE_NAME-key.pem
/etc/wazuh-dashboard/certs/dashboard-key.pem
$> chmod 500 /etc/wazuh-dashboard/certs
$> chmod 400 /etc/wazuh-dashboard/certs/*
$> chown -R wazuh-dashboard:wazuh-dashboard
/etc/wazuh-dashboard/certs

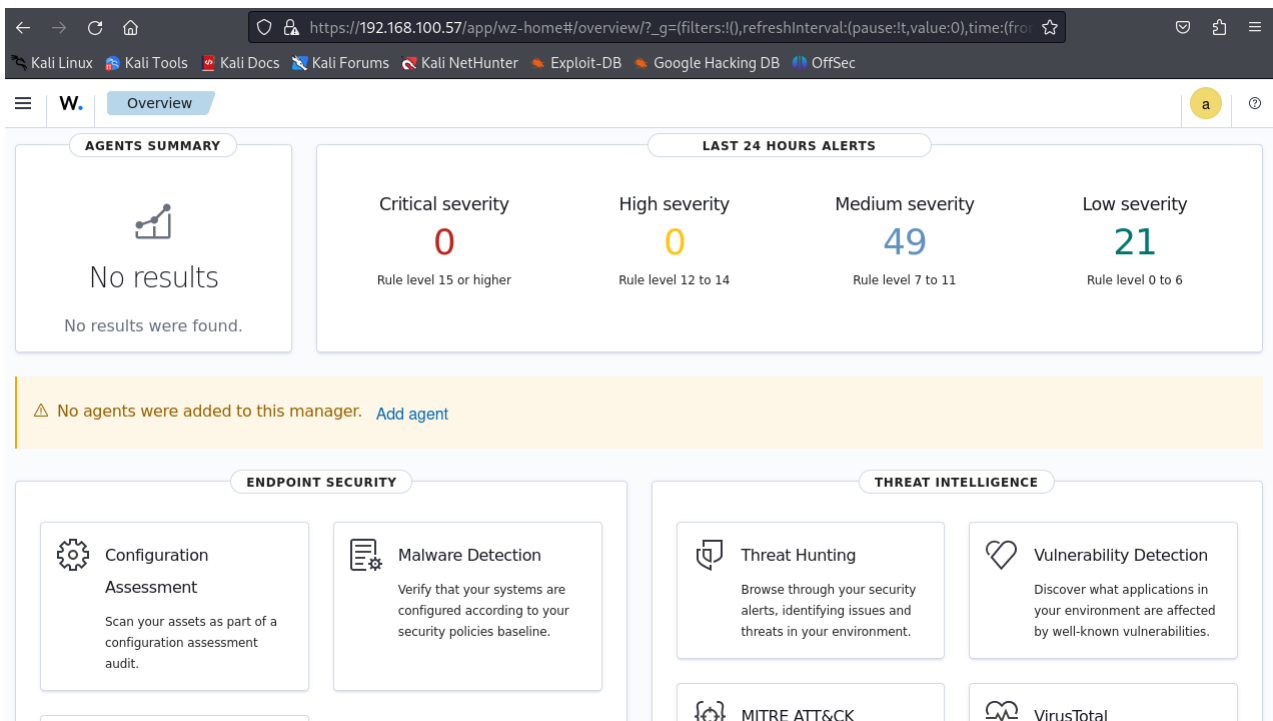
```

4. Start the services:

```
$> systemctl daemon-reload
$> systemctl enable wazuh-dashboard
$> systemctl start wazuh-dashboard
```

5. Edit the `/usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml` and enter the server machine IP address in the `URL` field.

Now enter the IP address of the server machine in the browser (of server). Enter the **username:password** `admin:admin`. If everything works fine, you'll be able to see this dashboard.





## Chapter 2

# Integrate Windows 7:

Windows 7 VM as an endpoint would be integrated with the Wazuh server by installing a Wazuh agent in Windows 7 that will transfer data i.e. Logs, from the endpoint to the server.

Official guides for integrating endpoints to servers are available at [\[wazuh-agent/index.html\]](https://wazuh-agent/index.html).

Download the wazuh agent here [\[wazuh-agent-4.8.2-1.msi\]](#).

We will enroll agents via agent configuration. The agent needs a Server IP address and Authentication key. In agent configuration, we will provide the IP address and it will automatically request the authentication key.

Edit *C:/Program Files (x86)/ossec-agent/ossec.conf* file and enter the server IP address in the address field in *client server address* section.

Finally, restart the agent in CMD:

```
$> net stop wazuh
```

```
$> net start wazuh
```

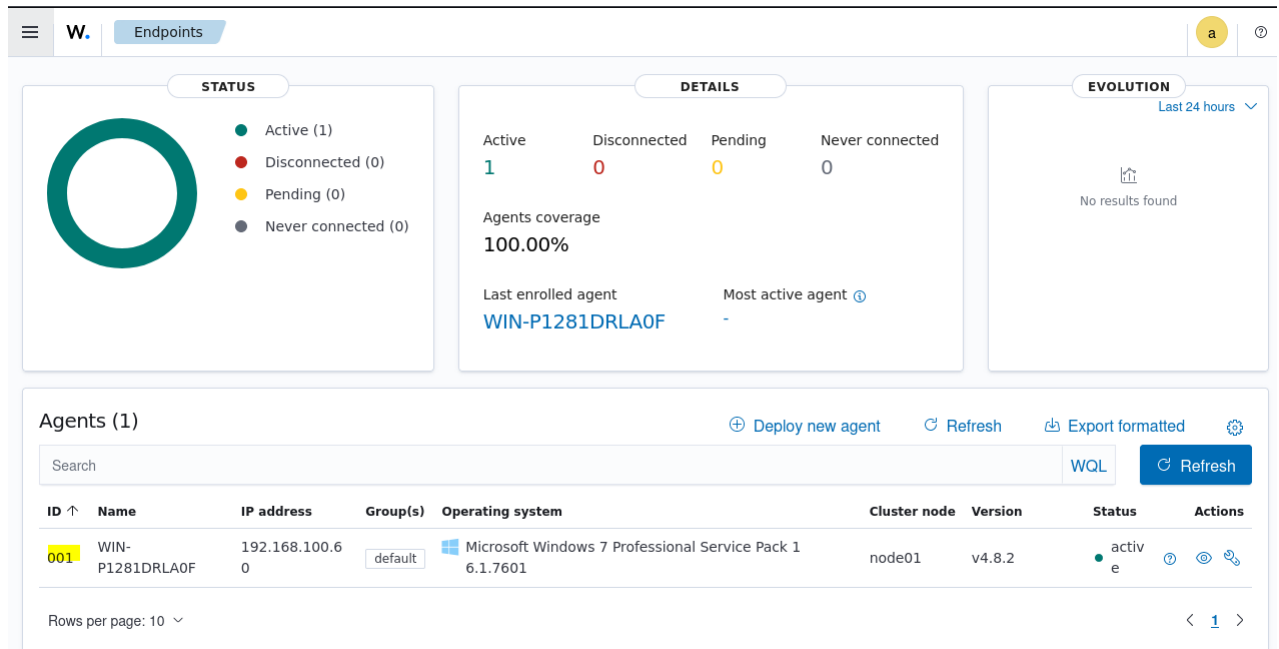
```
C:\Windows\system32>net stop wazuh
The Wazuh service is not started.

More help is available by typing NET HELPMSG 3521.

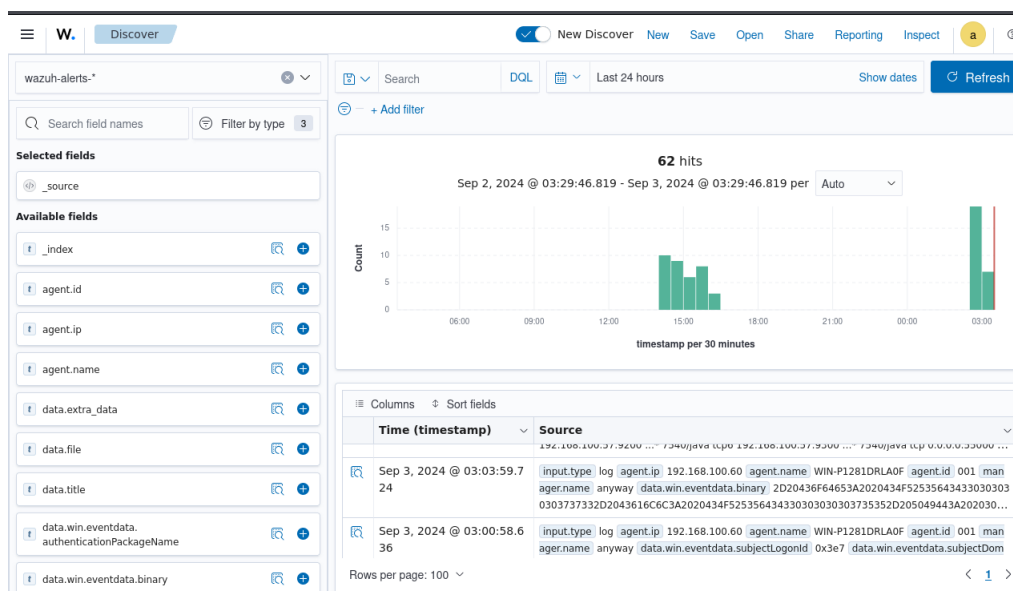
C:\Windows\system32>
C:\Windows\system32>net start wazuh
The Wazuh service is starting.
The Wazuh service was started successfully.
```

## 2.1 Verification of Integration:

To verify that the wazuh agent is enrolled in Windows 7, go to the Endpoint Summary tab in the side panel of the Wazuh dashboard:



To verify that the agent is sending logs to the server, go to the discover tab under explore, on the side panel:



# Chapter 3

## Creating Custom Rules:

In this part, we will create a custom rule on the wazuh server for wazuh agents. The rules are some criteria that if executed in Windows 7, the wazuh agent sends the alert logs to the wazuh server. In custom rule, we could create any type of rule to control the agent environment.

To create a simple rule we could modify the *local-rule.xml* file. We can combine two or more rules in it. We can also modify a rule. And if we want to create a rule on some large scale, we should create a new rule then.

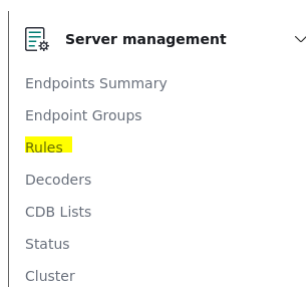
Below is the creation and testing of two custom rules.

### Create a rule file

This rule will be if the Windows login attempt exceeds to amount the 10, the wazuh agent sends the alert logs to the server.

First, let's understand the rule structure in wazuh. The rule file is *.xml* file, it defines how logs are detected. This enables the wazuh manager to identify the log's pattern and generate alerts. Each rule has a specific ID and a security level from 0 to 16, indicating the alert level. Rules file include some elements, which you can find here </rules.html>

To find a rule, go to *rule* tab under *Server Management*.



To create a rule, go to `/var/ossec/etc/rules/` and edit the `local-rules.xml`. It is where we will save our new custom rules.

### 3.1 Rule

This rule is called "**Successful login during non-business hours**", which will trigger the alert when an account is logged in during the non-business hour specified.

Here is the .XML lines of the rule:

---

```
<group name="off_time_login">
  <rule id="17101" level="16">
    <if_group>authentication_success </if_group>
    <time>1 pm - 10 am</time>
    <description>Successful login during non business
    hours.</description>
    <group>login_time , pci_dss_10.2.5 , pci_dss_10.6.1 ,
    gpg13.7.1 , gpg13.7.2 , gdpr_IV_35.7.d , gdpr_IV_32.2 , hipaa_164.312.b ,
    nist_800_53_AU.14 , nist_800 >
  </rule>
</group>
```

---

Rule ID can be anything within the limit specified by Wazuh. Level specifies the alert, ranging from 0 to 16 and higher. If the Group specifies that the rule will only be triggered if a successful login event is detected. The group indicates that the rule is related to various security compliance standards.

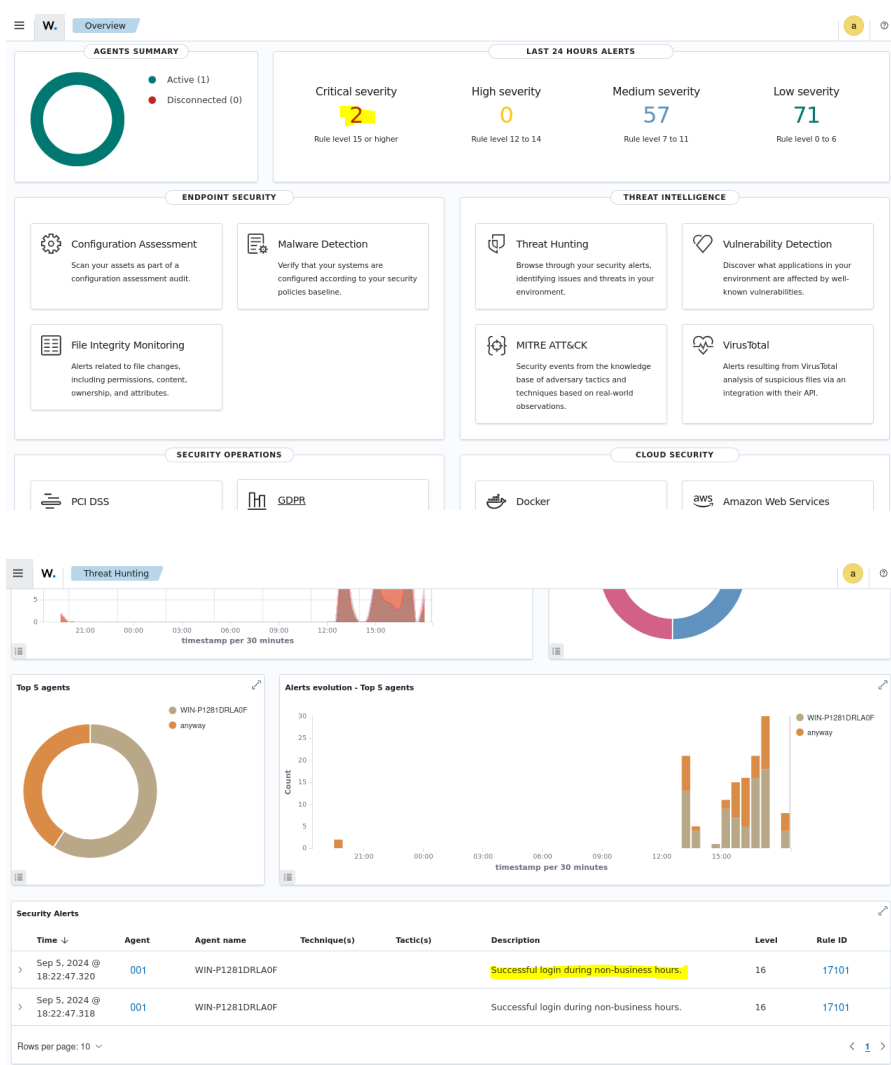
Add those lines in the `local-rules.xml`, this file holds the custom rule.



```
<group name="off_time_login">
  <rule id="17101" level="17">
    <install>
      <if_group>authentication_success</if_group>
      <time>1 pm - 10 am</time>
      <description>Successful login during non-business hours.</description>
    </rule>
  </group>
</group>
```

After editing the file, make sure to restart the *wazuh-manager* so that it will take effect.

Now in the time specified in the rule, log to the windows. And here’s the result in wazuh dashboard:



Rule triggers based on the specific conditions defined.

# Chapter 4

## Property Extraction:

In this part, we have to select the log payload, which refers to the data captured during logging operations, which includes the body contents along with metadata, from Windows 7 logs and extract relevant properties such as timestamps, event types, and user IDs.

Logs can be collected from `/var/ossec/logs/alerts/2024/Sep` or under Threat Intelligence, Threat Hunting, and find the log.

Here is the extracted regular expressions and relevant properties:

- **Event:** Successful login during non-business hours
- **Event Type:** Windows Security Audit (Event ID: 4624)
- **Timestamp:** 2024-09-05T13:21:41.909935400Z (UTC)

- **Agent:**

IP Address: 192.168.100.60

Name: WIN-P1281DRLA0F

ID: 001

Manager: anyway

- **Subject:**

Security ID: S-1-5-18

Account Name: WIN-P1281DRLA0F

Account Domain: WORKGROUP

Logon ID: 0x3e7

- **New Logon:**

Security ID: S-1-5-21-481202490-646887213-679027004-1000

Account Name: Windows

Account Domain: WIN-P1281DRLA0F

Logon ID: 0x2a54be

Logon GUID: 00000000-0000-0000-0000-000000000000

- **Process Information:**

Process ID: 0x378

Process Name: C:/Windows/System32/winlogon.exe

- **Network Information:**

Workstation Name: WIN-P1281DRLA0F

Source Network Address: 127.0.0.1

Source Port: 0

- **Authentication Information:**

Logon Process: User32

Authentication Package: Negotiate

- **Rule:**

ID: 17101

Description: Successful login during non-business hours

Groups: off-time=loginlogin-time

Fired Times: 2

Level: 16

Mail: True

Decoder: windows-eventchannel

- **Location:** EventChannel

- **Timestamp (Local):** 2024-09-05T18:22:47.320+0500

# Chapter 5

## Conclusion

This assignment is submitted to Bytewsie Cybersecurity fellowship, Sep 2024. The goal of this assignment is to gain hands-on experience with deploying Wazuh SIEM, integrating it with other machines, and creating custom security rules within Wazuh. Additionally, learn to utilize regular expressions for extracting properties from log payloads.

There is four part of this project. First is to deploy the Wazuh. Second is to integrate it with other machines. Third is to create a new custom security rule and fourth is to extract the information from a log.