Security Tools Lab 2
Project 1
SOC: Security Ops Centre

Student Name: Gowtham Baskar

Student ID: 1006523

Table of Content

1. Introduction

In today's world, organizations have multiple endpoints/machines running multiple services to perform business operations efficiently. At present, even a small-scale company has a minimum of ten systems in its infrastructure and they are constantly collecting data and generating logs. The biggest challenge that today's organizations have is analysing and utilizing these big data. These endpoints generate multiple records crucial for analysing and creating alerts for critical security violations and attacks. Organizations use SIEM (Security Information Event Management) solutions to handle this problem.

It is important to know two key terms and their definitions to understand the meaning of SIEM:

SEM – Security Event Management deals with collecting logs from endpoints.

SIM – Security Information Management deals with analysing the collected logs.

SEM + SIM = SIEM (Security Information Event Management)

SIEM is a solution that helps organizations in collecting logs and converting the records into useful information that can be analysed. It also provides real-time monitoring & analysing capabilities that helps in creating alerts when any security violation or attack occurs.

The most popular and widely used SIEM out there is WAZUH. Wazuh is used worldwide by many companies, starting from small firms to big corporations.

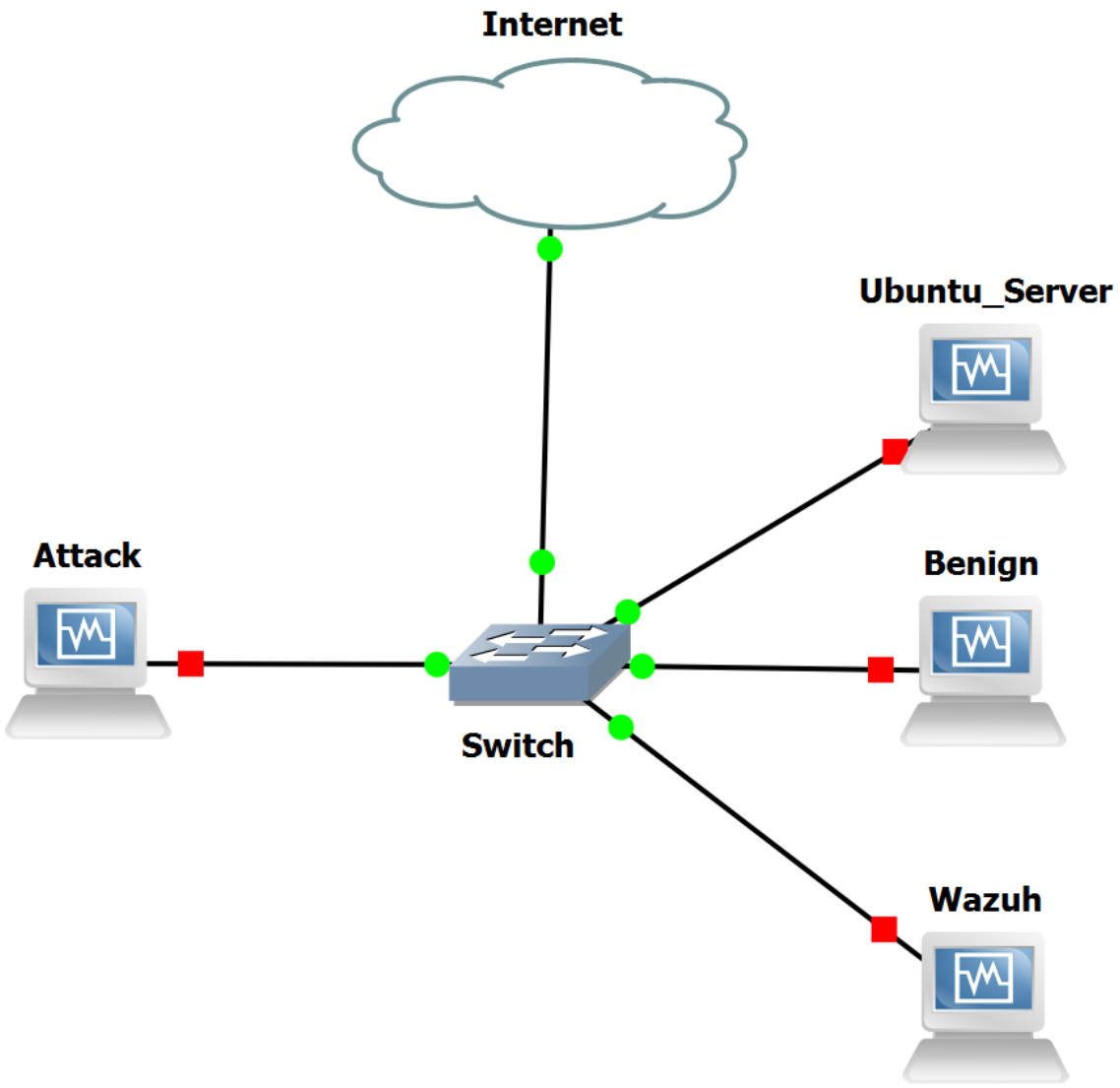
Wazuh is a free and open-source platform that is used for threat detection, prevention, and response.

It is typically used to protect networks, virtualized environments, containers, and cloud environments.

Wazuh is a SIEM system used for collecting, aggregating, indexing, and analysing security related data that allows you to detect attacks, intrusions, threats, vulnerabilities, and malicious activities.

We will be using Wazuh in our project to detect attack in our internal network that was set up in GNS3.

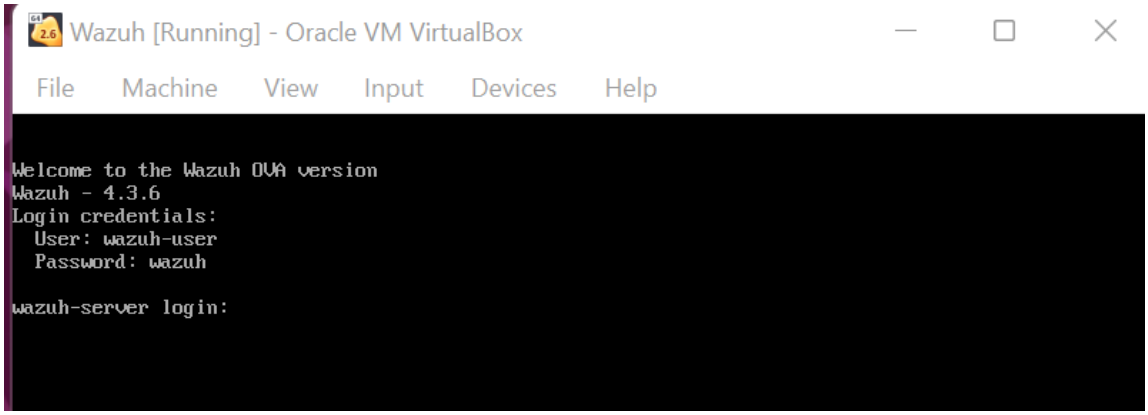
2. Network Topology



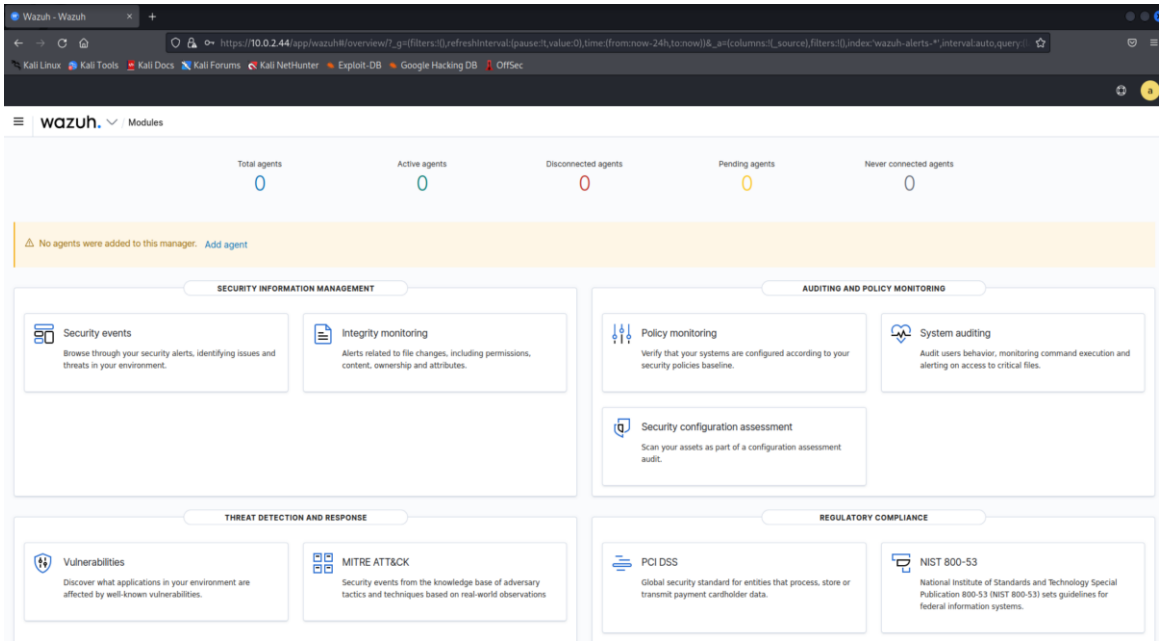
Host	IP (NAT Network is set as 10.0.2.0/24)
Security Agent	
Wazuh	10.0.2.44 (Server)
External Network	
Malicious	10.0.2.47 (
Internal Network	
Benign_STL2	10.0.2.42 (Wazuh Agent Installed)
Ubuntu Server_STL2	10.0.2.46 (Wazuh Agent Installed)

On the Ubuntu server, these are the following services that are running, the FTP, SSH, http, Domain, MariaDB and the SMTP.

Wazuh Server

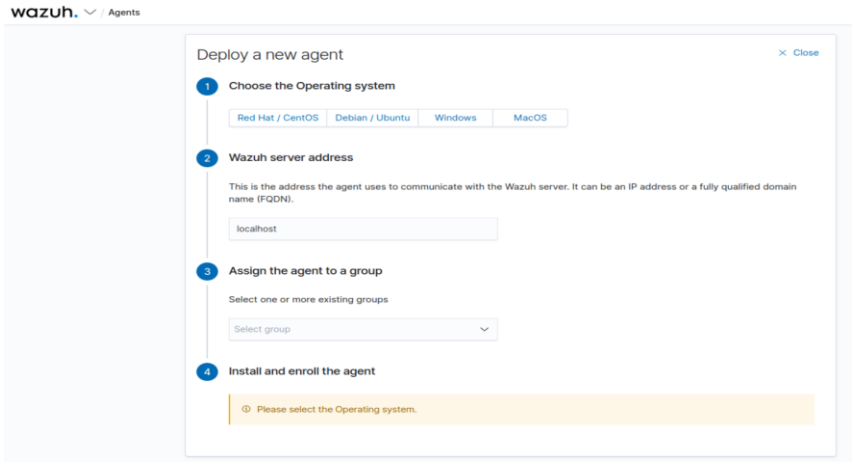


Wazuh Dashboard



Installing Agents

I have installed the agents to my Benign Machine and my ubuntu Server. Installation is guided by Wazuh as below



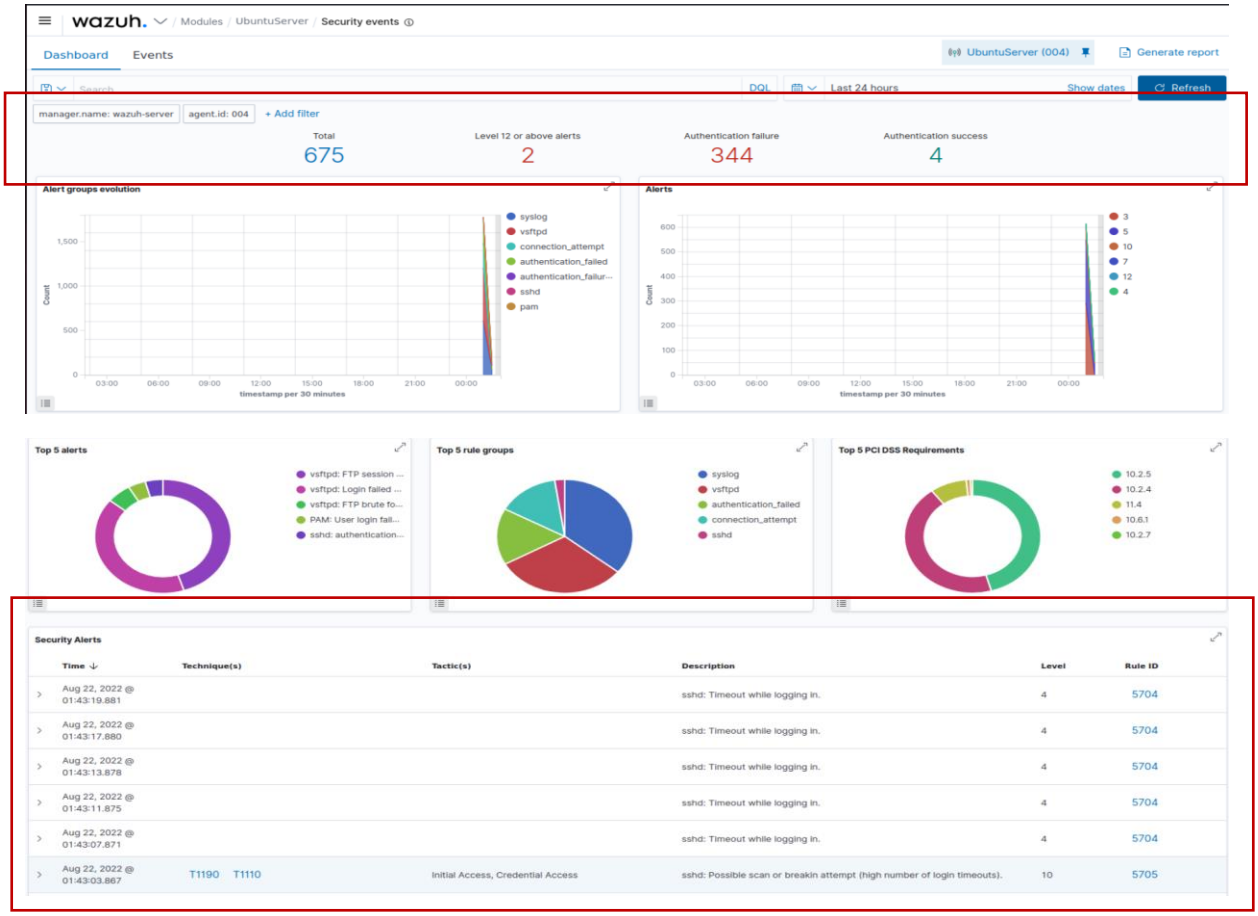
3. Attack Scenario

3.1 Dictionary Attack on SSH

I've attached the code along with the project folder as "Dictionary_SSH.py".

```
┌──(kali㉿kali)-[~/Desktop/Project/Attacks]
└─$ python3 Dictionary_SSH.py 10.0.2.46 -u mssd -P wordlist.txt
[!] Invalid credentials for mssd:123456
[!] Invalid credentials for mssd:12345
[!] Invalid credentials for mssd:123456789
[!] Invalid credentials for mssd:password
[!] Invalid credentials for mssd:iloveyou
[!] Invalid credentials for mssd:princess
[!] Invalid credentials for mssd:12345678
[!] Invalid credentials for mssd:1234567
```

After running the code, I was able to detect the events in my security alert dashboard as shown below in red.



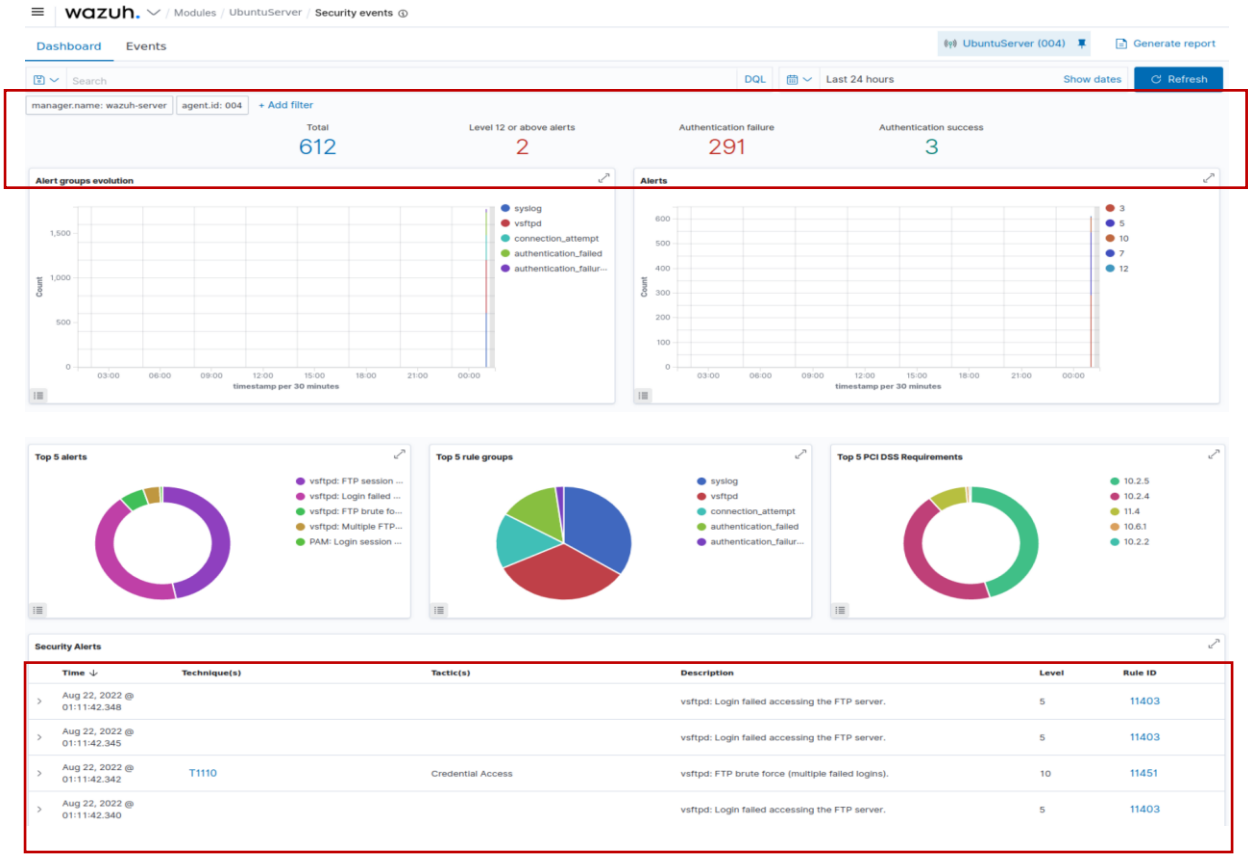
We have successfully managed to attack the ubuntu server from the attack machine and was successfully able to detect the Dictionary attempt (Credential Guessing).

3.2 Brute force FTP

I've attached the code along with the project folder as "Bruteforce_FTP.py".

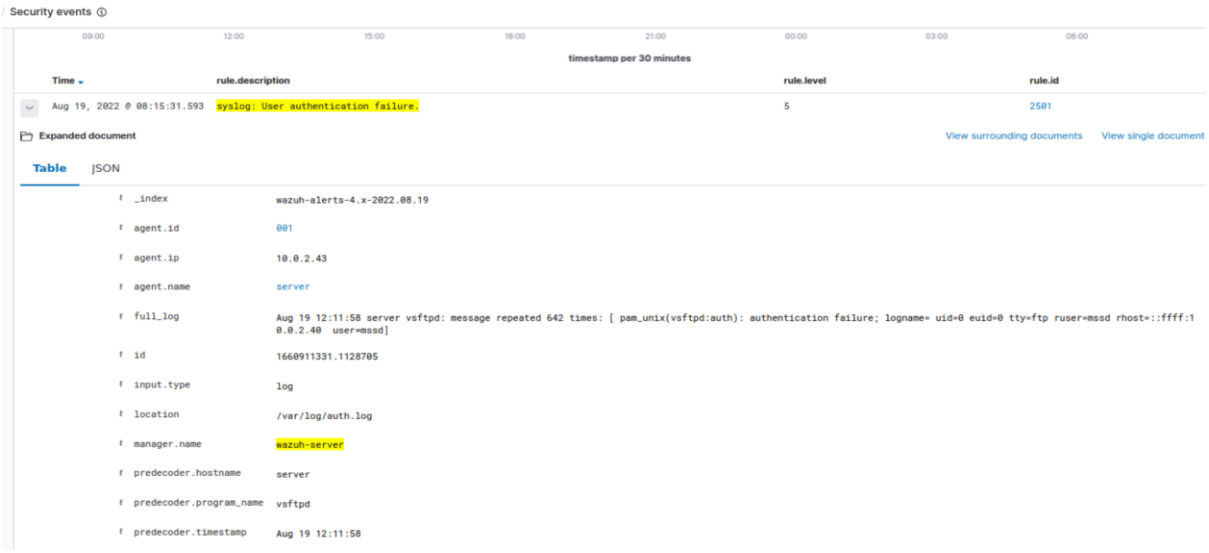
After executing the code, I was able to find the ftp credential as shown below

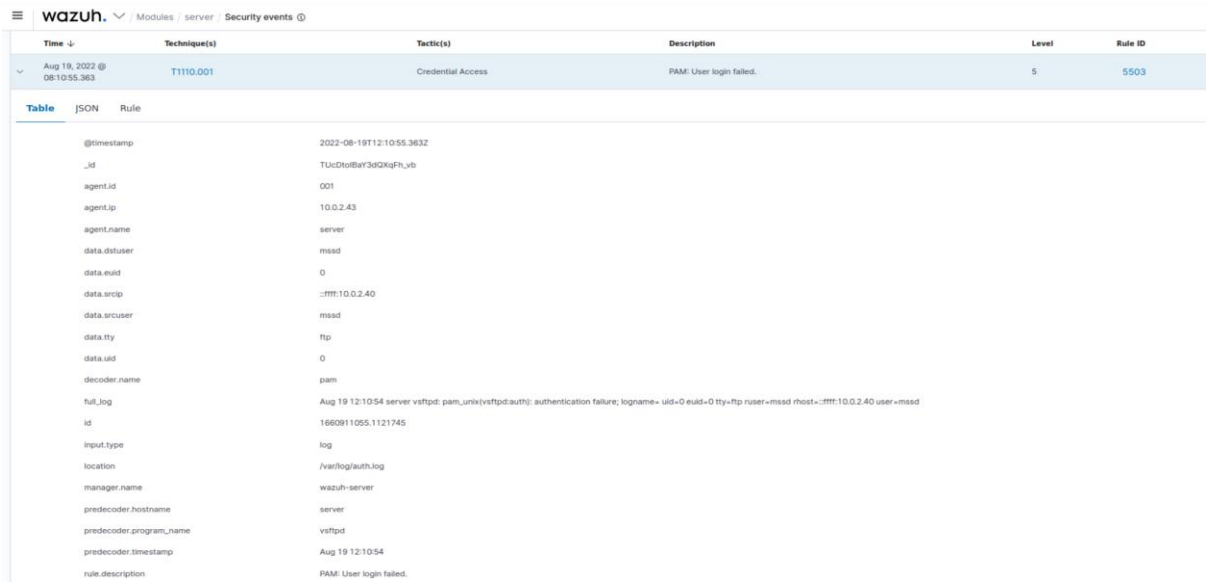
After running the code, I was able to detect the events in my security alert dashboard as shown below in red.



The scores in the dashboard are different as I've first executed the Brute force FTP attack followed by the Dictionary SSH Attack.

We have successfully managed to attack the ubuntu server from the attack machine and was successfully able to detect the brute force attempt.

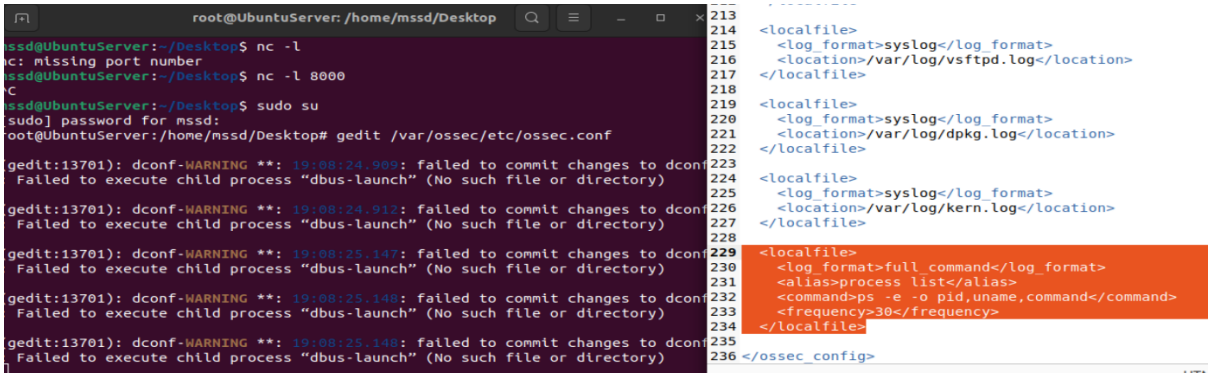




I was able to get more information by opening each security alerts as shown above.

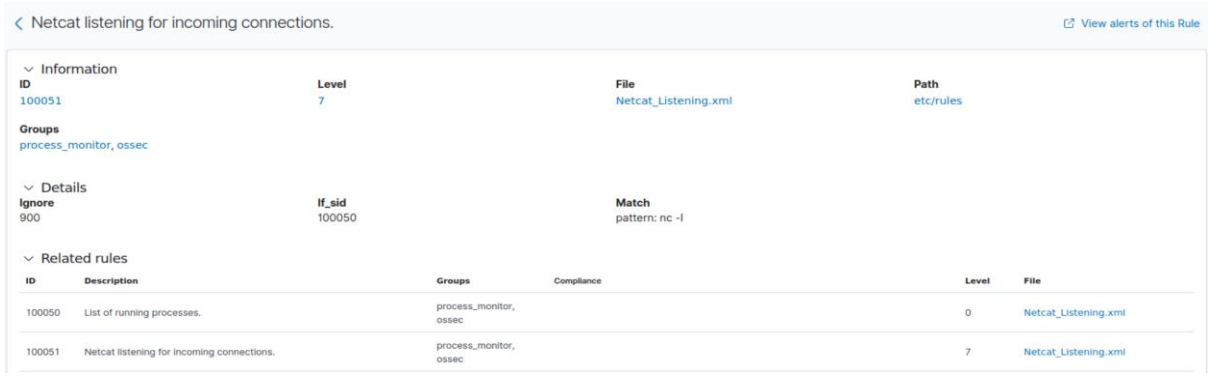
3.3 Netcat Listening rule (Rule Based)

Firstly, I've added the following rule which periodically gets a list of running processes in the local agent file as shown in the below



After adding here, I restarted the agent and proceeded to add another set of rules in the Wazuh server as show below.

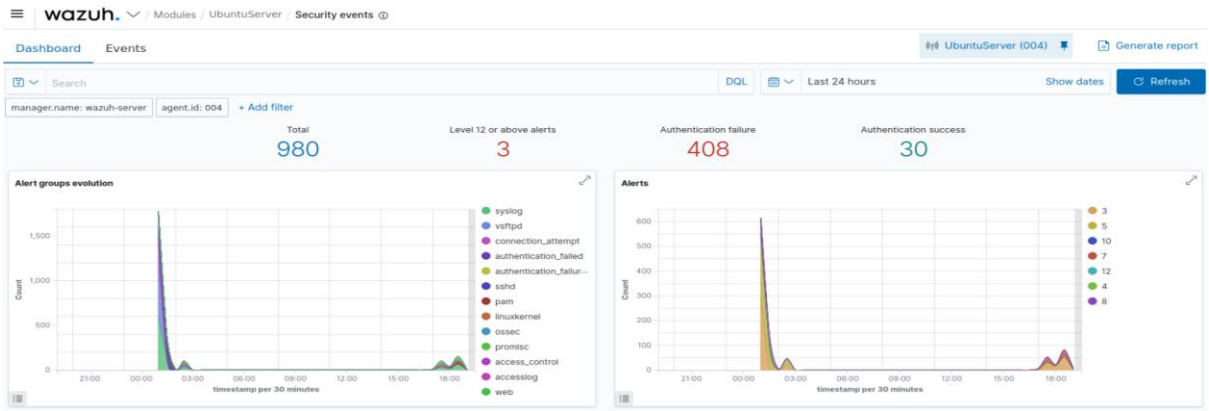
This is the detailed view of the rule. (I've changed the name to Netcat_listening)



After this, I restarted the Wazuh server and ran a netcat command <nc -l 8000> for more than 30s

```
mssd@UbuntuServer:~/Desktop$ nc -l 8000
```

I refreshed the security events in the agents



Here I'm able to detect the rule I've just set up

Security Alerts

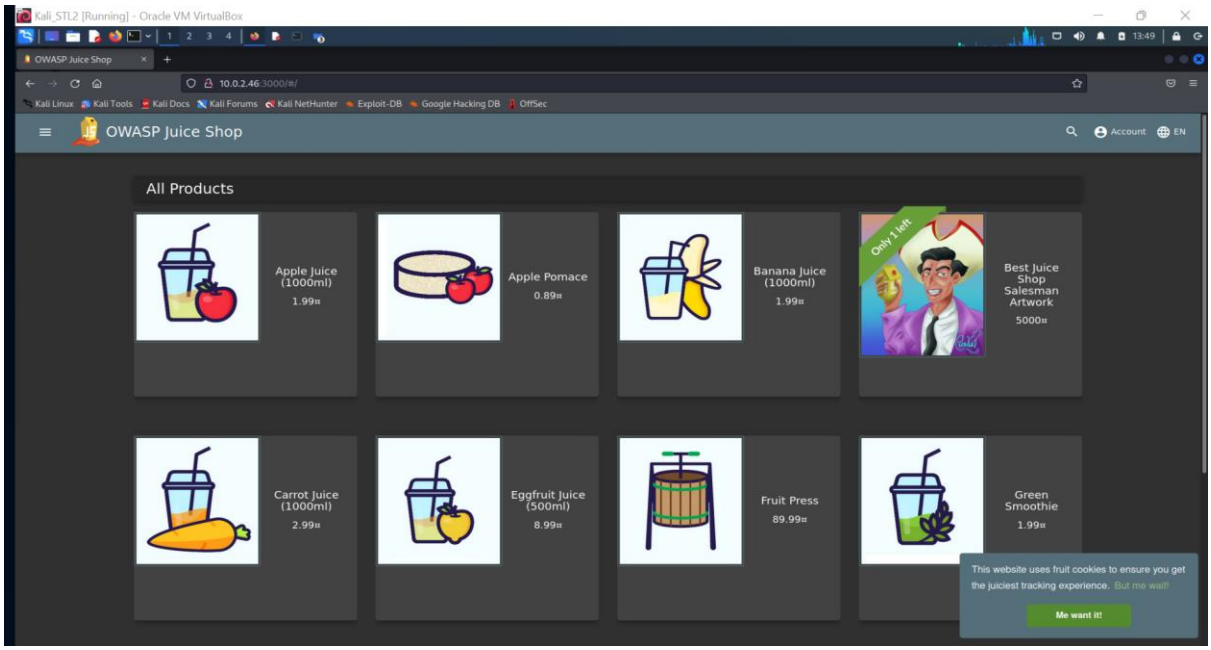
Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Aug 22, 2022 @ 19:10:54.153			Listened ports status (netstat) changed (new port opened or closed).	7	533
Aug 22, 2022 @ 19:08:19.992	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
Aug 22, 2022 @ 19:08:19.992	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
Aug 22, 2022 @ 19:08:19.929	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402
Aug 22, 2022 @ 19:04:53.803			Listened ports status (netstat) changed (new port opened or closed).	7	533
Aug 22, 2022 @ 19:04:33.703			Netcat listening for incoming connections.	7	100051
Aug 22, 2022 @ 19:01:53.544			Netcat listening for incoming connections.	7	100051
Aug 22, 2022 @ 19:01:27.563			PAM: Login session closed.	3	5502

We have successfully managed to detect the Netcat Listening command in the Wazuh Server as shown in red above. This was done as an understanding on how a rule-based detection works.

In the next scenario, I will be focusing more on the attacks.

3.4 SQL Injection Attack

I've enabled a docker which runs a OWASP Juice Website in my ubuntu server as shown below



I've added the below code in the local ubuntu server at /var/ossec/etc/ossec.conf

```
<localfile>
  <log_format>apache</log_format>
  <location>/var/log/apache2/access.log</location>
</localfile>
```

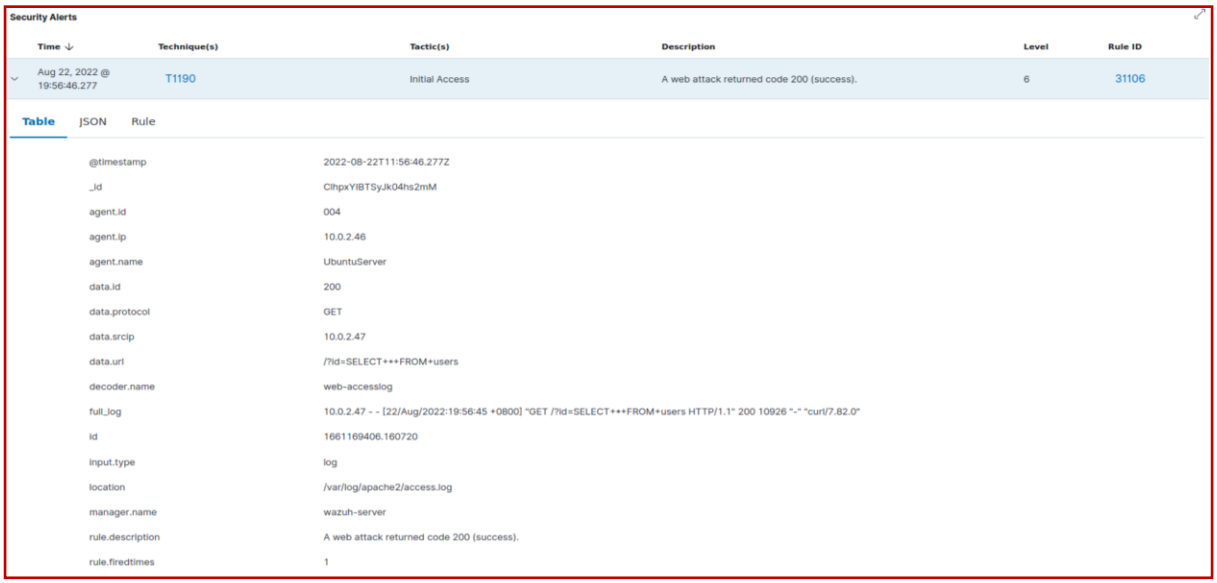
This code will monitor the access logs of the Apache server. After adding the code, I restarted the agent.

This is the details of the ruleset

Then I performed SQL Injection attack in my attacker's machine as shown below

```
┌──(kali㉿kali)-[~]
└─$ curl -XGET "http://10.0.2.46/?id=SELECT+*+FROM+users";
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }
```

After that, I refreshed my Wazuh server, and I was able to detect SQL Injection as shown below in red



We have successfully managed to attack the ubuntu server from the attack machine and was successfully able to detect the SQL Injection. (Web Attack – in this scenario)

3.5 NMAP Probing Attack (NMAP Script Engine Detect)

I have added the below code in my ruleset.

NMAP_Detection.xml

```
<!-- Modify it at your will. -->
<!-- # NMAP Detection Rule  #  -->
<!-- ################################## -->
<group name="NMAP_Detection,">
  <rule id="100100" level="6">
    <if_matched_sid>31101</if_matched_sid>
    <match>Nmap Scripting Engine</match>
    <description>NMap Scripting Engine Detected</description>
    <location>/var/log/apache2/access.log</location>
  </rule>
</group>
```

I have also checked if this code as implemented is available in my local agent path. (We did this during SQL injection)

```
<localfile>
  <log_format>apache</log_format>
  <location>/var/log/apache2/access.log</location>
</localfile>
```

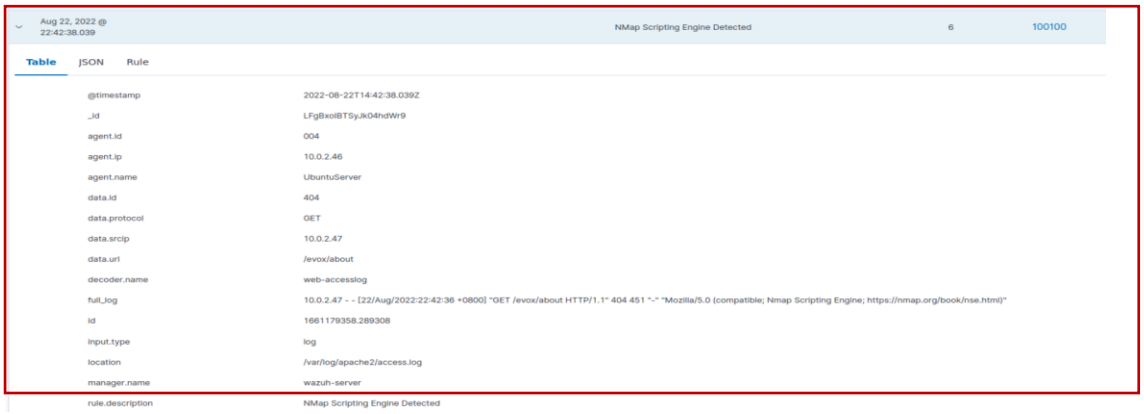
I ran an NMAP script in my attacker machine

```
┌──(kali㉿kali)-[~/Desktop/Project/Attacks]
└─$ sudo nmap -sV -A -O -Pn -sS 10.0.2.46
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-22 10:36 EDT
Nmap scan report for 10.0.2.46
Host is up (0.00030s latency).
Not shown: 996 closed tcp ports (reset)
```

I was able to detect the NMAP scripting rule in my Wazuh as shown below.

Security Alerts

Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Aug 22, 2022 @ 22:42:38.087	T1110.001	Credential Access	PAM: User login failed.	5	5503
Aug 22, 2022 @ 22:42:38.082	T1110.001	Credential Access	PAM: User login failed.	5	5503
Aug 22, 2022 @ 22:42:38.077	T1110.001	Credential Access	PAM: User login failed.	5	5503
Aug 22, 2022 @ 22:42:38.041			Web server 501 error code (Not Implemented).	4	31121
Aug 22, 2022 @ 22:42:38.039			NMap Scripting Engine Detected	6	100100
Aug 22, 2022 @ 22:42:38.033			NMap Scripting Engine Detected	6	100100
Aug 22, 2022 @ 22:42:38.029			NMap Scripting Engine Detected	6	100100
Aug 22, 2022 @ 22:42:38.025			NMap Scripting Engine Detected	6	100100



We have successfully managed to run an NMAP Engine from the attack machine and was successfully able to detect the NMAP Probing Attack that was set on rule 100100 as shown in the rule code above.

4. Conclusion

As cyber threats are becoming more sophisticated in today's world, real-time monitoring and security analysis are required for quick threat detection and remediation. In this PoC, I performed NMAP Probing attacks and they were all successfully detected by Wazuh. Hence, from this PoC, we can conclude that Wazuh is really a good SIEM platform for threat detection.

I have uploaded the Demo Video in OneDrive. Link to the video

<https://sutdapac-my.sharepoint.com/:v:/g/personal/gowtham_baskar_mymail_sutd_edu_sg/EcsTc0fDnXVLpRpJpwRyGpIBhJfeTrf6G9siuO-ZAaEszQ?e=KStRuB>

5. References

Wazuh

1. https://wazuh.com/
2. https://documentation.wazuh.com/current/index.html
3. https://attack.mitre.org/
4. https://www.youtube.com/c/HackerSploit (Wazuh)
5. https://mangolassi.it/topic/21941/wazuh-when-i-write-the-rule-i-encounter-with-a-problem-nmap-scripting/9 (NMAP Scripting)
6. https://www.geeksforgeeks.org/introduction-to-wazuh/

Server Services

1. https://ubuntu.com/server/docs

OS's

1. https://www.kali.org/
2. https://ubuntu.com/

Attacks

1. https://www.thepythoncode.com/article/brute-force-ssh-servers-using-paramiko-in-python
2. https://www.thepythoncode.com/article/brute-force-attack-ftp-servers-using-ftplib-in-python
3. https://documentation.wazuh.com/current/proof-of-concept-guide/