Hardit Singh
Cyber Operations Intern

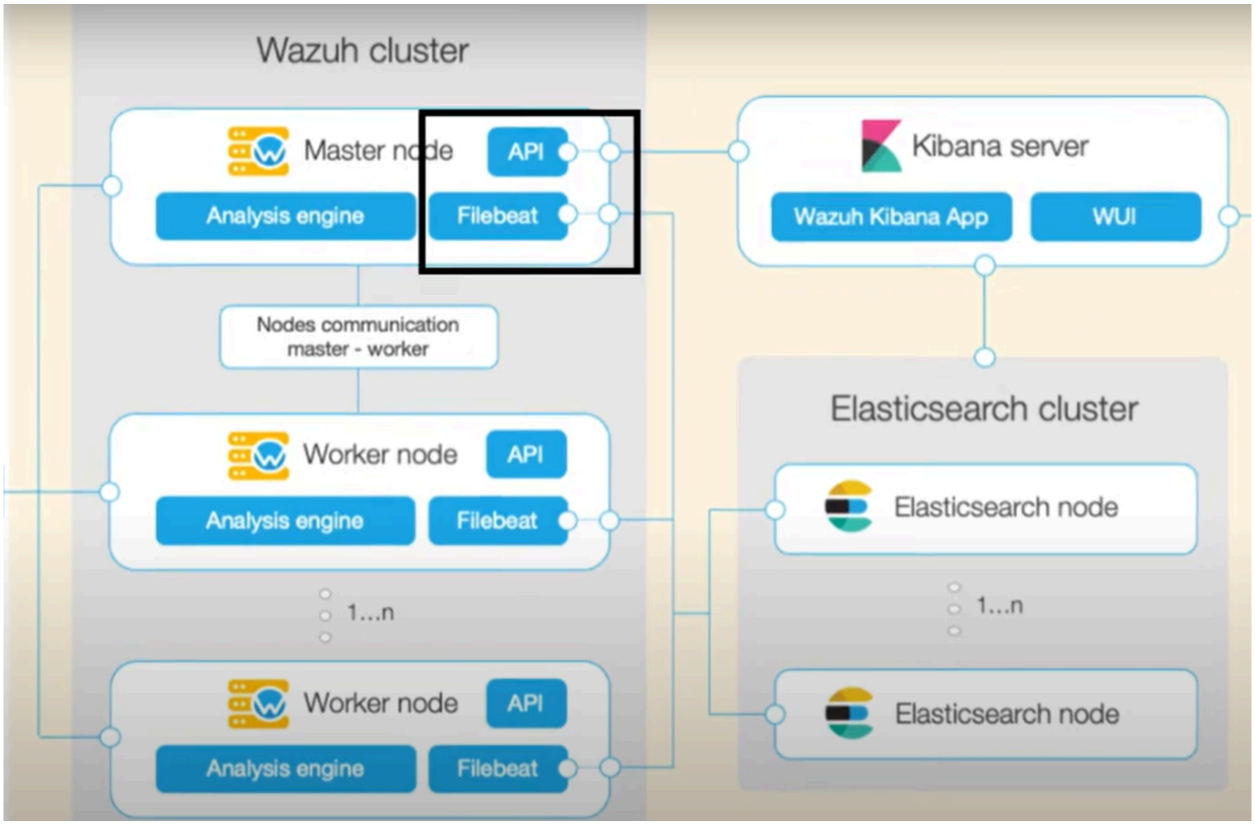
Elastic Stack Capabilities With Wazuh

*Disclaimer: This was created in a testing environment, there is no PII in this report

Introduction:

Wazuh is an open-source security information and event management (SIEM) platform that provides security monitoring and threat detection capabilities. Wazuh integrates with the Elastic Stack, allowing users to leverage the powerful data processing and visualization capabilities of Elasticsearch, Logstash, and Kibana. By integrating with the Elastic Stack, Wazuh enables users to centralize and analyze security-related data from various sources, including logs, events, and alerts, providing a unified view of the security posture. The integration allows Wazuh to send data to Elasticsearch, where it can be indexed, stored, and queried, while Logstash can be used to process and enrich the data before it is ingested into Elasticsearch. Kibana, the visualization and dashboard component of the Elastic Stack, can be used in conjunction with Wazuh to create custom dashboards, visualizations, and reports, providing users with a comprehensive and intuitive security monitoring and analysis experience.

Procedure:

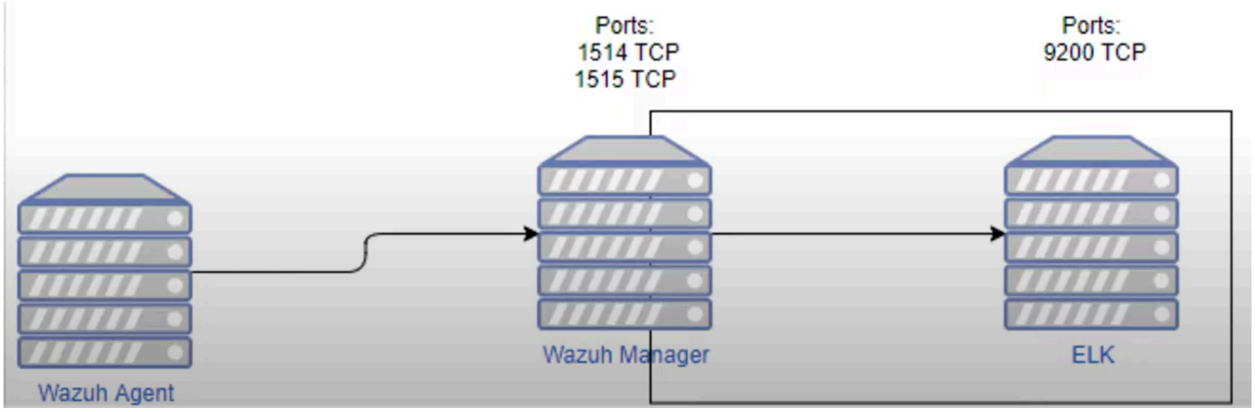


There are two methods of deployment that can be applied -

- All-in-one Deployment
- Distributed Deployment

For the testing environment, the all in one deployment method will be configured, meaning that elasticsearch, kibana, and the Wazuh manager are all running on the same server. For a production environment, a distributed deployment is recommended because you can then allocate more resources to individual servers. If only one resource is being used, then tasks like maintenance, storage and potentially compliance becomes easier. For example, if there are a lot of agents and the logs of those agents are being kept in elasticstack, a separate server that has a lot of storage capability for the elasticstack should be used.

Single or multi node cluster options
For the testing environment we will be using the single node cluster, but in a production environment a multi cluster provides more redundancy. If the Wazuh manage goes down or crashes, a multi-cluster deployment will still receive logs from the agent and this prevents downtime.



Pre-Installation Notes:

- 8GB of Ram should be enough on the Wazuh server - although elastic stack and kibana are memory intensive. In a production environment more memory will be needed.
- All in one deployment being configured
- Single-Node being configured
- Assisted installation method

- Wazuh version 4.5
- Curl needs to be installed

Documentation:
Wazuh provides excellent documentation.
All-in-one deployment - Installing Wazuh with Elastic Stack
Provides an assisted installation method which will be used

1. Add GPG Key:

```
rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

2. Add the repository

```
cat > /etc/yum.repos.d/elastic.repo << EOF
[elasticsearch-7.x]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
EOF
```

3. Install ElasticSearch package:

```
apt-get install elasticsearch=7.17.13
```

4. Download the configuration file:

```
curl -so /etc/elasticsearch/elasticsearch.yml
https://packages.wazuh.com/4.5/tpl/elastic-basic/elasticsearch_all_in_one.yml
```

5. Certificate creation:

```
curl -so /usr/share/elasticsearch/instances.yml
https://packages.wazuh.com/4.5/tpl/elastic-basic/instances_aio.yml
```

6. Certificates are created using elasticsearch-certutil

```
/usr/share/elasticsearch/bin/elasticsearch-certutil cert ca --pem --in instances.yml --keep-ca-key
--out ~/certs.zip
```

7. Extract Certificate

8. Create the directory /etc/elasticsearch/certs, and then copy the CA file, the certificate and the key there:

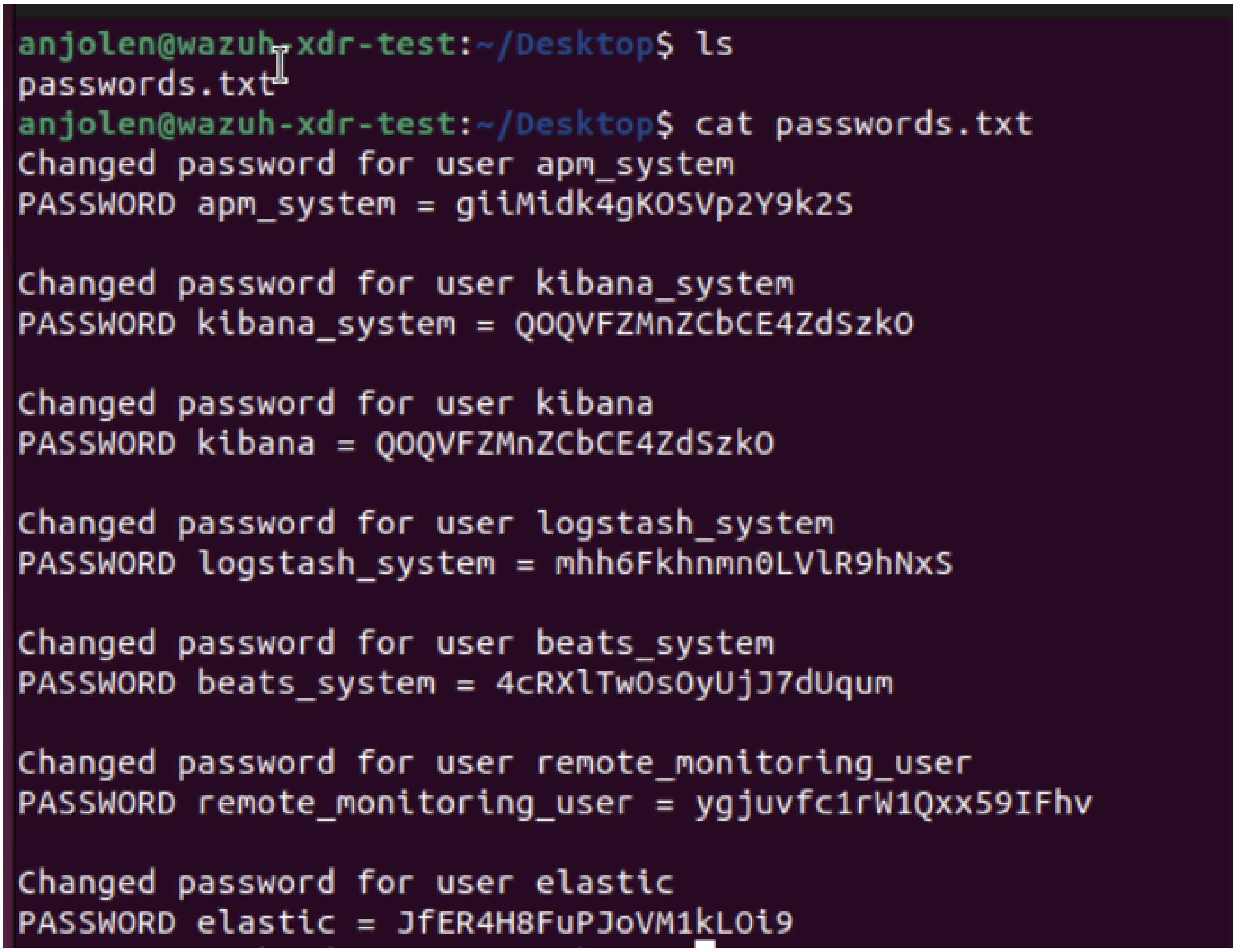
```
mkdir /etc/elasticsearch/certs/ca -p
cp -R ~/certs/ca/ ~/certs/elasticsearch/* /etc/elasticsearch/certs/
chown -R elasticsearch: /etc/elasticsearch/certs
chmod -R 500 /etc/elasticsearch/certs
chmod 400 /etc/elasticsearch/certs/ca/ca.* /etc/elasticsearch/certs/elasticsearch.*
rm -rf ~/certs/ ~/certs.zip
```

9.Enable and start the Elasticsearch service:

```
systemctl daemon-reload
systemctl enable elasticsearch
systemctl start elasticsearch
```

10. Generate credentials for all the Elastic Stack pre-built roles and users:

```
/usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
```

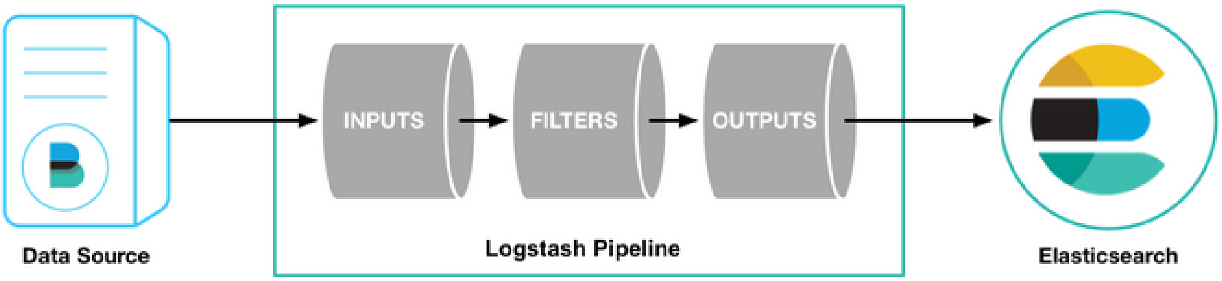


```
anjolen@wazuh-xdr-test:~/Desktop$ ls
passwords.txt
anjolen@wazuh-xdr-test:~/Desktop$ cat passwords.txt
Changed password for user apm_system
PASSWORD apm_system = giiMidk4gKOSVp2Y9k2S

Changed password for user kibana_system
PASSWORD kibana_system = QOQVFZMnZCbCE4ZdSzkO

Changed password for user kibana
PASSWORD kibana = QOQVFZMnZCbCE4ZdSzkO

Changed password for user logstash_system
PASSWORD logstash_system = mhh6Fkhnmn0LVlR9hNxS

Changed password for user beats_system
PASSWORD beats_system = 4cRXlTwOsOyUjJ7dUqum

Changed password for user remote_monitoring_user
PASSWORD remote_monitoring_user = ygjuvfc1rW1Qxx59IFhv

Changed password for user elastic
PASSWORD elastic = JfER4H8FuPJoVM1kLOi9
```

Follow the documentation provided by the link above. It is made by Wazuh and will step by step guide you through an assisted installation and integration of Wazuh and Elastic Stack

The above procedure and commands only creates the elastic user:

- The user must install the Wazuh Server
- Configure Filebeat - this is responsible for shipping the alerts .json to elasticsearch Elastic search will store and query these logs
- Install Kibana which will be used as the web UI to view and query calls to elastic search.



Agent Configuration:

Now that the Wazuh Server and elastic stack integration is complete, agents must be added in order for the Server to aggregate real data. In the testing environment, a Ubuntu agent will be used as well as a windows agent as the test subjects. These agents will be exposed to various penetration testing methods such as using an nmap scan or even a brute force attack. The objective of this is to understand the capabilities of Wazuh and see how well these attacks are being documented.

Wazuh has brilliant documentation for configuring a new agent. It is very simple and it guides the user step by step as well as providing the command that has to be used on the agent's side to complete the connection.
[Deploying Wazuh agents on Linux endpoints - Wazuh agent]

Here is the step by step commands to add a agent to the Wazuh/Elastic Stack SIEM

apt-get install gnupg apt-transport-https curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -

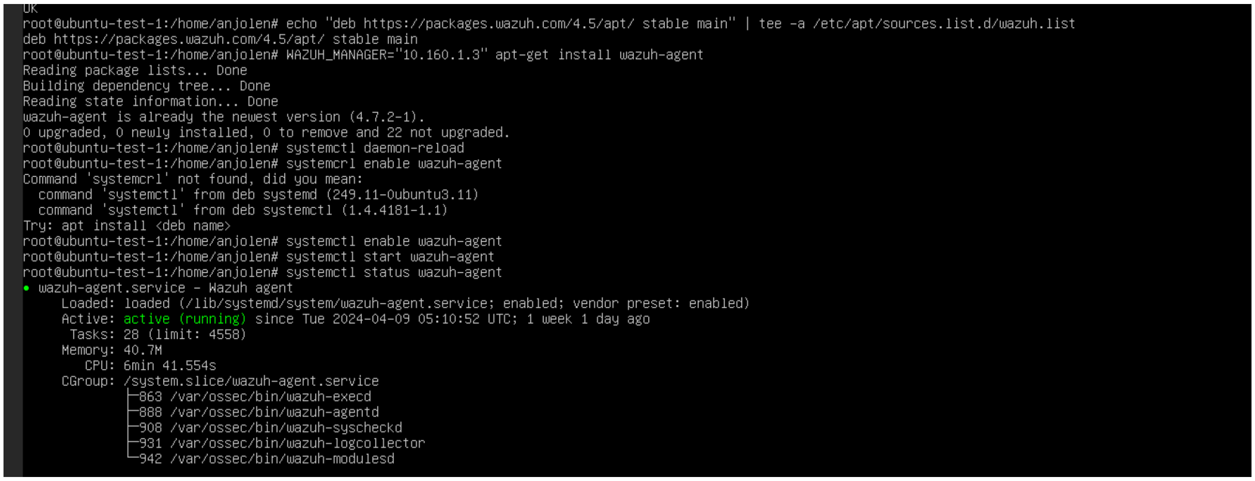
echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list

Then deploy the Wazuh Agent on the endpoint:

WAZUH_MANAGER="10.160.1.3" apt-get install wazuh-agent

Then restart services:

systemctl daemon-reload
systemctl enable wazuh-agent
systemctl start wazuh-agent

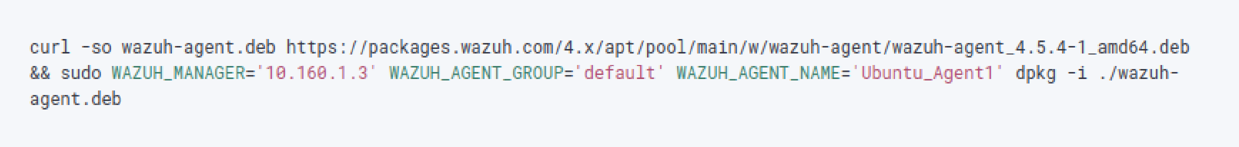


```
OK
root@ubuntu-test-1:/home/anjolen# echo "deb https://packages.wazuh.com/4.5/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
deb https://packages.wazuh.com/4.5/apt/ stable main
root@ubuntu-test-1:/home/anjolen# WAZUH_MANAGER="10.160.1.3" apt-get install wazuh-agent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wazuh-agent is already the newest version (4.7.2-1).
0 upgraded, 0 newly installed, 0 to remove and 22 not upgraded.
root@ubuntu-test-1:/home/anjolen# systemctl daemon-reload
root@ubuntu-test-1:/home/anjolen# systemcrl enable wazuh-agent
Command 'systemcrl' not found, did you mean:
  command 'systemctl' from deb systemd (249.11-0ubuntu3.11)
  command 'systemctl' from deb systemctl (1.4.4181-1.1)
Try: apt install <deb name>
root@ubuntu-test-1:/home/anjolen# systemctl enable wazuh-agent
root@ubuntu-test-1:/home/anjolen# systemctl start wazuh-agent
root@ubuntu-test-1:/home/anjolen# 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 Tue 2024-04-09 05:10:52 UTC; 1 week 1 day ago
      Tasks: 28 (limit: 4558)
     Memory: 40.7M
        CPU: 6min 41.554s
     CGroup: /system.slice/wazuh-agent.service
             ├─863 /var/ossec/bin/wazuh-execd
             ├─888 /var/ossec/bin/wazuh-agentd
             ├─908 /var/ossec/bin/wazuh-syscheckd
             ├─931 /var/ossec/bin/wazuh-logcollector
             └─942 /var/ossec/bin/wazuh-modulesd
```

*As seen the wazuh agent service is active and running

In further findings, there is an easier way to configure an agent on the server GUI directly. Under the Agents tab, click on add a new agent.

It then gives step by step documentation and a singular command that is to be used on the endpoint.



```
curl -so wazuh-agent.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.5.4-1_amd64.deb
&& sudo WAZUH_MANAGER='10.160.1.3' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='Ubuntu_Agent1' dpkg -i ./wazuh-
agent.deb
```

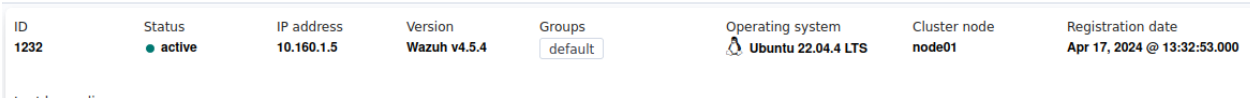
The command is specific to what I wanted - the manager IP is my Wazuh server's and I wanted the default group as well as the name Ubuntu_Agent1

Recommended action - Disable Wazuh updates

For smoothest operation, the Wazuh manager needs to be the same version or a newer version than the Wazuh agent. Upgrading the agent can cause issues, it is recommended pausing updates from the Wazuh repository. Here's the command to do that:

```
echo "wazuh-agent hold" | dpkg --set-selections
```

Agent connected to SIEM –



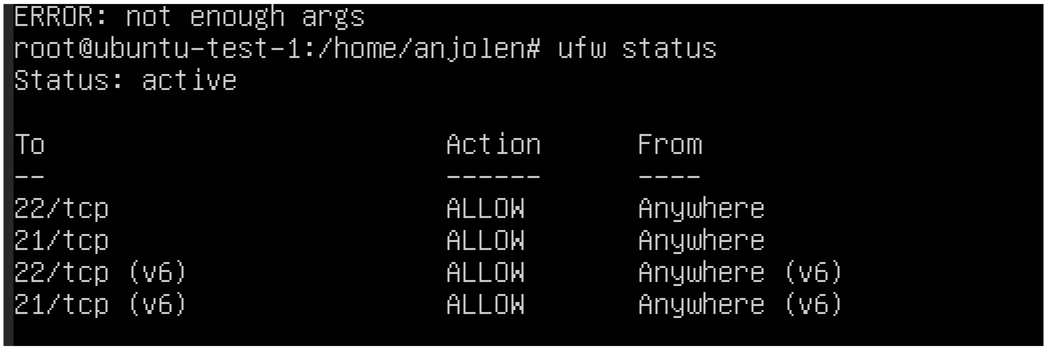
ID	Status	IP address	Version	Groups	Operating system	Cluster node	Registration date
1232	● active	10.160.1.5	Wazuh v4.5.4	default	Ubuntu 22.04.4 LTS	node01	Apr 17, 2024 @ 13:32:53.000

Testing Log Aggregation with Penetration Testing:

Now a small White box Penetration Test will be conducted in order to test log aggregation and the SIEM's functionality.

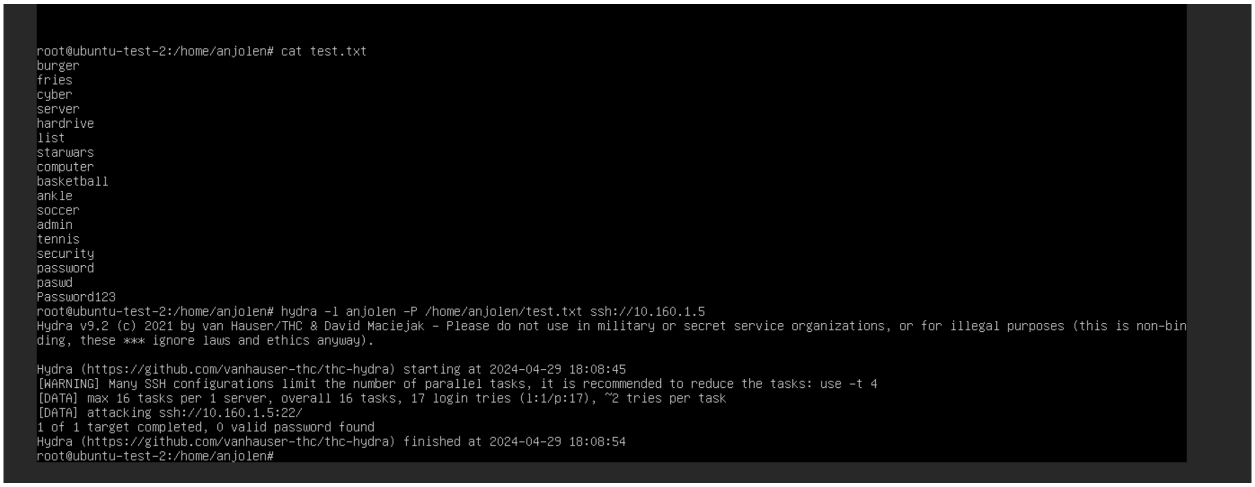
On the Ubuntu Agent, UFW had to be enabled and the ports 21 for FTP and 22 for SSH were opened. On another Ubuntu machine a brute force attack will be conducted, one with the correct password and one without. The objective is to see how well the agent communicated with the server and displays that an attack is happening. Wazuh is known for its extended detection and response capabilities so we will test its capabilities with this brute force attack. I will make a password.txt list and use that for the hydra attack – the correct ssh password will not be in it to test how well the SIEM can log the attack.

UFW Status on Target 10.160.1.5/24



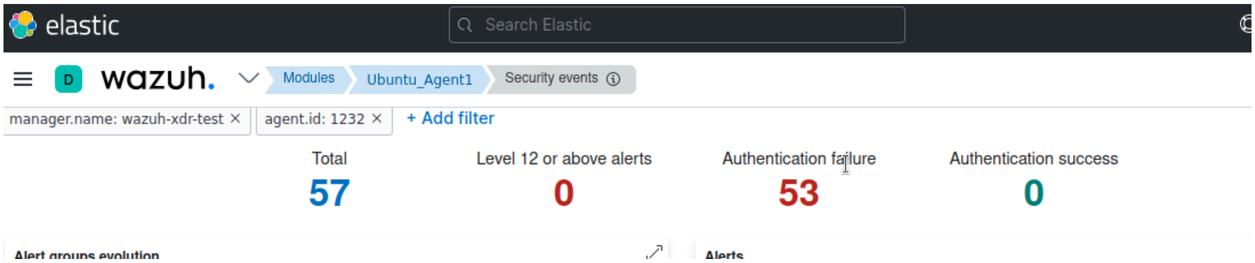
```
ERROR: not enough args
root@ubuntu-test-1:/home/anjolen# ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
```

Creation of the test.txt and the following Hydra command:
Hydra -l anjolen -P /home/anjolen/test.txt ssh://10.160.1.5



```
root@ubuntu-test-2:/home/anjolen# cat test.txt
burger
fries
cyber
server
hardrive
list
starwars
computer
basketball
ankle
soccer
admin
tennis
security
password
paswd
Password123
root@ubuntu-test-2:/home/anjolen# hydra -l anjolen -P /home/anjolen/test.txt ssh://10.160.1.5
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-29 18:08:45
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 17 login tries (l:1/p:17), ~2 tries per task
[DATA] attacking ssh://10.160.1.5:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-29 18:08:54
root@ubuntu-test-2:/home/anjolen#
```

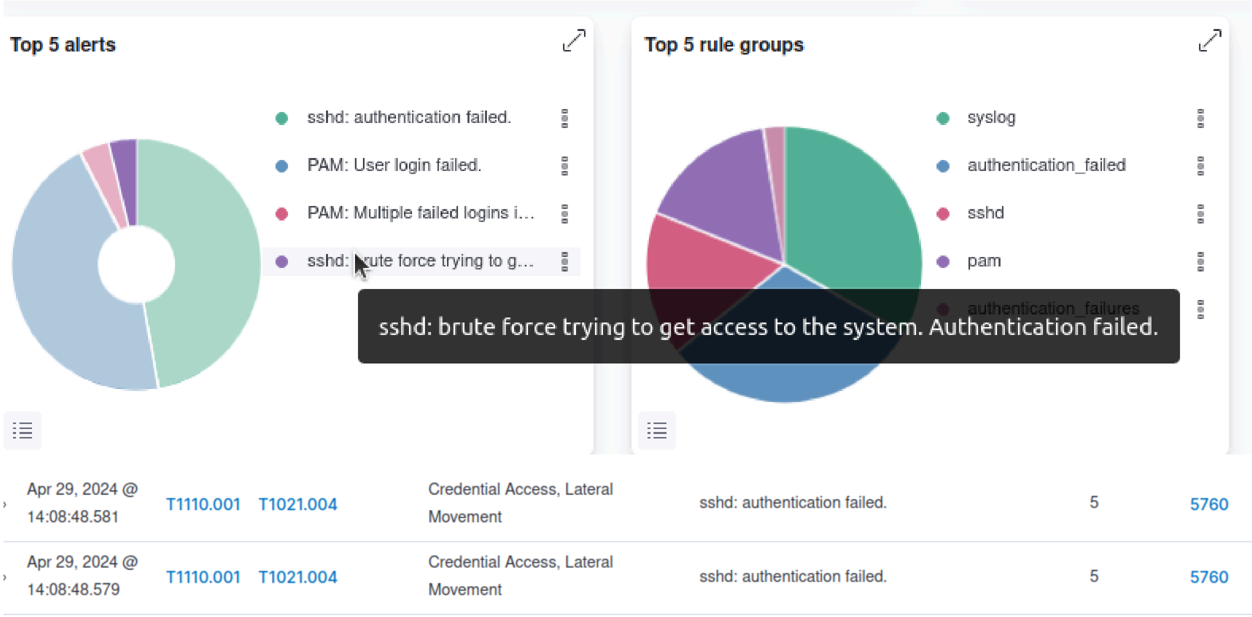
Results Under the Wazuh Manager:



manager.name: wazuh-xdr-test ✕ agent.id: 1232 ✕

Total 57
Level 12 or above alerts 0
Authentication failure 53
Authentication success 0

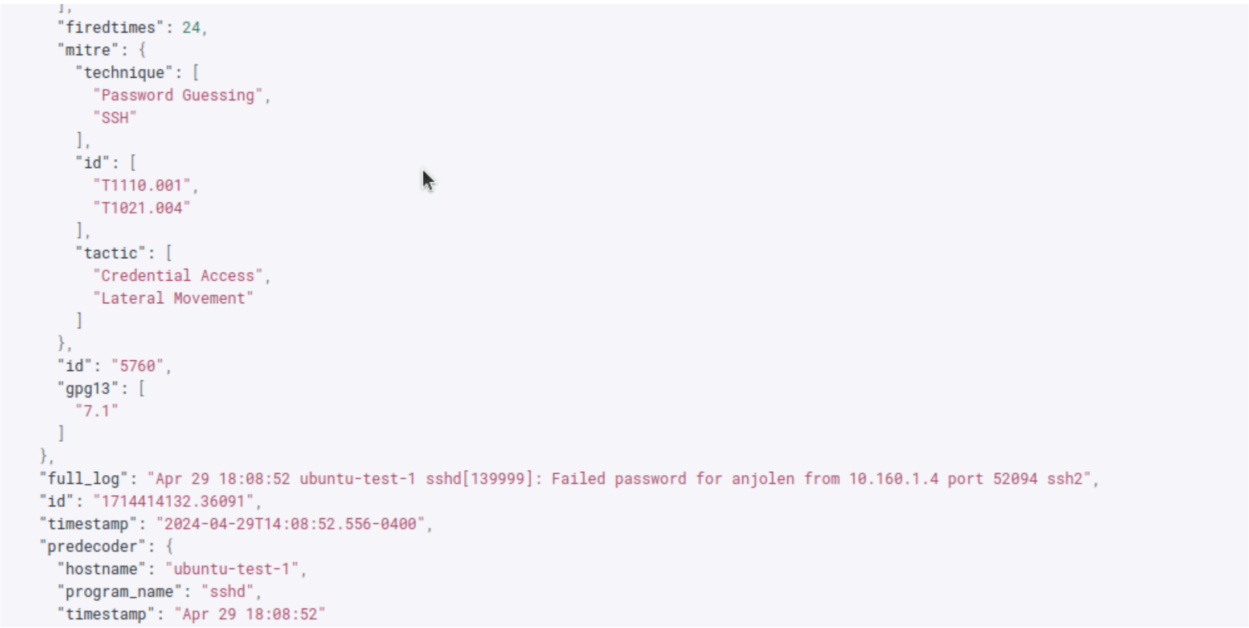
The SIEM caught the Brute Force Attack:


Top 5 alerts
● sshd: authentication failed.
● PAM: User login failed.
● PAM: Multiple failed logins i...
● sshd: brute force trying to g...
Top 5 rule groups
● syslog
● authentication_failed
● sshd
● pam
● authentication_failures

sshd: brute force trying to get access to the system. Authentication failed.

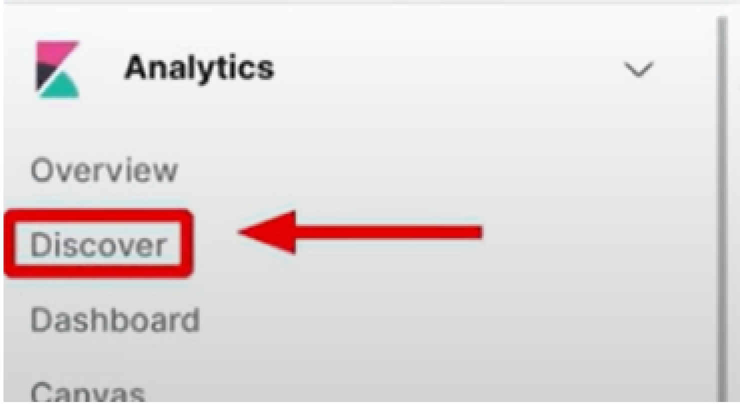
Apr 29, 2024 @ 14:08:48.581	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: authentication failed.	5	5760
Apr 29, 2024 @ 14:08:48.579	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: authentication failed.	5	5760

When clicking on a specific event and then moving over to the JSON tab. Results including attacker IP address, time, tactics, methods and attempt numbers are logged and stored. In the JSON screenshot below, it shows the amount of passwords attempted which was 24, the tactics which is credential access/lateral movement, and finally the most important one being the attackers IP address which is 10.160.1.4 on port 52094.

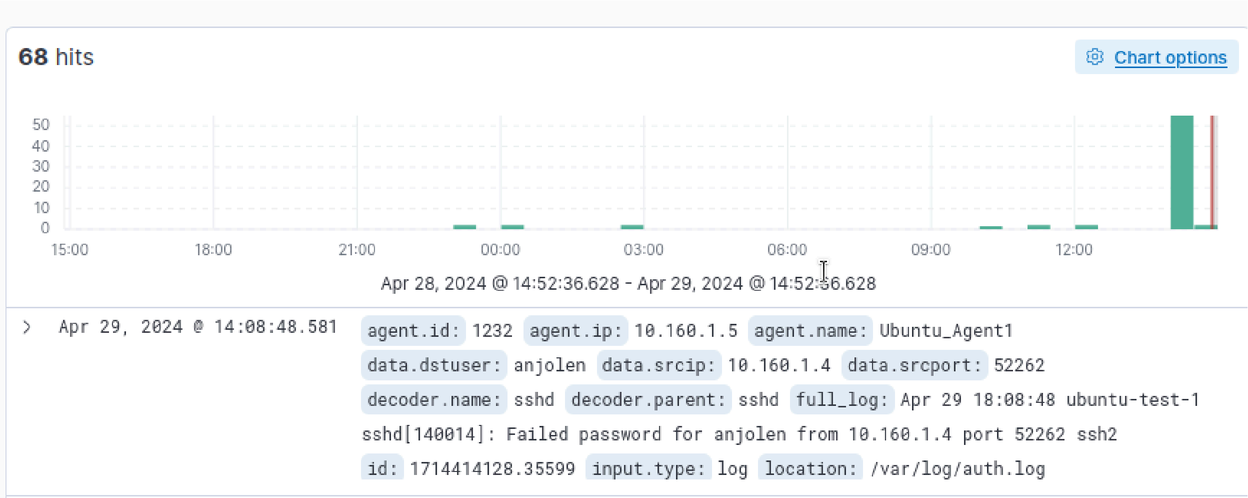


```
    ],
    "firedtimes": 24,
    "mitre": {
      "technique": [
        "Password Guessing",
        "SSH"
      ],
      "id": [
        "T1110.001",
        "T1021.004"
      ],
      "tactic": [
        "Credential Access",
        "Lateral Movement"
      ]
    },
    "id": "5760",
    "gpg13": [
      "7.1"
    ]
  },
  "full_log": "Apr 29 18:08:52 ubuntu-test-1 sshd[139999]: Failed password for anjolen from 10.160.1.4 port 52094 ssh2",
  "id": "1714414132.36091",
  "timestamp": "2024-04-29T14:08:52.556-0400",
  "predecoder": {
    "hostname": "ubuntu-test-1",
    "program_name": "sshd",
    "timestamp": "Apr 29 18:08:52"
```

Under the Kibana Analytics tab we can view the same JSON files but with data processing and visualization capabilities of Kibana



Here is the Kibana entry:
An item to notice is that it shows the directory in which the log/event is kept
location: /var/log/auth.log



```
> Apr 29, 2024 @ 14:08:48.581   agent.id: 1232 agent.ip: 10.160.1.5 agent.name: Ubuntu_Agent1
                                data.dstuser: anjolen data.srcip: 10.160.1.4 data.srcport: 52262
                                decoder.name: sshd decoder.parent: sshd full_log: Apr 29 18:08:48 ubuntu-test-1
                                sshd[140014]: Failed password for anjolen from 10.160.1.4 port 52262 ssh2
                                id: 1714414128.35599 input.type: log location: /var/log/auth.log
```

Conclusion:

Overall, with its extended detection and response capabilities, Wazuh can definitely be something used in a production environment. Wazuh is a robust solution for modern security information and event management (SIEM) needs, particularly owing to its integration with the Elastic Stack. Its compatibility not only ensures efficient data management but also enhances the scalability and effectiveness of security operations in a production environment. By leveraging the comprehensive features of Wazuh alongside Elastic Stack's powerful analytics and visualization capabilities, a company can significantly improve its cybersecurity posture. The biggest upside lies in its cost as it is an open source and free tool. The integration of the elastic stack allows for scalability if one chooses that route as the elastic stack offers more tools to pair with the SIEM, but unfortunately it is at a price. There have been security vulnerabilities mentioned including CVE-2024-32038 but this is fixed in Wazuh versions greater than 4.7.