1.18. Learning outcome Security

Proposal

The challenge that I'm proposing is Sec1 because you can start with the basics and slowly expand to a more difficult challenge. In my challenge, I'll take measures. With SOAR I'll try to protect the DNS, the mail, and all unwanted ports.

ID	Level	Topic	Description	Capacity
Sec1	Basic Advanced	Deploy a SOAR infrastructure.	Deploy a SOAR (Security Orchestration, Automation and Response) infrastructure and demonstrate that core functionality works properly a.o. by emulating an attack. For advanced: add endpoint security, automated vulnerability scanning and threat intel feeds. A "Getting started with SOAR" guide can be found in the Security resources section.	5 VM
Sec2a	Basic	Applied security concepts: DNS security	Make an informed choice for one or more mechanisms (DoH, DoT, DNSSEC) for securing DNS traffic and realising the solution. The measure sent data before and after securing DNS and used the data to explain why it is now more secure against specific threats (e.g. by using the CIA triad).	

Challenge 1

Sec1

1.18.1.1. Analysis

In this challenge we get the options to make a SOAR environment and we get multiple options:

- Elk Stack
- OpenVAS
- Wazuhmanager
- The HIVE and Cortex
- MISP
- Snort

Their functions

ELK Stack:

ELK stands for Elasticsearch Logstash Kibana.

ELK STACK: Kibana (Visualizes the data) + Elasticsearch (Search and analytics engine) + Logstash (Processing pipeling -> collects data from multiple sources, transforms/changes it and sends it to a stash like Elasticsearch)

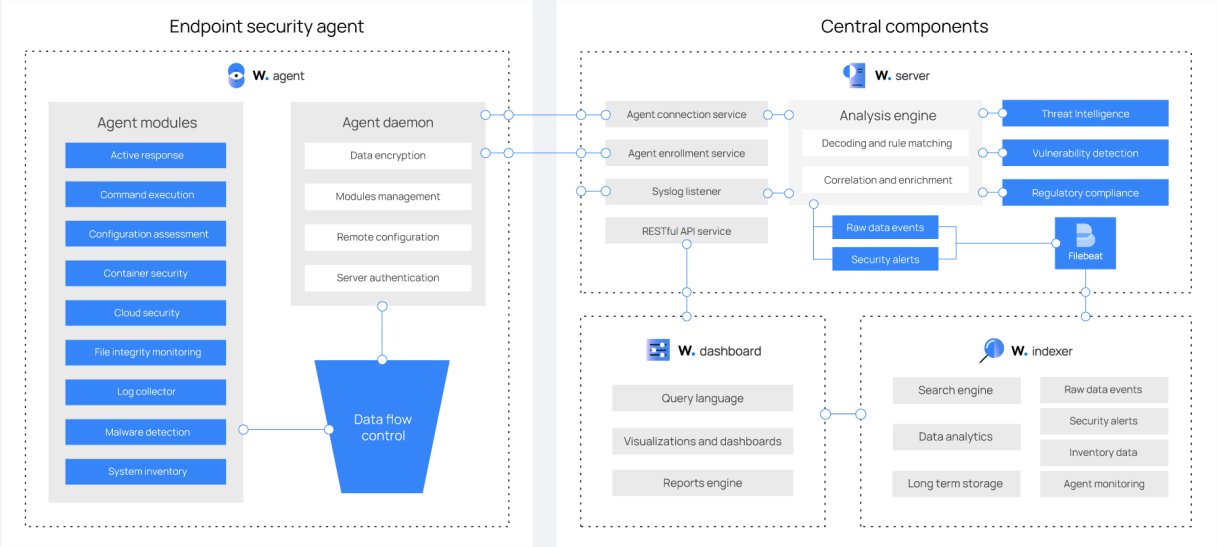
OpenVAS:

OpenVAS: OpenVAS is a vulnerability scanner. It has a database of known weaknesses and exploits and searches for weaknesses/exploits. This can be done by a penetration testing tool that OpenVAS has built-in Snort: Network Intrusion Detection System and Intrusion Prevention System. Snort IPS uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generates alerts for users. Snort can be deployed inline to stop these packets, as well. Source: "https://snort.org/"

Wazuhmanager:

The Wazuh platform helps organizations and individuals protect their data assets through threat prevention, detection, and response. Besides, Wazuh is also employed to meet regulatory

compliance requirements, such as PCI DSS or HIPAA, and configuration standards like CIS hardening guides.



Source "https://wazuh.com/", "https://documentation.wazuh.com/current/getting-started/index.html"

The HIVE and Cortex:

TheHive

TheHive is a 4 in 1 incident response could be handy in a SOAR environment

Cortex

The Cortex is responsible for scanning for any possible threats from mail to IPs from websites. It is quite a handy tool to scan if something malicious is going on the company offers quotes from the company itself:

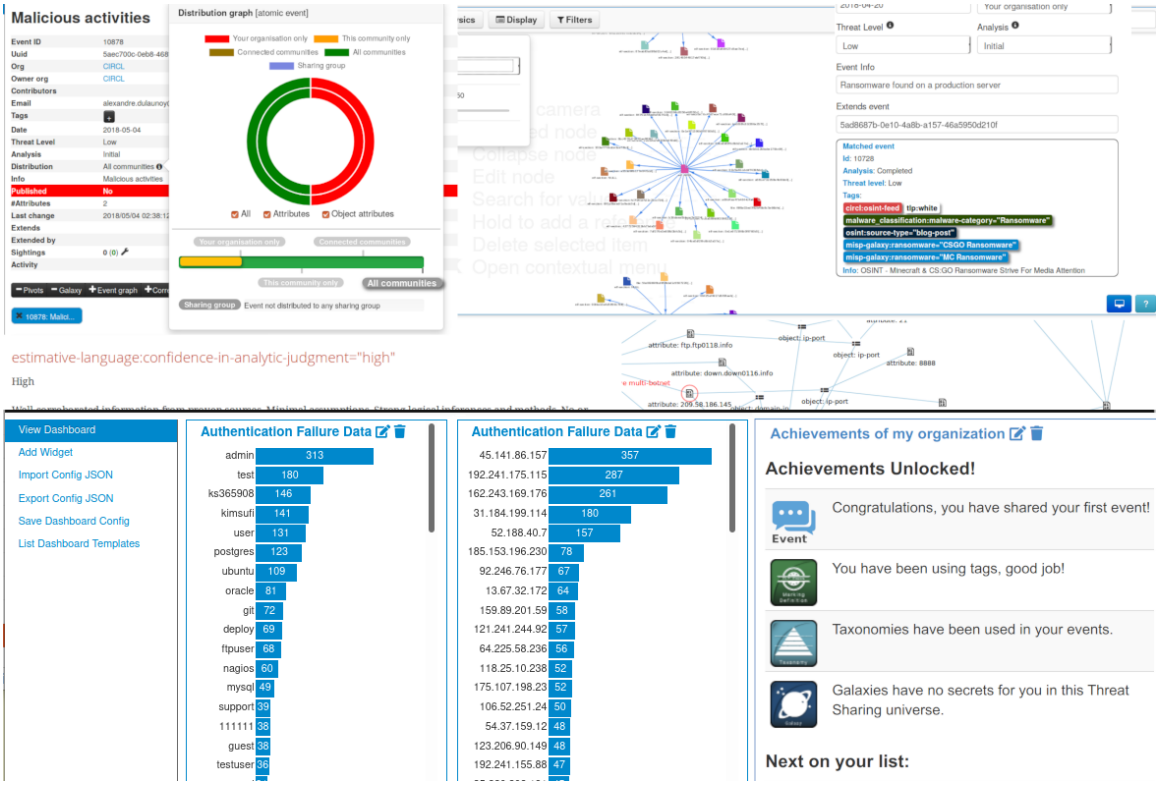
"Thanks to Cortex, observables such as IP and email addresses, URLs, domain names, files or hashes can be analyzed using a Web interface. Analysts can also automate these operations and submit large sets of observables from TheHive or through the Cortex REST API from alternative

SIRP platforms, custom scripts or MISP. When used in conjunction with TheHive, Cortex largely facilitates the containment phase thanks to its Active Response features.” - TheHive Project homepage (By T. Franco, S. Kadhi, J. Leonard, N. Adouani, D. Co, & N. Kuhert). (2020). TheHive Project. Retrieved September 28, 2022, from https://thehive-project.org/

Source “https://github.com/TheHive-Project/Cortex”, “https://github.com/TheHive-Project/TheHive”

MISP:

Malware Information Sharing Platform or MISP is an open source threat intelligent which means it can scan the metadata and tags for any malicious it can find.MISP also offers a dashboard/visualization to give you an overview of what it has found or just to monitor things.



This open-source project has its own build in automation feature which makes it a must for the automated environment, not only that there are a lot of features that won’t fit in this page quote from their website.

- *An efficient IoC and indicators database allowing to storage technical and non-technical information about malware samples, incidents, attackers and intelligence.*
- *Automatic correlation finding relationships between attributes and indicators from malware, attacks campaigns or analysis. Correlation engine includes correlation between attributes and more advanced correlations like Fuzzy hashing correlation (e.g. ssdeep) or CIDR block matching. Correlation can be also enabled or event disabled per attribute.*
- *A flexible data model where complex objects can be expressed and linked together to express threat intelligence, incidents or connected elements.*
- *Built-in sharing functionality to ease data sharing using different model of distributions. MISP can synchronize automatically events and attributes among different MISP. Advanced filtering functionalities can be used to meet each organization sharing policy including a flexible sharing group capacity and an attribute level distribution mechanisms.*
- *An intuitive user-interface for end-users to create, update and collaborate on events and attributes/indicators. A graphical interface to navigate seamlessly between events and their correlations. An event graph functionality to create and view relationships between objects and attributes. Advanced filtering functionalities and warning list to help the analysts to contribute events and attributes.*
- *storing data in a structured format (allowing automated use of the database for various purposes) with an extensive support of cyber security indicators along fraud indicators as in the financial sector.*
- *export: generating IDS (Suricata, Snort and Bro are supported by default), OpenIOC, plain text, CSV, MISP XML or JSON output to integrate with other systems (network IDS, host IDS, custom tools*
- *import: bulk-import, batch-import, free-text import, import from OpenIOC, GFI sandbox, ThreatConnect CSV or MISP format.*
- *Flexible free text import tool to ease the integration of unstructured reports into MISP.*
- *A gentle system to collaborate on events and attributes allowing MISP users to propose changes or updates to attributes/indicators.*
- *data-sharing: automatically exchange and synchronization with other parties and trust-groups using MISP.*

- *feed import: flexible tool to import and integrate MISP feed and any threatintel or OSINT feed from third parties. Many default feeds are included in standard MISP installation.*
- *delegating of sharing: allows a simple pseudo-anonymous mechanism to delegate publication of event/indicators to another organization.*
- *Flexible API to integrate MISP with your own solutions. MISP is bundled with PyMISP which is a flexible Python Library to fetch, add or update events attributes, handle malware samples or search for attributes.*
- *adjustable taxonomy to classify and tag events following your own classification schemes or existing taxonomies. The taxonomy can be local to your MISP but also shareable among MISP instances. MISP comes with a default set of well-known taxonomies and classification schemes to support standard classification as used by ENISA, Europol, DHS, CSIRTs or many other organisations.*
- *intelligence vocabularies called MISP galaxy and bundled with existing threat actors, malware, RAT, ransomware or MITRE ATT&CK which can be easily linked with events in MISP.*
- *expansion modules in Python to expand MISP with your services or activate already available misp-modules.*
- *sighting support to get observations from organizations concerning shared indicators and attributes. Sighting can be contributed via the MISP user interface, API as MISP document or STIX sighting documents. Starting with MISP 2.4.66, Sighting has been extended to support false-negative sighting or expiration sighting.*
- *STIX support: export data in the STIX format (XML and JSON) including export/import in STIX 2.0 format.*
- *integrated encryption and signing of the notifications via PGP and/or S/MIME depending on the user preferences.*
- *Real-time publish-subscribe channel within MISP to automatically get all changes (e.g. new events, indicators, sightings or tagging) in ZMQ (e.g. misp-dashboard) or Kafka.*

- *MISP features and functionalities. (n.d.). MISP Open Source Threat Intelligence Platform & Open Standards for Threat Information Sharing. Retrieved September 28, 2022, from https://www.misp-project.org/features/*

  Crazy if you ask me nowadays it's quite normal due to a lot of cyberattacks.

SNORT:

Snort is also an open-source project that originated from one person who was later bought by Cisco and took it over. The main use case for this software is intrusion detection/prevention. Snort has 3 main features:

Sniffer Mode

Like Wireshark reads packages that are being transmitted through your network traffic and displays it on the screen.

Packet Logger Mode

This is self-explanatory but if you don't know what it does it saves everything that ik found while sniffing or other things into one or more text files for you to read later or to do diagnostics on it which is default nowadays.

Network Intrusion Detection System Mode

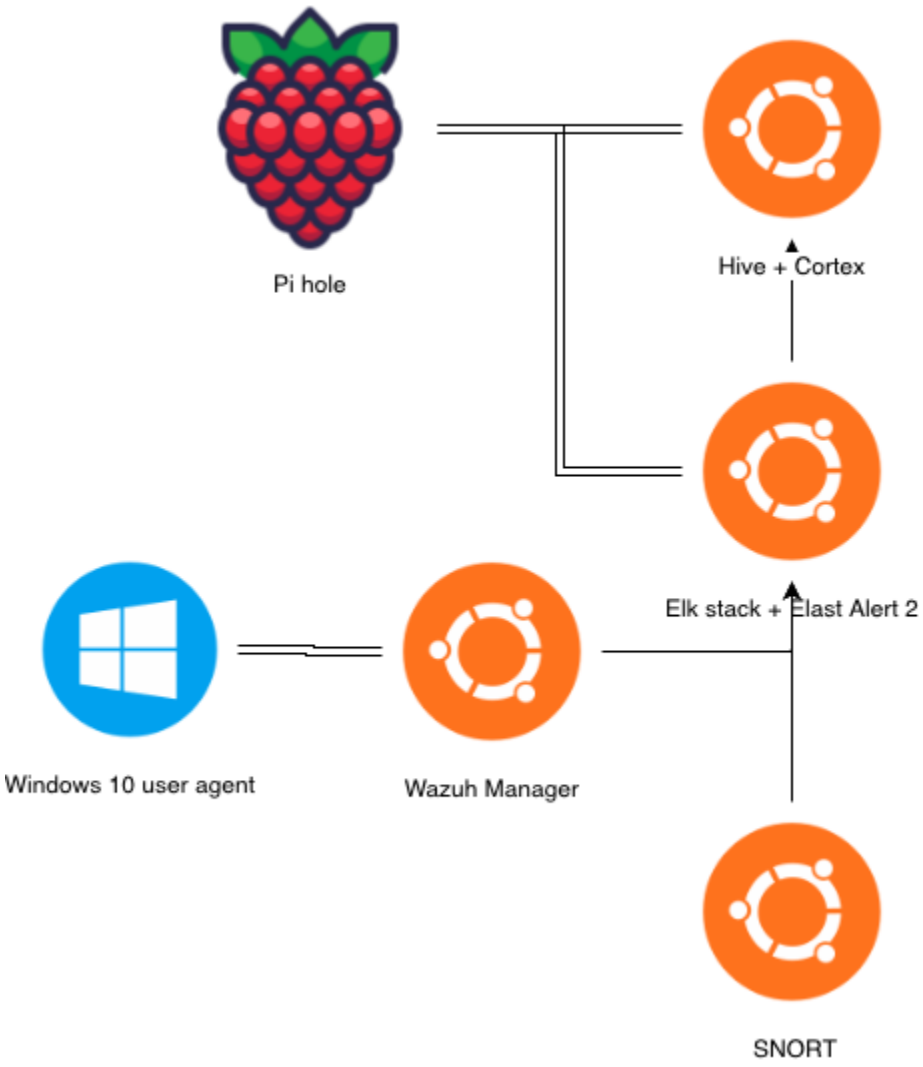
Network intrusion Detection System mode is when a user sets rules or blocks some of the data/packages the system takes a closer look and if it finds something that is not allowed it will notify you and it will write you the information in detail.

Source "https://en.wikipedia.org/wiki/Snort_(software)"

1.18.1.2. Design

My choice are:

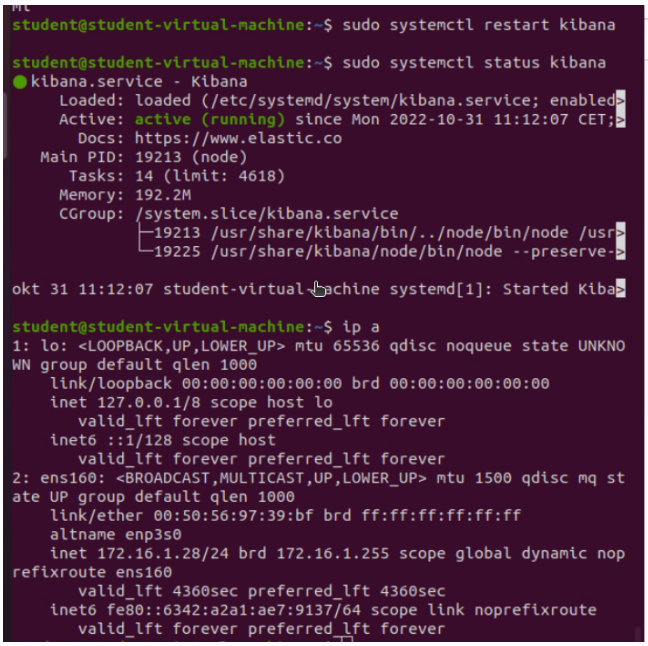
- ELK Stack
- Wazuhmanager
- Snort
- OpenVAS
- TheHive and Cortex

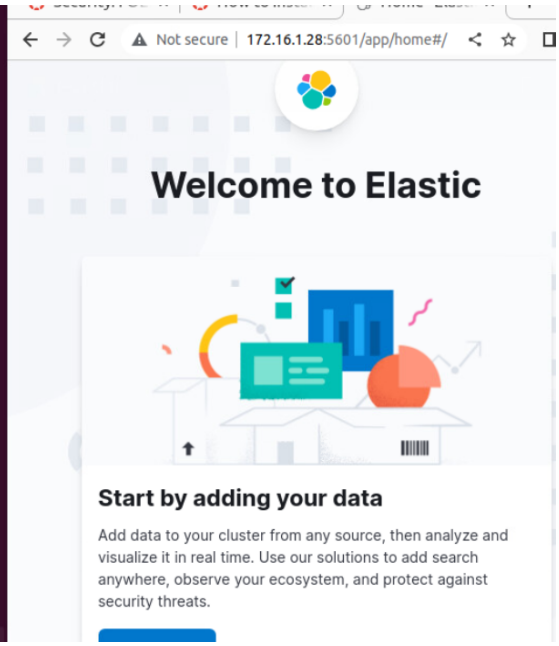


1.18.1.3. Realisation

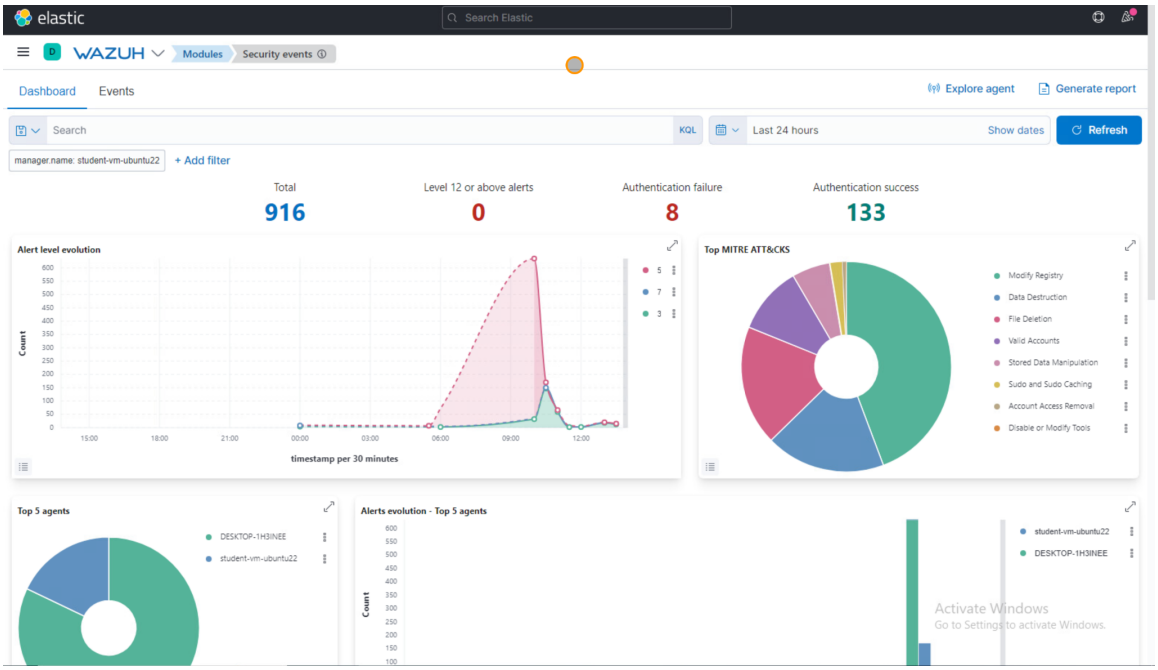
Elk stack

After following the guide I installed ELK stack without any issues. Next was installing a kibana. After following the guide and registering it to start from systemctl, you can see that kibana is up and running.

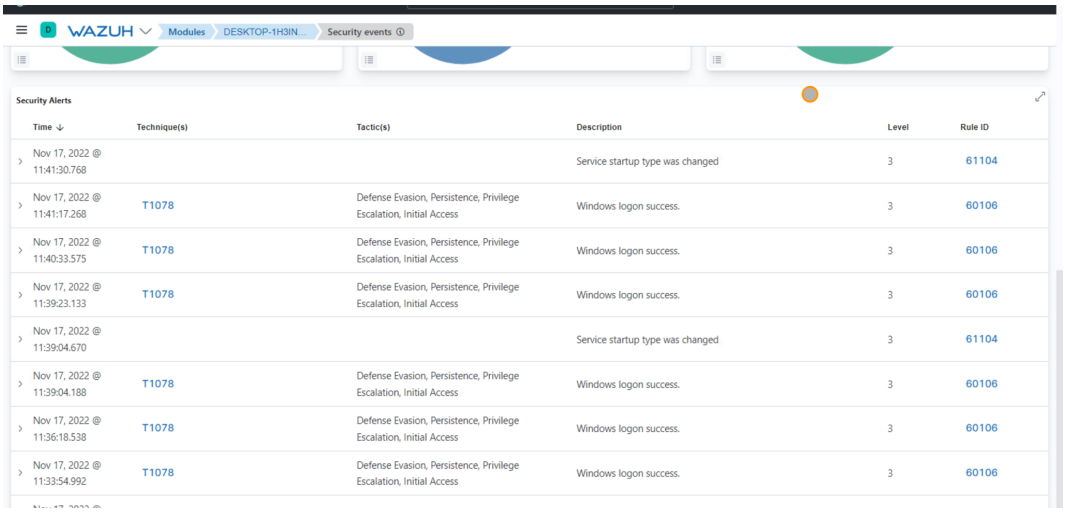


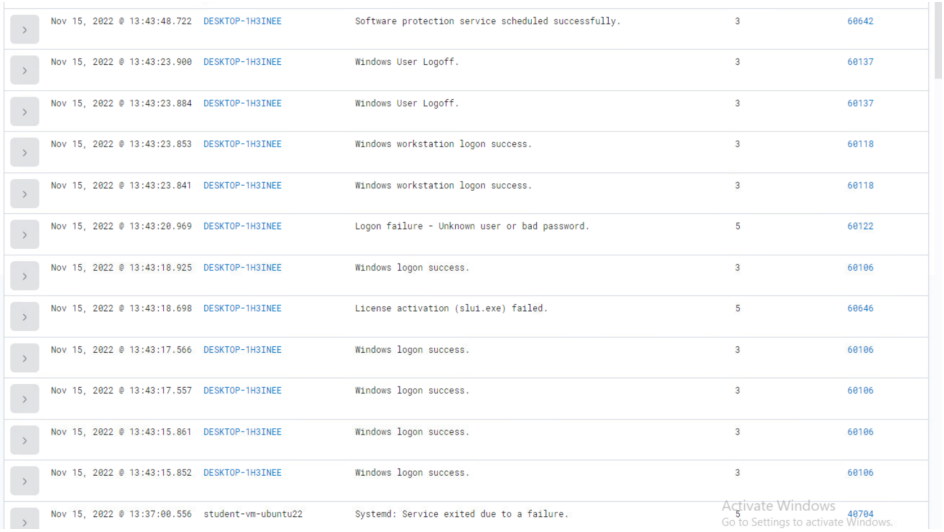


After installing WAZUH and connecting the api to ELK Stack which is to see in Once I installed my Wazuh agent on my Windows it started reporting Windows events.



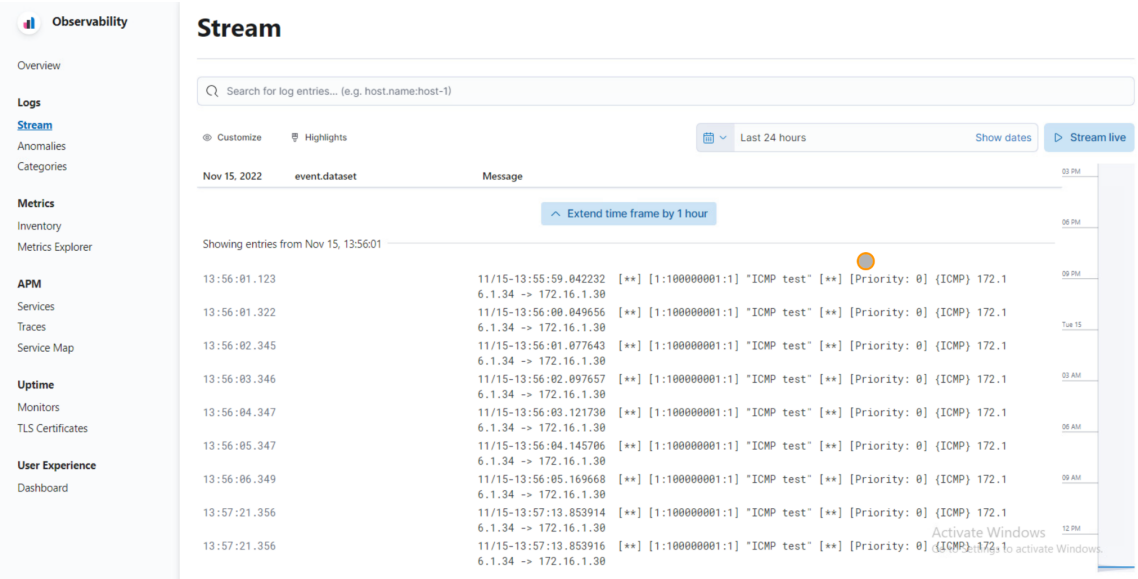
After selecting the agent it shows all the events it got received



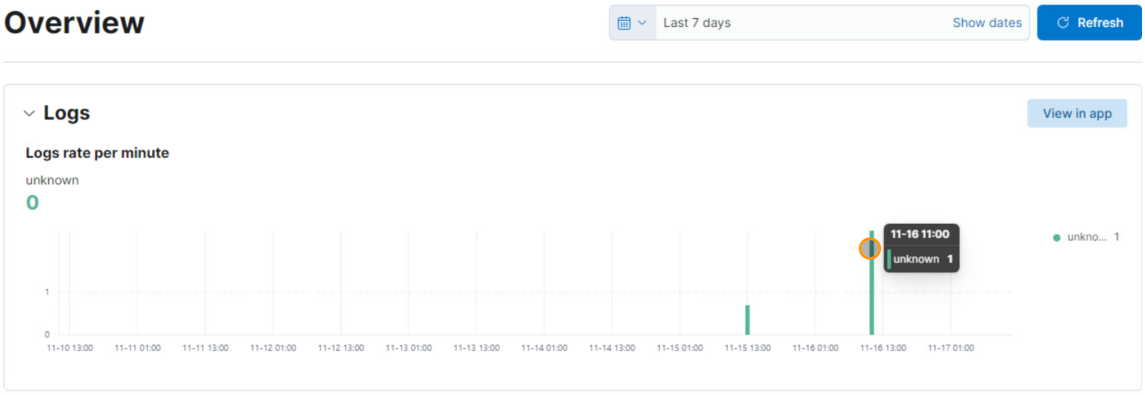


Snort logs in ELK STACK:

After installing Filebeat on snort the data and alerts are sent straight to logstash



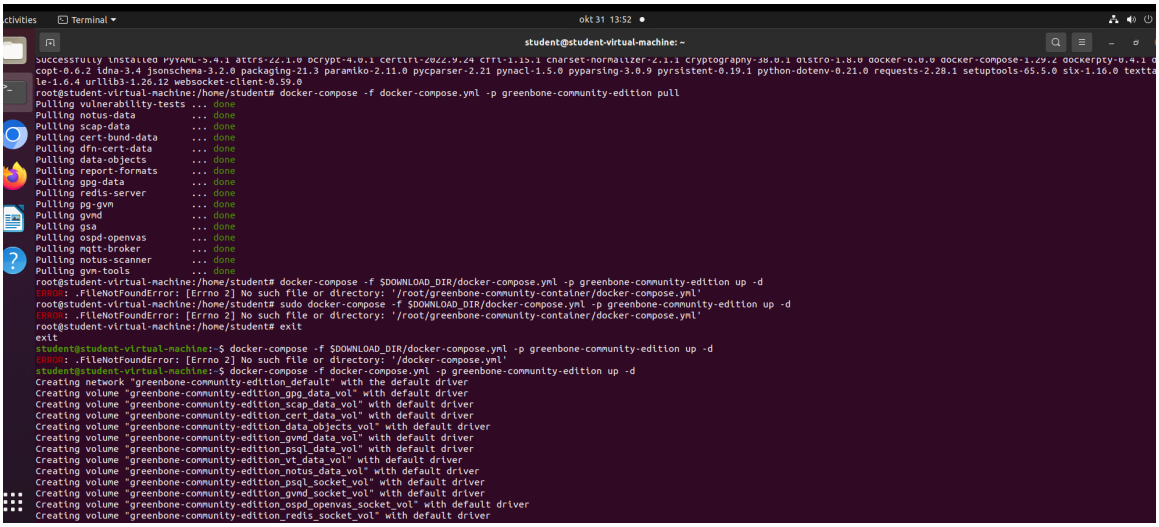
Here shows when Snort has an alert it sends straight with the logs to logstash



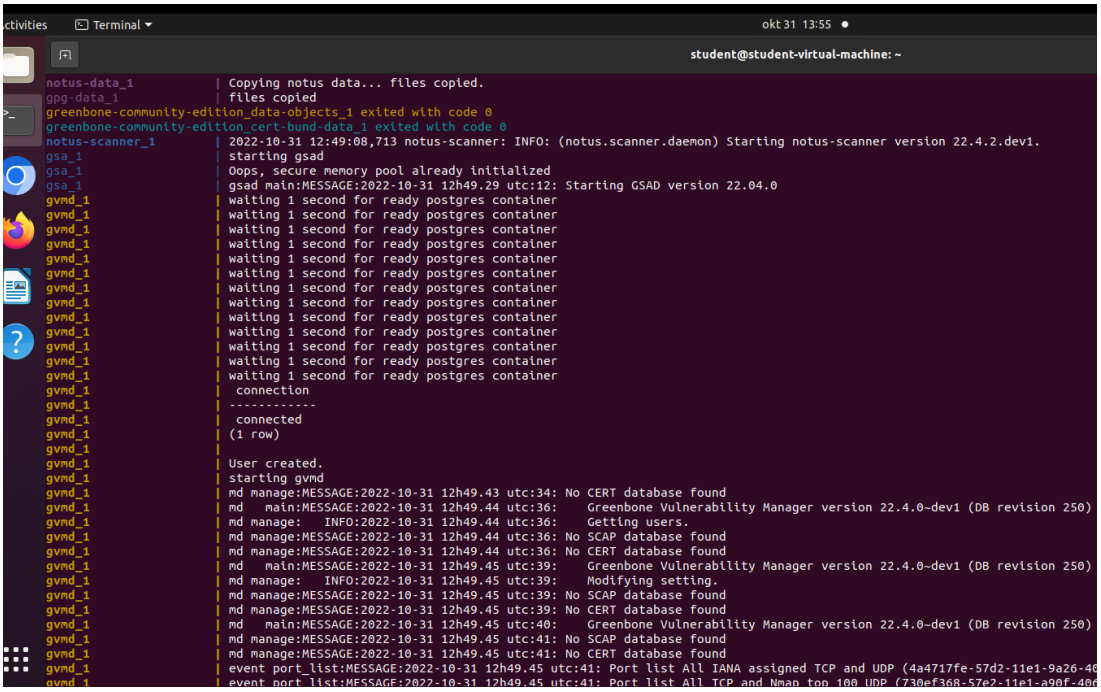
OpenVas

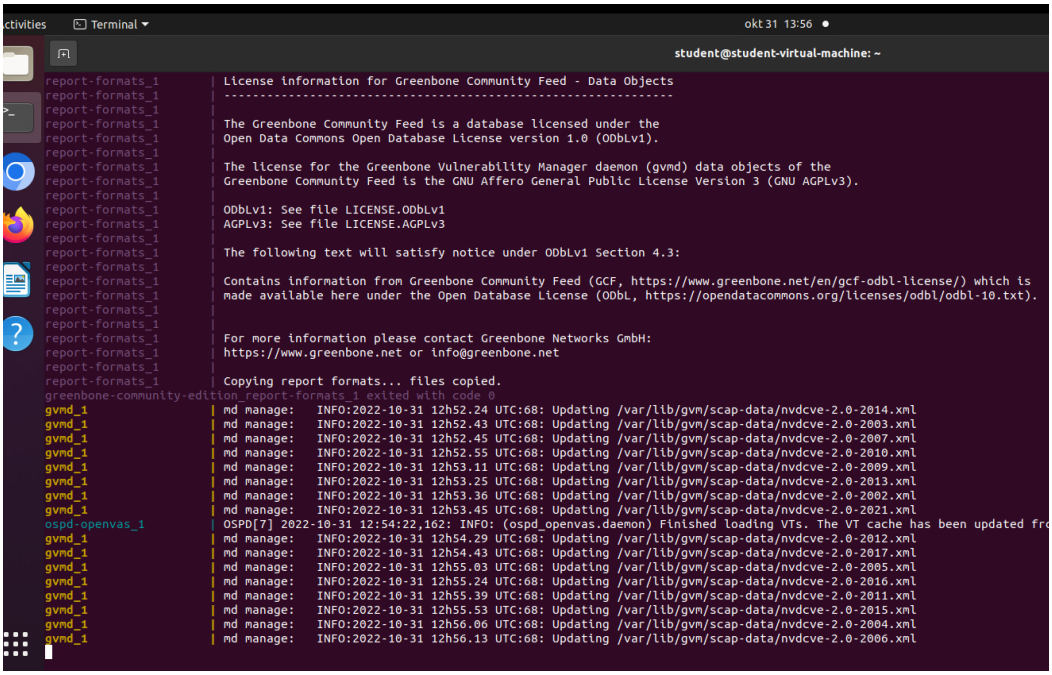
Installing openvas in docker compose with the help of this link (https://greenbone.github.io/docs/latest/22.4/container/index.html)

The reason why I chose the docker version instead of the manual install/guide is Because this saves massive time not only that the OPENVAS is pre-configured which reduces human errors. plus it is tested and verified by other people to be a superior method due to the redundancy of Docker

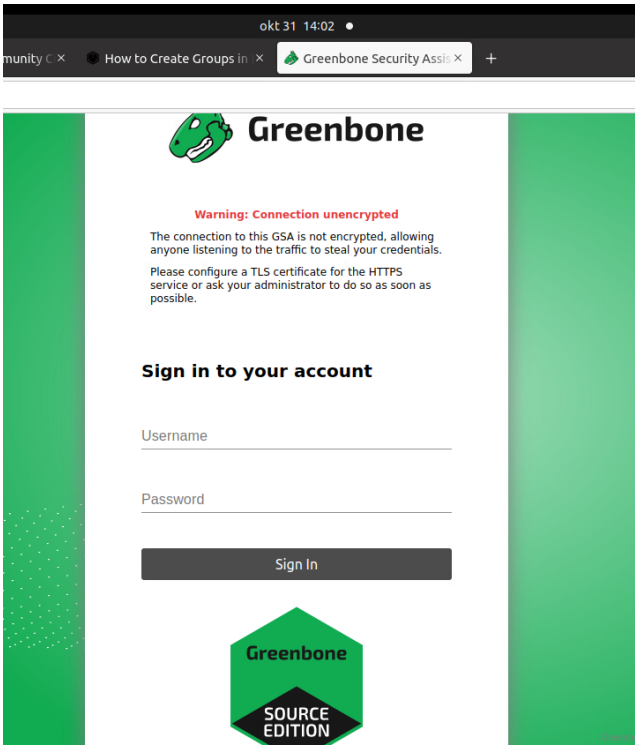

Activities
Terminal
okt 31 13:52
student@student-virtual-machine: ~
Successfully installed PyYAML-5.4.1 attrs-22.1.0 bcrypt-4.0.1 certifi-2022.9.24 cffi-1.15.1 charset-normalizer-2.1.1 cryptography-38.0.1 distro-1.8.0 docker-6.0.0 docker-compose-1.29.2 dockerpty-0.4.1 d
ocopt-0.6.2 idna-3.4 jsonschema-3.2.0 packaging-21.3 paramiko-2.11.0 pycparser-2.21 pynacl-1.5.0 pyparsing-3.0.9 pyrsistent-0.19.1 python-dotenv-0.21.0 requests-2.28.1 setuptools-65.5.0 six-1.16.0 textta
ble-1.6.4 urllib3-1.26.12 websocket-client-0.59.0
root@student-virtual-machine:/home/student# docker-compose -f docker-compose.yml -p greenbone-community-edition pull
Pulling vulnerability-tests ... done
Pulling notus-data ... done
Pulling scap-data ... done
Pulling cert-bund-data ... done
Pulling dfn-cert-data ... done
Pulling data-objects ... done
Pulling report-formats ... done
Pulling gpg-data ... done
Pulling redis-server ... done
Pulling pg-gvm ... done
Pulling gvmd ... done
Pulling gsa ... done
Pulling ospd-openvas ... done
Pulling mqtt-broker ... done
Pulling notus-scanner ... done
Pulling gvm-tools ... done
root@student-virtual-machine:/home/student# docker-compose -f $DOWNLOAD_DIR/docker-compose.yml -p greenbone-community-edition up -d
ERROR: .FileNotFoundError: [Errno 2] No such file or directory: '/root/greenbone-community-container/docker-compose.yml'
root@student-virtual-machine:/home/student# sudo docker-compose -f $DOWNLOAD_DIR/docker-compose.yml -p greenbone-community-edition up -d
ERROR: .FileNotFoundError: [Errno 2] No such file or directory: '/root/greenbone-community-contalner/docker-compose.yml'
root@student-virtual-machine:/home/student# exit
exit
student@student-virtual-machine:~$ docker-compose -f $DOWNLOAD_DIR/docker-compose.yml -p greenbone-community-edition up -d
ERROR: .FileNotFoundError: [Errno 2] No such file or directory: '/docker-compose.yml'
student@student-virtual-machine:~$ docker-compose -f docker-compose.yml -p greenbone-community-edition up -d
Creating network "greenbone-community-edition_default" with the default driver
Creating volume "greenbone-community-edition_gpg_data_vol" with default driver
Creating volume "greenbone-community-edition_scap_data_vol" with default driver
Creating volume "greenbone-community-edition_cert_data_vol" with default driver
Creating volume "greenbone-community-edition_data_objects_vol" with default driver
Creating volume "greenbone-community-edition_gvmd_data_vol" with default driver
Creating volume "greenbone-community-edition_psql_data_vol" with default driver
Creating volume "greenbone-community-edition_vt_data_vol" with default driver
Creating volume "greenbone-community-edition_notus_data_vol" with default driver
Creating volume "greenbone-community-edition_psql_socket_vol" with default driver
Creating volume "greenbone-community-edition_gvmd_socket_vol" with default driver
Creating volume "greenbone-community-edition_ospd_openvas_socket_vol" with default driver
Creating volume "greenbone-community-edition_redis_socket_vol" with default driver

These two images show that the docker is composing with Docker Compose

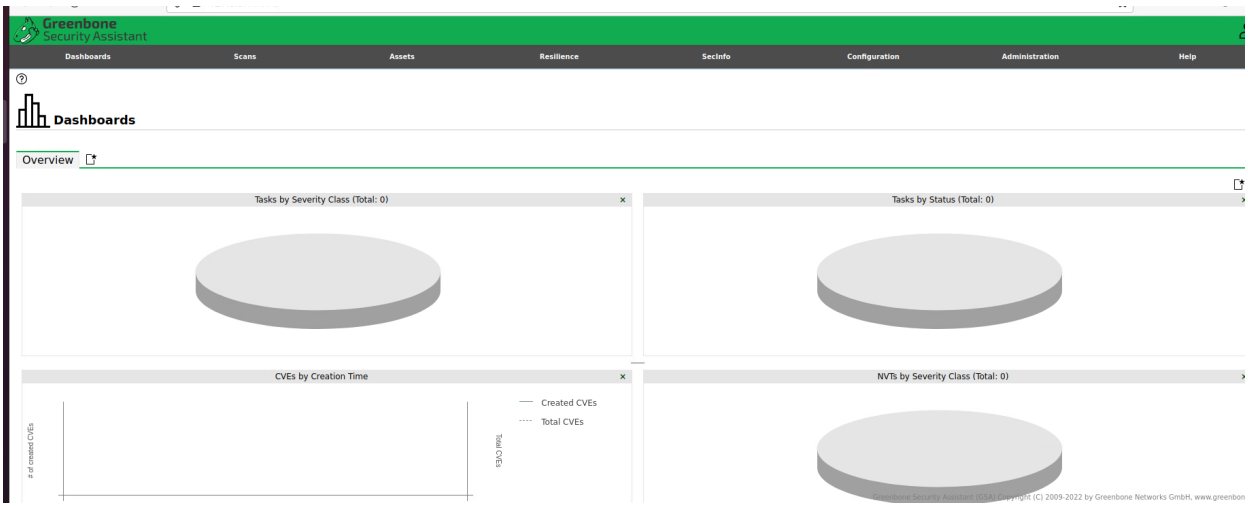




After successful install i can finally create a account to OPENVAS

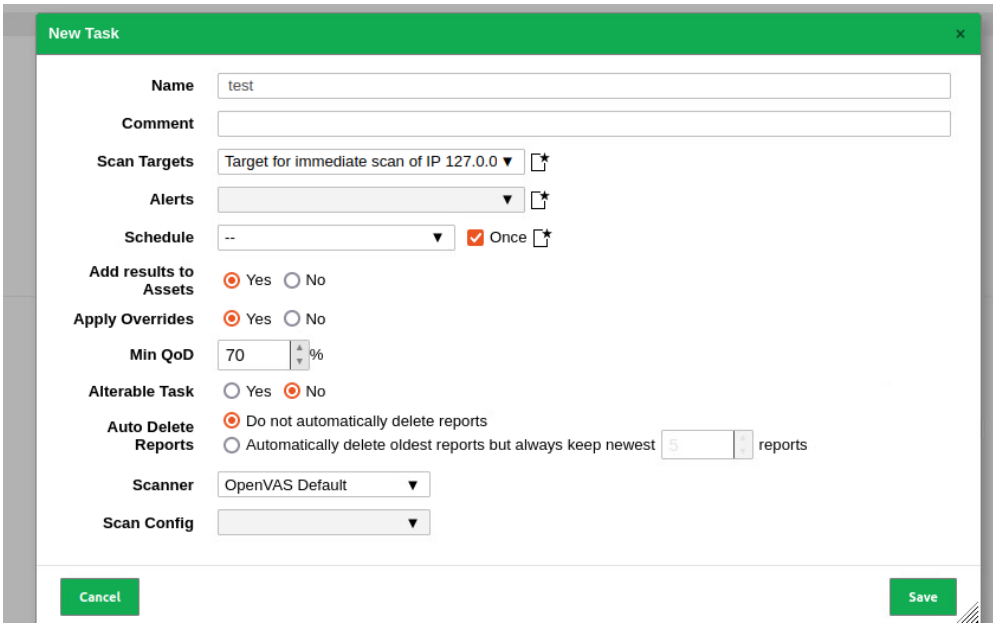


Now we need to create a task to scan our enviorement.

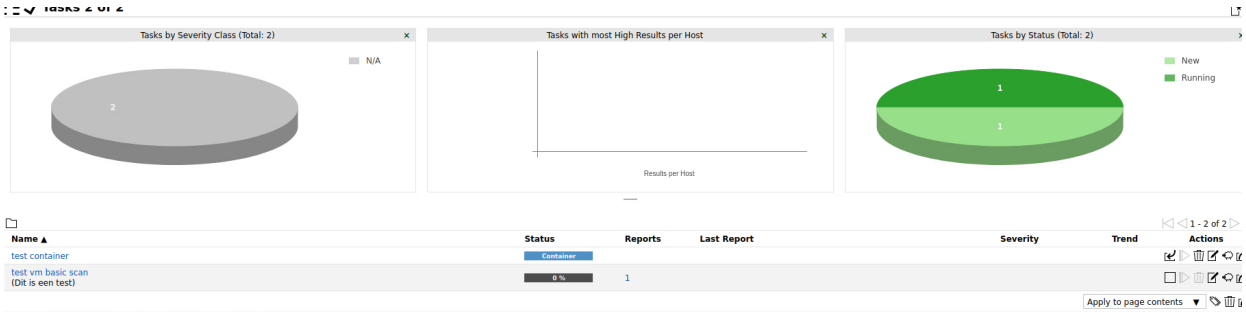


Openvas test scan

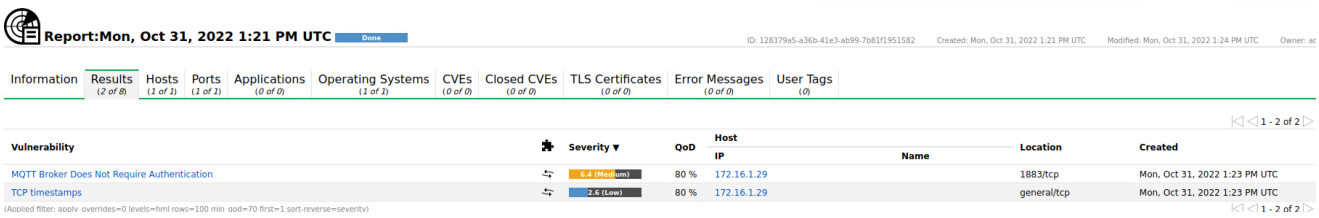
Here i'm trying to create a "test" scan to test if there are any vulnerabilities



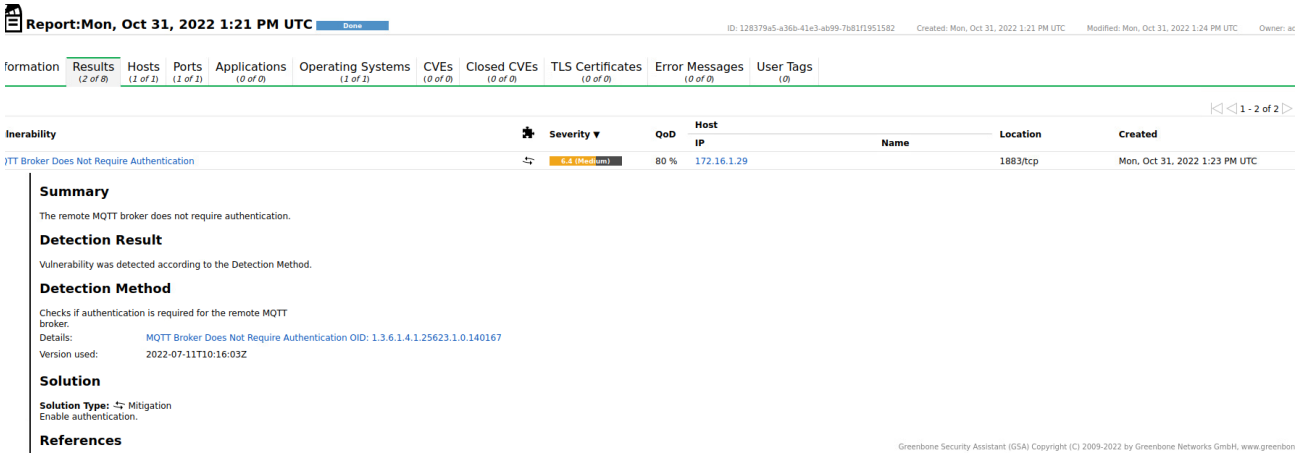
After creating the task the scan can begin. I chose the basic version just to show that it works. Once it's done you can run the full scan but right now the vcenter from FHICT Int is a bit unstable and my ELK STACK eats 9ghz of resources. Maybe another time.



The report is done and it looks like he found something



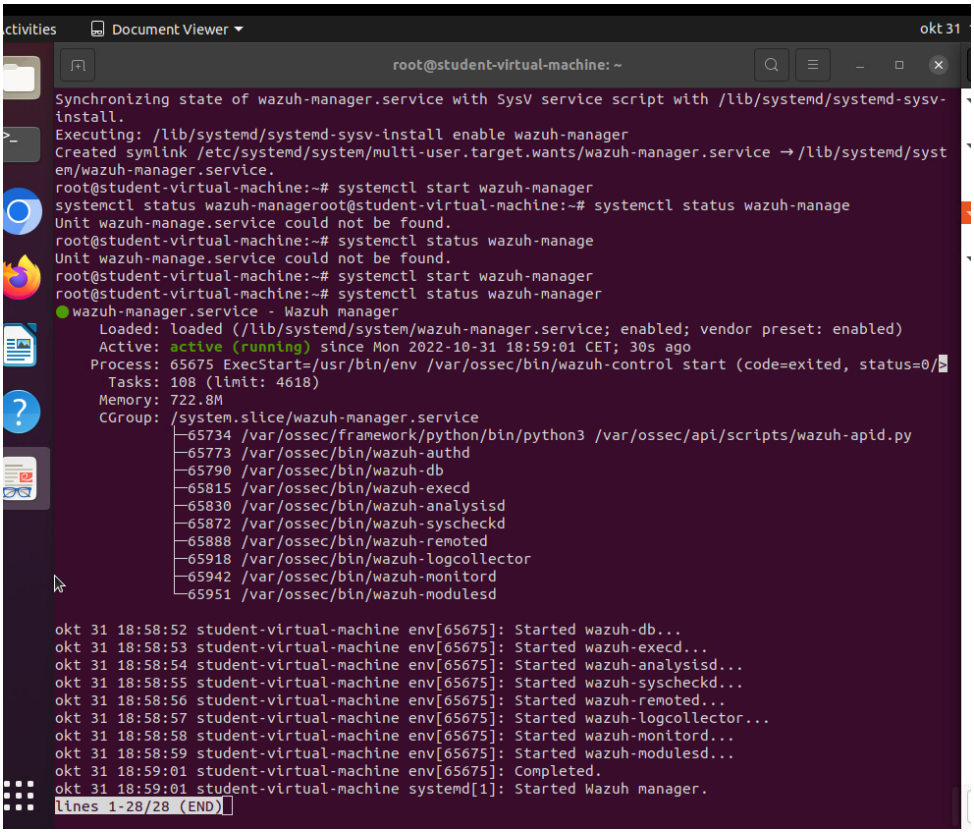
Apparently my MQTT is not configured right. Which is a medium issue



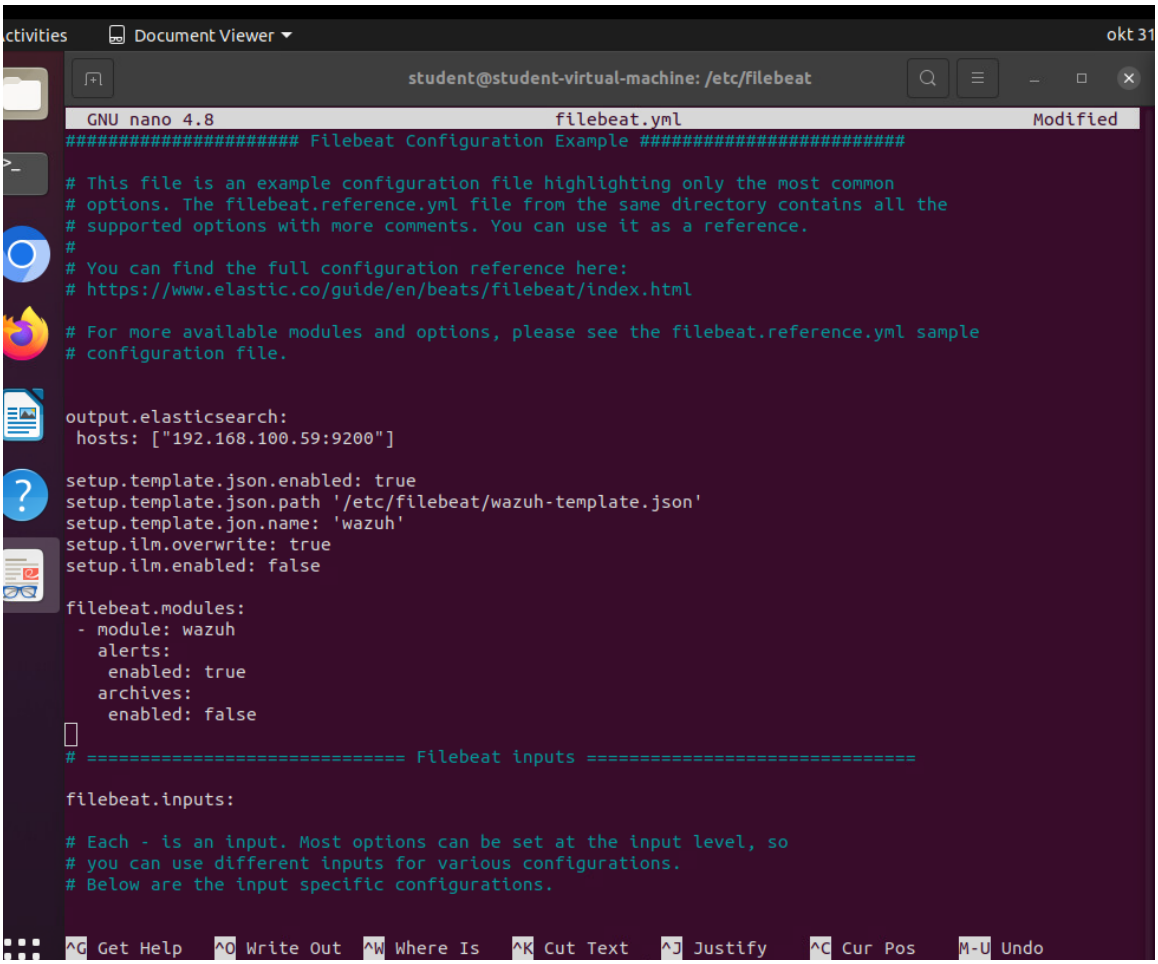
WAZUH manager

To install Wazuh manager i chose the official guide to install with assistance of Wazuh script found right here (https://documentation.wazuh.com/current/installation-guide/index.html). I recommend using the official Wazuh assisted installer which is better than the guide and also reduces the human errors.

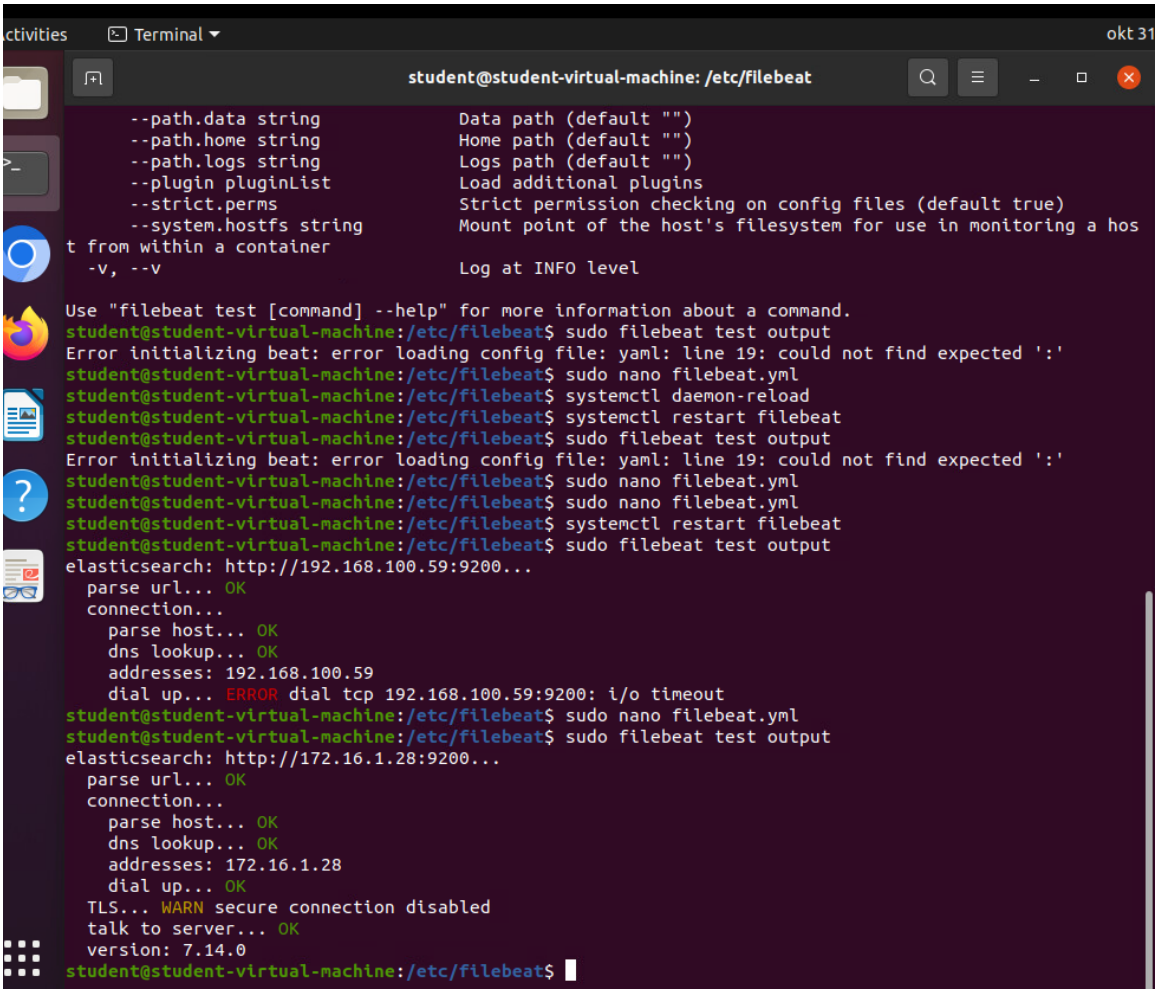
After installing Wazuh and Wazuh services and starting them to systemctl.



Ofcourse you need to install Filebeat and configure it for my wazuh manager to Elk stack

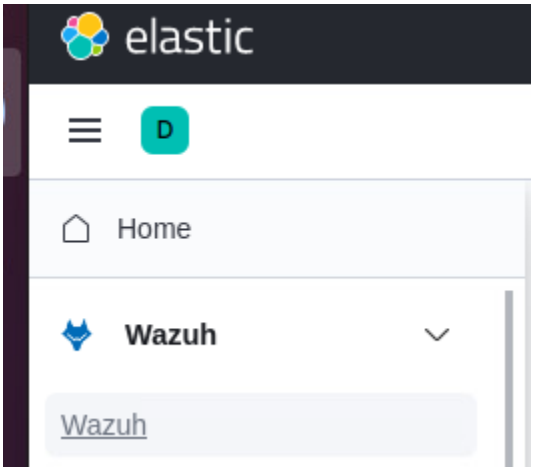


Filebeat is up and running (i accidently used 172.16.100:59:9200 but is must have been 172.16.1.28:9200 but it is fixed now)

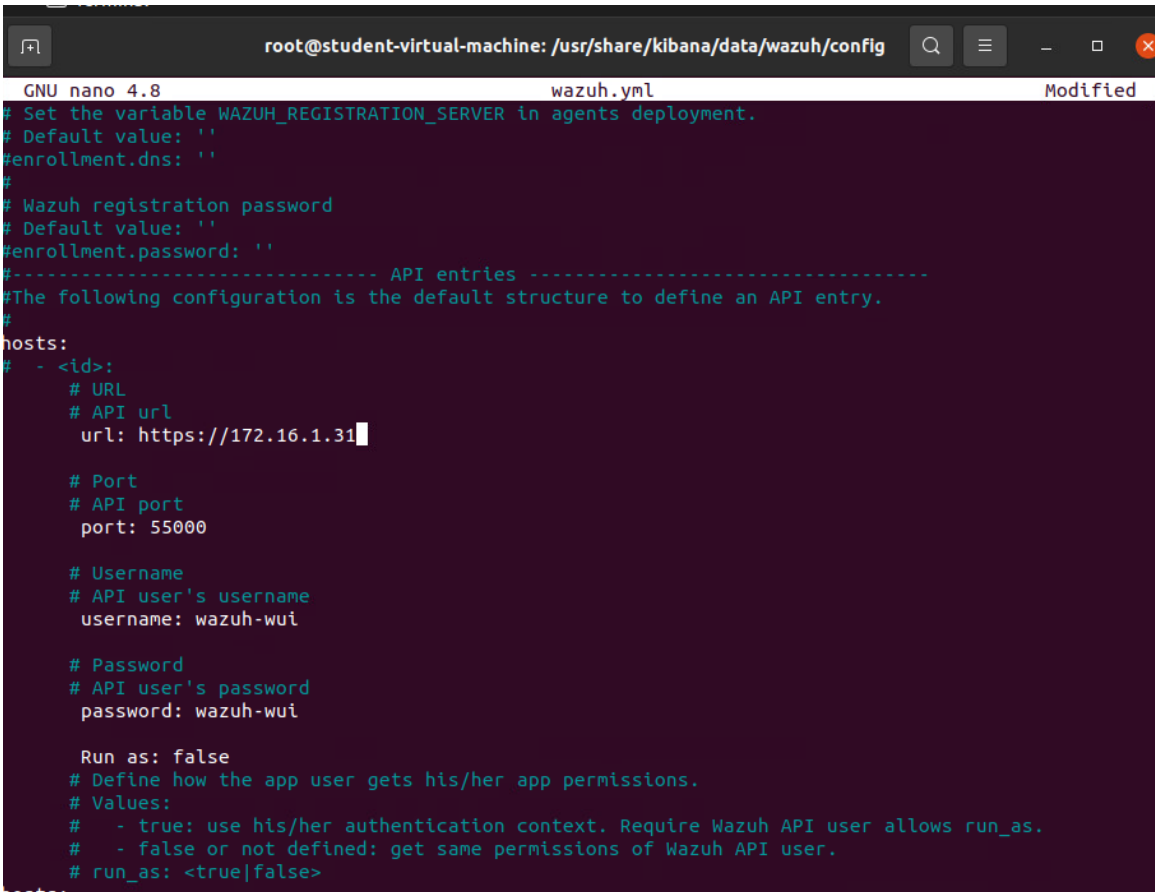


After running it couple of times the Wazuh has been added to kabana

Wazuh has been registered by ELK STACK!

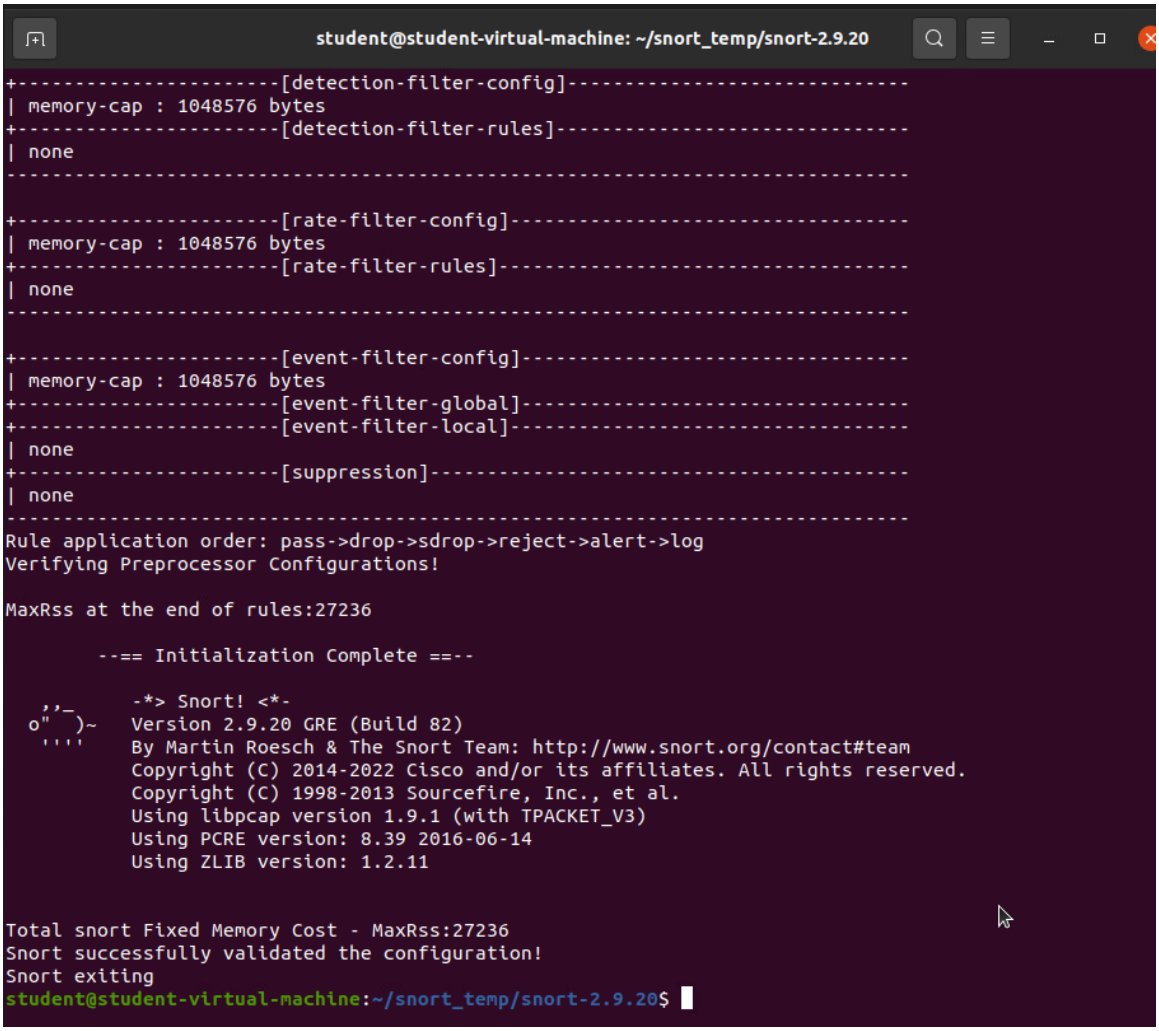


But wazuh is not working yet. We need to setup a api credentials before data comes



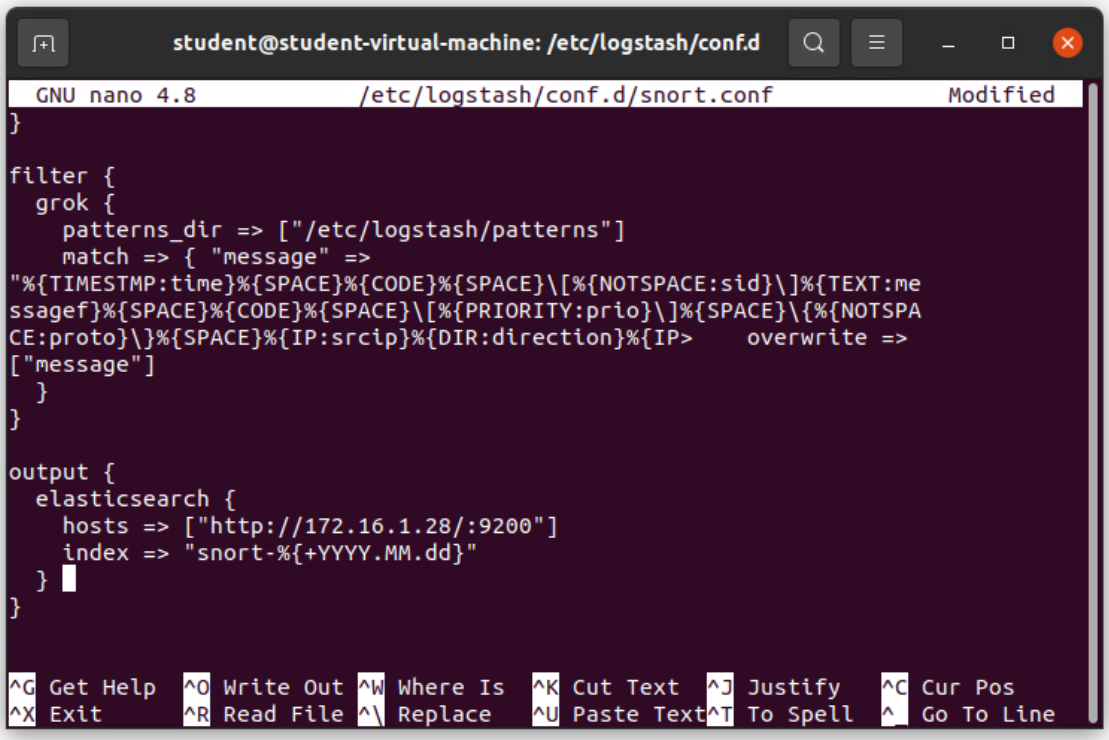
Snort

After following the guide i've installed my Snort and added my IONK code to snort



```
~
lines 1-10/10 (END)...skipping...
● snort.service - Snort
     Loaded: loaded (/lib/systemd/system/snort.service; disabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-11-02 16:10:40 CET; 50s ago
   Main PID: 56893 (snort)
      Tasks: 2 (limit: 4618)
     Memory: 23.0M
     CGroup: /system.slice/snort.service
             └─56893 /usr/local/bin/snort -A fast -c /etc/snort/snort.conf -q -i ens160

nov 02 16:10:40 student-virtual-machine systemd[1]: Started Snort.
~
```

In elk stack

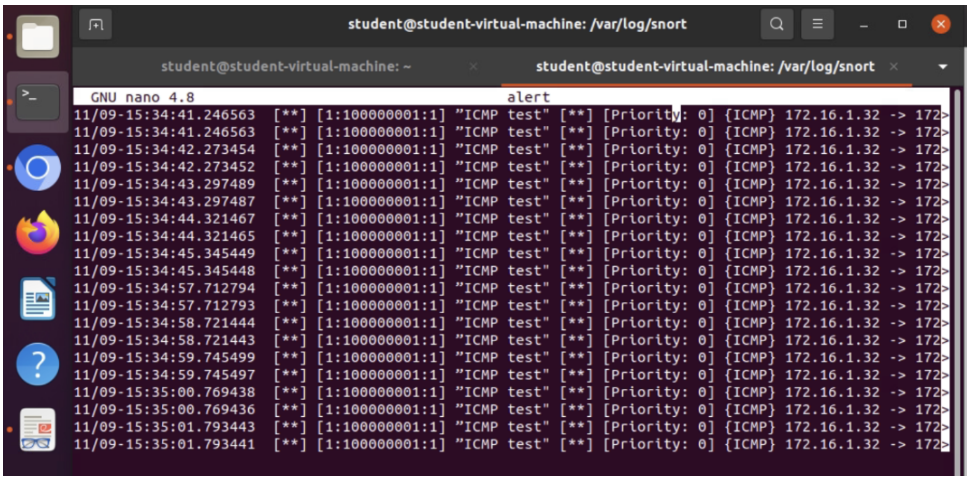


```
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
     Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
     Active: failed (Result: exit-code) since Wed 2022-11-02 16:31:09 CET; 5s ago
       Docs: https://www.elastic.co/beats/filebeat
    Process: 58204 ExecStart=/usr/share/filebeat/bin/filebeat --environment systemd $BEAT_LOG_OPTS $BEAT>
   Main PID: 58204 (code=exited, status=1/FAILURE)

nov 02 16:31:09 student-virtual-machine systemd[1]: filebeat.service: Failed with result 'exit-code'.
nov 02 16:31:09 student-virtual-machine systemd[1]: filebeat.service: Scheduled restart job, restart cou>
nov 02 16:31:09 student-virtual-machine systemd[1]: Stopped Filebeat sends log files to Logstash or dire>
nov 02 16:31:09 student-virtual-machine systemd[1]: filebeat.service: Start request repeated too quickly.
nov 02 16:31:09 student-virtual-machine systemd[1]: filebeat.service: Failed with result 'exit-code'.
nov 02 16:31:09 student-virtual-machine systemd[1]: Failed to start Filebeat sends log files to Logstash>
~
```

After fixing my mistype the logstash is up and running

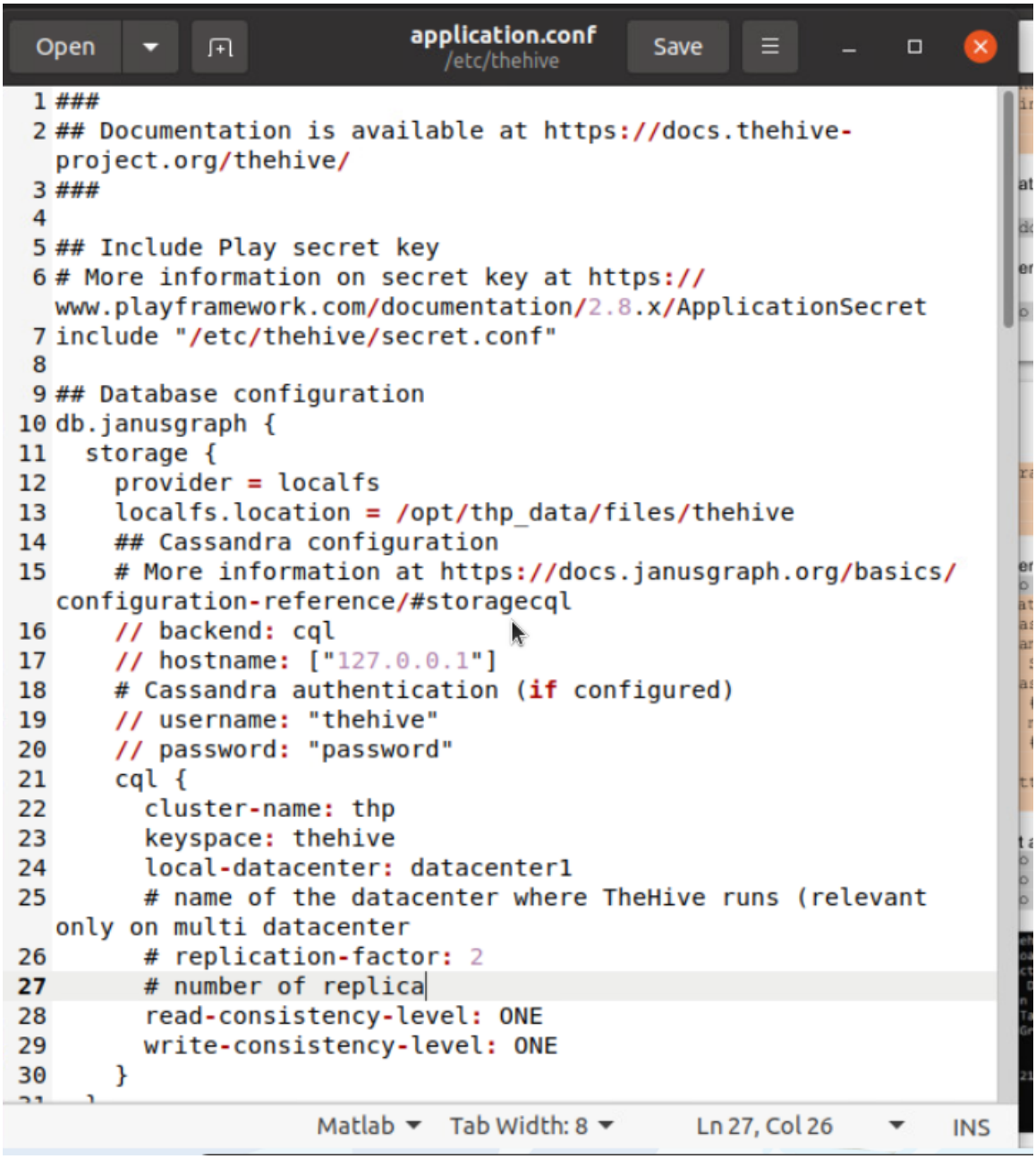
```
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-11-02 16:40:10 CET; 23s ago
   Main PID: 537358 (java)
      Tasks: 15 (limit: 4618)
     Memory: 485.0M
     CGroup: /system.slice/logstash.service
             └─537358 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=>

nov 02 16:40:10 student-virtual-machine systemd[1]: logstash.service: Failed with result 'exit-code'.
nov 02 16:40:10 student-virtual-machine systemd[1]: Stopped logstash.
nov 02 16:40:10 student-virtual-machine systemd[1]: Started logstash.
nov 02 16:40:10 student-virtual-machine logstash[537358]: Using bundled JDK: /usr/share/logstash/jdk
nov 02 16:40:11 student-virtual-machine logstash[537358]: OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprec>
nov 02 16:40:20 student-virtual-machine logstash[537358]: /usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/bundler-1.17.3/lib/>
~
~
```

The logs of my snort when it was recording



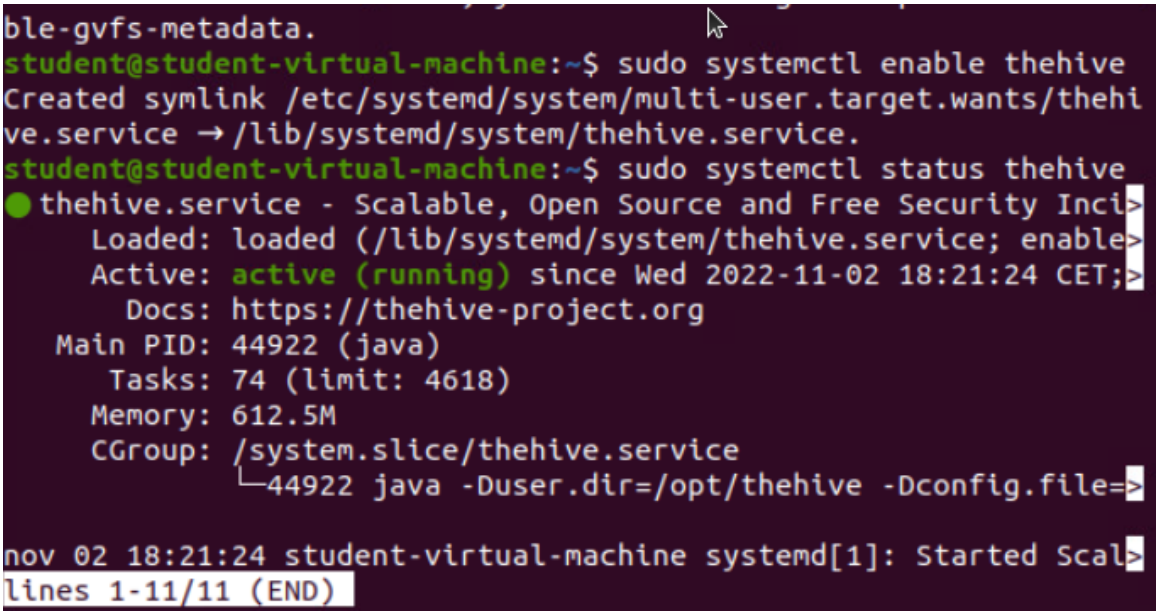
The Hive and Cortex

Application.conf to make this running I needed to set up the application to make HIVE running and then add Cortex. This time I used the guide from FHICT.



After configuring the application.conf, now to start my hive in systemctl

The Hive did not crash!



After installing Cortex it was i needed to use the cortex api and assign it to the Hive

Cortex is up and running!

```
Creating system group: cortex
Creating system user: cortex in cortex with cortex daemon-user and shell /bin/false
Processing triggers for systemd (245.4-4ubuntu3.18) ...
student@student-virtual-machine:~$ sudo nano /etc/cortex/application.conf
student@student-virtual-machine:~$ sudo systemctl start cortex
student@student-virtual-machine:~$ sudo systemctl enable cortex
Synchronizing state of cortex.service with SysV service script with /lib/systemd/systemd-sysv-instal
.
Executing: /lib/systemd/systemd-sysv-install enable cortex
Created symlink /etc/systemd/system/multi-user.target.wants/cortex.service → /etc/systemd/system/cor
ex.service.
student@student-virtual-machine:~$ sudo systemctl status cortex
● cortex.service - cortex
     Loaded: loaded (/etc/systemd/system/cortex.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-11-02 18:49:28 CET; 15s ago
       Docs: https://thehive-project.org
   Main PID: 48359 (java)
      Tasks: 36 (limit: 4618)
     Memory: 504.6M
     CGroup: /system.slice/cortex.service
             └─48359 java -Duser.dir=/opt/cortex -Dconfig.file=/etc/cortex/application.conf -Dlogger

nov 02 18:49:28 student-virtual-machine systemd[1]: Started cortex.
lines 1-11/11 (END)
```

1.18.1.4. Problems

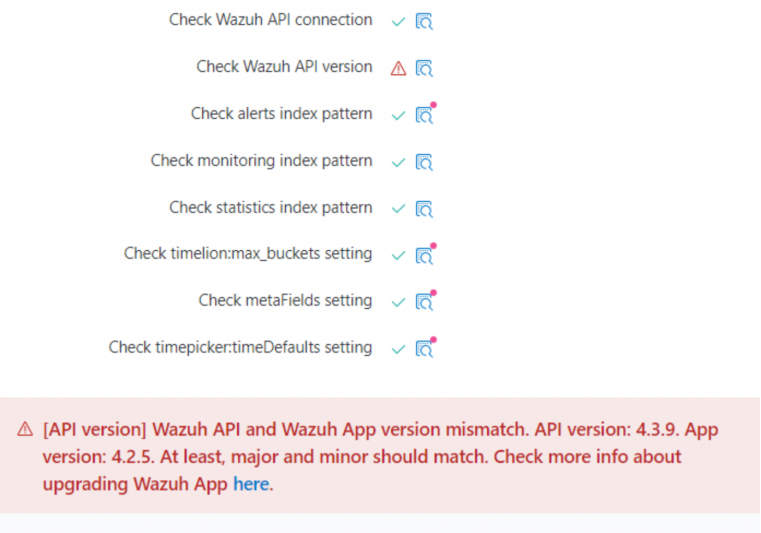
Right now my logstash does not accept anything through port 5044. This is important because my Snort's filebeat needs to be able to send to my logstash in ELK Stack. Not only that but my Wazuh Manager sends warnings from the user agent that a user tries to log in but it fails. That notification must also be sent from ELK stack and HIVE/Cortex.

```
student@student-virtual-machine:/etc/filebeat$ curl http://172.16.1.28:5044
curl: (7) Failed to connect to 172.16.1.28 port 5044: Connection refused
student@student-virtual-machine:/etc/filebeat$ sudo nmap 172.16.1.28 -p 5044
Starting Nmap 7.80 ( https://nmap.org ) at 2022-12-02 13:04 CET
Nmap scan report for 172.16.1.28
Host is up (0.00051s latency).

PORT     STATE  SERVICE
5044/tcp closed lxi-evntsvc
MAC Address: 00:50:56:97:39:BF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

After consulting with Peter we could not find any issues in the logs that implies misconfiguration or is wrong.

There is another issue which does not cause any issues but more of annoyance and that is that Wazuh API and Wazuh manager are not the same version which everytime i log into ELK elk

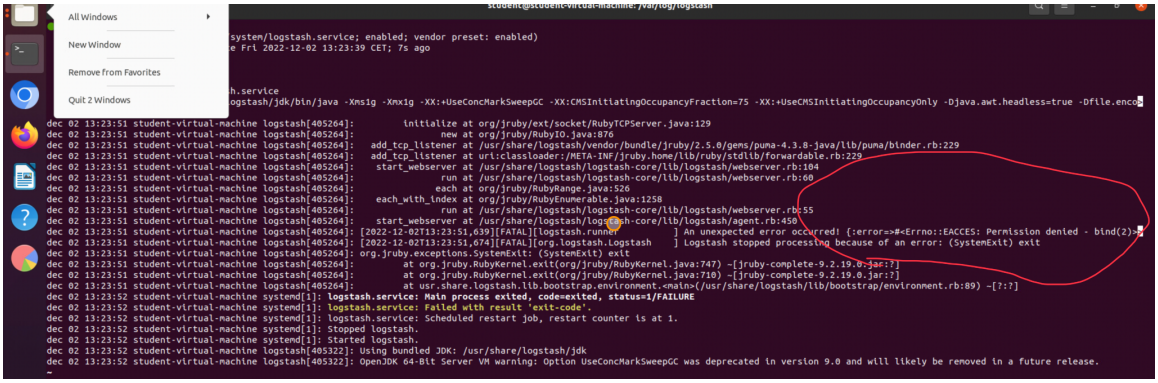
elkstack > Wazuh tab then i get a error but you can just continue fine without any issues.



You can just bypass this by pressing the Wazuh home button and you get to see the dashboard.

1.18.1.5. Troubleshooting

After asking my classmates and teachers we could not make it to work. Even comparing our configs and scripts. To no avail, it did not work as I was hoping. So I went for the last time to Peter and tried to make my SNORT to communicate with my ELK stack. With the help of NMAP we could try and ping the port for any response but the port was closed and wouldn't open for some reason. So digging further i saw a stackoverflow post about this issue and apparently java does not give the permission to that port. This is a bug found in february and got notified in github. The bug is that "eacces permission denied"



To fix the issue they said you need to configure the config/jvm options file and add this command " ***-Djdk.io.File.enableADS=true*** ". I applied it and did not change anything at all. There was also another solution with binding java to netwerk service.

Peter told me to try again but this time with CLI or non gui which does not make any difference in my opinion. The only differice it makes is that one uses alot of resources and the other uses less. But the core of the distro is still the same at the end of the day it is Ubuntu 20.04 LTS.

My guess is that I had bad luck installing it. Maybe it got corrupted on the way. The other applications are running fine and doing their jobs fine but connecting it to eachother was a nightmare even with help of my classmates/teachers we couldn't solve the issue.

1.18.1.6. Reflection

I've learned that backups are VERY important because I broke most of my vms bij not taking any snapshot. I've learned from my mistake and made snapshots because I don't want to make the same mistake, but besides that the ELK stack was fun building (if it worked) and seeing the data from snort and wazuh flowing through. OpenVAS was also cool to see that it can see vulnerabilities. I think that openvas can be useful later in our group project.