Ministry of Communications and Information Technology

Digital Egypt Pioneers

Cybersecurity Incident Response Analyst

SIEM Configuration and Monitoring Report

Team Members:

Esraa Matarawy Abdelmoniem
Nada Saleh Mohamad
Omar Abdelrahman Ahmed
Kholoud Khaled Mohamed
Fahd Mahmoud Abdelkhalek

Supervised by:

Eng.Nour Eldeen Essam

Table of Contents

Table of Figures

Network Overview

A basic network comprises three hosts operating on the 192.168.188.0/24 subnet, along with a network firewall.

1. **Three PCs**:
   - PC-1(192.168.188.156/24): a Linux OS which hosts the SIEM solution and its different components like Fleet server, the Logstash server as well as treated as a normal PC.
   - PC-2 (192.168.188.157/24): Windows OS machine, a sample of assets that need to be monitored and defended against attacks.
   - PC-3 (192.168.188.158/24): Windows OS machine, a sample of assets that need to be monitored and defended against attacks.
2. **Firewall**:
   - Forti-Firewall to route and monitor network traffic generated by hosts in the network.

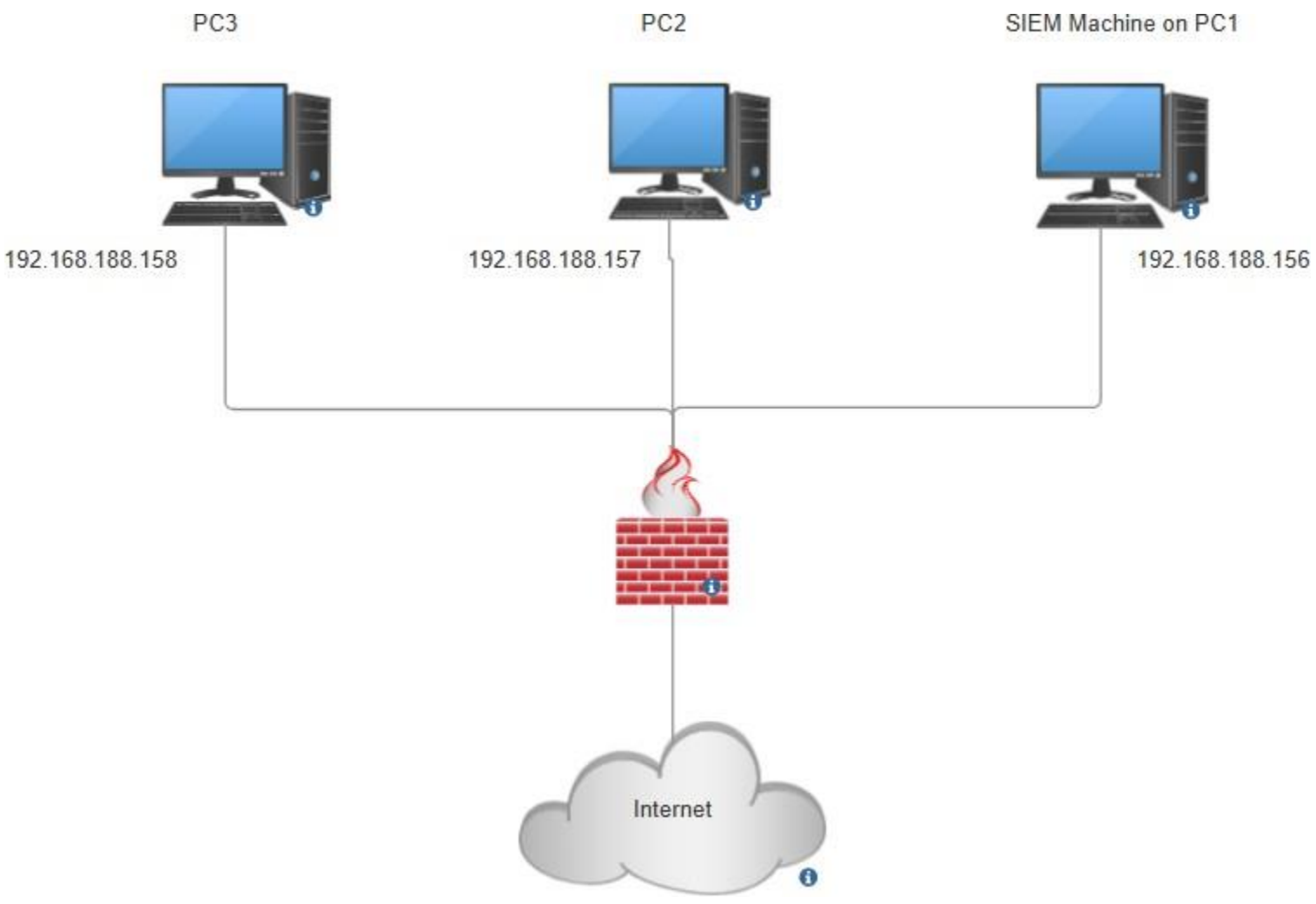

Figure 1:Netwok Overview

In configuring the firewall, I set up Port 1 to function as the WAN interface and Port 2 as the LAN interface, establishing Port 2 as the gateway for devices within the network. To enhance security, I created a straightforward profile designed to block

access to the well-known website, https://www.wicar.org/test-malware.html, effectively preventing the download of malicious files. The alerts and logs generated from this configuration will be presented in the report, accompanied by corresponding screenshots for visual reference.

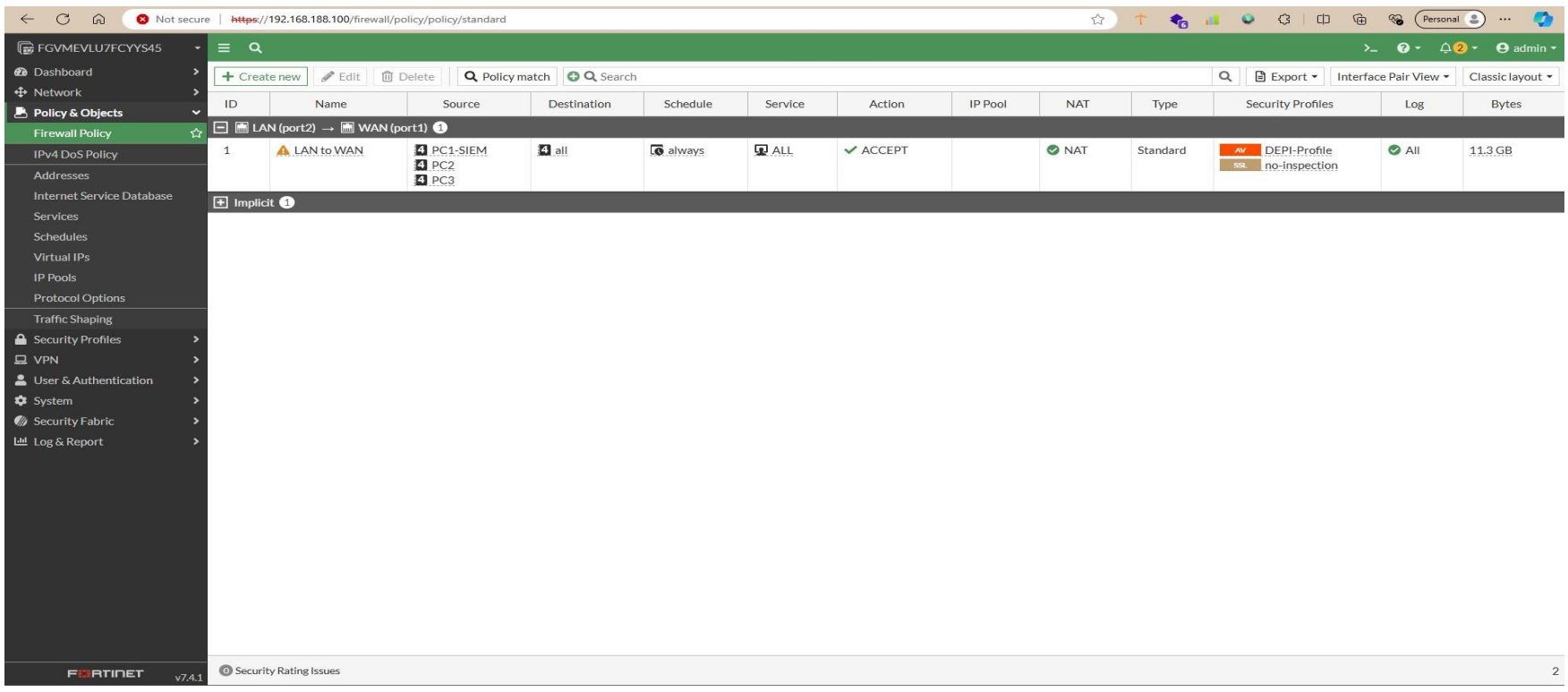

Figure 3: Simple Firewall Policy

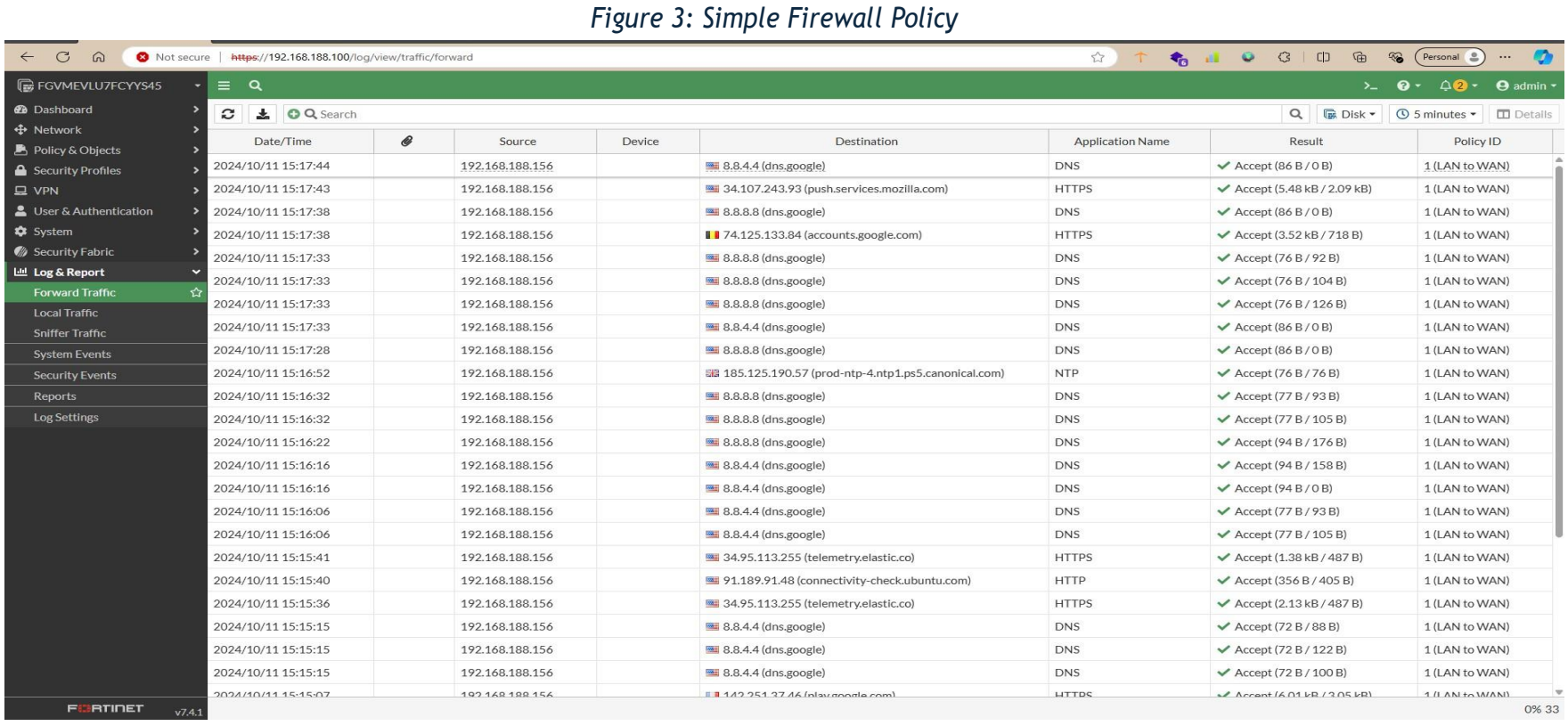

Figure 2: Firewall Logs

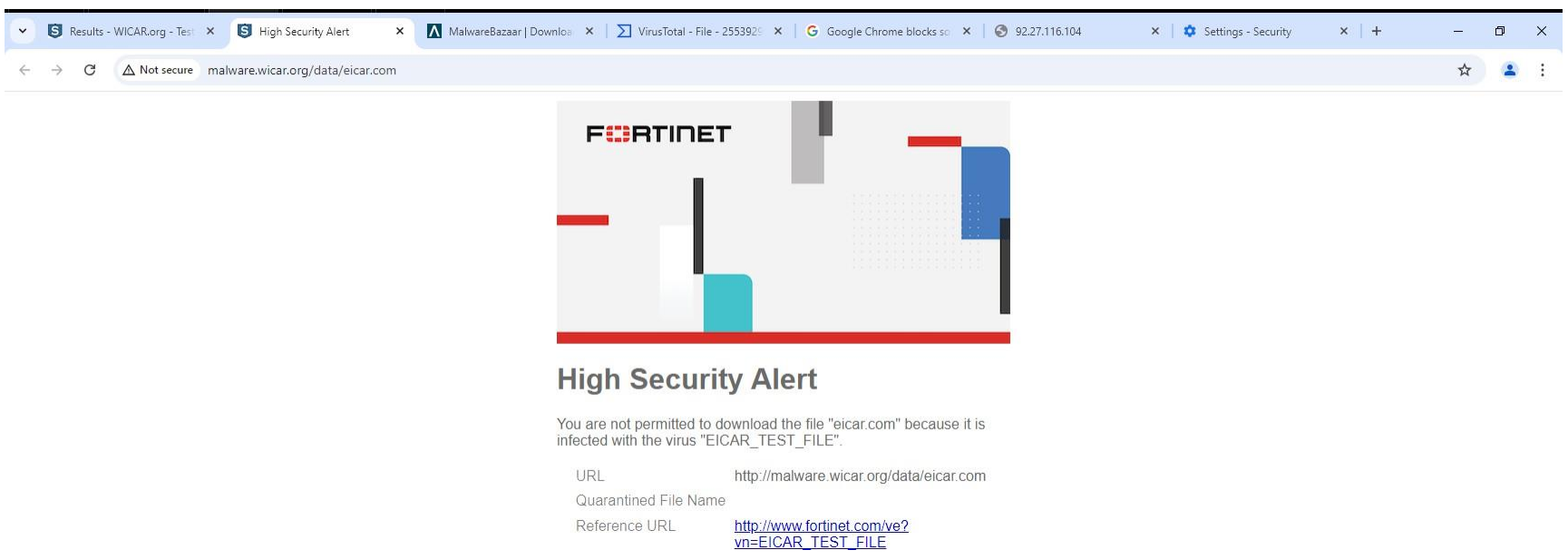

Figure 4: Malware Prevention

Intro To Elastic Security SIEM

Elastic Security SIEM (Security Information and Event Management) is a product built on top of the Elastic Stack, which provides security insights and real-time threat detection. As a modern SIEM solution, it collects, normalizes, and analyzes data from various sources within an organization's IT environment, such as logs, network traffic, and endpoint data.

The primary function of Elastic Security SIEM is to offer a centralized platform for monitoring and managing security events. It enhances an organization's ability to detect unusual or potentially malicious activity quickly. Elastic SIEM provides advanced correlation techniques and machine learning algorithms that assess risk levels, spot anomalies, and prioritize alerts based on their potential security impact.

Technically, Elastic SIEM uses a different component to perform its job correctly, These components are as follows:

- Elasticsearch: **The heart** of Elastic Stack, Elasticsearch is a distributed, RESTful search and analytics engine, scalable data store, and vector database capable of addressing a growing number of use cases. As the heart of the Elastic Stack, it centrally stores your data for lightning-fast search.
- Kibana: Kibana is a **user interface** that lets you visualize your Elasticsearch data and navigate the Elastic Stack.
- **Integrations:** Like Elastic Agent which is a single, unified way to add monitoring for logs, metrics, and other types of data to a host.
  Logstash, which is a server-side **data processing pipeline** that ingests data from a multitude of sources, transforms it, and then sends it to your favorite "stash."
  Beats **data shippers** that you install as agents on your servers to send operational data to Elasticsearch.

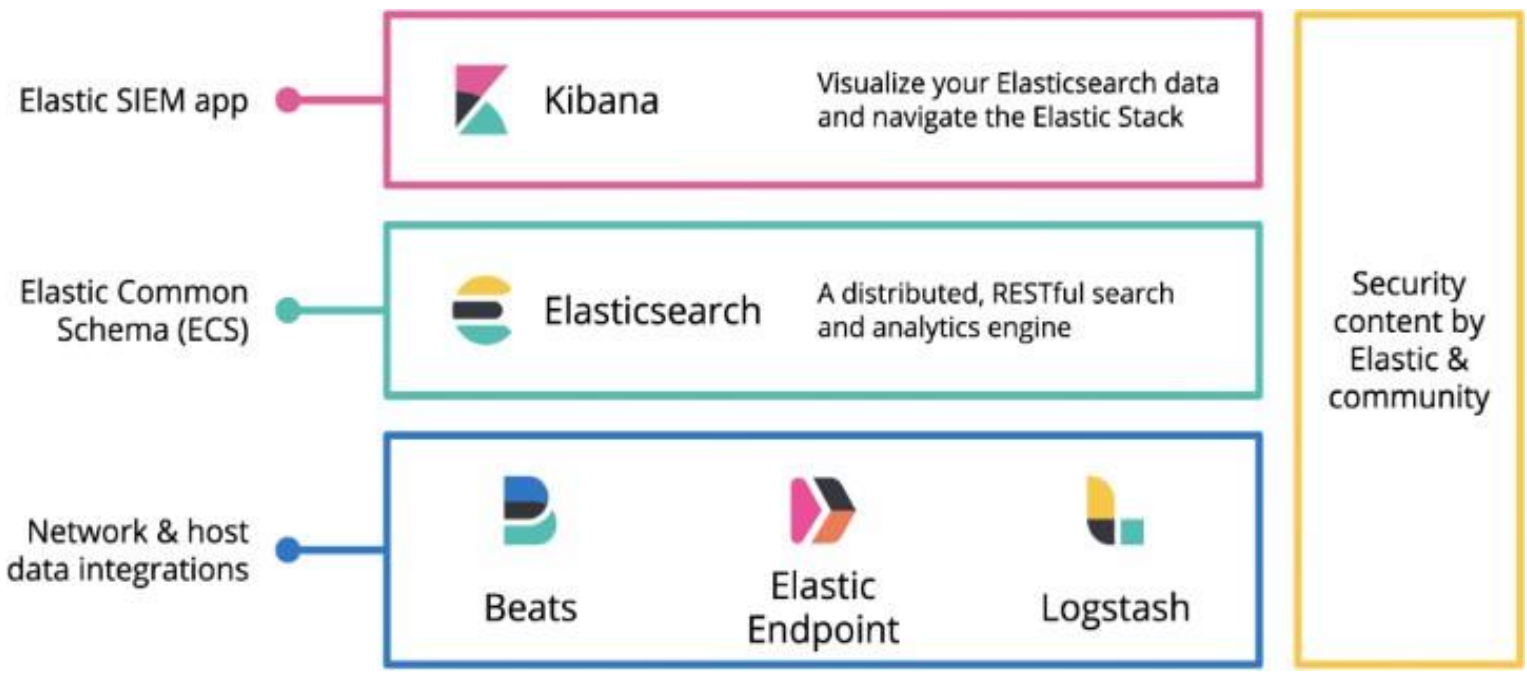

Figure 5: Elastic Stack

Install and Set Up Elastic Stack

Installing and setup elastic stack step by step and configuring our network to ensure that all logs and alerts are received by SIEM, generally Elastic Stack components are distributed in different servers, in our network we will install and setup all main components of Elastic Stack in the same Ubuntu machine (PC-1) with IP address 192.168.188.156, this is because the simplicity of our network is.

In this section we will setup and configure each component of Elastic Stack to run our SIEM.

Install and Set Up Elasticsearch

We should first import the Elasticsearch PGP Key using this command:

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg

```
root@nada-VMware-Virtual-Platform:/home/nada/Desktop# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elast
icsearch-keyring.gpg
root@nada-VMware-Virtual-Platform:/home/nada/Desktop# apt-get update
```

Figure c: Import PGP Key

Then install apt-transport-http package before installation and save the repo definition to **/etc/apt/sources.list.d/elastic-8.x.list** and Update.

```
root@nada-VMware-Virtual-Platform: /home/nada/Desktop
** (wireshark:5063) 12:47:20.774391 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
root@nada-VMware-Virtual-Platform:/home/nada/Desktop# sudo apt-get install apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 3,974 B of archives.
After this operation, 35.8 kB of additional disk space will be used.
Get:1 http://eg.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3,974 B]
Fetched 3,974 B in 0s (12.6 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 149585 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Setting up apt-transport-https (2.7.14build2) ...
root@nada-VMware-Virtual-Platform:/home/nada/Desktop# echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/a
pt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list
deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main
root@nada-VMware-Virtual-Platform:/home/nada/Desktop# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://eg.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://eg.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 https://artifacts.elastic.co/packages/8.x/apt stable InRelease [10.4 kB]
Err:4 https://artifacts.elastic.co/packages/8.x/apt stable InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY D27D666CD88E42B4
Hit:5 http://eg.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:6 http://eg.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [475 kB]
Get:7 http://eg.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [120 kB]
Get:8 http://eg.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8,156 B]
```

Figure 7: apt-transport-http package

Update and Install Elasticsearch packages we can see here the generated password for Elasticsearch and elastic as a Username.

```
root@nada-VMware-Virtual-Platform:/home/nada/Desktop# sudo apt-get install elasticsearch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 606 MB of archives.
After this operation, 1,168 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 elasticsearch amd64 8.15.1 [606 MB]
Fetched 606 MB in 6min 29s (1,558 kB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 149597 files and directories currently installed.)
Preparing to unpack .../elasticsearch_8.15.1_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (8.15.1) ...
Setting up elasticsearch (8.15.1) ...
--------------------------- Security autoconfiguration information ------------------------------

Authentication and authorization are enabled.
TLS for the transport and HTTP layers is enabled and configured.

The generated password for the elastic built-in superuser is : ISJ=YTQjycJ*HChwvDJp

If this node should join an existing cluster, you can reconfigure this with
'/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <token-here>'
```

Figure 8: Elasticsearch Installation

Checking the configuration file of Elasticsearch (elasticsearch.yml), Elasticsearch is using localhost and port 9200 by default, we change it the machine IP 192.168.188.156 to access it from different machine, and checking the network host to be 0.0.0.0 which means accepting connection coming from any IP address.

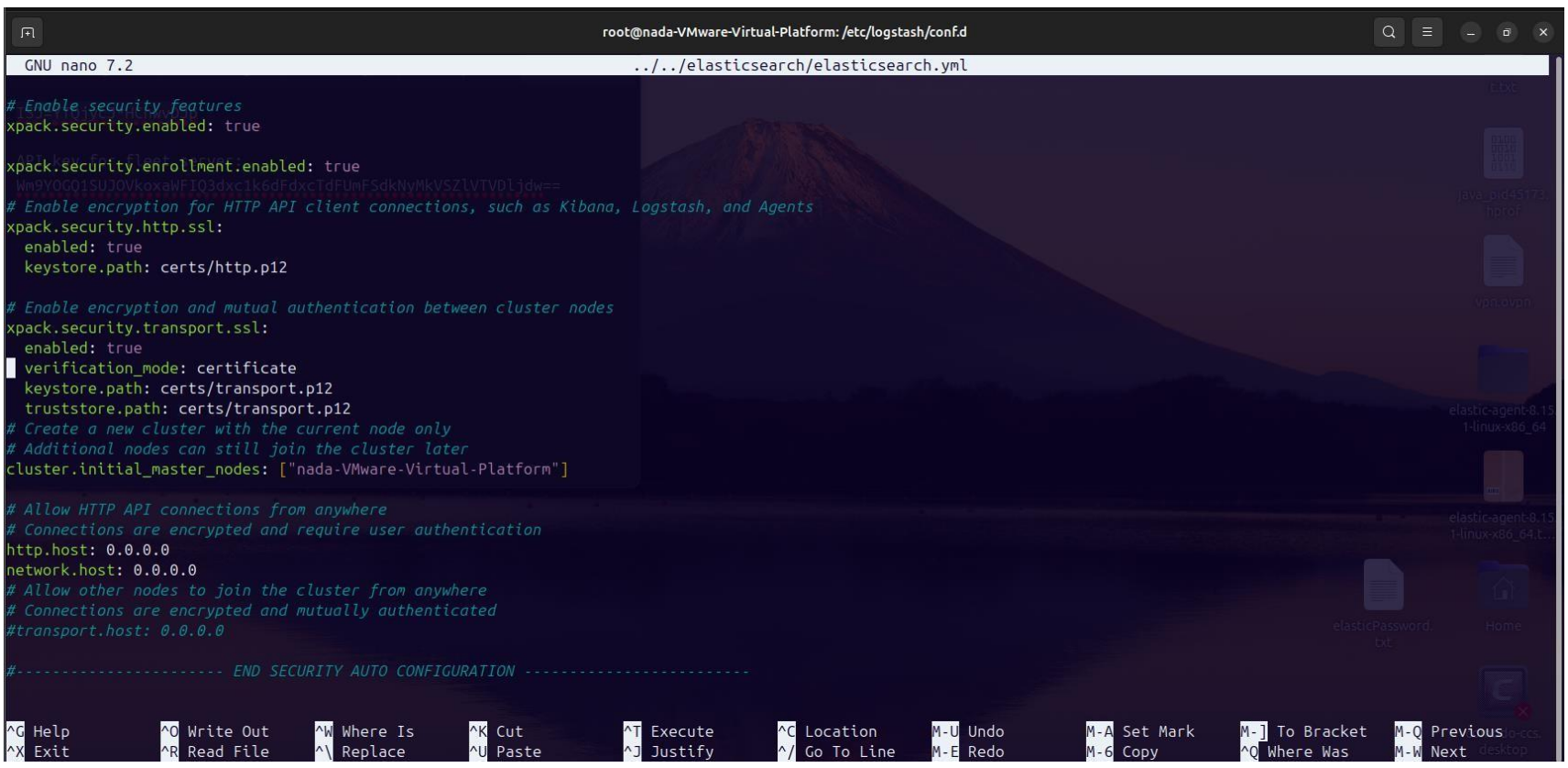

Figure S: Elasticsearch Configuration File

Now we can start Elasticsearch using command : **sudo systemctl start elasticsearch.service** and connect to Elasticsearch on browser at https://localhost:9200 , we can see that we successfully connect to Elasticsearch and it works correctly.

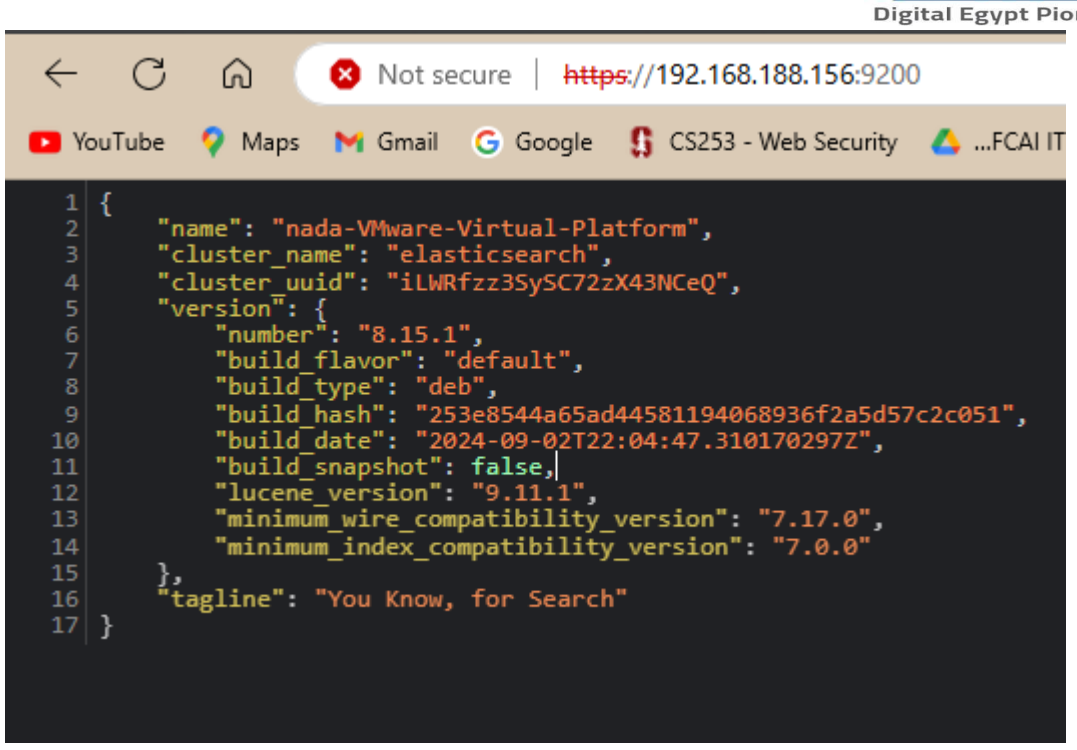

Figure 10: Elasticsearch

Install and Setup Kibana

The installation of Kibana is not different from Elasticsearch as it is same steps of importing the PGP Key, Update the repo and install Kibana Package.

As previously mentioned, we installed all components of the Elastic Stack on a single machine. Therefore, after installing Elasticsearch, we did not need to import the PGP key for Kibana, as we had done in previous installations. Importing the PGP key again would create a conflict due to the duplication of the same key. Thus, it is crucial to skip this step and proceed to install the Kibana package directly using the following command: **sudo apt-get install kibana**

And checking the configuration file at **/etc/kibana/kibana.yml**, Kibana is reached at http://localhost:5601 by default so we had to change it to http://192.168.188.156:5601, we can see here the token that we used before logging on to Kibana and the Elasticsearch host on port 9200.

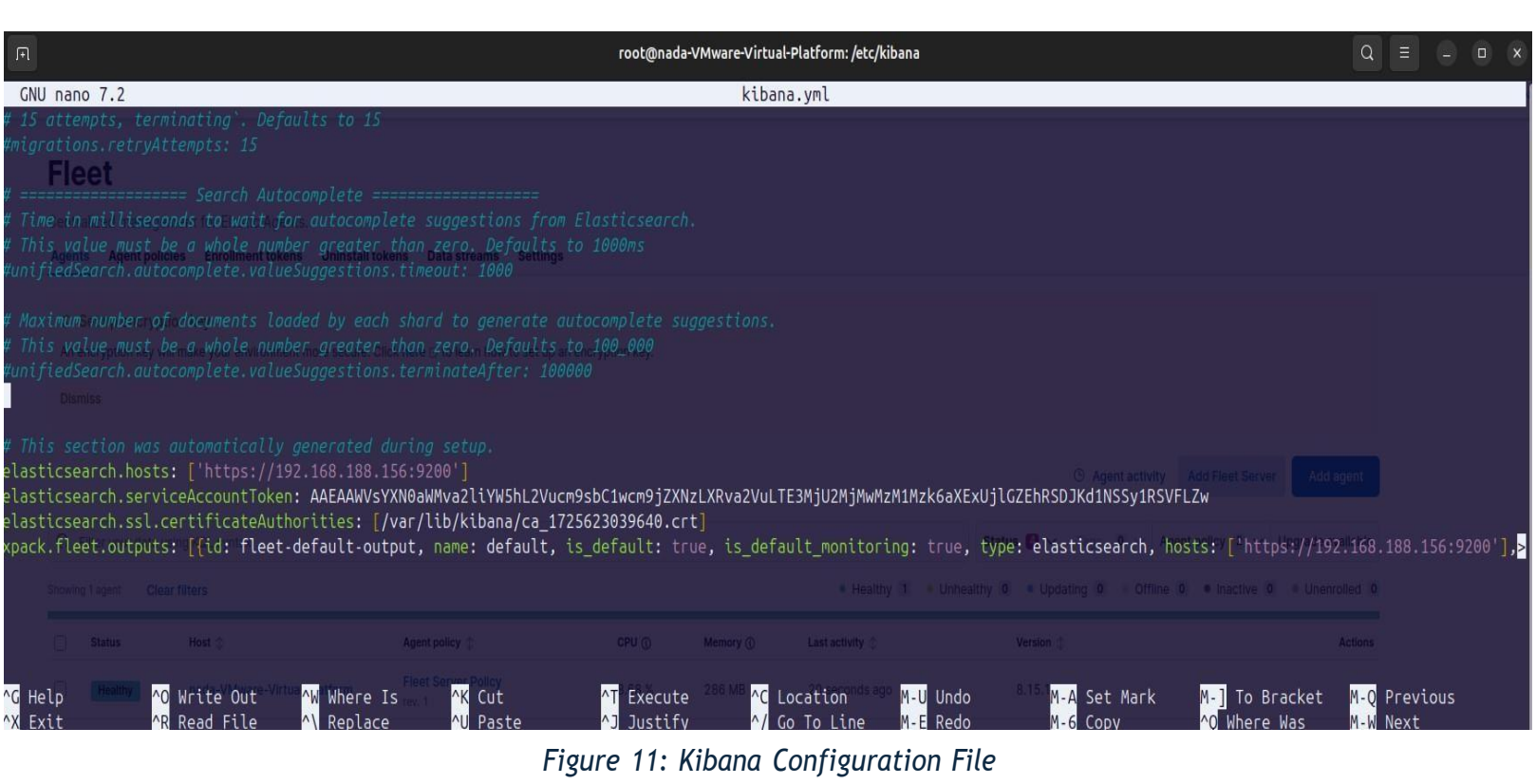

Figure 11: Kibana Configuration File

Connecting to Kibana on port 5601 and enter the Username (elastic) and password(ISJ=YTQjycJ*HChwvDJp)

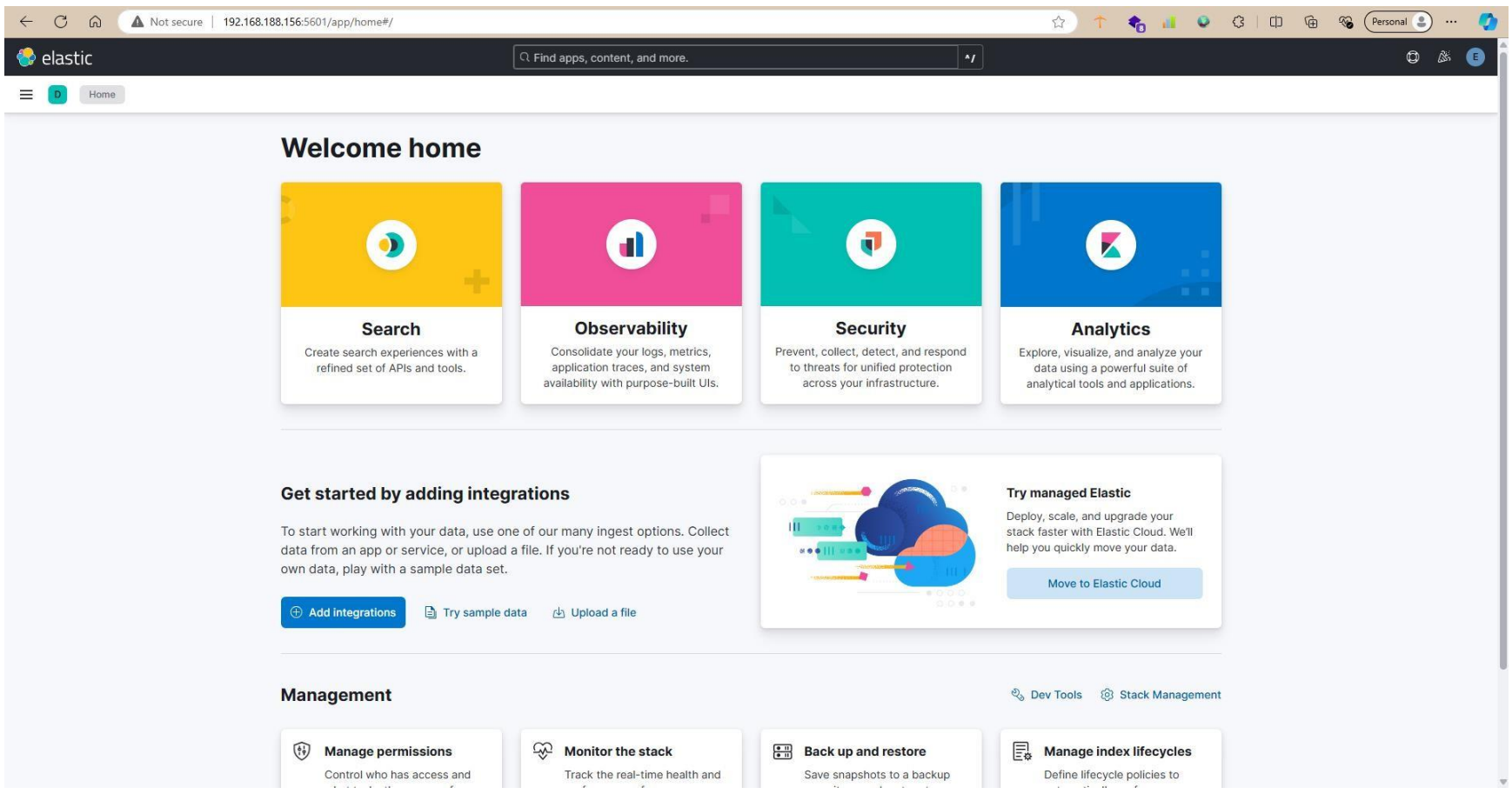

Figure 12: Kibana

Now we know that Elasticsearch and Kibana are working well. Next step will be the installation of the Fleet server and Agents on each machine for logging and controlling other Endpoints.

Agent Enrollment

The Agent is installed to collect logs from endpoints, while Fleet acts as a centralized management interface that controls and oversees multiple agents. This is particularly beneficial for large environments, allowing for streamlined management rather than manual oversight of each agent.

Agents retrieve their policies through the Fleet Server at https://<Agent-IP>:8220, which specifies the logs to be collected and the destinations for these logs.

Initially, we installed a Fleet Agent on PC-1, which runs Ubuntu OS and hosts the Elastic SIEM. The Fleet Server's IP address is set to 192.168.188.156, utilizing port 8220. To set this up, we navigated to **Management > Fleet**, selected **Add Fleet Server**, and followed the on-screen instructions.

The commands executed included the necessary configurations for the Fleet, detailing the Elasticsearch server for communication, the required token for authentication, and the standard port for the Fleet Server.

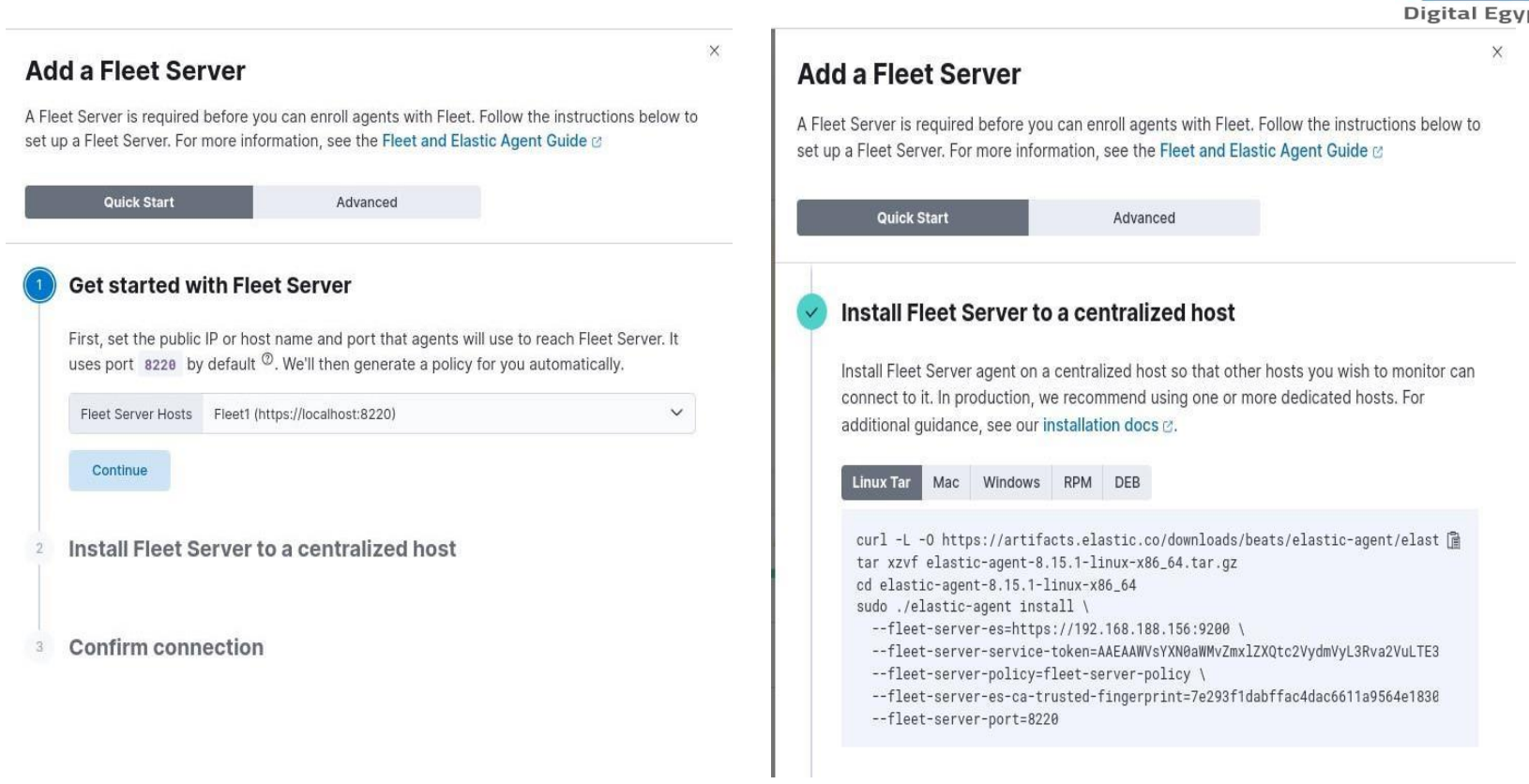

Figure 13: Agent Enrollment

Installing the Fleet Agent alone is insufficient for sending logs to Elasticsearch; we need to verify the configuration file to identify any issues. Upon review, we discovered that the Fleet Server requires an API Key, along with the username and password for Elasticsearch.

We proceeded to add the necessary username and password. For the API Key, we generated it through Kibana by navigating to **Management** > **Stack Management** > **API Keys**, where we created an API key named **fleet-server**.

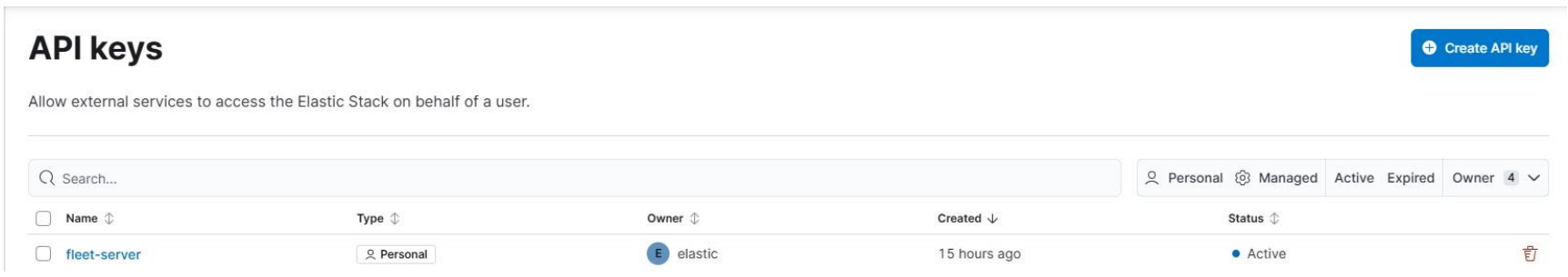

Figure 14: API Key Creation

And Add the API Key to configuration file of fleet server (elastic-agent.yml), Then restart the agent using the following command:

sudo systemctl restart elastic-agent

(OR)

Just provide username and password of Elasticsearch to the Agent

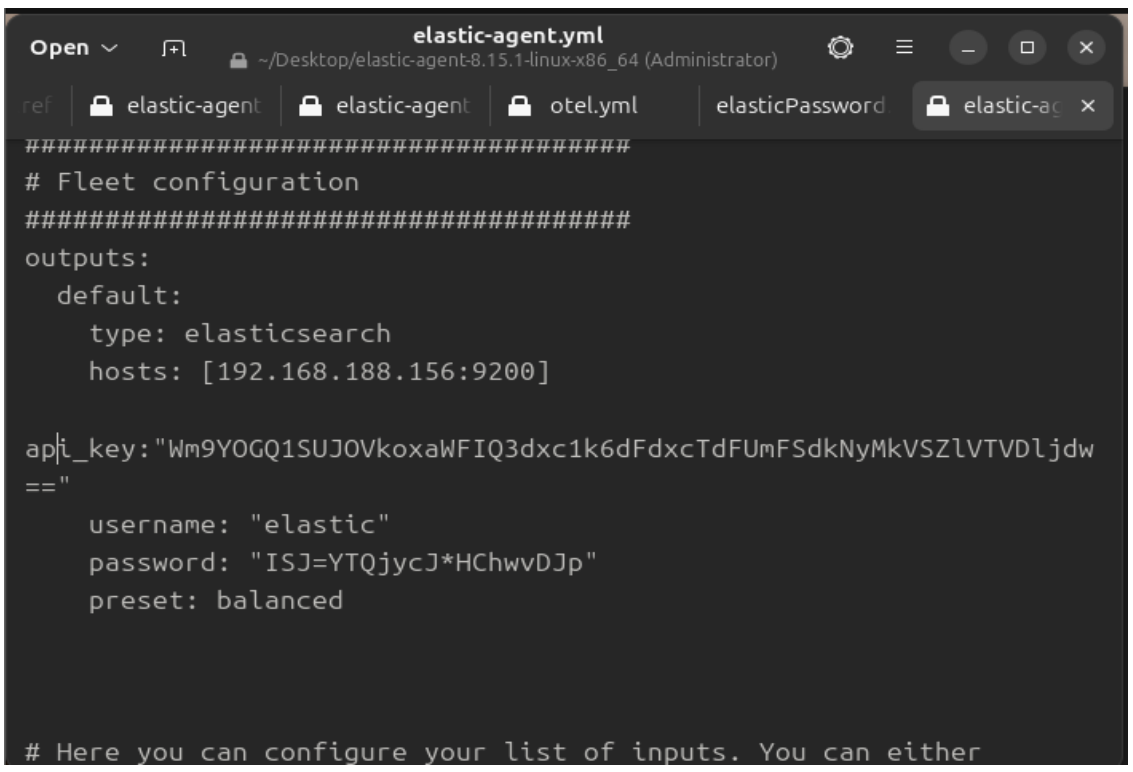

Figure 15:Agent's Configuration File

Fleet server before and after we configure the file:

```
root@nada-VMware-Virtual-Platform:/etc/logstash/conf.d# sudo elastic-agent status
┌ fleet
└  └ status: (HEALTHY) Connected
└ elastic-agent
   └ status: (HEALTHY) Running
root@nada-VMware-Virtual-Platform:/etc/logstash/conf.d# sudo systemctl restart elastic-agent
root@nada-VMware-Virtual-Platform:/etc/logstash/conf.d# sudo elastic-agent status
┌ fleet
└  └ status: (STARTING)
└ elastic-agent
   ├ status: (STARTING) Waiting for initial configuration and composable variables
   ├ beat/metrics-monitoring
   │  ├ status: (STARTING) Starting: spawned pid '88982'
   │  ├ beat/metrics-monitoring
   │  │  └ status: (STARTING) Starting: spawned pid '88982'
   │  └ beat/metrics-monitoring-metrics-monitoring-beats
   │     └ status: (STARTING) Starting: spawned pid '88982'
   ├ filestream-monitoring
   │  ├ status: (STARTING) Starting: spawned pid '88953'
   │  ├ filestream-monitoring
   │  │  └ status: (STARTING) Starting: spawned pid '88953'
   │  └ filestream-monitoring-filestream-monitoring-agent
   │     └ status: (STARTING) Starting: spawned pid '88953'
   ├ log-default
   │  ├ status: (STARTING) Starting: spawned pid '88945'
   │  ├ log-default
   │  │  └ status: (STARTING) Starting: spawned pid '88945'
   │  └ log-default-logfile-system-d678dc99-5e72-4031-b704-de5ec9c4d67d
   │     └ status: (STARTING) Starting: spawned pid '88945'
   └ system/metrics-default
      ├ status: (STARTING) Starting: spawned pid '88951'
      ├ system/metrics-default
      │  └ status: (STARTING) Starting: spawned pid '88951'
      └ system/metrics-default-system/metrics-system-d678dc99-5e72-4031-b704-de5ec9c4d67d
         └ status: (STARTING) Starting: spawned pid '88951'
```

Figure 1c: Logstash Server

The fleet server now sends the logs to Elasticsearch successfully.

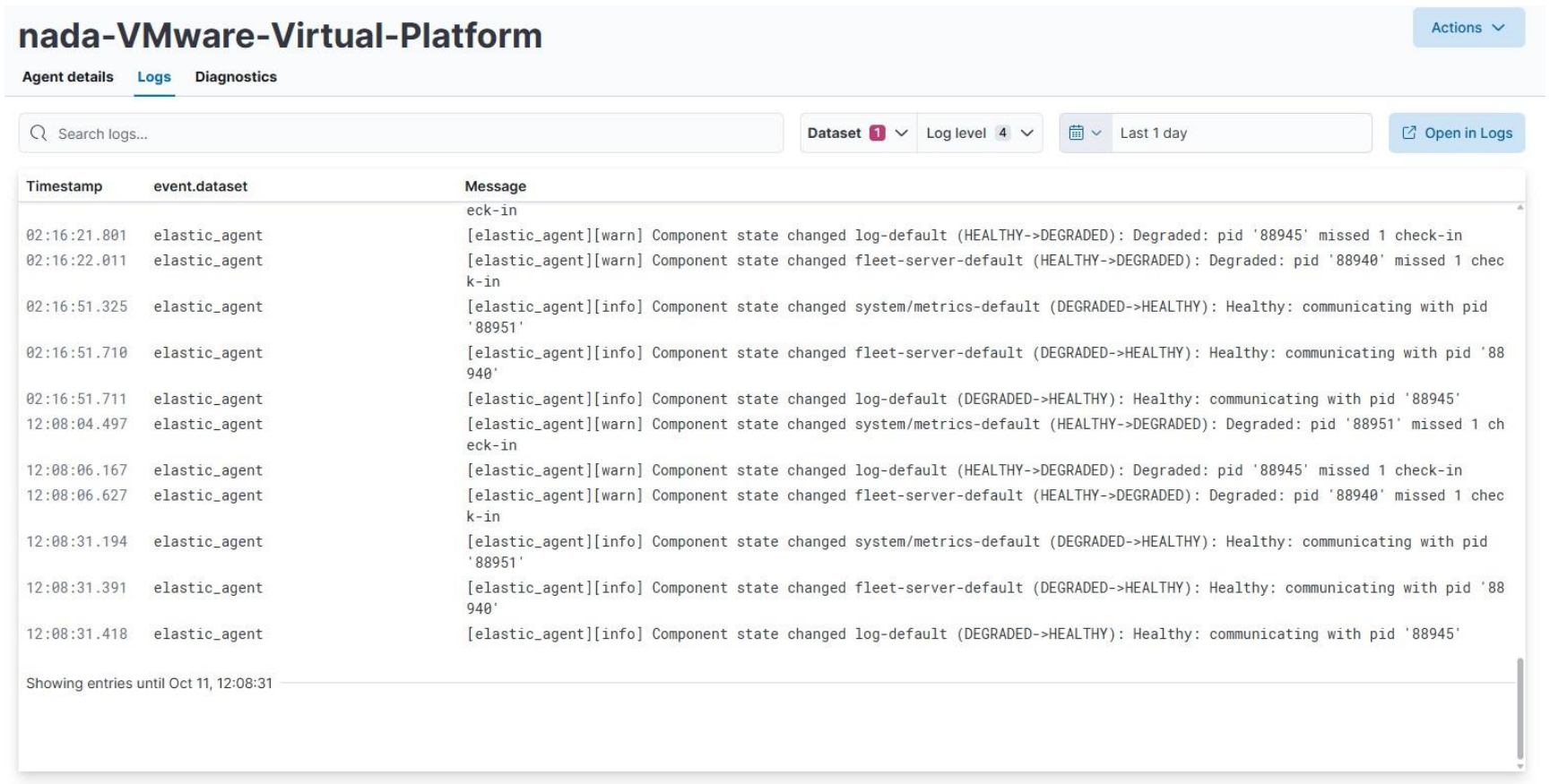

Figure 17: Fleet Agent Logs

Like how we enrolled the fleet, we will also enroll the other agents on PC-2 and PC-3. For the desktops that require management, we will select **Add Agent**, followed by **Enrollment in Fleet**. Next, we will download the necessary packages, extract them, and run the service using the following command:

.\elastic-agent.exe install --url=https://192.168.188.156:8220 --enrollment-token=SzZEclhaSUJxU3dkNnpyQkFEaF86U0dpSTZIb2lRa1dvQnpZSWF0UDQzUQ== --insecure

Checking each agent and ensure that it sends the logs appropriately from choosing the agent and then Logs section:

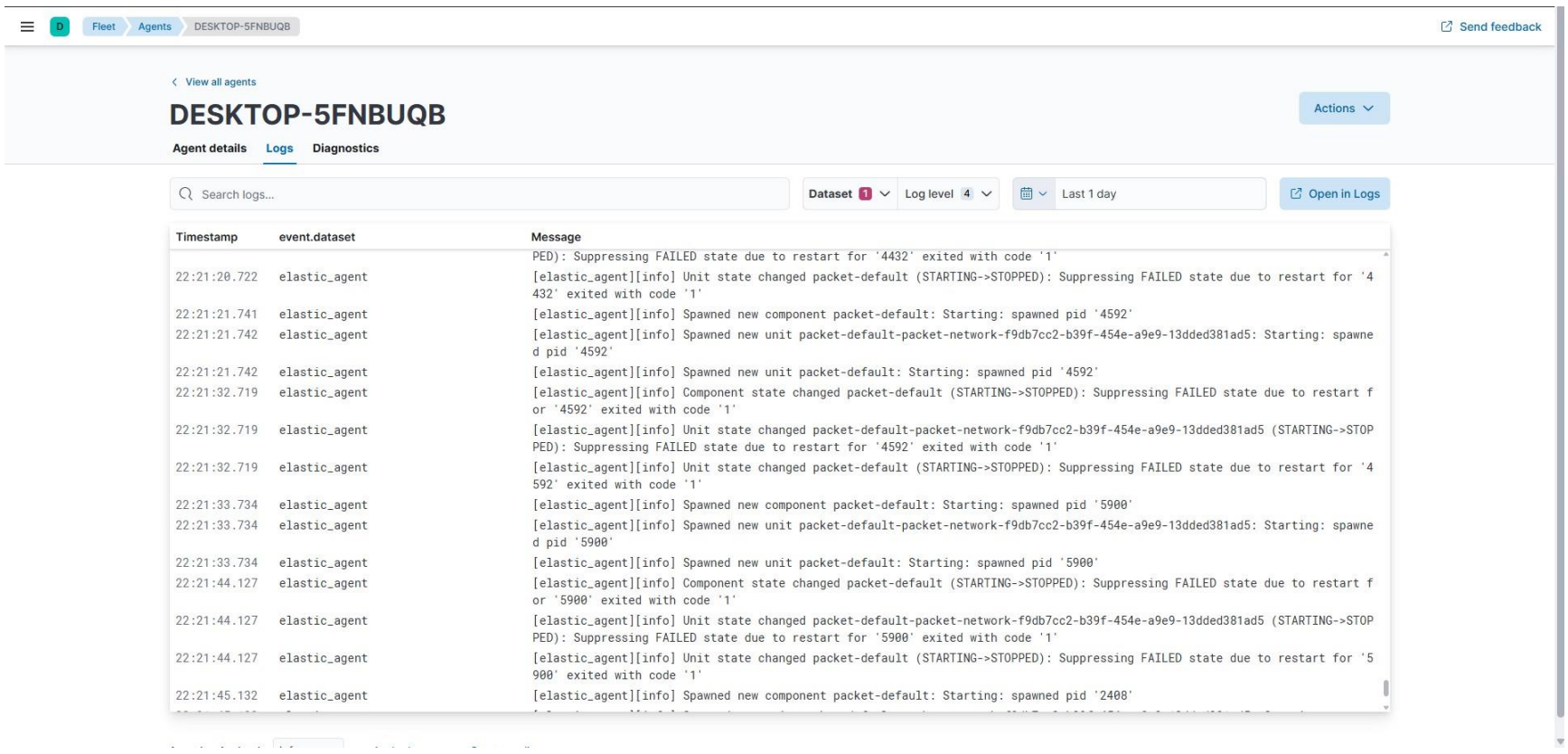

Figure 18: Agent Logs

Agentless Devices:

We cannot install agents on certain devices, such as network equipment including routers, switches, and firewalls.

To ensure effective monitoring of our firewall, we need to collect the logs it generates and transmit them to Elasticsearch for comprehensive investigation. To achieve this, we utilized the syslog protocol to push the logs directly to our Logstash server, which filters, parses, and forwards the logs to Elasticsearch for further analysis.

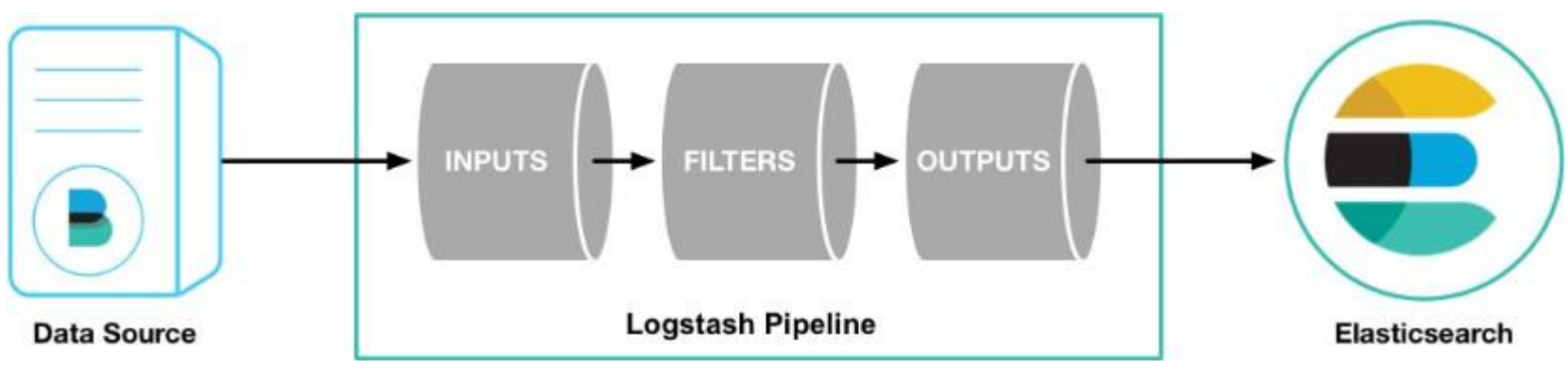

Figure 1S: Logstash

Installing Logstahs server using the command: **sudo apt-get install logstash**
After installing the Logstash server, we created a configuration file for Forti-Firewall logs at **/etc/logstash/conf.d,** we named the file (firewall.conf) and add a configuration to listen to syslog port 514 to receive logs coming from the firewall and throw the outputs to Elasticsearch as follows:

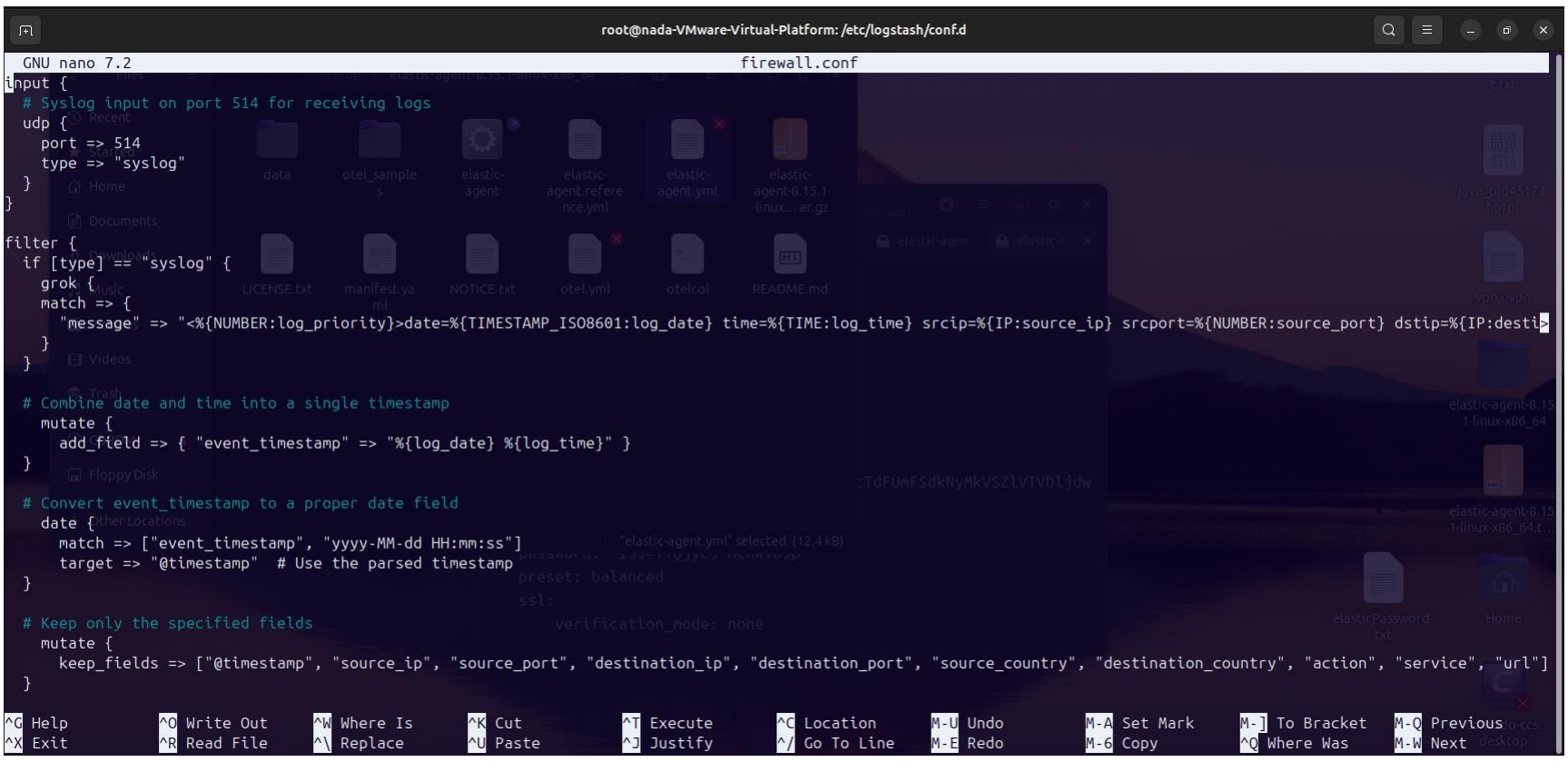

Figure 21: firewall configuration file

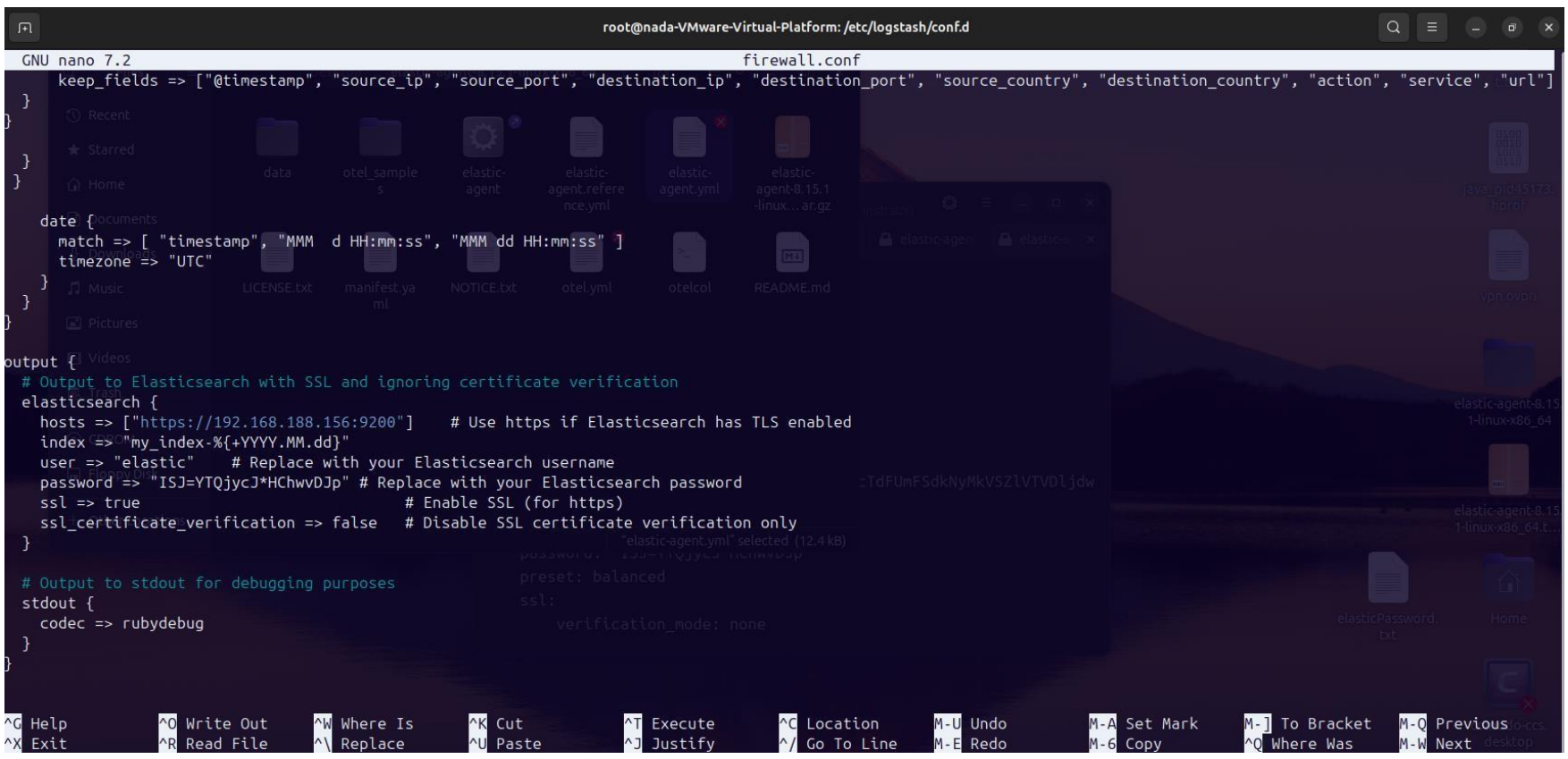

Figure 20: Output of firewall Conf file

Configure the Firewall to send its logs through syslog port, then apply:

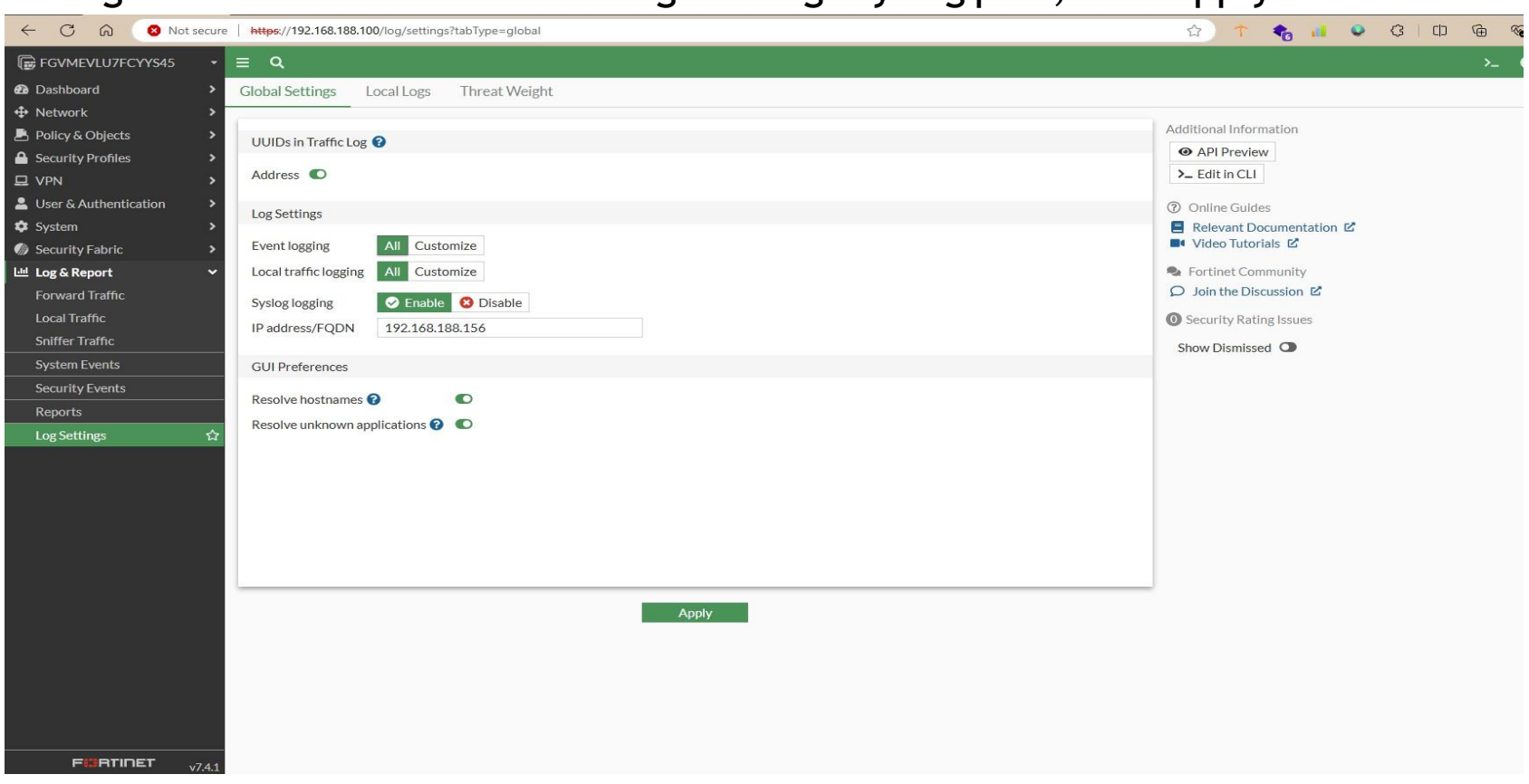

Figure 22:Pushing logs

Running the Logstash server using the command: **sudo /usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/firewall.conf,** to see if it is receiving

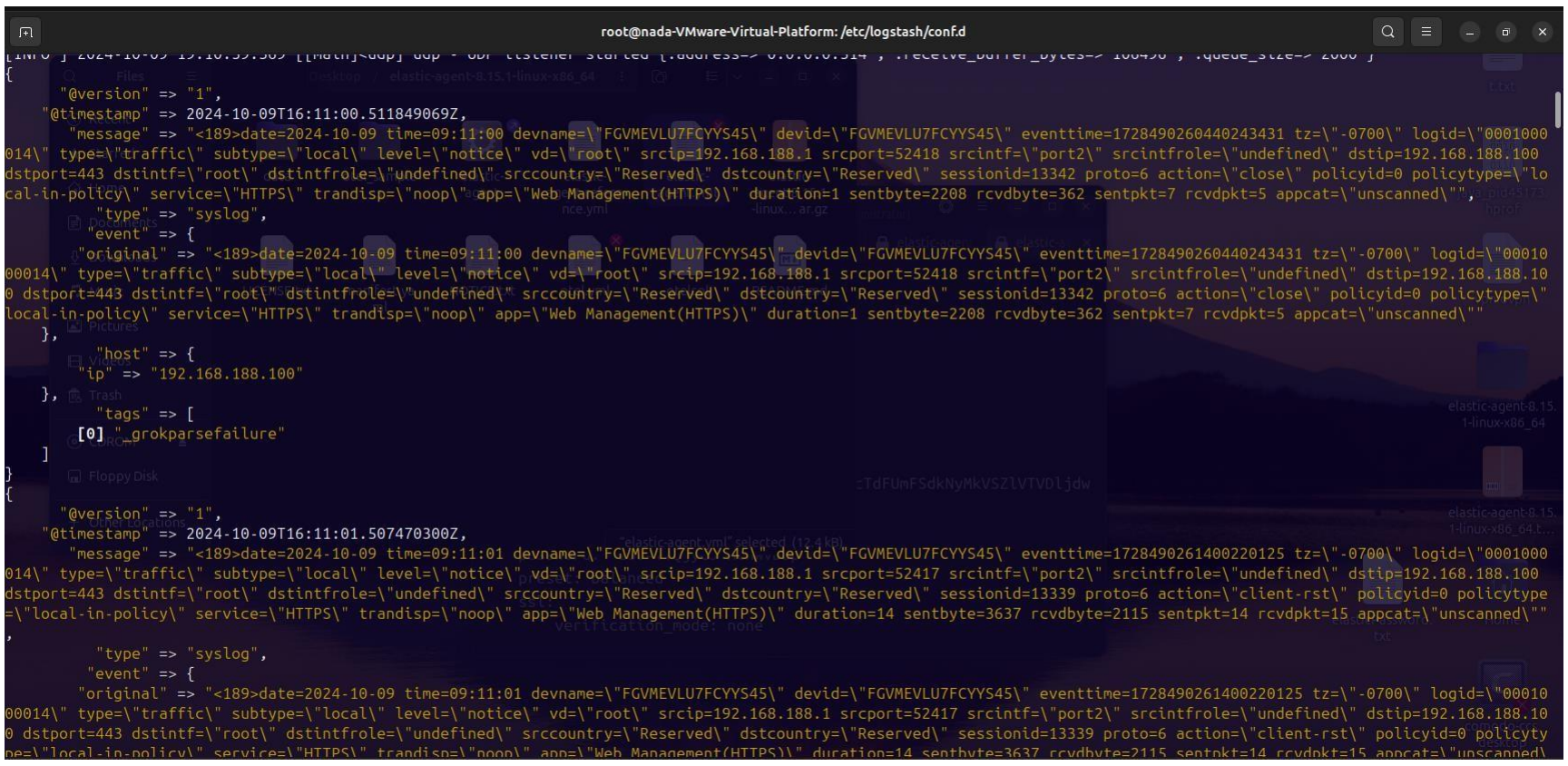

Figure 23: Debuging firewall.conf

the logs correctly.

The index we created previously (forti-logs*) to store the logs coming from Logstash:

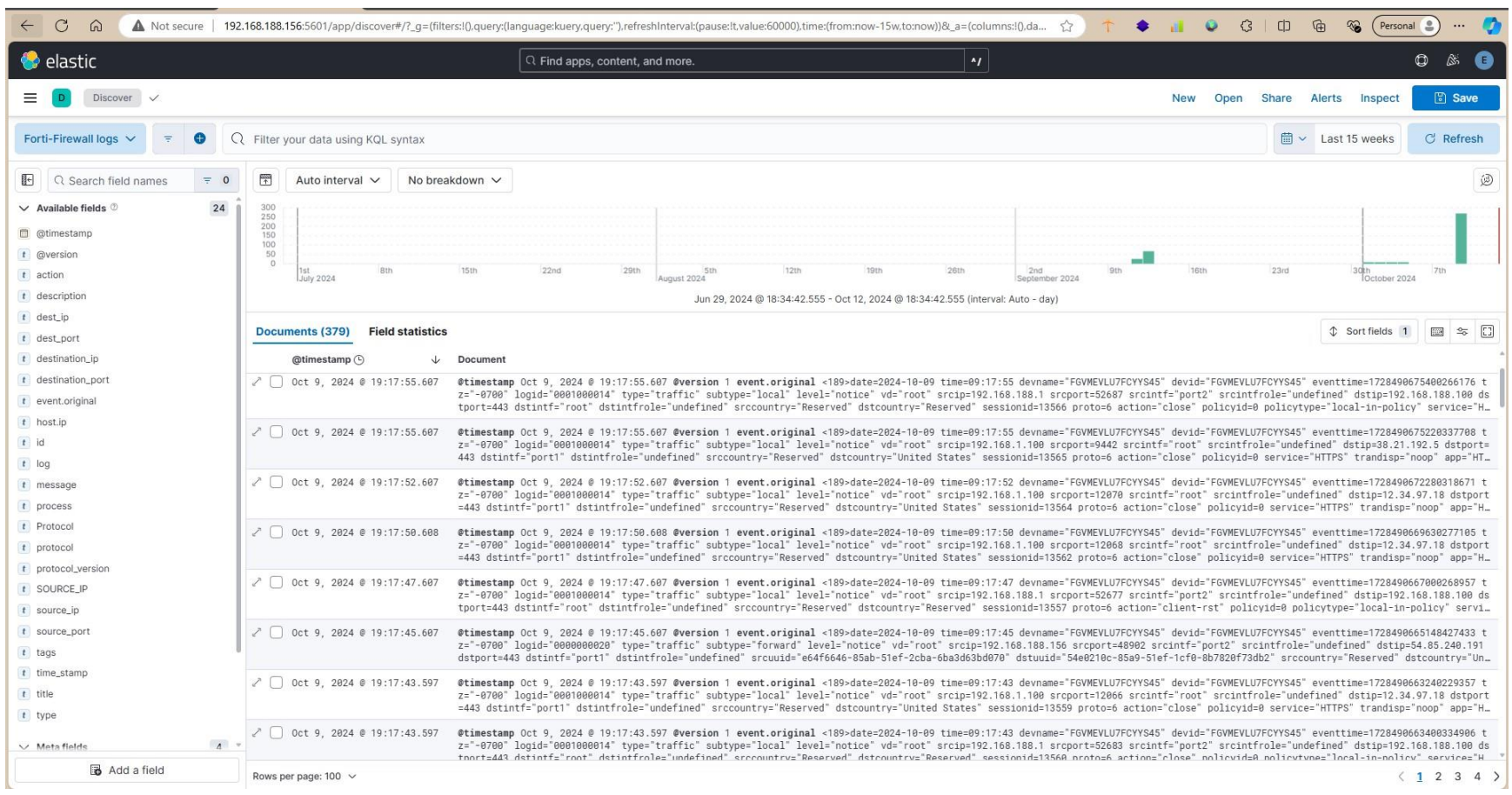

Figure 24: Firewall logs

Monitoring and Alerting

To effectively monitor our devices and endpoints, we established an index that integrates with the previously created policies. This integration enables us to collect diverse logs from various sources. By defining specific alert rules, we can access comprehensive and relevant logs, facilitating thorough investigations.

Integrations:

We incorporated several integrations into Agent Policy 1 to enhance our log collection capabilities.

Network Packet Capture: This integration sniffs network packets on a host and dissects known protocols.

System: The System integration allows you to monitor servers, personal computers, and more.

File Integrity Monitoring: This integration sends events when a file is changed (created, updated, or deleted) on disk. The events contain file metadata and hashes.

Elastic Defend: Elastic Defend provides organizations with prevention, detection, and response capabilities with deep visibility for EPP, EDR, SIEM, and Security Analytics use cases across Windows, macOS, and Linux operating systems running on both traditional endpoints and public cloud environments.

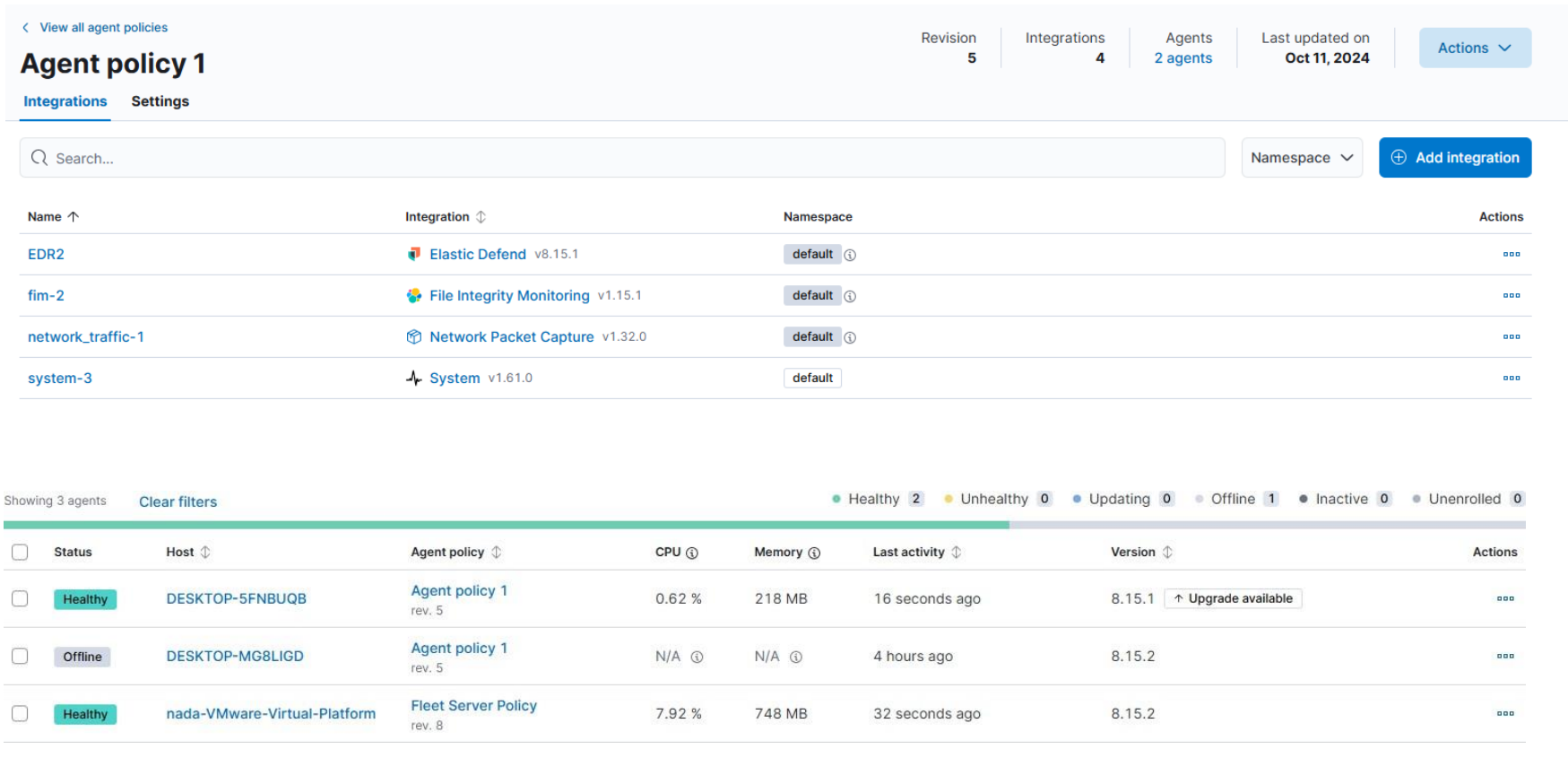

Figure 25: Agents and policy

Rules and Alerts:

In this section, we focus on the generation and management of alerts derived specifically from Endpoint Detection and Response (EDR) systems. By establishing targeted rules, we ensure that any suspicious activities or potential threats are promptly detected and reported. This proactive approach allows for effective monitoring and rapid incident response, enabling us to maintain a robust security posture and safeguard our network from evolving threats.

Initially, we implemented several rules designed to detect incidents aligned with the MITRE ATTCCK framework. These rules specifically target threats such as suspicious child processes related to privilege escalation, unauthorized copying of SAM files, Remote Desktop Protocol (RDP) attacks, EDR alerts, and reverse shell activities.

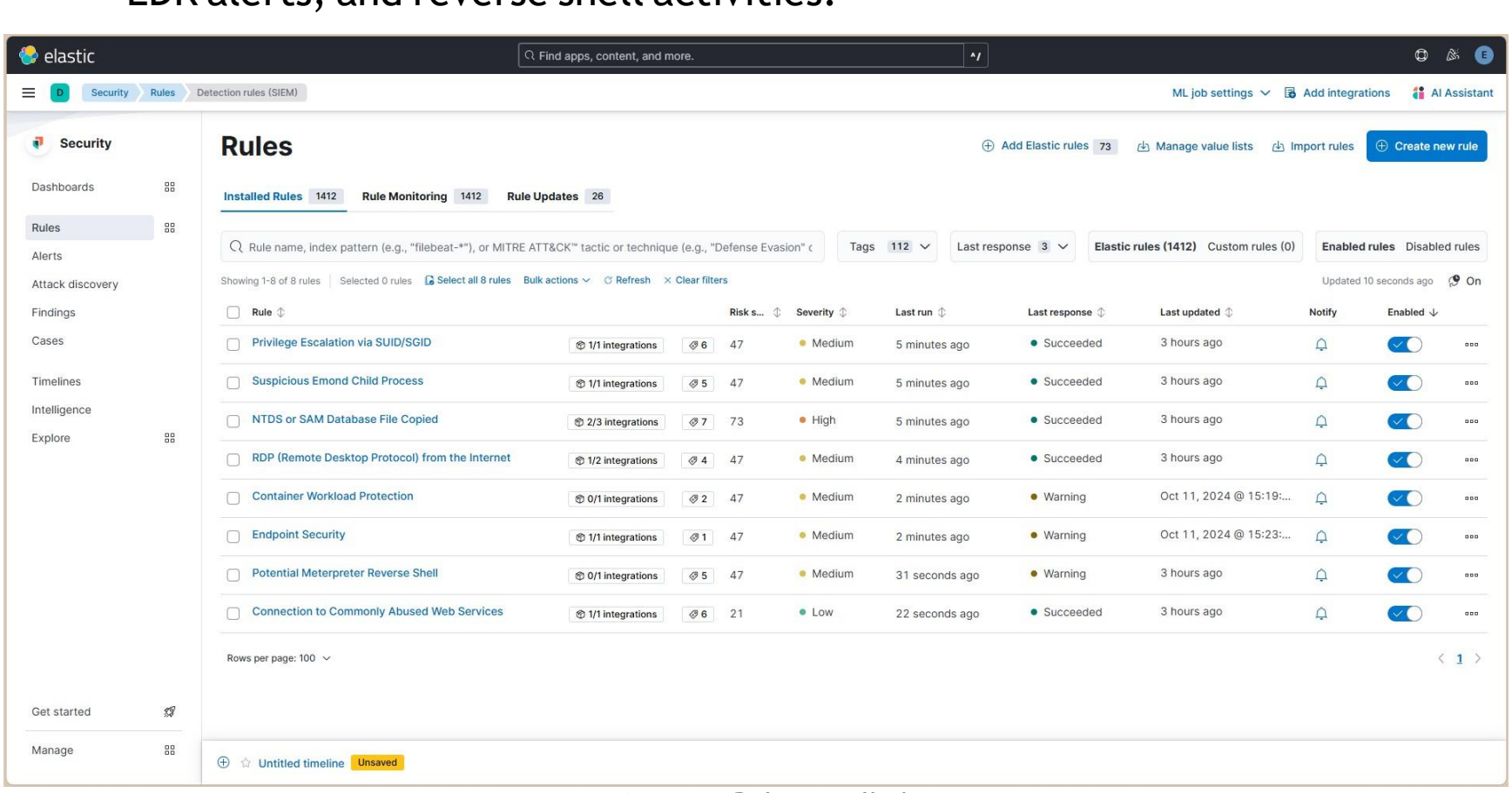

Figure 2c: Rules Installed

Some Techniques covered in MITRE ATTCCK:

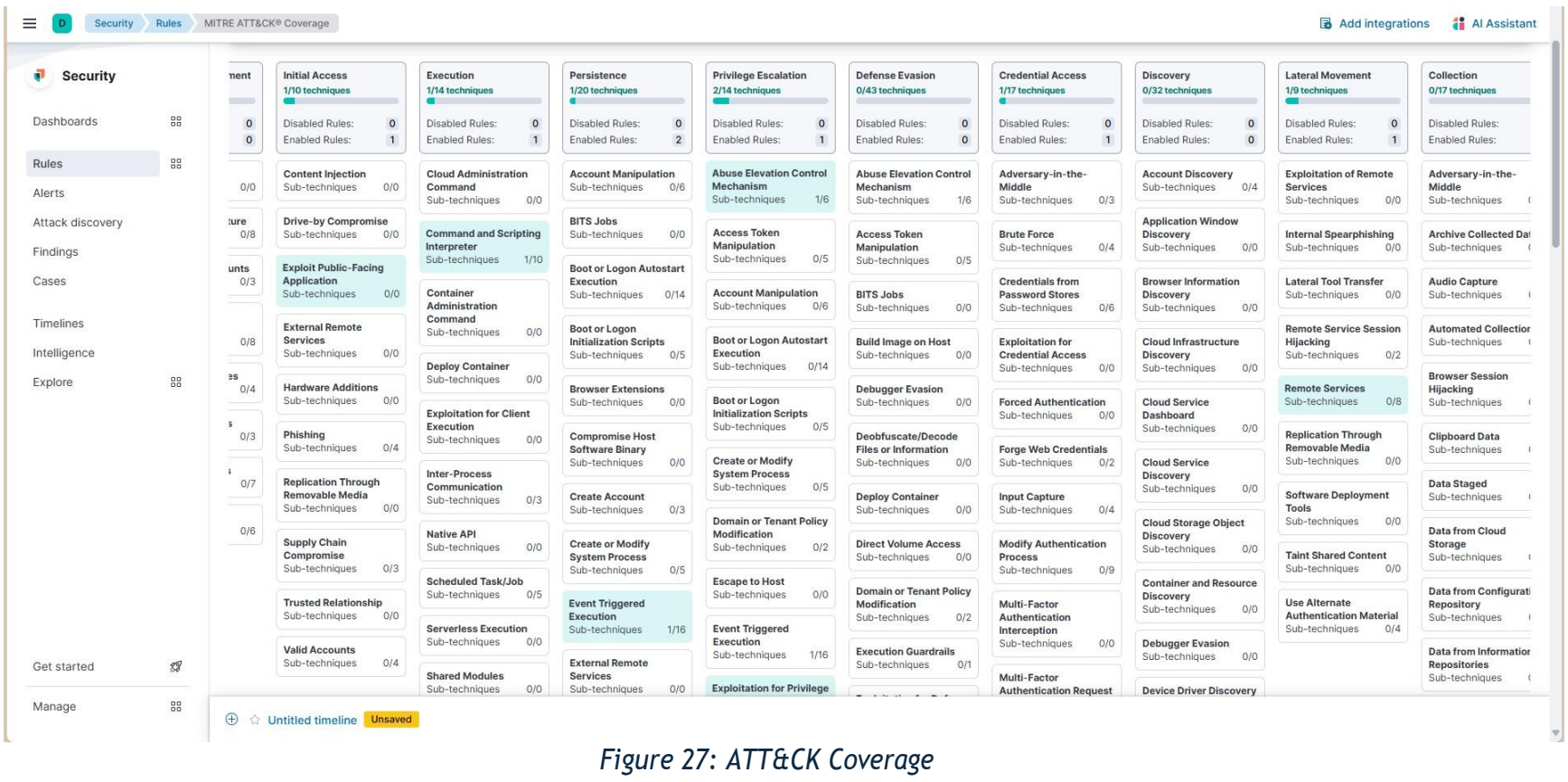

Figure 27: ATT&CK Coverage

To validate the rule, we installed a malicious ZIP file with the hash value **cddeGG520664ac313d43G6462001Gc61** and subsequently extracted its contents. Prior to opening the file, we observed its presence.

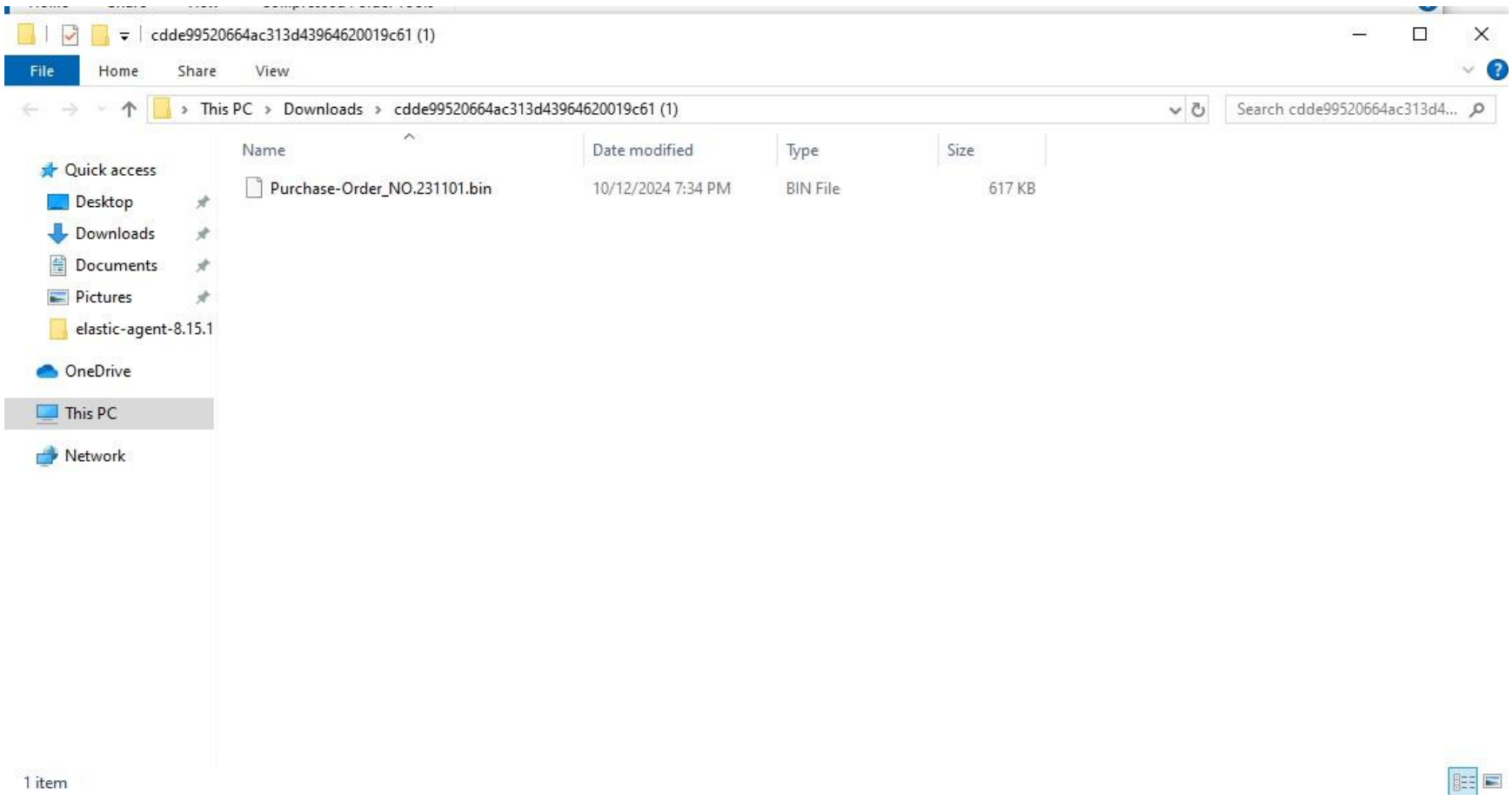

Figure 28: Malware

and shortly after, the EDR detected the threat, removing the file and generating an alert from Elastic Defender.

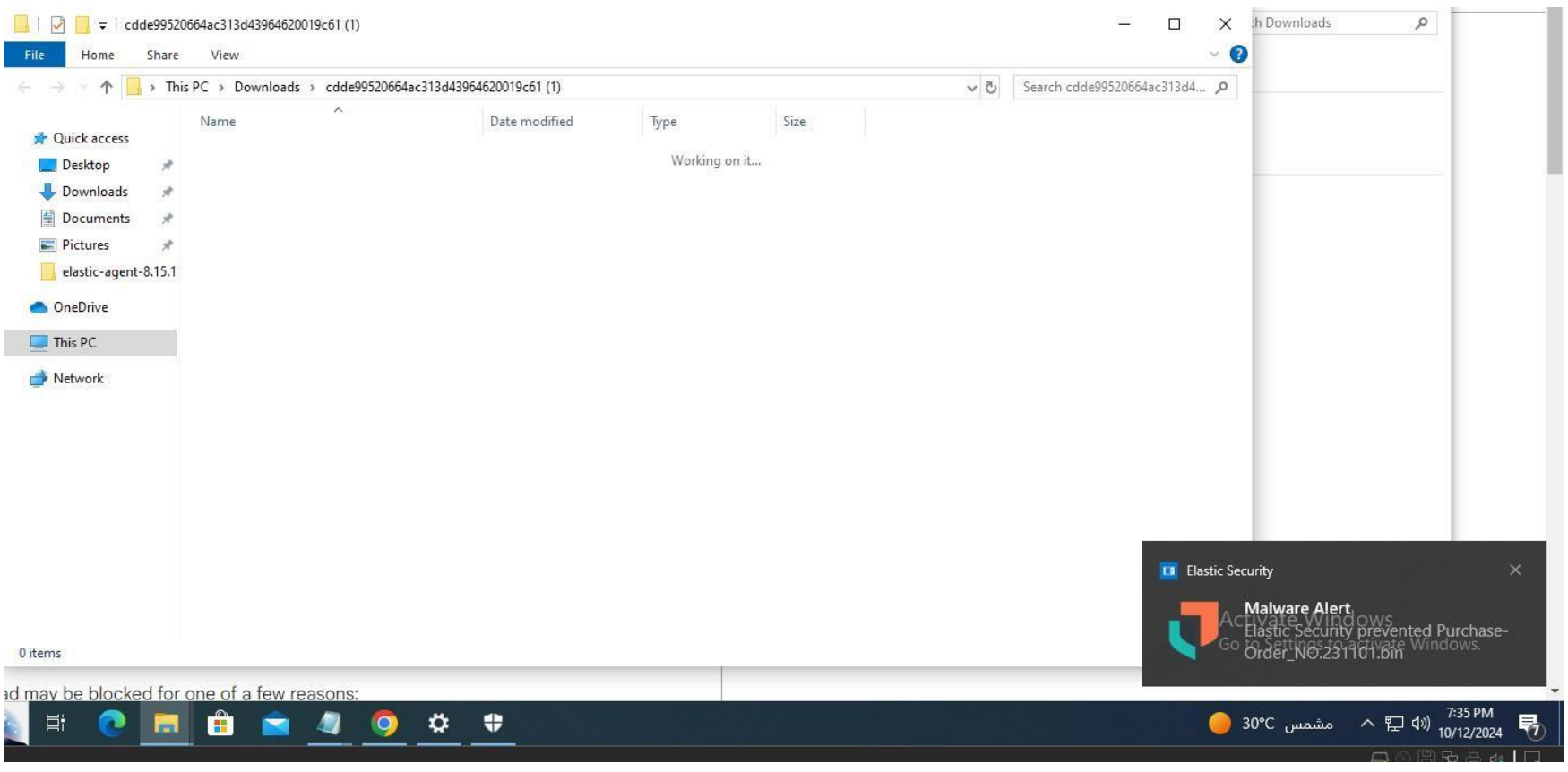

Figure 2S: Malware Prevention

In the Alerts section located within Security, we can view the alerts generated by the Malware Prevention Alerts rule.

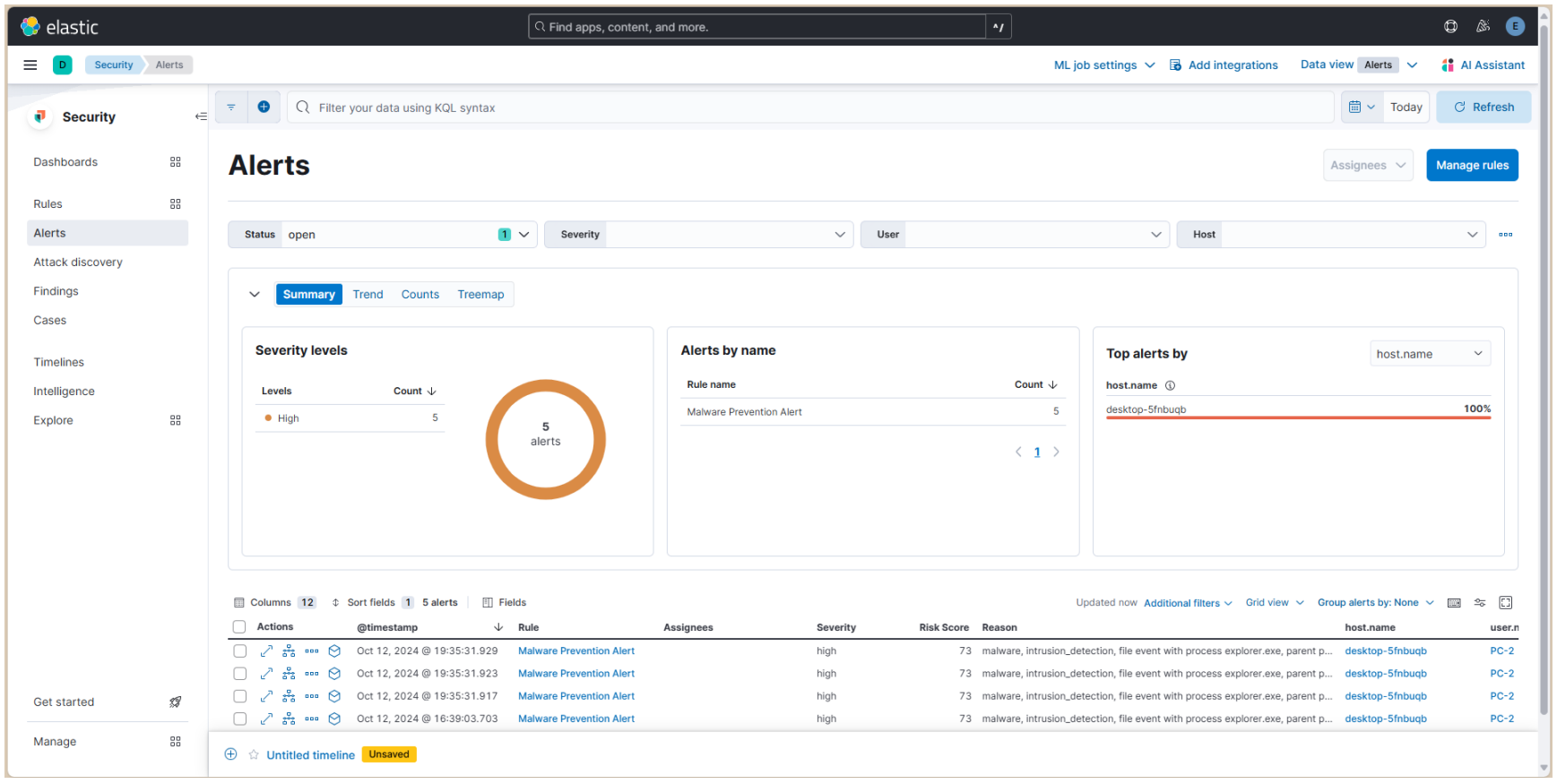

Figure 30: SIEM Alerts

By selecting one of the alerts, we can access detailed information about the incident, including the operating system, host name, malicious process, file path, and other relevant details.

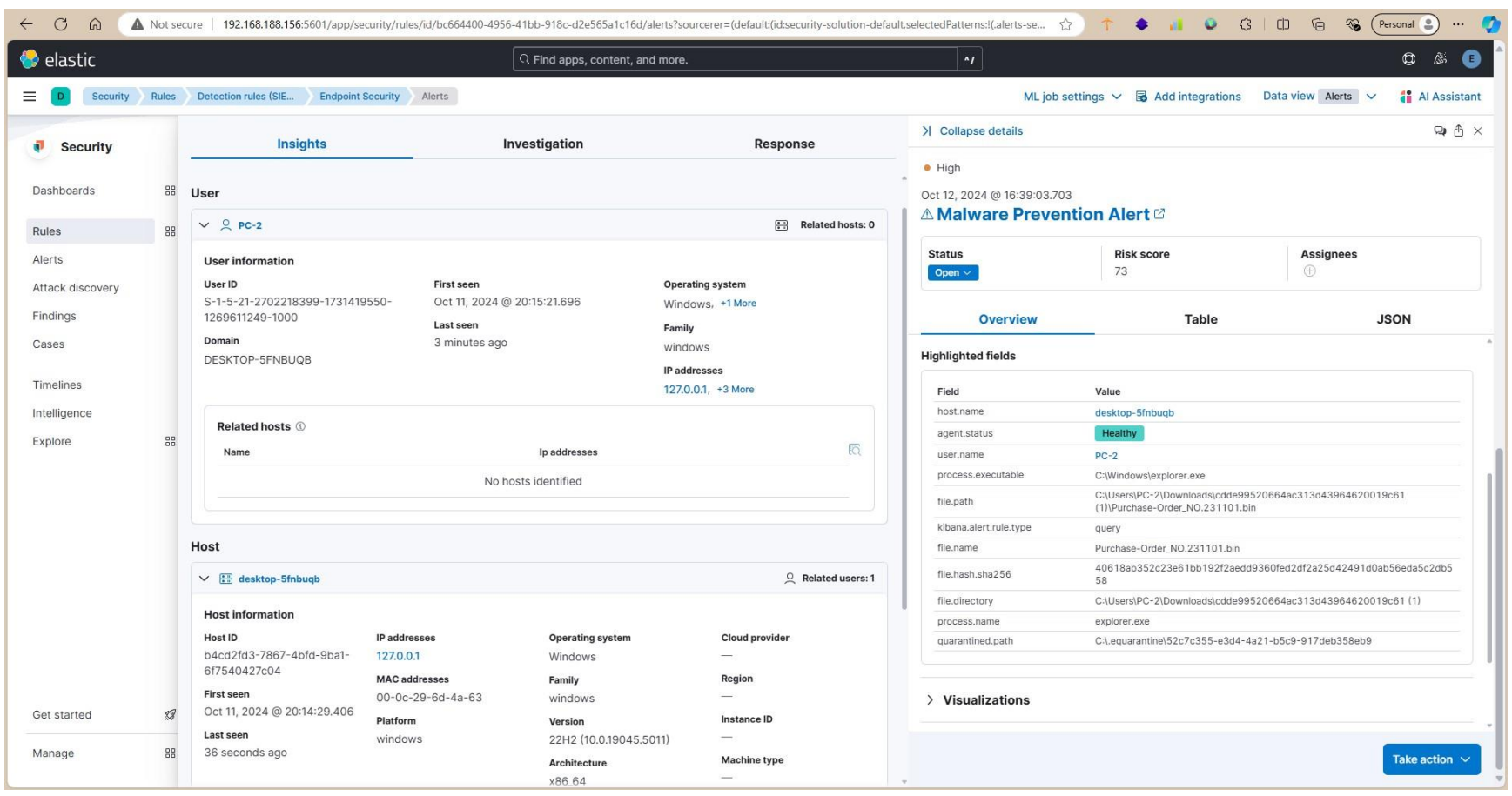

Figure 31: Alert Details

Conclusion

In conclusion, the installation and configuration of Elastic SIEM have proven to be a vital step in enhancing our cybersecurity posture. By effectively integrating various components of the Elastic Stack, we established a robust platform for monitoring and alerting that is capable of detecting and responding to potential threats in real-time. The careful configuration of agents and the implementation of tailored rules, based on the MITRE ATTCCK framework, have enabled us to identify incidents such as privilege escalation and malicious file downloads efficiently.

Moreover, the incorporation of a network firewall for log collection further strengthens our ability to monitor and analyze security events across the entire network. The alerts generated from our configurations provide actionable insights, allowing for timely investigations and responses to incidents.

Overall, this project has successfully laid the groundwork for an effective monitoring and alerting system, ensuring that we can proactively safeguard our environment against emerging threats and vulnerabilities.