Digital Egypt Pioneers

وزارة الاتصالات وتكنولوجيا المعلومات
Ministry of Communications and Information Technology

Digital Egypt Pioneers Initiative (DEPI)

Malware Analysis

Cyber Security Incident Response Analyst Track

Team Members:

1. *Omar Abdelrahman Ahamed*
2. *Fahd Mahmoud Abdelkhalek*
3. *Kholoud Khaled Mohamed*
4. *Nada Saleh Mohamed*
5. *Esraa Matarawy Abdelmoniem*

Supervised by:

Eng. Nour Eldin Essam

Table of Contents

Malware Analysis

Ministry of Communications and Information Technology

Digital Egypt Pioneers

SIEM Configuration and Monitoring

Table of Figures

Prevention Strategy and Training

1. Introducti

Malware is short for "malicious software", it is any software that one intends and develops to cause damage to the computer, server, client, or network. Some cybercriminals use malware in attempting to steal sensitive information, gain access to systems, or disrupt operations.

From simple worms to the complex malware of today, cybersecurity has always been an evolving area. It is the leading cyber threat after ransomware.

Basically, protection against malware-both for individual users and within whole organizations-relies on awareness and the sense of responsibility to take proper security measures regarding systems, platforms, and data in use. Accordingly, everybody must take proactive steps in this respect, as without such engagement, one can hardly achieve proper protection.

The following detailed report looks at the analysis of different types of malwares, their impact on systems and networks, and the methods to detect and avoid them.

2. Overview of Malware Types

Malware is classified into types based on its behavior, method of infection, and purpose. The following are some of the key categories of malware:

2.1. Viruses

Description: Viruses attach themselves to a clean file or program, often damaging or destroying it. They, themselves, cannot proliferate and depend on some user interference-such as executing infected software-to spread.

Behavior: It deletes files, corrupts files, consumes system resources, and leads to system crashes.

Propagation: Common methods include email attachments, infected USB drives, and downloads from dubious websites. Once a virus infects a system, it can use that machine to spread to others, often through shared networks.

Example: The Melissa Virus (1999), which infected email systems and crashed corporate servers due to the high volume of emails created by the virus to spread itself.

2.2. Worms

Description: Worms are self-replicating malware propagating without the need for user interaction. The propagation of worms relies on the existence of software and network vulnerabilities.

Behavior: Worms consume network bandwidth and possibly cause network congestion or even a network outage.

Propagation: Spread via networks by taking advantage of security flaws in network protocols, operating system or application vulnerabilities.

Example: The ILOVEYOU Worm (2000) infected millions of computers worldwide, crashed into e-mail systems, and resulted in US$10 billion of damages.

2.3. Trojan Horses

Description: Trojans pretend to be some form of legitimate software but are really a malicious program. They don't replicate but provide unauthorized access or run other malware programs on the system.

Behavior: Trojans can steal sensitive data, create backdoors, and give opportunities for the attacker to control systems remotely.

Propagating: Most of them spread through malicious e-mail attachments, fake software updates, or compromised websites.

Example: The Zeus Trojan (2007): used for banking credential theft using keystroke logging, causing massive losses. Restoring access requires paying the ransom, usually in cryptocurrency.

2.4. Ransomware

Description: Ransomware is a class of malware that either encrypts the files or locks users out of their system and demands an exchange-usually cryptocurrency-for return of access.

Behavior: Makes data and systems unusable until a ransom is paid.

Propagation: It spreads by way of phishing emails, malicious links, or by trying to exploit software vulnerabilities, particularly in remote desktop protocols.

Example: WannaCry Ransomware (2017) spread quickly due to an exploited vulnerability against Windows, affecting over 240,000 computers in 160 countries and causing losses of up to $5 billion.

2.5. Spyware

Description: It gathers information from an infected system in secrecy, which the user is unaware of. Examples include login credentials, tracking of keystrokes, and browsing habits.

Behavior: It monitors user activity as well as sensitive information, resulting in breaches of privacy and identity theft.

Propagation: Spyware normally infects your device through fake software bundling, phishing emails, or security vulnerabilities.

Example: Pegasus Spyware, 2017, had been used to track and collect personal information of targets, including calls and messages, through their mobile devices.

2.6. Adware

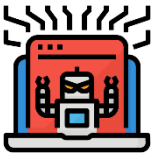
Description: Adware is a malicious program that pops up unwanted ads; sometimes it gathers user browsing data to display targeted advertisements. Generally, adware is not harmful but is intrusive and slow to the system performance.

Behavior: It pops up ads, slows down the performance of a system, and sometimes redirects browsers to malicious sites.

Propagation: It is mostly installed with free software installation or is bundled into some legitimate-appearing downloads.

Example: Fireball Adware (2017) hijacked web browsers and generated revenue out of the fake online traffic.

2.7. Rootkits



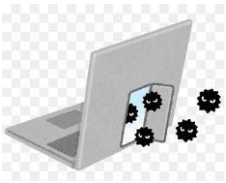
Description: The rootkit provides an attacker with administrative-level access to the system and hides the very existence of the malicious program itself. They are also capable of editing core system functions to disguise other malicious activities of the attacker.

Behavior: Rootkits allow unauthorized access to a system and can't be detected by antivirus.

Propagation: It is mostly installed through Trojans or some other malware that receives administrative privilege.

Example: Sony BMG Rootkit (2006) tracked secretly the media use of users, compromising the security of millions of systems.

2.8. Backdoors



Description: A backdoor provides illegitimate access to an attacker to a system or network bypassing normal authentication procedures.

Behavior: Allows sustained remote accessibility to infected systems, which are often used in conjunction with other malware to reach permanent control.

Propagation: Generally installed as part of another malware or by exploiting security weaknesses.

Example: The Back Orifice Tool (1998) was used to gain a remote control over infected Windows systems.

3. Detection Methods

Detection methods are crucial for identifying and mitigating malware threats.

3.1. Signature-based Detection

Detection Technique: Signature-based detection is the most traditional method. It identifies malware by comparing a file's binary structure or unique signatures (e.g., hash values) against a database of known malware signatures. Each signature is a unique code pattern associated with a specific piece of malware.

Tools:

- **ClamAV**: Open-source antivirus engine.
- **Kaspersky**, **Norton**: Commercial antivirus solutions with extensive signature databases.
- **Virustotal**: An online platform that checks files and URLs against multiple antivirus engines.

How It Works: When a file enters the system, the antivirus scans its code and compares it against an internal signature database. If the code matches a known signature, the system flags it as malware and typically isolates or removes the file.
Pros:

- **Efficiency**: Fast and accurate for identifying well-known malware.

- **Low resource usage**: Because it simply matches patterns, it requires minimal system resources.

Cons:

- **Inability to detect zero-day threats**: Signature-based systems cannot identify new or unknown malware that doesn't have a known signature.
- **Frequent updates needed**: The database must constantly be updated with new malware signatures, making it reactive rather than proactive.

Best Use Case:

- **Well-established malware detection**: Suitable for identifying old or well-known malware, typically in environments with legacy systems or basic threat protection needs.

3.2. Behavior-based Detection (Anomaly Detection)

Detection Technique: This method focuses on identifying unusual or suspicious behaviors in applications and system processes. It monitors real-time activities such as file modifications, network connections, and system resource consumption. When abnormal behavior is detected—such as a program attempting to escalate privileges or modify system files—the system flags it as suspicious.

Tools:

- **CrowdStrike Falcon**: Monitors real-time behavior and can quickly detect anomalous activities.
- **CylancePROTECT**: Focuses on detecting malicious behavior without needing a signature database.

How It Works: Behavior-based detection does not rely on predefined malware signatures. Instead, it creates a baseline of normal activities on a system or network. Any deviation from this norm, such as unusual network traffic or changes to critical system files, triggers an alert. For example, ransomware encrypting files will trigger the system to detect this behavior as abnormal.

Pros:

- **Effective against zero-day and file-less malware**: It can detect unknown malware based on its actions rather than its code.

- **Real-time threat detection**: Provides immediate responses to potentially malicious activities.

Cons:

- **False positives**: Legitimate software may be flagged as suspicious if it behaves in an unusual way, leading to unnecessary alerts.
- **Resource-intensive**: Monitoring the system in real-time can use significant system resources and impact performance.

Best Use Case:

- **Advanced Threat Protection (ATP)**: Ideal for environments that need protection from sophisticated attacks, such as zero-day malware, file-less threats, and advanced persistent threats (APTs).

3.3. Static Analysis

Detection Technique: Static analysis involves examining the malware without executing it. This method inspects the binary code, file structure, headers, and other properties of the file to identify any malicious characteristics. It checks for things like packed or obfuscated code, suspicious API calls, or strange imports that could signal malicious intent.

Tools:

- **IDA Pro**: A powerful disassembler that provides deep insights into malware binaries.
- **Radare2**: An open-source framework for reverse engineering and analyzing binary files.
- **PEiD**: A tool that identifies packers and cryptors used to hide malware.

How It Works: Analysts or automated tools analyze a file's static properties—such as examining the code for known malicious patterns, strings, or signatures—without running it. For example, a file might have encrypted sections or make calls to dangerous APIs. Analysts can detect these issues by carefully inspecting the file structure.

Pros:

- **No risk of execution**: Since the file is never run, there’s no risk of infecting the system during analysis.

- **Detailed code inspection**: Allows for deep inspection of a file's code, helping identify potential threats.

Cons:

- **Cannot detect runtime behavior**: Some sophisticated malware alters its behavior during execution (e.g., polymorphic malware), which static analysis cannot detect.
- **Requires expertise**: It can be time-consuming and complex, requiring skilled analysts to perform effectively.

Best Use Case:

- **Malware Reverse Engineering**: Ideal for analyzing malware samples without executing them, especially when conducting forensic investigations or developing malware signatures.

3.4. Dynamic Analysis

Detection Technique: Dynamic analysis (also called runtime analysis) observes how a file behaves when executed in a controlled environment. It watches for malicious actions such as file encryption, unauthorized data access, or network connections. This method is useful for detecting malware that hides its true nature until it runs.

Tools:

- **Cuckoo Sandbox**: A popular open-source sandbox environment used to execute and analyze malware in isolation.
- **Process Monitor (Sysinternals)**: A tool for real-time system monitoring of file, registry, and process activities.

How It Works: The suspected malware is executed in a sandbox, which is an isolated environment that mimics the target system but does not affect the actual network or system. As the malware runs, dynamic analysis tools log its actions, such as modifying system files, accessing sensitive data, or attempting to communicate with command and control servers.

Pros:

- **Detects runtime behavior**: Can uncover malicious activities that static analysis might miss, such as dynamic code execution or network attacks.

- **Effective against obfuscated malware**: Many malware families attempt to disguise their code, but dynamic analysis can detect their actual behavior during execution.

Cons:

- **Time and resource-intensive**: Running files in a sandbox or dynamic analysis environment requires more system resources and time.
- **Evasion techniques**: Some malware can detect when it's being run in a sandbox and modify its behavior to avoid detection.

Best Use Case:

- **Malware Behavior Analysis**: Ideal for understanding how malware behaves in real-world execution, especially for advanced threats that use polymorphism or code obfuscation.

3.5. Reputation-based Detection

Detection Technique: Reputation-based detection focuses on assessing the trustworthiness of a file or URL by comparing it to a vast database of known malicious and benign files. It uses attributes like file hashes, file size, origin, and digital certificates to score the file's reputation.

Tools:

- **Symantec Insight**: A reputation-based security tool that assesses files based on their prevalence and origin.
- **Norton Safe Web**: Checks URLs for safety by assessing their reputation against known bad sites.

How It Works: A file or URL is checked against a cloud database or community-reported threat intelligence platform. The database contains a history of previous interactions with the file or URL and gives it a reputation score. Files with poor reputation (e.g., those from suspicious origins or rarely seen by other users) are flagged as malicious or suspicious.

Pros:

- **Quick and lightweight**: It doesn't require in-depth analysis of the file's behavior or code and is fast at detecting previously flagged threats.

- **Effective against known threats**: Ideal for environments that require fast detection of well-known threats or phishing sites.

Cons:

- **Ineffective against brand-new malware**: If a file has no history or reputation, it may not be flagged despite being dangerous.
- **Relies on external sources**: Requires an up-to-date and reliable reputation database, often from external sources.

Best Use Case:

- **Web and file downloads protection**: Best for quickly assessing the safety of files or links before allowing user interaction, particularly in enterprise or user-driven environments.

3.6. Hybrid Detection

Detection Technique: Hybrid detection combines two or more techniques to offer comprehensive malware detection. For instance, it may merge signature-based detection with behavior-based monitoring, or static analysis with dynamic analysis. By combining methods, hybrid detection can detect a wider variety of malware, including zero-day threats.

Tools:

- **Palo Alto WildFire**: Combines static and dynamic analysis, as well as behavioral detection, to uncover sophisticated threats.
- **Bitdefender GravityZone**: Employs hybrid techniques like signature-based scanning, heuristic analysis, and behavior monitoring.

How It Works: A hybrid system first attempts to identify malware using quick methods, such as signature-based detection. If the file is unknown or shows suspicious behavior, it will trigger further in-depth analysis using behavioral or dynamic analysis. The system dynamically switches between methods to maximize detection rates and minimize system load.

Pros:

- **High detection accuracy**: Combining multiple techniques allows for both quick detection of known threats and deep analysis of unknown ones.
- **Flexible and adaptive**: It can

Flexible and adaptive: It can adapt its detection method based on the nature of the threat, using different techniques for different malware types.

Cons:

- **Resource-intensive**: Since it may use multiple methods in tandem, hybrid detection can be more demanding in terms of processing power and memory.
- **Complex to implement**: Creating a balanced hybrid detection system requires expertise to ensure the right combination of techniques and minimize false positives or system slowdowns.

Best Use Case:

- **Advanced Threat Protection for Enterprises**: Ideal for businesses that need comprehensive protection against a wide array of threats, including zero-day vulnerabilities, fileless malware, and advanced persistent threats (APTs).

4. Impact Analysis

Malware impacts the organizations and individuals regarding system performance, network integrity, data confidentiality, and user experience. Understanding the impact is essential in developing effective means for prevention and response.

4.1. System-Level Impact

Performance Degradation: Malware hijacks system resources to cause slow system performance. CPU usages increase, malicious processes consume memory, and system responsiveness is reduced.

File Corruption: Most types of malwares, especially viruses and ransomware, often corrupt or destroy critical files, making them unusable or irretrievable sans backups.

System Instability: Malware causes frequent system crashes or forced reboots. These disrupt user activity and business operations.

4.2. Network-Level Impact

Bandwidth Consumption: Worms and some types of malwares replicate and proliferate across the network. It does so by consuming bandwidth, hence reducing overall network performance.

Unauthorized Access: Malware of types like Trojans or backdoors provide unauthorized access to sensitive networks to the attacker, thus enabling them to exfiltrate data, manipulate systems, or further spread infections.

Network Disruption: Most malware variants are designed to conduct DDoS attacks through infecting hundreds of devices to overwhelm target networks, a process often known as botnets.

4.3. Data Impact

Data Theft: Spyware and backdoors steal sensitive information related to finance, personal identity, intellectual property, or trade secrets.

Data Corruption/Deletion: Viruses and worms may cause corruption or permanent deletion of some valuable data. For instance, ransomware encrypts data and then asks for a ransom in exchange for decryption.

Data Privacy Violations: Malware collecting data without consent can lead to severe breaches of data privacy, exposing sensitive personal or business information.

4.4. User Impact

Financial Loss: Ransomware can cause the loss of great financial resources by either paying ransoms or causing data loss and downtime for the user or organization concerned.

Identity Theft: Spyware would normally target personal information to be used in identity theft or fraud. In turn, stolen credentials could be sold in the dark web or be leveraged in unauthorized transactions.

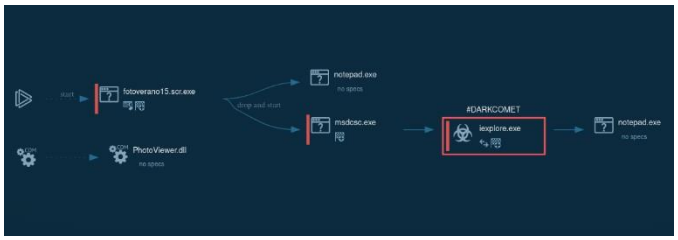
Productivity Loss: System slowdowns and crashes, data unavailability-these are productivity losses, especially in organizations where uptime means business.

5. Case Studies

5.1. Case Study 1: “DarkComet”

Type: Remote Access Trojan (RAT).

Impact: Enabled attackers to gain complete control over the victim's system, allowing for keystroke logging, screenshot capture, and webcam access.



Static Analysis:

- ✓ Using tools like "Strings," we can extract hardcoded URLs or suspicious IP addresses intended for command-and-control (C2).
- ✓ Disassembling with IDA Pro reveals functions used for process manipulation and methods for maintaining persistence through registry changes.

Dynamic Analysis:

- ✓ In a sandbox environment, "DarkComet" connects to its C2 server, transmits system data, and attempts to disable security measures.
- ✓ Analyzing network traffic shows encrypted communication over specific ports.

Consequences: Infected systems were often repurposed into botnets, utilized for launching distributed denial-of-service (DDoS) attacks or spying on users.

5.2. Case Study 2: "CryptoLocker"

Type: Ransomware.

Impact: Encrypted user files, demanding a ransom in Bitcoin for decryption.

Static Analysis:

- ✓ Disassembly indicates the use of robust RSA encryption to lock files.
- ✓ Inspection of the binary shows the creation of a ransom note, typically displayed post-encryption.

Dynamic Analysis:

- ✓ Executing "CryptoLocker" in a virtual environment reveals its immediate scanning of local drives for target file types (.docx, .pdf, etc.).
- ✓ It encrypts files using an RSA public key and contacts a remote server to retrieve the private key.

Consequences: Victims faced limited choices: pay the ransom or restore data from backups, often resulting in significant financial repercussions.

6. Conclusion

Understanding malware types, their various impacts, ways of effective detection, and prevention strategies helps an organization reduce its potential for vulnerability to some specific kinds of cyberattacks and strengthens its cybersecurity posture.

Any organization should be focusing on a multilayered security approach, including routine updates, exceptional antivirus solutions, training employees, and adequate monitoring systems. Besides, the development of a solid incident response and recovery plan means that, should an intrusion occur, damage will be minimal, and operations can return to normal as soon as possible.

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.

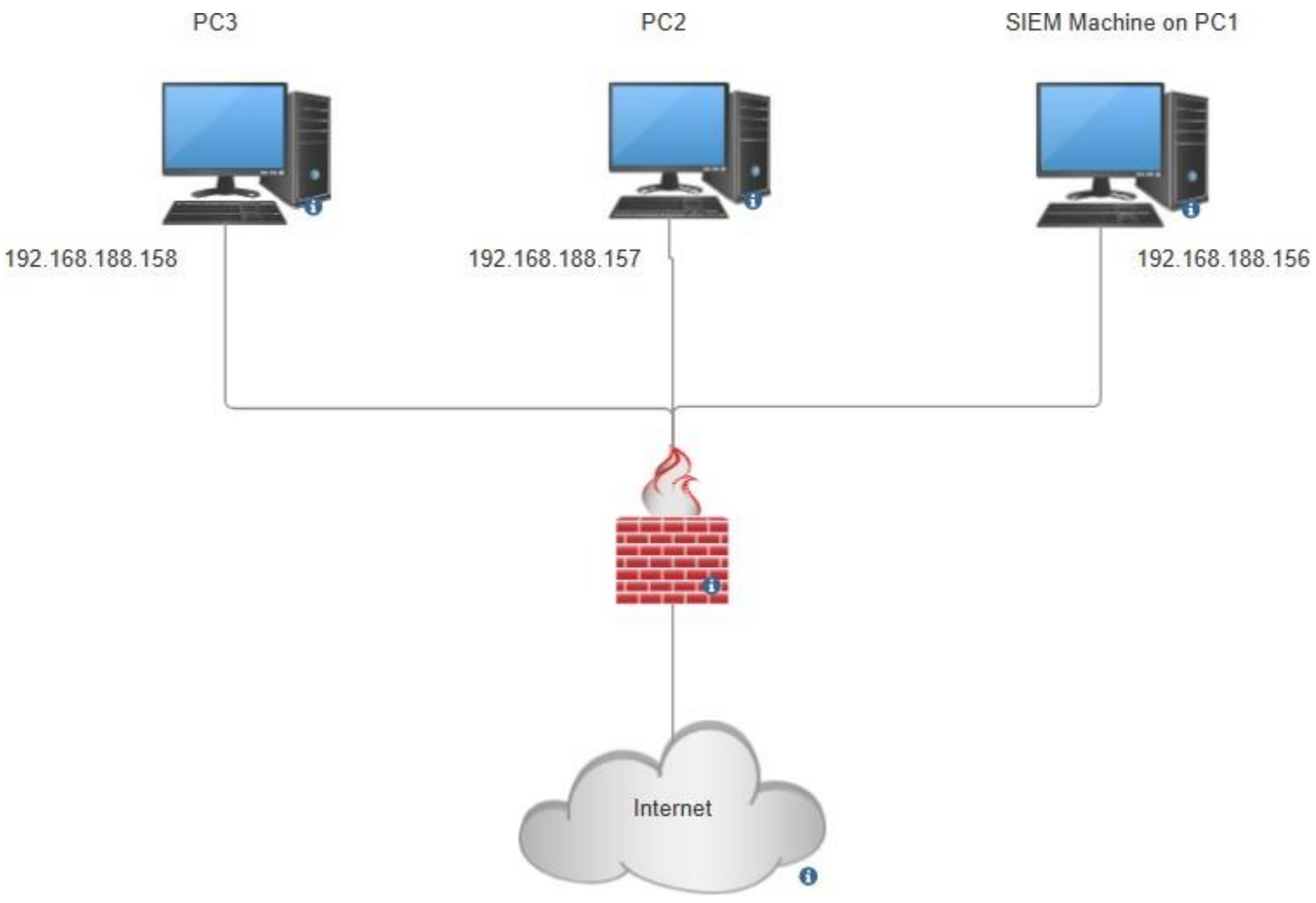


Figure 1:Netwok Overview

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.

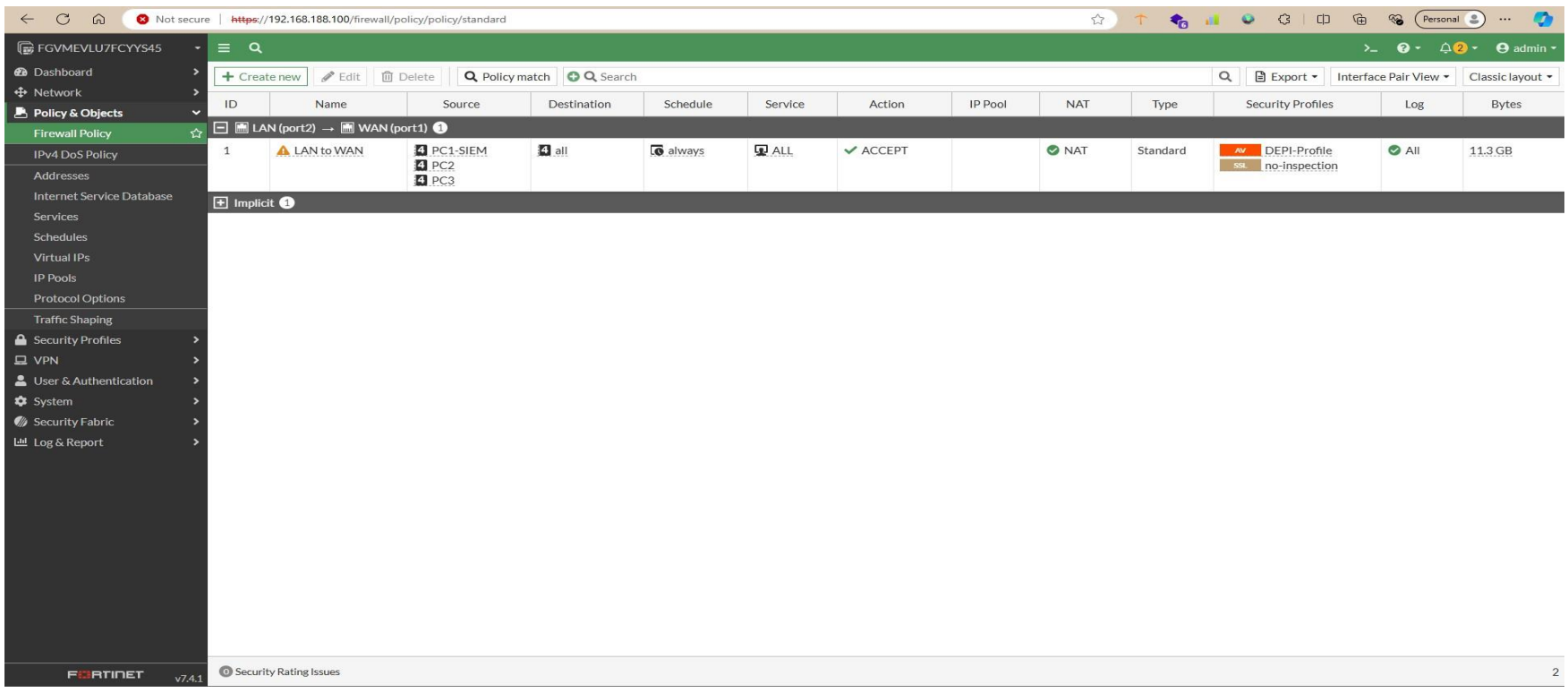


Figure 3: Simple Firewall Policy

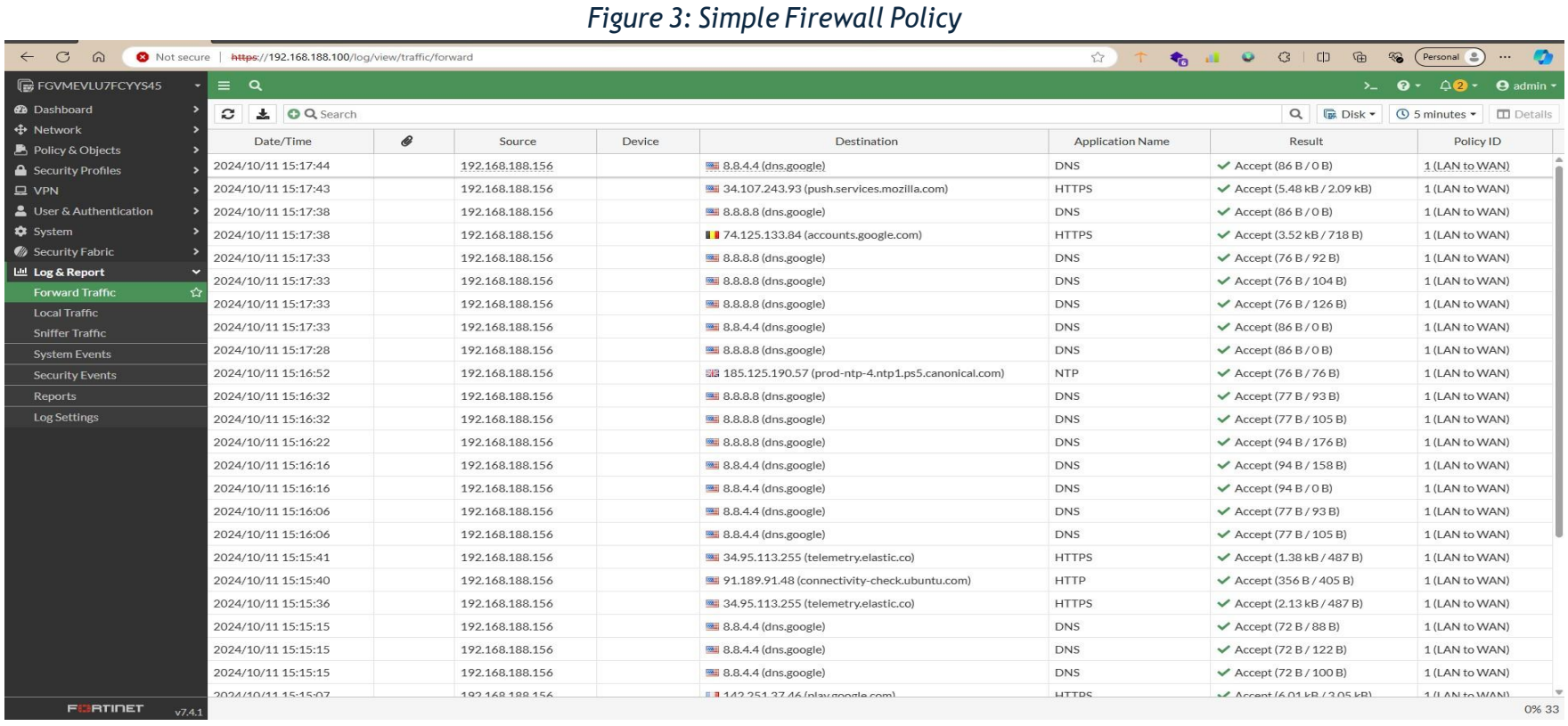


Figure 2: Firewall Logs

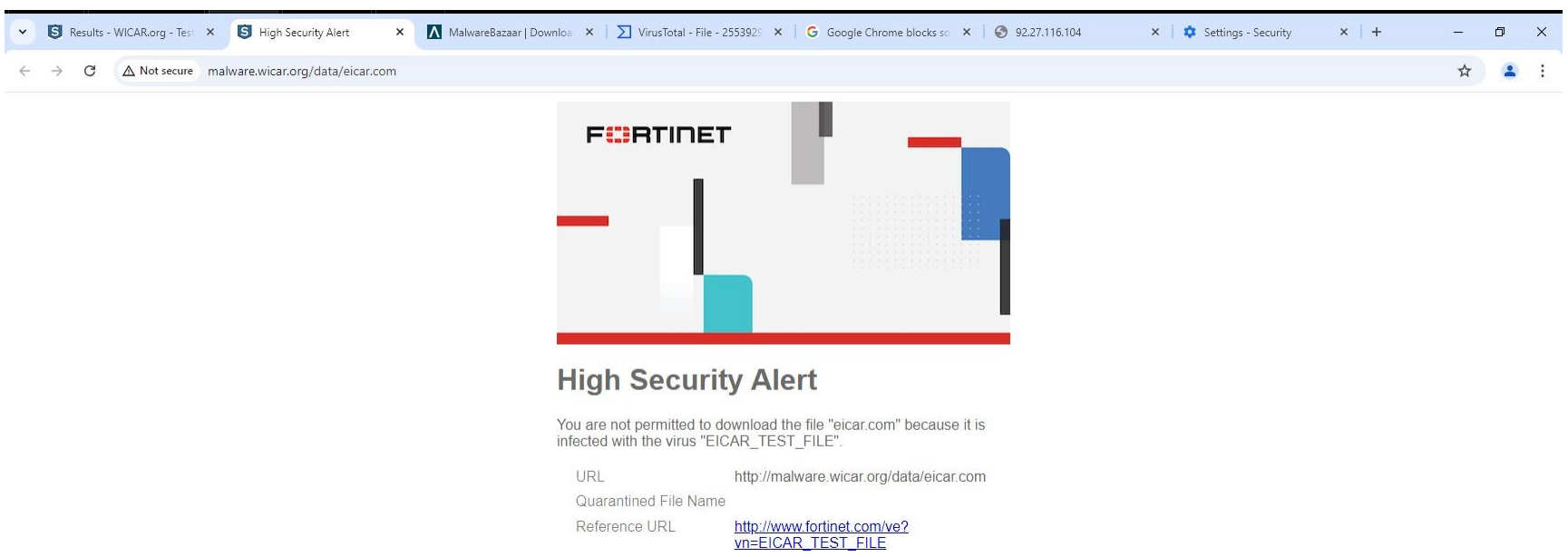


Figure 4: Malware Prevention

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.

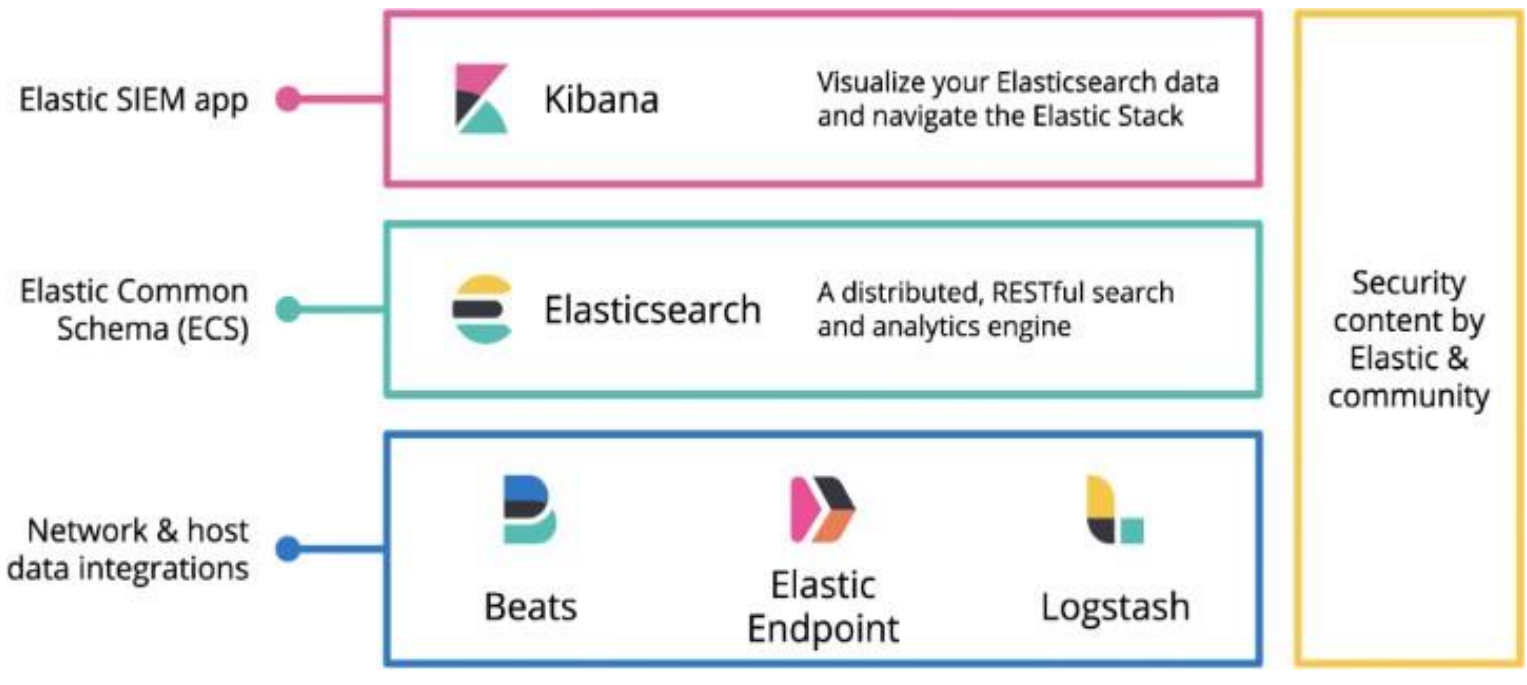


Figure 5: Elastic Stack

Ministry of Communications and Information Technology

Digital Egypt Pioneers

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 6: 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.

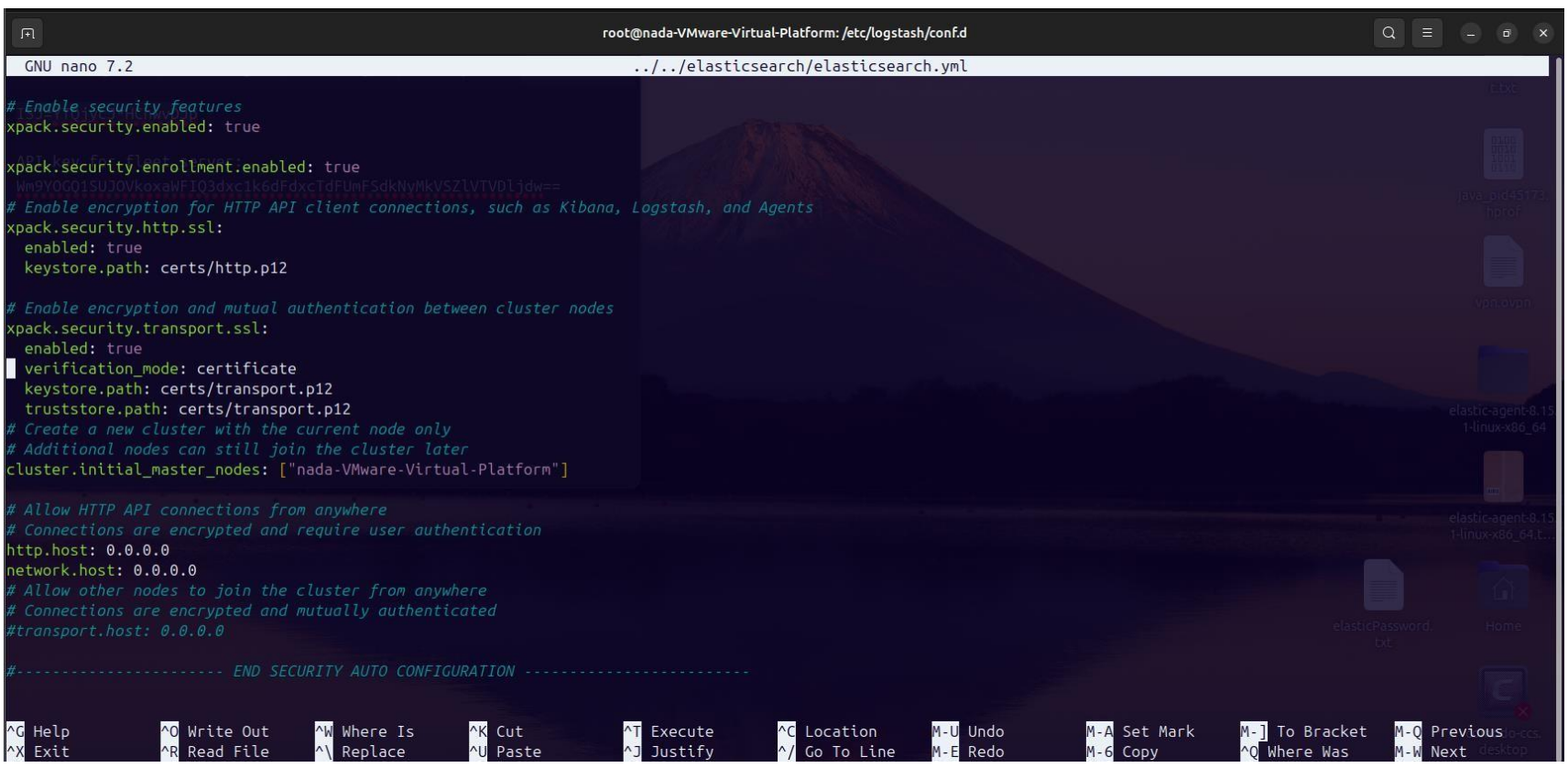


Figure 9: Elasticsearch Configuration File

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.

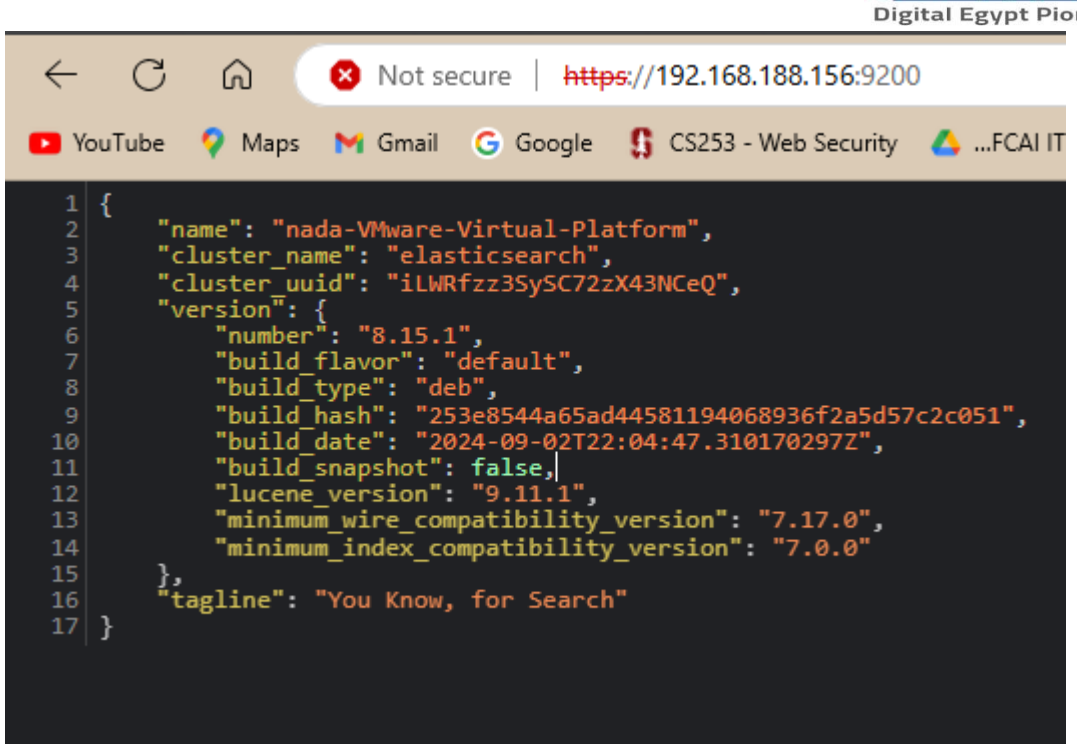


Figure 10: Elasticsearch

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.

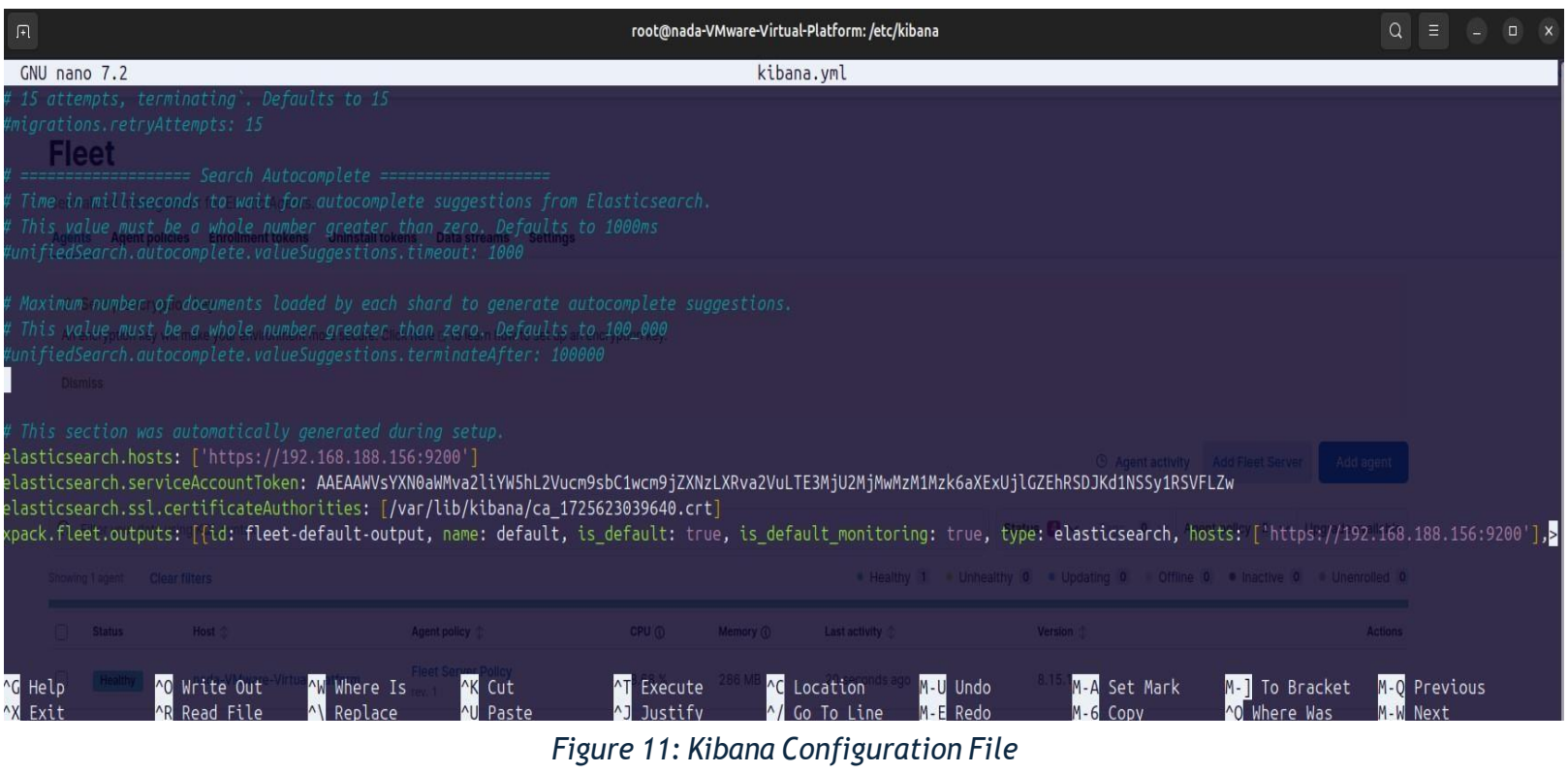


Figure 11: Kibana Configuration File

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

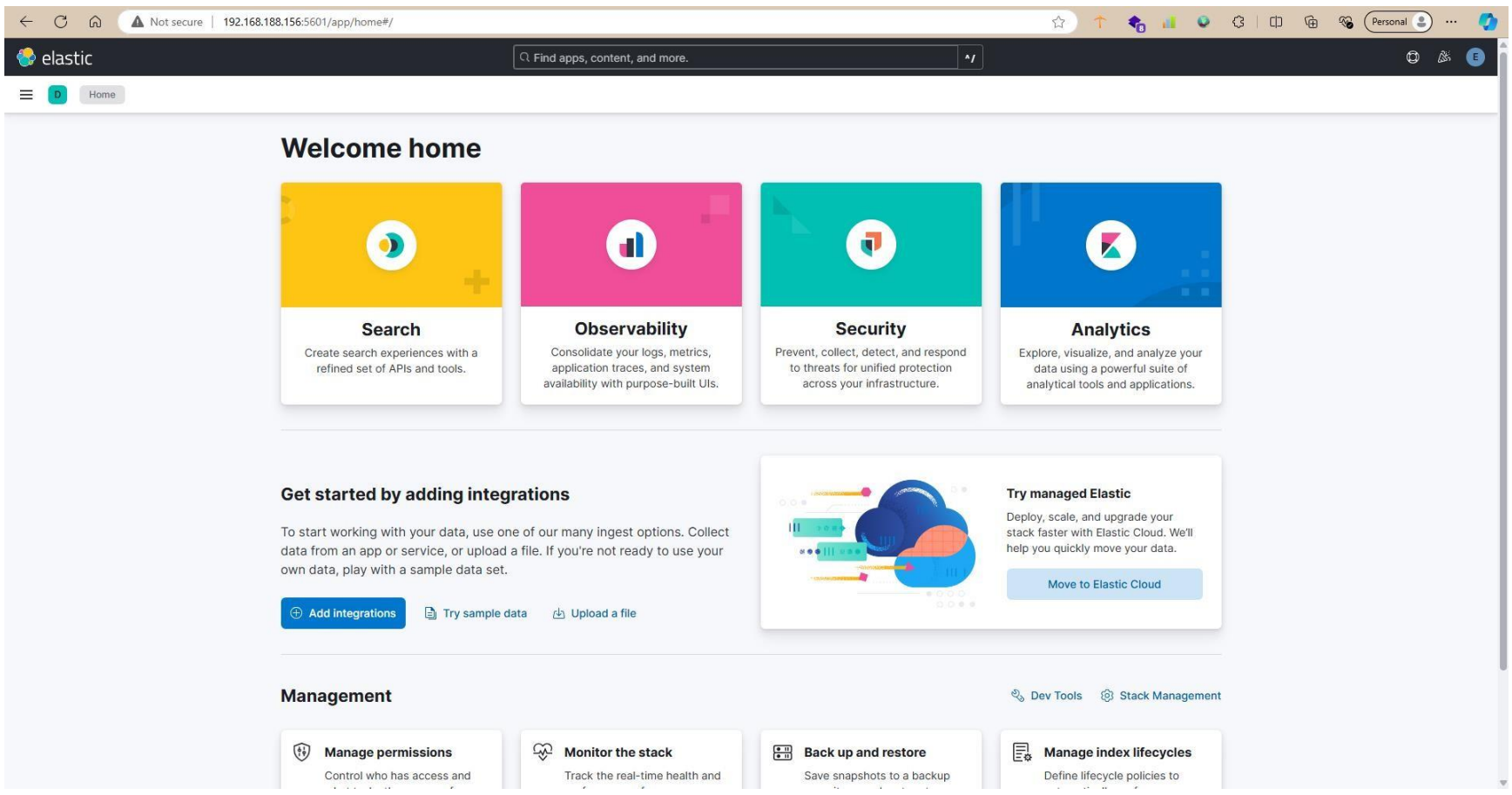


Figure 12: Kibana

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.

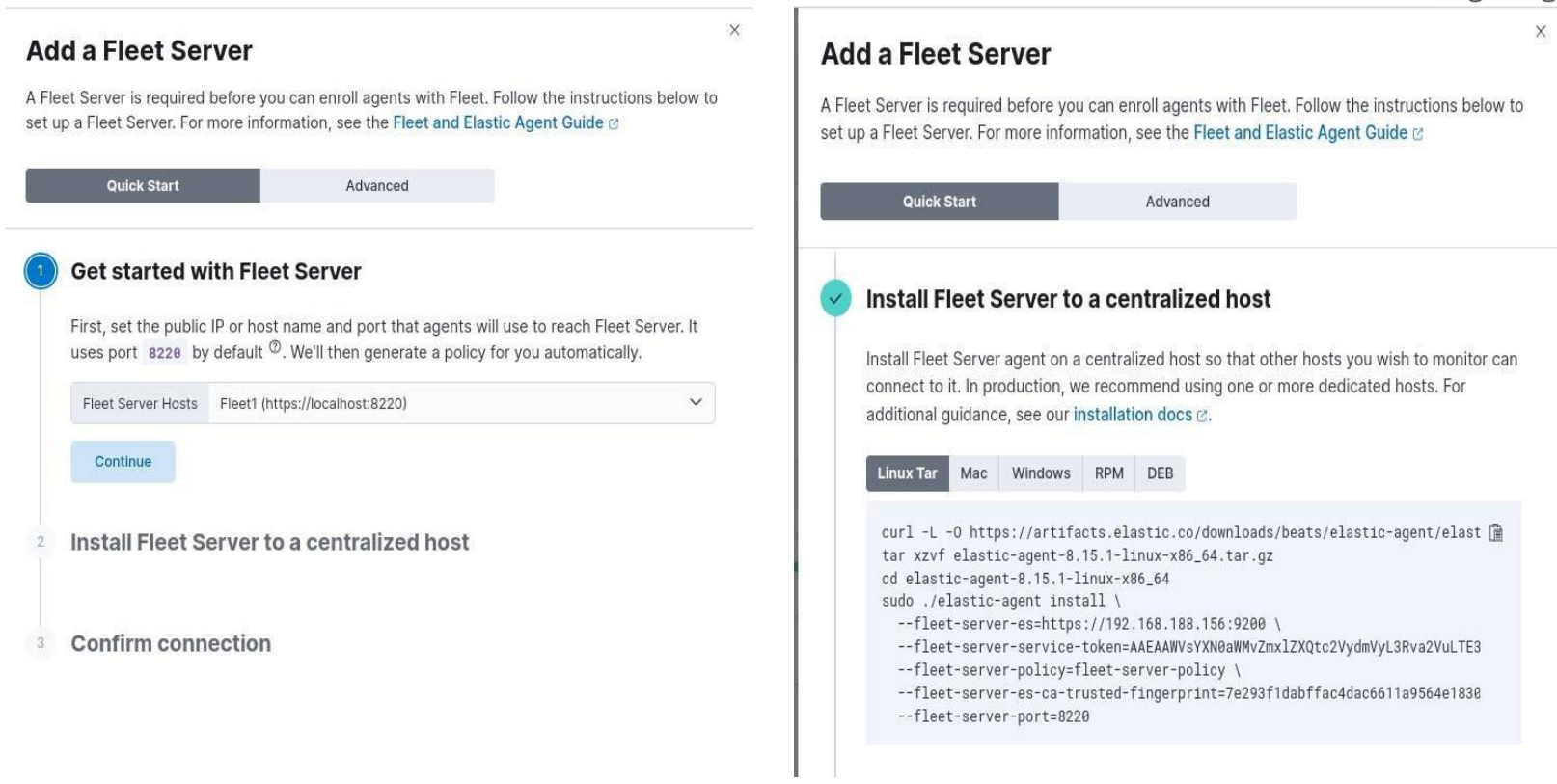


Figure 13: Agent Enrollment

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**.

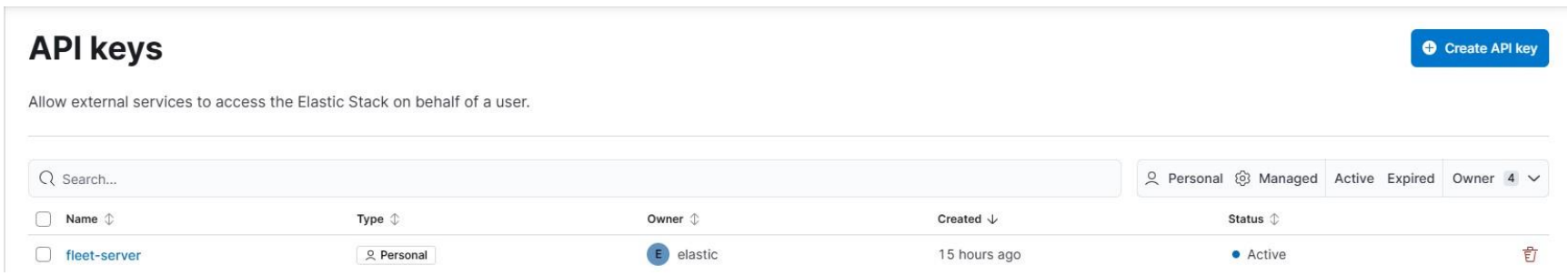


Figure 14: API Key Creation

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

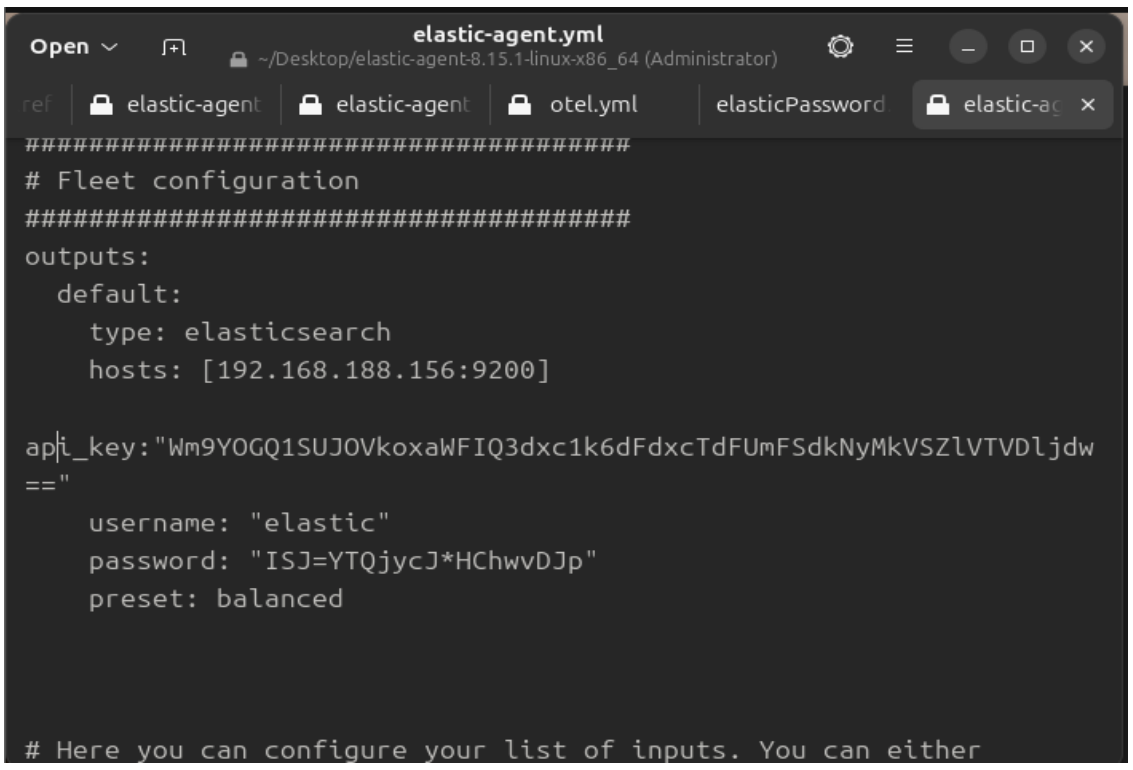


Figure 15:Agent's Configuration File

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 16: Logstash Server

The fleet server now sends the logs to Elasticsearch successfully.

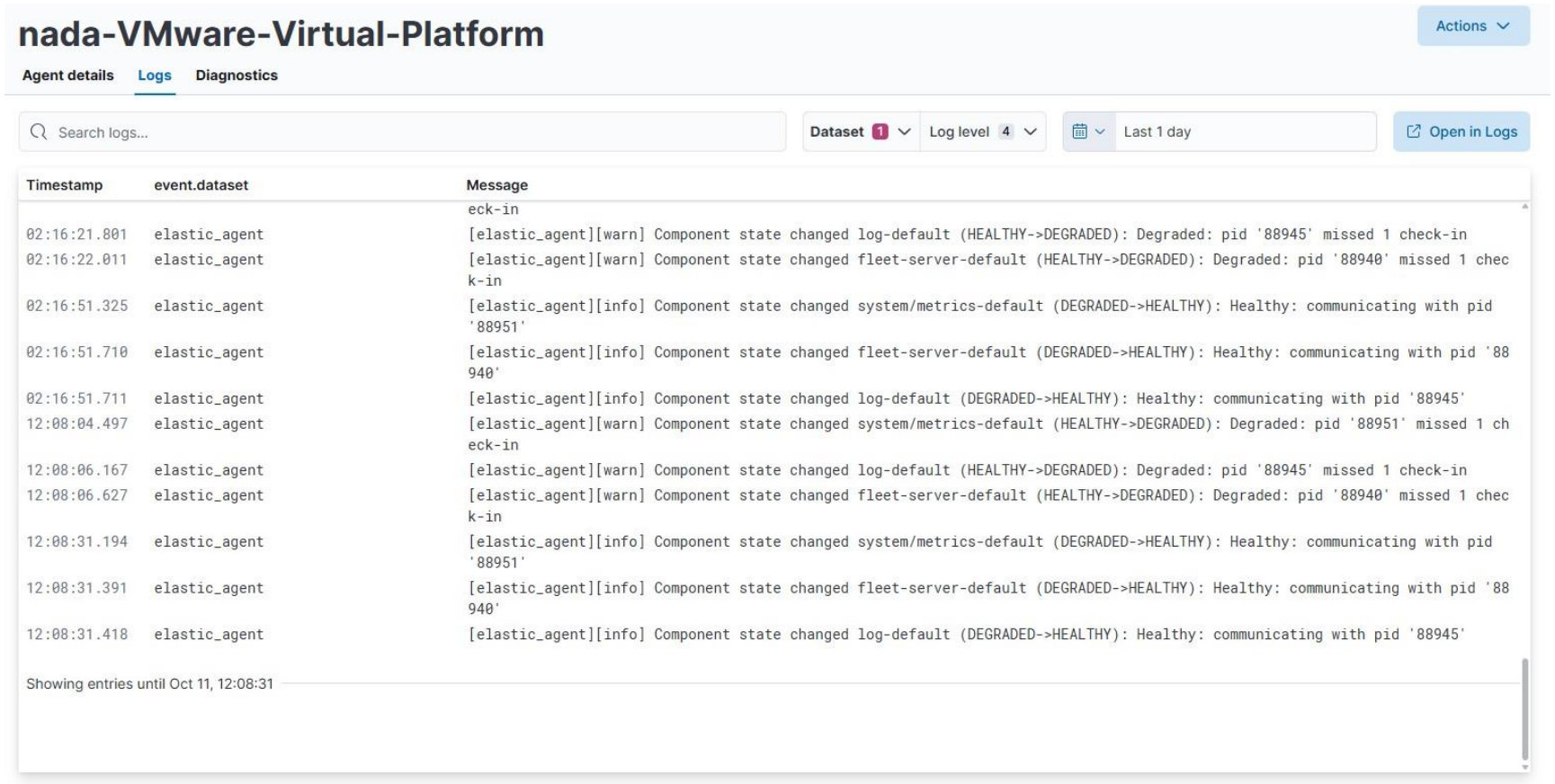


Figure 17: Fleet Agent Logs

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:

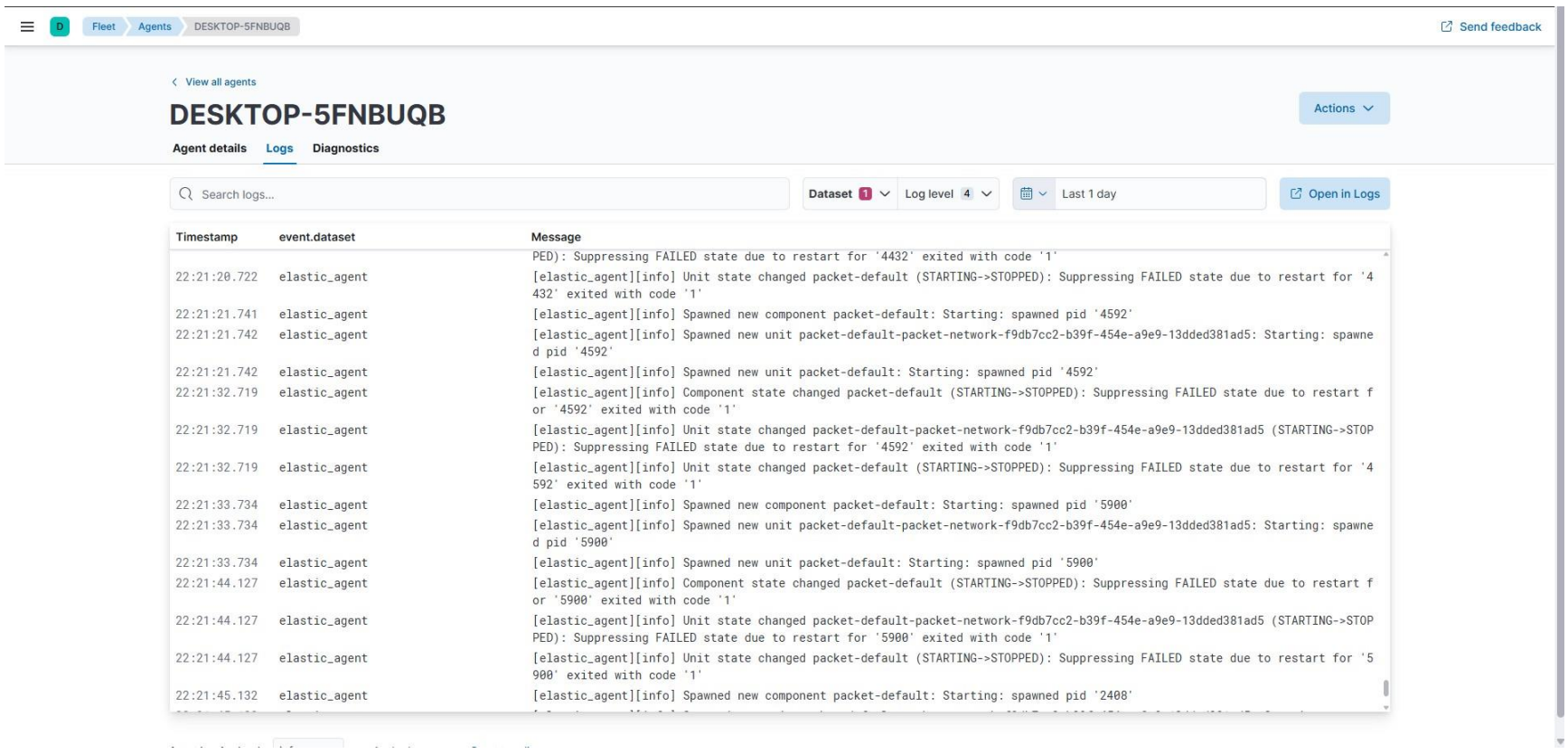


Figure 18: Agent Logs

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.

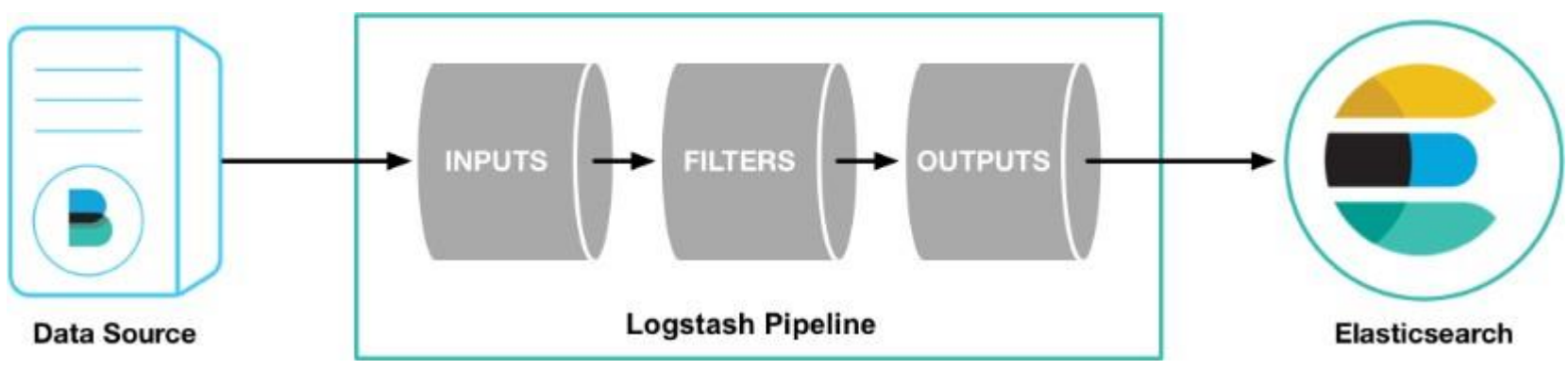


Figure 19: Logstash

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:

```
root@nada-VMware-Virtual-Platform: /etc/logstash/conf.d
 GNU nano 7.2                                                  firewall.conf
input {
  # Syslog input on port 514 for receiving logs
  udp {
    port => 514
    type => "syslog"
  }
}

filter {
  if [type] == "syslog" {
    grok {
    match => {
      "message" => "<%{NUMBER:log_priority}>date=%{TIMESTAMP_ISO8601:log_date} time=%{TIME:log_time} srcip=%{IP:source_ip} srcport=%{NUMBER:source_port} dstip=%{IP:desti>
    }
  }

  # Combine date and time into a single timestamp
    mutate {
      add_field => { "event_timestamp" => "%{log_date} %{log_time}" }
  }

  # Convert event_timestamp to a proper date field
    date {
      match => ["event_timestamp", "yyyy-MM-dd HH:mm:ss"]
      target => "@timestamp"  # Use the parsed timestamp
  }

  # Keep only the specified fields
    mutate {
      keep_fields => ["@timestamp", "source_ip", "source_port", "destination_ip", "destination_port", "source_country", "destination_country", "action", "service", "url"]
  }

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket   M-Q Previous
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was     M-W Next
```

Figure 21: firewall configuration file

```
root@nada-VMware-Virtual-Platform: /etc/logstash/conf.d
 GNU nano 7.2                                                  firewall.conf
    keep_fields => ["@timestamp", "source_ip", "source_port", "destination_ip", "destination_port", "source_country", "destination_country", "action", "service", "url"]
  }
}

  }
 }

   date {
     match => [ "timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
     timezone => "UTC"
   }
  }
}

output {
  # Output to Elasticsearch with SSL and ignoring certificate verification
  elasticsearch {
    hosts => ["https://192.168.188.156:9200"]    # Use https if Elasticsearch has TLS enabled
    index => "my_index-%{+YYYY.MM.dd}"
    user => "elastic"    # Replace with your Elasticsearch username
    password => "ISJ=YTQjycJ*HChwvDJp" # Replace with your Elasticsearch password
    ssl => true                          # Enable SSL (for https)
    ssl_certificate_verification => false   # Disable SSL certificate verification only
  }

  # Output to stdout for debugging purposes
  stdout {
    codec => rubydebug
  }
}

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket   M-Q Previous
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was     M-W Next
```

Figure 20: Output of firewall Conf file

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

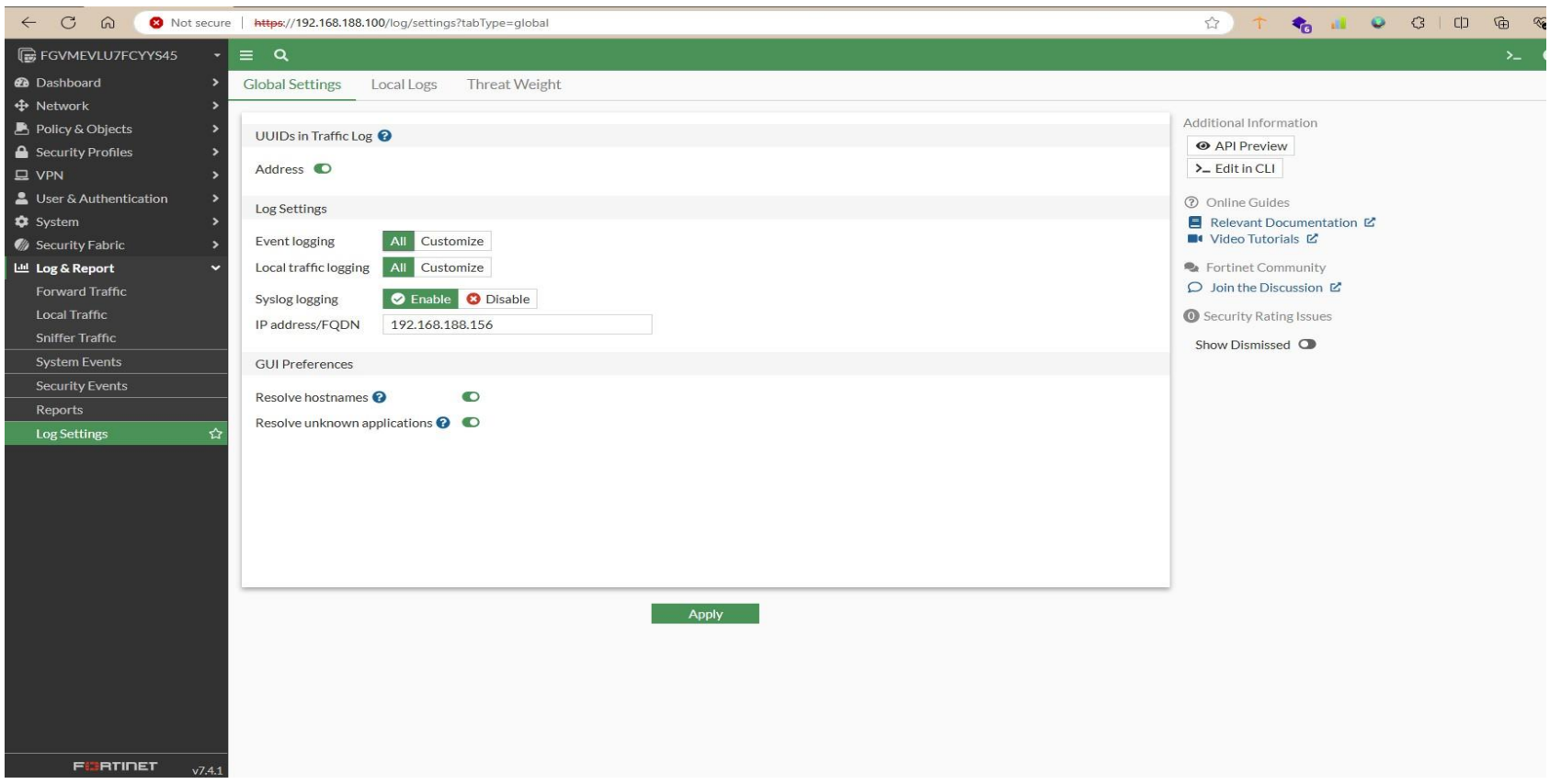


Figure 22:Pushing logs

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

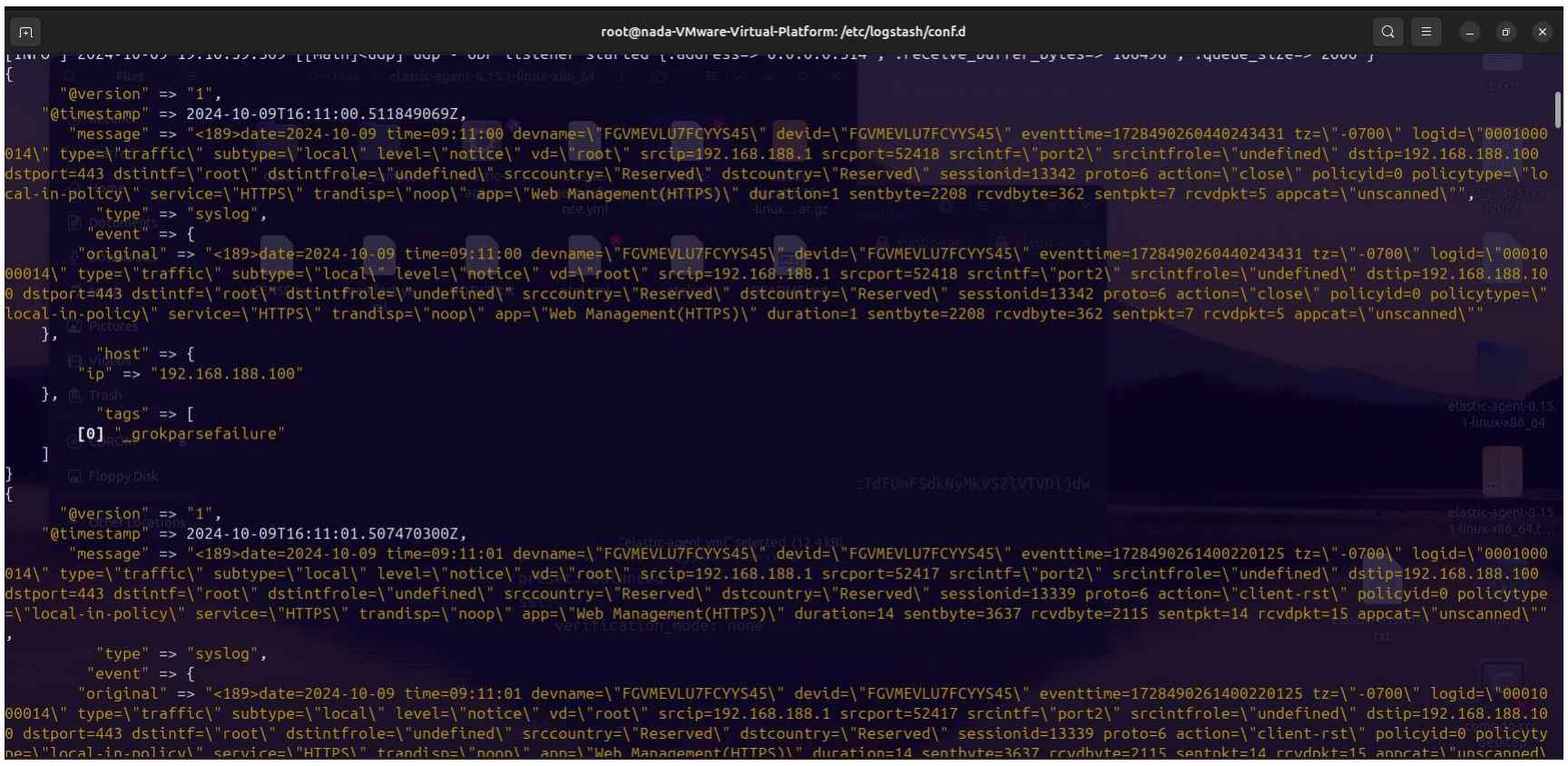


Figure 23: Debuging firewall.conf

the logs correctly.

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

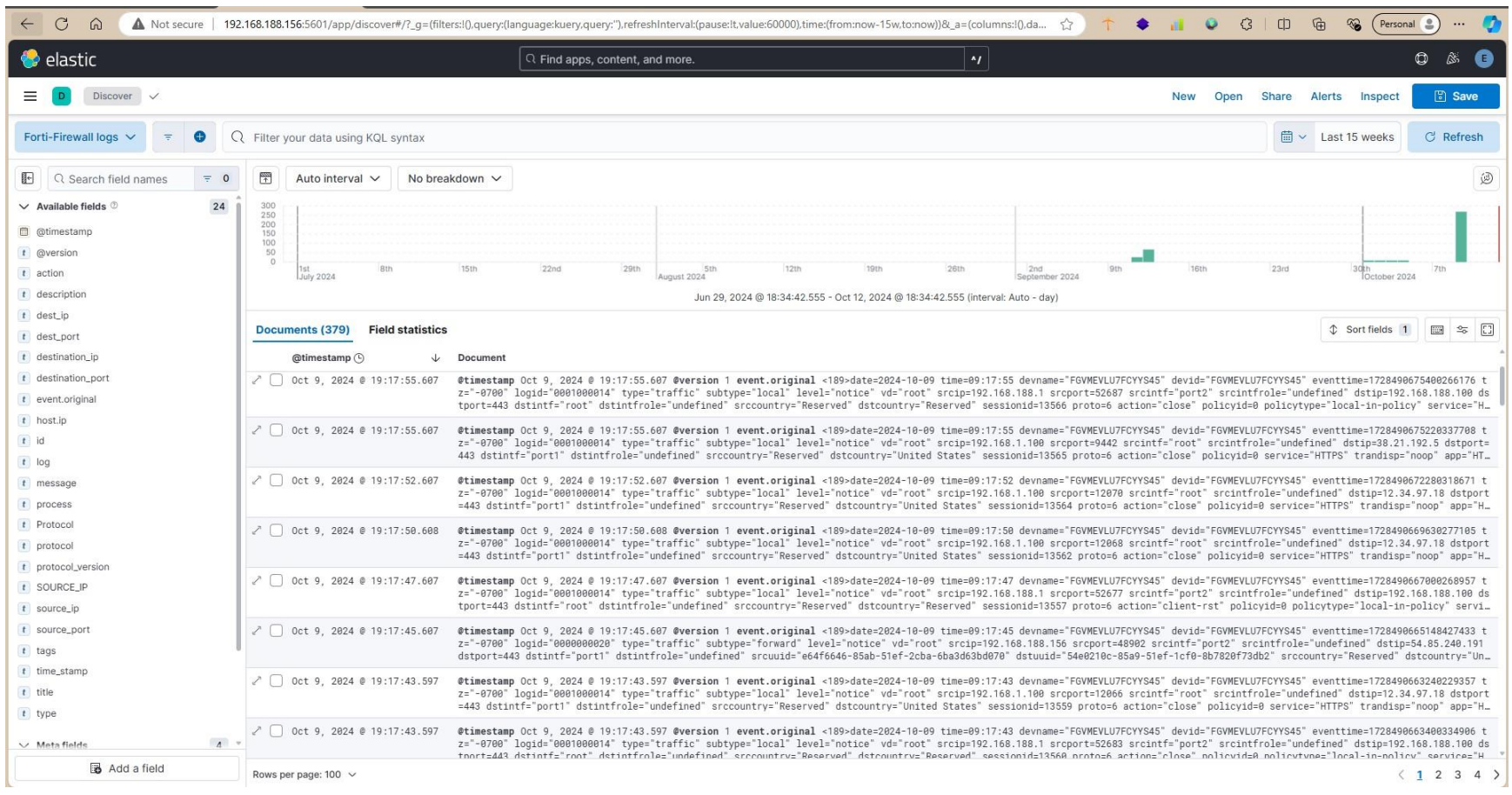


Figure 24: Firewall logs

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.

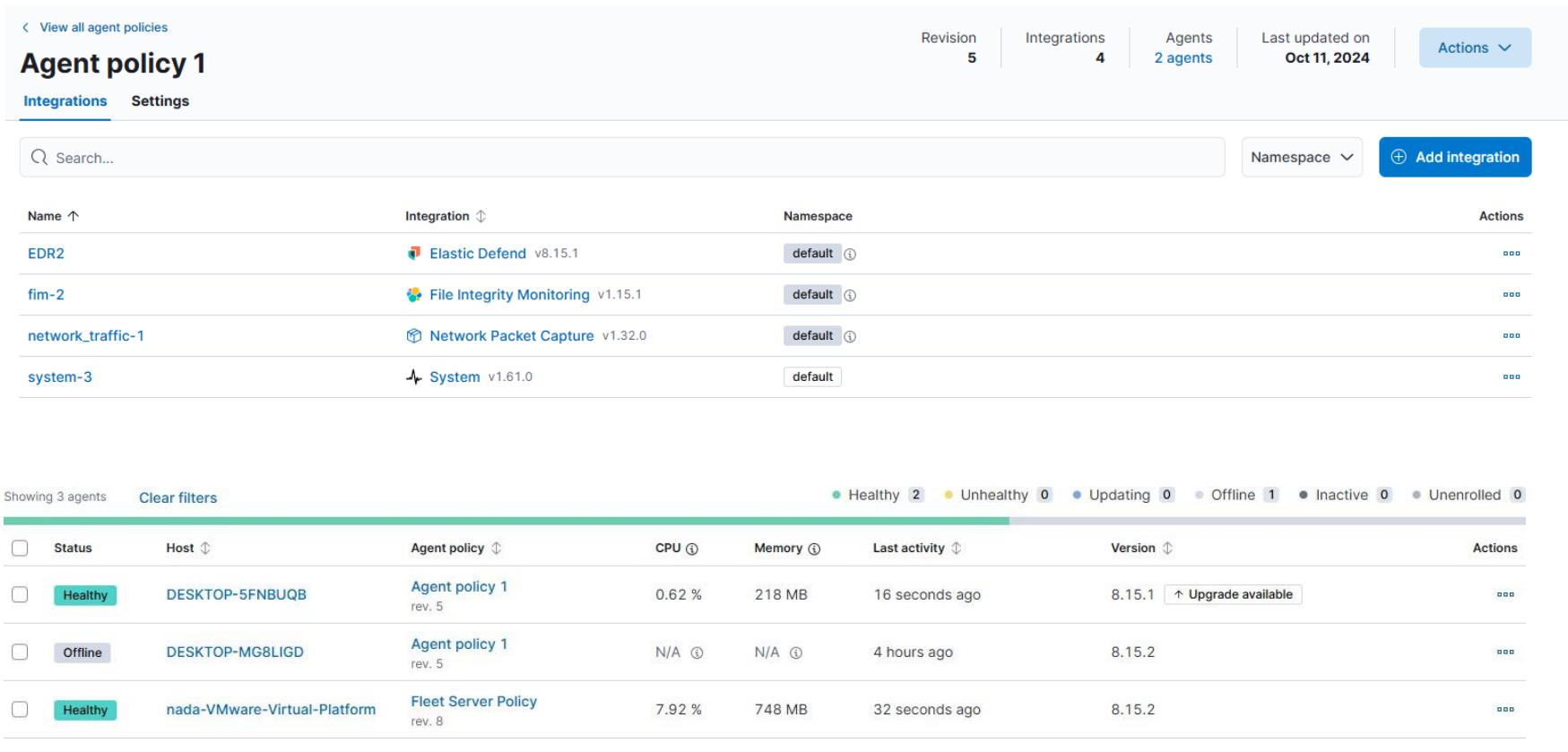


Figure 25: Agents and policy

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 ATT&CK 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.

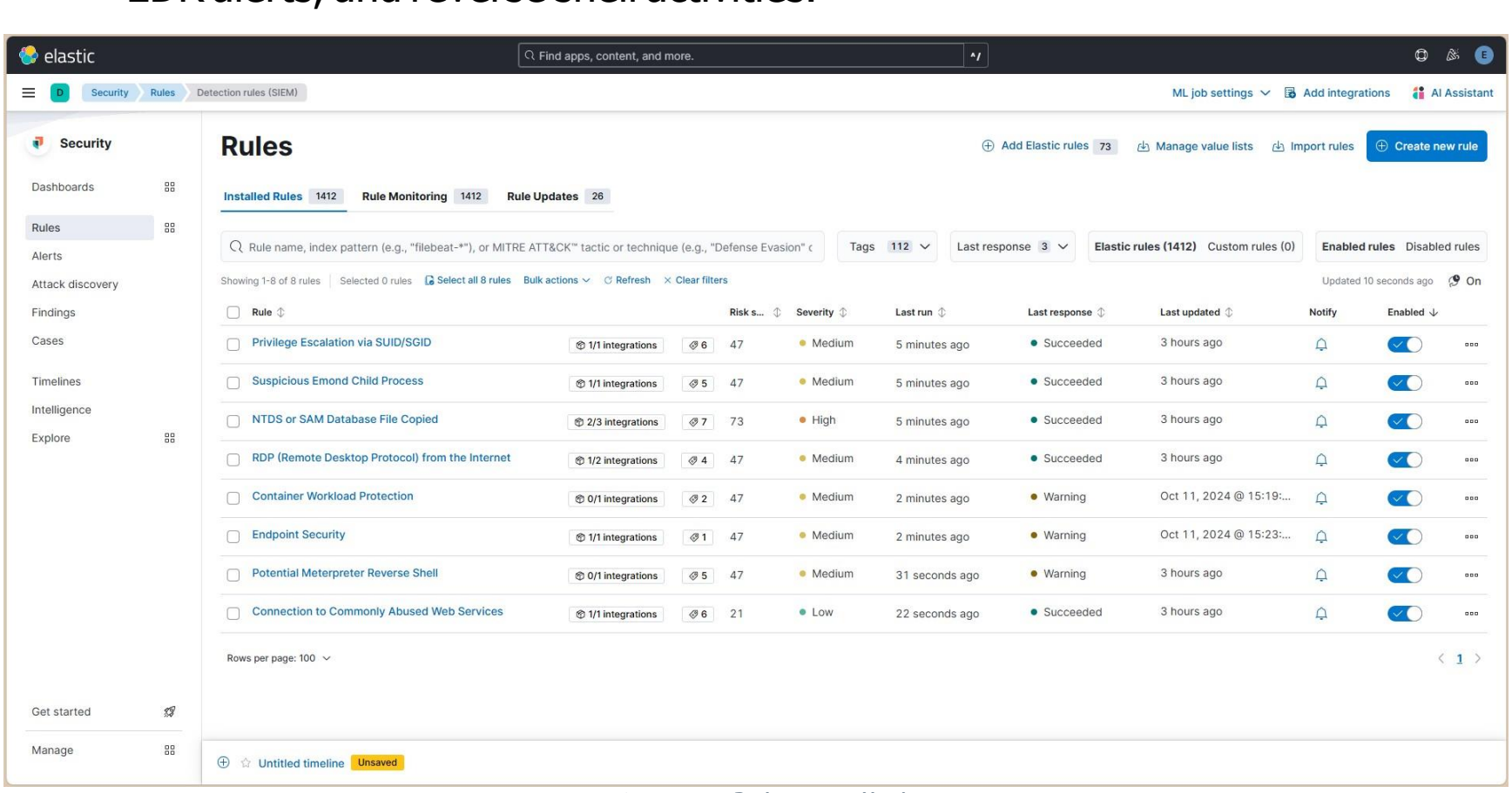


Figure 26: Rules Installed

Some Techniques covered in MITRE ATT&CK:

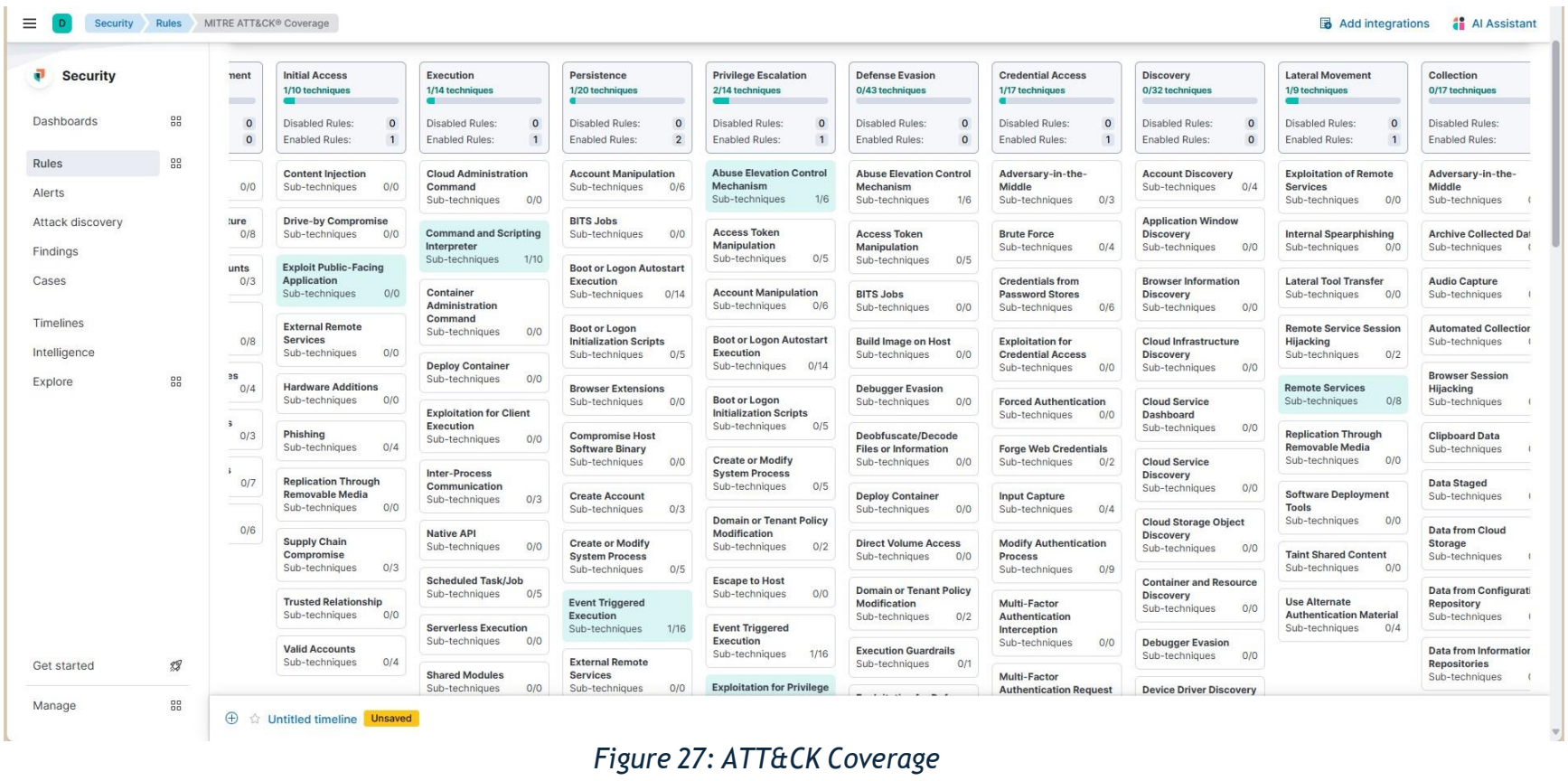


Figure 27: ATT&CK Coverage

To validate the rule, we installed a malicious ZIP file with the hash value **cdde99520664ac313d43964620019c61** and subsequently extracted its contents. Prior to opening the file, we observed its presence.

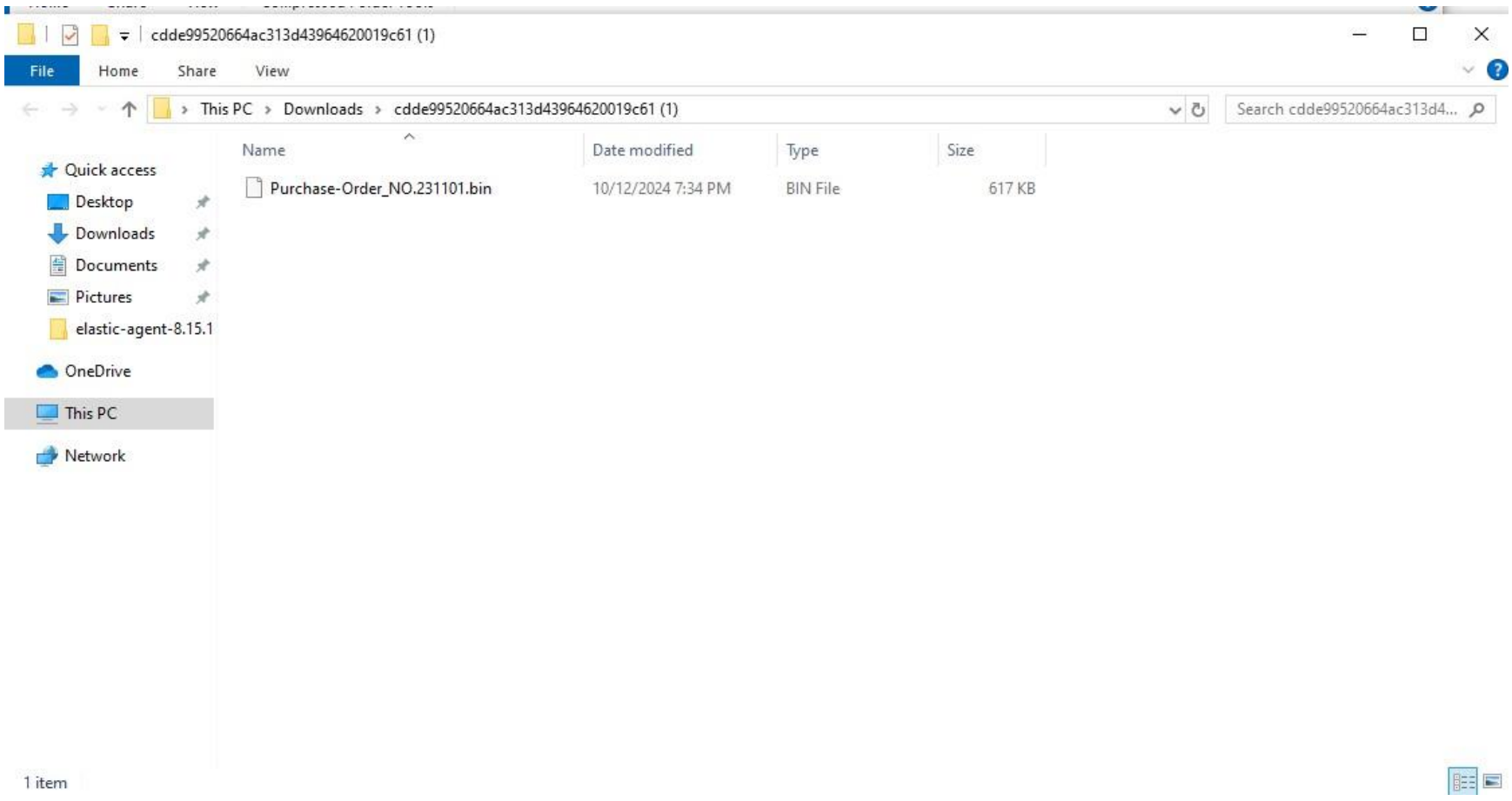


Figure 28: Malware

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

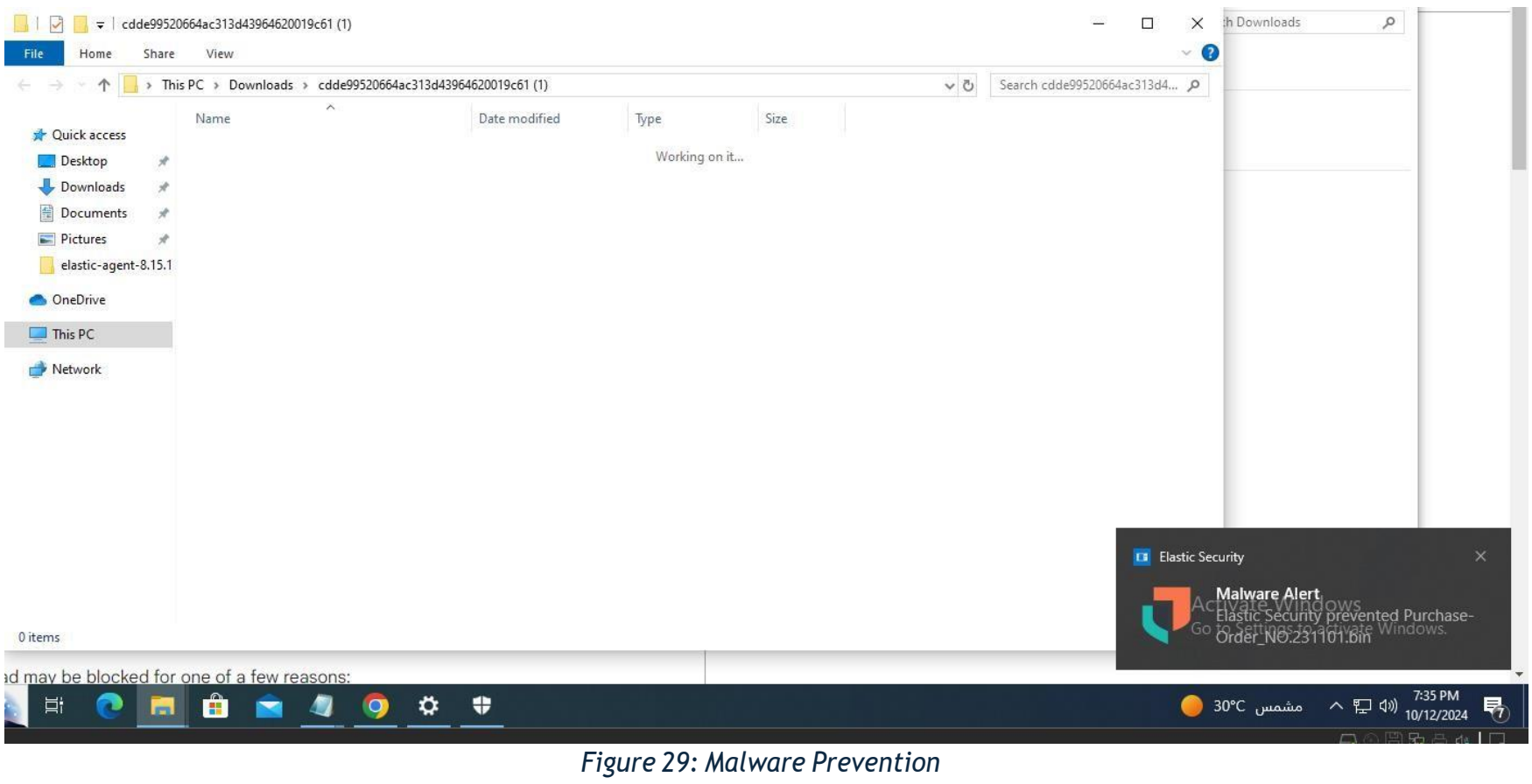


Figure 29: Malware Prevention

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

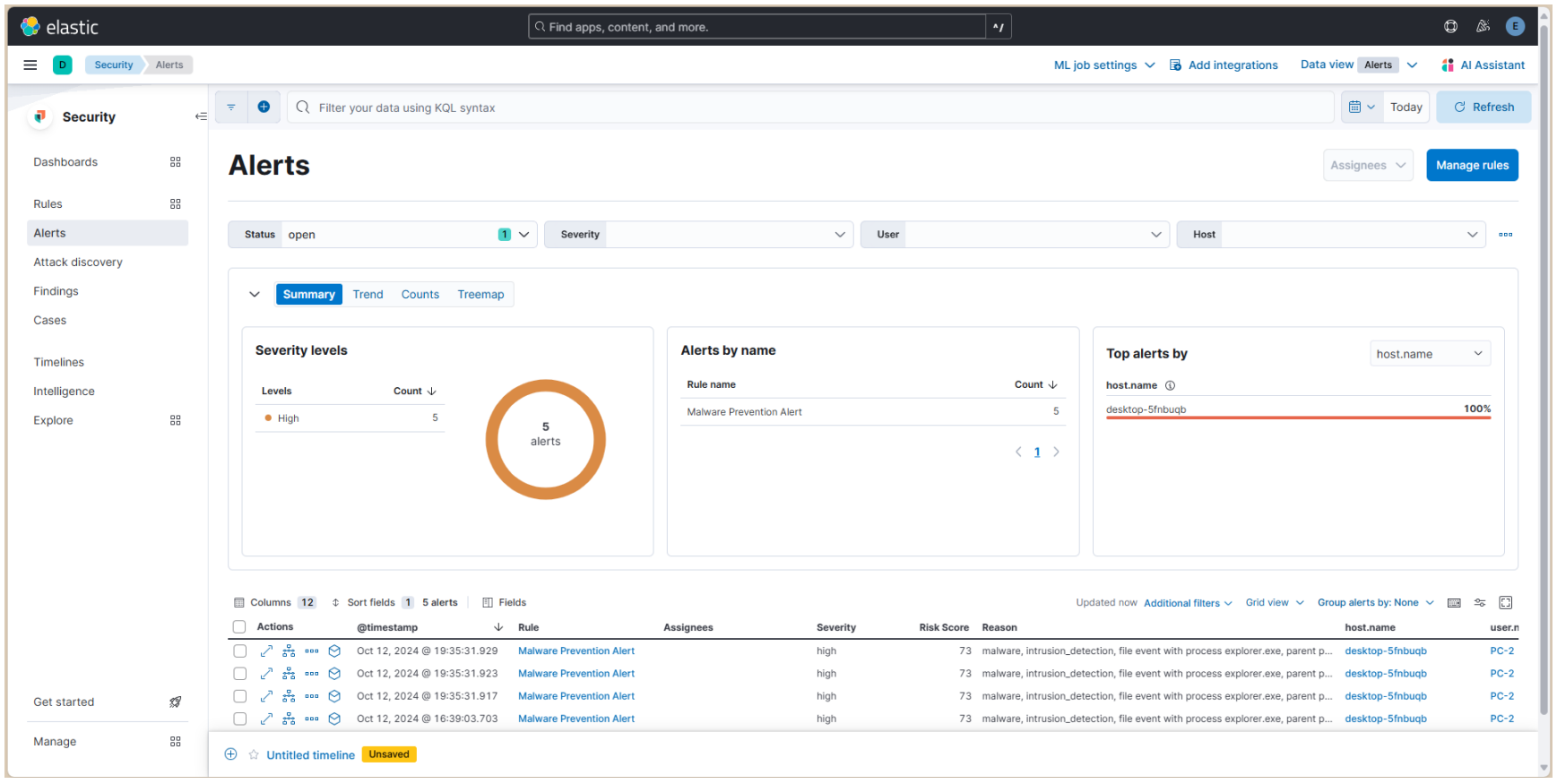


Figure 30: SIEM Alerts

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.

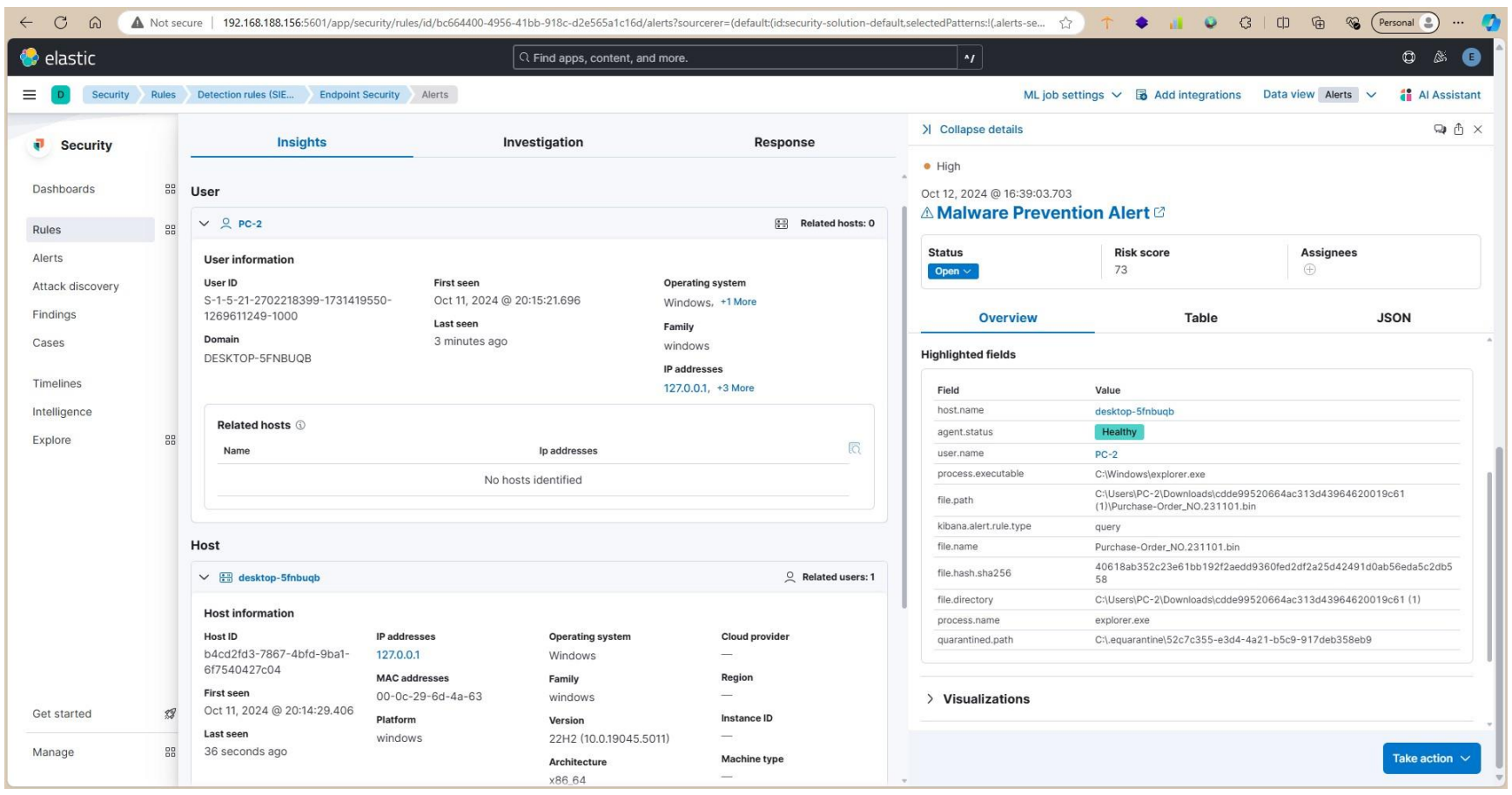


Figure 31: Alert Details

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 ATT&CK 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.

Comprehensive Malware Prevention Strategy

Introduction:

Malware prevention is an essential aspect of cybersecurity, aiming to stop malicious software from infiltrating and compromising networks, systems, and data. This strategy provides a layered approach, ensuring security across endpoints, networks, user behavior, and systems. By addressing prevention at multiple levels, this plan minimizes risk and enhances the ability to detect and mitigate threats effectively.

Key Pillars of the Strategy:

- Patch Management and Vulnerability Scanning
- Endpoint Protection
- Network Segmentation
- Perimeter Defense (Firewalls and Intrusion Prevention)
- Secure Email Gateway
- User Awareness and Training
- Access Control and Privilege Management
- Data Backup and Recovery
- Security Information and Event Management (SIEM)
- Incident Response Plan

---

1. Patch Management and Vulnerability Scanning

Objective:

To ensure that systems are up-to-date and vulnerabilities are identified and addressed proactively.

Detailed Plan:

1. **Asset Inventory**:
   Create an accurate and up-to-date inventory of all systems and software within the network. This includes:
   - **Operating Systems** (Windows, Linux, macOS)
   - **Software Applications** (Enterprise software, custom applications, etc.)
   - **Firmware** (Routers, switches, and other network appliances)
   - **Third-party libraries or open-source components**

2. **Patch Management Schedule**:
   Establish a patching cadence:
   - **Critical Systems**: Apply security patches within 24-48 hours of release.
   - **Non-Critical Systems**: Weekly patching or based on risk assessment.
   - **Automated Patching**: Use patch management solutions to automate patch application.
3. **Patch Testing**:
   Before deploying patches across production systems, establish a controlled testing environment:
   - Create a **sandbox environment** that mirrors production.
   - Test all new patches here to ensure no business-critical functions are affected.
4. **Vulnerability Scanning**:
   Conduct regular vulnerability scans to detect and address system weaknesses:
   - **Weekly internal vulnerability scans**: For identifying security gaps in workstations, servers, and network devices.
   - **Monthly external vulnerability scans**: Focus on publicly accessible systems such as web servers, VPNs, and firewalls.
5. **Reporting and Prioritization**:
   - Review scan results and prioritize vulnerabilities based on severity.
   - Apply the **CVSS** (Common Vulnerability Scoring System) to rate and address critical issues first.

Recommended Tools:

- **Patch Management**: Microsoft SCCM (System Center Configuration Manager)
- **Vulnerability Scanning**: QualysGuard, Nessus (Tenable)

Best Practices:

- Establish automated alerts for missed patches or failed scans.
- Maintain detailed documentation of patch deployments and vulnerability remediation activities.

2. Endpoint Protection

Objective:

To prevent malware from infecting individual endpoints (laptops, desktops, servers) by deploying advanced security solutions.

Detailed Plan:

1. **Antivirus and Antimalware Deployment**:
   Implement endpoint protection software across all devices:
   - **Real-time protection**: Ensure that endpoint protection is active 24/7, scanning all files in real-time.
   - **Scheduled full scans**: Set a weekly full system scan on all devices to detect dormant threats.
   - Ensure **automatic updating** of malware signatures to protect against the latest threats.
2. **Behavioral Monitoring**:
   Enable behavioral analytics within your endpoint protection solution:
   - Monitor processes for suspicious activity (e.g., unexpected file encryption or unusual network connections).
   - Configure alerts for processes exhibiting malware-like behavior.
3. **Application Whitelisting/Blacklisting**:
   Limit the execution of unauthorized or untrusted software:
   - **Whitelisting**: Allow only pre-approved applications to run on endpoints. Unauthorized apps are blocked by default.
   - **Blacklisting**: Continuously update the blacklist to block known malicious apps.
4. **Advanced EDR Solutions (Endpoint Detection & Response)**:
   Implement EDR solutions to enhance visibility and incident response capabilities:
   - **Threat Hunting**: Use EDR to actively search for malware or suspicious activities within the network.
   - **Automated Response**: Automatically isolate infected endpoints or kill malicious processes upon detection.

Recommended Tools:

- **Antivirus/Antimalware**: Symantec Endpoint Protection (SEP), McAfee Total Protection
- **Advanced EDR**: CrowdStrike Falcon, Carbon Black

Best Practices:

- Perform **daily health checks** on endpoint protection solutions to ensure they are running smoothly.
- Create a **centralized dashboard** for real-time monitoring of all endpoint protection activities.

3. Network Segmentation

Objective:

To limit the spread of malware and enhance security by isolating critical network zones and implementing strict traffic controls between them.

Detailed Plan:

1. **Network Segmentation Design**:
    - **Classify network zones**:
        - Critical infrastructure (databases, file servers)
        - User networks (endpoints, workstations)
        - Guest networks (visitors, IoT devices)
    - Isolate each zone using firewalls or VLANs to prevent lateral movement in the event of malware infection.
2. **Access Control Implementation**:
    - **Network Access Control (NAC)**: Ensure that only authorized devices are allowed to access specific network segments.
    - **ACLs (Access Control Lists)**: Create rules to restrict communication between different segments. For example:
        - Only allow workstations to access the file server on specific ports.
        - Deny guest devices from accessing sensitive internal networks.
3. **Firewall Rule Configuration**:
   Configure firewalls with strict traffic rules:
    - **Whitelist essential traffic**: Only allow necessary traffic between network segments (e.g., file sharing, web access).
    - **Deny all other traffic**: Block any non-essential communication, especially between high-risk zones (e.g., guest network to internal network).
4. **Micro-Segmentation for Critical Assets**:
   Use micro-segmentation to isolate specific applications or workloads, reducing their attack surface:

- Implement **software-defined networking (SDN)** to create fine-grained segmentation policies.
- For example, isolate individual VMs or containers based on their role and restrict their network access accordingly.

Recommended Tools:

- **Network Firewalls**: Cisco Firepower, Palo Alto Networks NGFW
- **Software-Defined Networking (SDN)**: VMware NSX

Best Practices:

- Regularly review segmentation policies and firewall rules to ensure they are still relevant and secure.
- Perform internal penetration tests to validate the effectiveness of network segmentation and access controls.

4. Perimeter Defense (Firewalls and Intrusion Prevention)

Objective:

To monitor and control inbound and outbound traffic, protecting the network perimeter from malware threats and other intrusions.

Detailed Plan:

1. **Next-Generation Firewall (NGFW) Setup**:
   - **Deep Packet Inspection (DPI)**: Enable DPI on all perimeter firewalls to inspect traffic for malware or suspicious payloads.
   - **Geo-blocking**: Restrict traffic from regions or countries with high malware activity, as per global threat intelligence.
2. **Intrusion Detection/Prevention Systems (IDS/IPS)**:
   - **Intrusion Detection**: Configure IDS systems to monitor and detect unusual behavior in network traffic.
   - **Intrusion Prevention**: Set IPS to block known malicious traffic using pre-configured signatures and heuristics.
3. **Sandboxing**:
   Enable sandboxing features to inspect and analyze suspicious attachments or executables before allowing them into the network.
   - Sandbox suspicious files in an isolated environment to observe their behavior before releasing them to end-users.

4. **Security Logging and Monitoring**:
    - Set up comprehensive logging for all firewall and IDS/IPS activities.
    - Regularly review logs for anomalies and correlate with your SIEM solution (discussed below).

Recommended Tools:

- **Firewalls**: Fortinet FortiGate, Palo Alto Networks NGFW
- **IDS/IPS**: Snort, Suricata

Best Practices:

- Keep IDS/IPS signatures updated regularly to detect the latest threats.
- Conduct **regular firewall audits** to ensure that all rules and configurations adhere to security policies.

5. Secure Email Gateway

Objective:

To prevent malware and phishing threats from entering the organization through email channels.

Detailed Plan:

1. **Email Filtering**:
   Deploy a secure email gateway to scan all inbound and outbound emails for malware and phishing:
    - Block attachments with executable content (.exe, .js).
    - Use **heuristic scanning** to identify malicious attachments and links.
    - Enable **advanced threat protection (ATP)** to detect zero-day malware.
2. **URL and Link Analysis**:
   Automatically scan email URLs and rewrite them to pass through a secure web proxy:
    - Block known phishing sites and prevent users from clicking on malicious links.
    - Use real-time scanning to identify new, previously unknown malicious URLs.
3. **Attachment Sandboxing**:
    - Use email sandboxing to open and analyze attachments in an isolated environment before they reach the recipient.
    - This helps in detecting malware embedded in documents (like malicious macros).

4. **Phishing Simulation and Reporting**:
   Enable a one-click reporting feature in email clients for users to flag suspicious emails.
   - Use this data to refine email filtering rules and improve employee awareness.

Recommended Tools:

- **Email Security Gateway**: Proofpoint, Barracuda Email Security Gateway
- **Advanced Threat Protection**: Microsoft Defender for Office 365

Best Practices:

- Regularly review email filtering logs and adjust policies to block newly identified threats.
- Train users on how to recognize phishing and malicious emails (discussed in the training section below).

6. User Awareness and Training

Objective:

To educate users on how to identify and avoid malware, phishing attacks, and other threats.

Detailed Plan:

1. **Phishing Simulations**:
   Conduct regular phishing simulations to test users' ability to recognize malicious emails:
   - Track metrics on who falls for phishing attempts and provide them with targeted training.
   - Run at least quarterly phishing tests and increase complexity over time (e.g., using spear-phishing scenarios).
2. **Regular Security Training**:
   Organize mandatory security awareness sessions to educate employees on the following:
   - Safe internet browsing practices.
   - Identifying suspicious emails or websites.
   - Safe use of external media (e.g., USB drives).

Deliver this training in various formats (workshops, webinars, quizzes) and tailor it for different departments (e.g., Finance, IT).

3. **Security Awareness Campaigns**:
   - Launch an internal awareness campaign with posters, intranet articles, and newsletters.
   - Provide ongoing tips on recognizing malware, ransomware, and phishing threats.

Recommended Tools:

- **Phishing Simulation**: KnowBe4, Cofense PhishMe
- **User Training**: Wombat Security

Best Practices:

- Use metrics from phishing simulations to measure improvement in user behavior over time.
- Continuously update training materials to reflect the latest threats.

7. Access Control and Privilege Management

Objective:

To prevent unauthorized users from accessing critical systems and data, minimizing the risk of malware spread due to compromised accounts.

Detailed Plan:

1. **Principle of Least Privilege (PoLP)**:
    - Ensure that all users, processes, and systems are granted the minimum access necessary to perform their tasks.
    - Regularly review and adjust user privileges to ensure no excessive access is granted.
2. **Multi-Factor Authentication (MFA)**:
    - Implement MFA for all critical systems, especially for administrators and privileged users.
    - Use MFA for remote access systems, VPNs, and email to prevent unauthorized access due to stolen credentials.
3. **Privileged Access Management (PAM)**:
    - Implement PAM solutions to monitor and control privileged accounts.
    - Use session recording for critical administrative activities, providing an audit trail for accountability.

Recommended Tools:

- **MFA**: Duo Security, Microsoft Authenticator
- **Privileged Access Management**: CyberArk, BeyondTrust

Best Practices:

- Conduct regular audits of user privileges to identify and remove any excess permissions.

- Monitor all privileged account activities using a centralized dashboard for tracking.

8. Data Backup and Recovery

Objective:

To ensure that data is securely backed up and can be restored quickly in the event of a ransomware or malware attack.

Detailed Plan:

1. **Backup Strategy**:
    - Implement **daily incremental backups** and **weekly full backups** for critical systems and data.
    - Ensure backups are stored in a **secure offsite location** (either in the cloud or physically).
    - Use **air-gapped** backups to protect against ransomware that targets backup systems.
2. **Immutable Backups**:
    - Implement backup systems that create **immutable copies** of your data. These copies cannot be altered or deleted, even by administrators.
    - Use **versioning** to maintain multiple copies of backup files, allowing recovery from a point-in-time before malware infection occurred.
3. **Disaster Recovery Plan**:
    - Regularly test the recovery of critical data from backups.
    - Simulate disaster scenarios (ransomware, system crashes) and test the time required to restore data.

Recommended Tools:

- **Backup Solutions**: Veeam Backup & Replication, Acronis Cyber Backup
- **Cloud Backup**: AWS Backup, Azure Backup

Best Practices:

- Follow the **3-2-1 backup rule**: 3 copies of data, 2 different storage types, 1 offsite backup.
- Encrypt all backup data to ensure its integrity and confidentiality.

9. Security Information and Event Management

(SIEM) Objective:

To provide centralized monitoring and alerting for all security events across the network, allowing for real-time detection of potential malware threats.

Detailed Plan:

1. **SIEM Deployment**:
    - Integrate all security devices (firewalls, IDS/IPS, endpoint protection, network traffic monitors) with your SIEM platform.
    - Configure log collection and correlation rules to identify suspicious activities or malware indicators.
    - Set up **real-time alerts** for critical security events such as malware detection, unusual login attempts, or unauthorized file access.
2. **Threat Intelligence Integration**:
    - Integrate external threat intelligence feeds into your SIEM to enrich detection capabilities.
    - Automatically correlate events with known malware indicators from threat intelligence databases.
3. **Incident Response Automation**:
    - Use **SOAR (Security Orchestration, Automation, and Response)** features to automatically respond to certain incidents (e.g., isolating compromised endpoints, blocking malicious IP addresses).

Recommended Tools:

- **SIEM**: Splunk, IBM QRadar
- **SOAR**: Palo Alto Cortex XSOAR

Best Practices:

- Regularly review and update SIEM detection rules to account for emerging threats.
- Use SIEM-generated reports to conduct post-incident analysis and improve defenses.

10. Incident Response

Plan Objective:

To provide a structured approach for handling malware incidents quickly and effectively to minimize damage and recover systems.

Detailed Plan:

1. **Incident Detection and Analysis**:
    - Use SIEM and EDR alerts to detect malware incidents in real-time.
    - Analyze the scope and impact of the malware using automated threat intelligence and incident response tools.
2. **Containment and Eradication**:
    - **Contain the spread**: Isolate infected systems from the network to prevent further malware propagation.
    - **Remove malware**: Use endpoint protection and specialized tools to clean the malware from infected systems.
3. **Recovery**:
    - Restore infected systems from **clean, recent backups**.
    - Verify system integrity after the malware has been eradicated and before bringing systems back online.
4. **Post-Incident Review**:
    - Conduct a full incident analysis to determine the root cause of the malware infection.
    - Implement additional prevention measures based on lessons learned from the incident.

Recommended Tools:

- **Incident Response**: Cisco AMP, CrowdStrike Falcon
- **Forensics Tools**: EnCase, FTK Imager

Best Practices:

- Update incident response playbooks regularly to reflect new malware tactics and attack vectors.
- Train the incident response team on how to handle various types of malware threats (ransomware, Trojans, worms).

User Awareness Materials:

1. Videos:

   - ✓ How To Stay Safe Online: https://drive.google.com/file/d/1VSShNDMsSfxaBvpCG8Znz_nth1aOLs6f/view?usp=drive_link

   - ✓ Alex's Phishing Story: https://drive.google.com/file/d/1y13DyEbGxQQaTuqRuSZ55gKNxVBB8vvw/view?usp=drive_link

2. Infographics:

   - ✓ Malware at a Glance: https://infograph.venngage.com/pl/qITivENl1CE

   - ✓ How To Protect Yourself Against Malwares: https://infograph.venngage.com/pl/aJS0fI7w0HA

   - ✓ Guard Steps Against Malwares: https://infograph.venngage.com/pl/AT6vQyN5g14

3. Awareness Game:

   - ✓ Scape Room: A Cyber Game Waiting For You! https://infograph.venngage.com/pl/rgho56aaOyA

4. Presentation:

   - ✓ Presentation Link: https://app.presentations.ai/view/8GmwjI