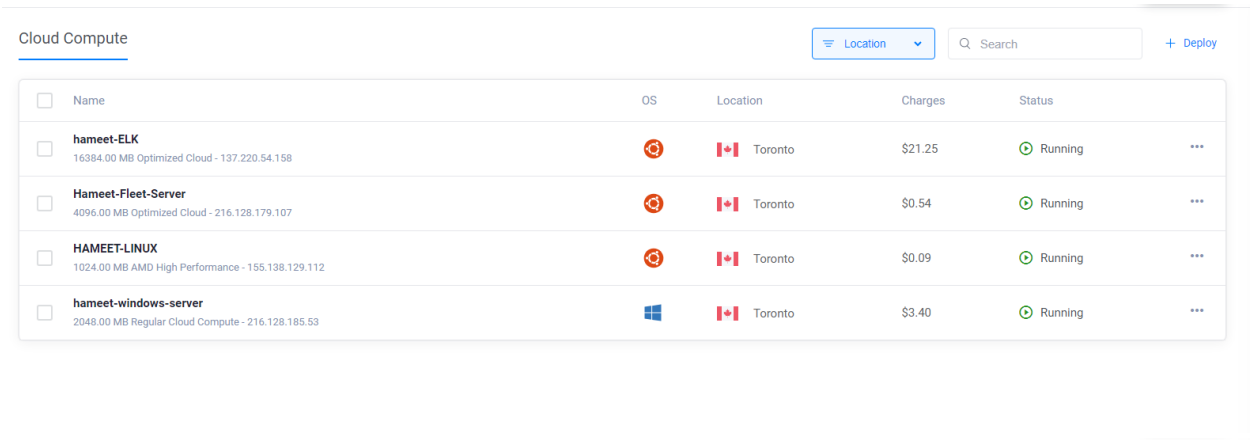
SOC Analyst Simulation Project

Hameet Benipal

The SOC Analyst Simulation Project provided practical, hands-on experience replicating the core responsibilities of a Security Operations Center (SOC) analyst. This project involved setting up a virtualized environment comprising multiple Linux and Windows VMs, including an ELK stack for log collection and analysis, a fleet server for telemetry management, Kali Linux for conducting simulated attacks, and a Mythic server to execute command-and-control (C2) operations. Using tools like Crowbar for brute-force SSH and RDP attacks and Mythic agents for C2 attacks, the environment was configured to mimic real-world cyber threats. Elastic SIEM was leveraged to monitor and detect these attacks through custom detection rules, with logs and telemetry analyzed and visualized using Elastic dashboards. To simulate real-world incident response workflows, OSTicket was integrated with Elastic to automatically generate incident tickets for detected alerts, enabling the assignment, tracking, and resolution of security incidents. This project provided valuable experience in security monitoring, log analysis, incident response, and cybersecurity automation, aligning closely with the daily tasks of a modern SOC analyst.

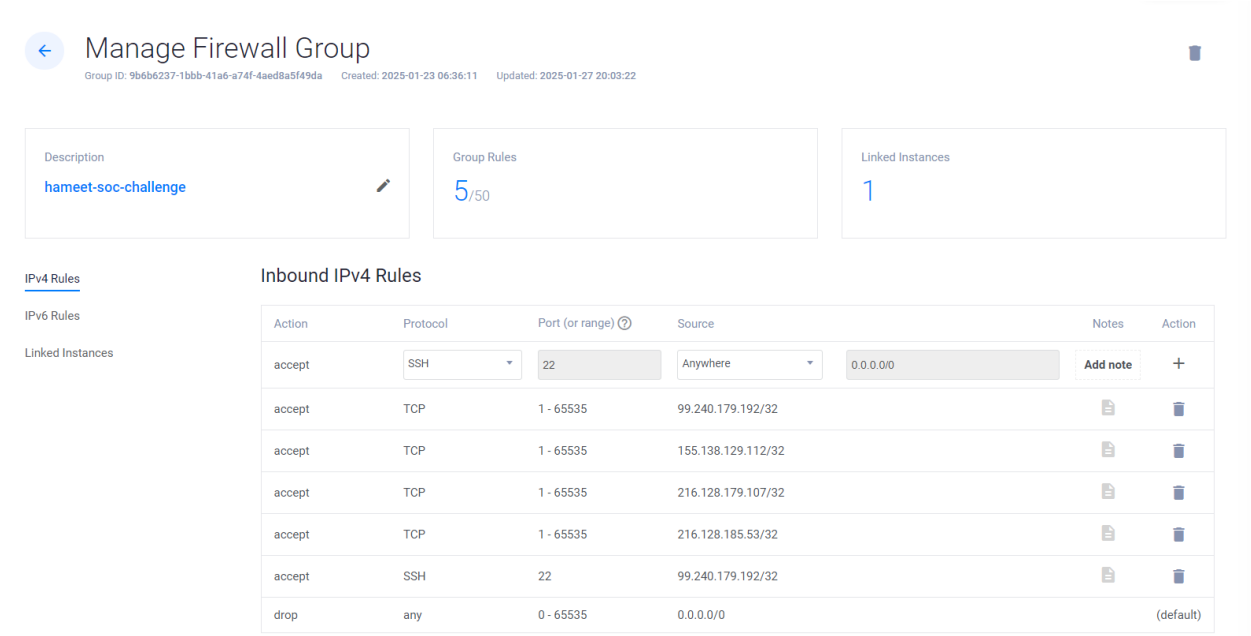
VM Configuration and Elastic SIEM Setup



Cloud Compute

Location | Search | + Deploy

Name	OS	Location	Charges	Status
hameet-ELK 16384.00 MB Optimized Cloud - 137.220.54.158		Toronto	$21.25	Running
Hameet-Fleet-Server 4096.00 MB Optimized Cloud - 216.128.179.107		Toronto	$0.54	Running
HAMEET-LINUX 1024.00 MB AMD High Performance - 155.138.129.112		Toronto	$0.09	Running
hameet-windows-server 2048.00 MB Regular Cloud Compute - 216.128.185.53		Toronto	$3.40	Running



Manage Firewall Group

Group ID: 9b6b6237-1bbb-41a6-a74f-4aed8a5f49da Created: 2025-01-23 06:36:11 Updated: 2025-01-27 20:03:22

Description: hameet-soc-challenge

Group Rules: 5/50

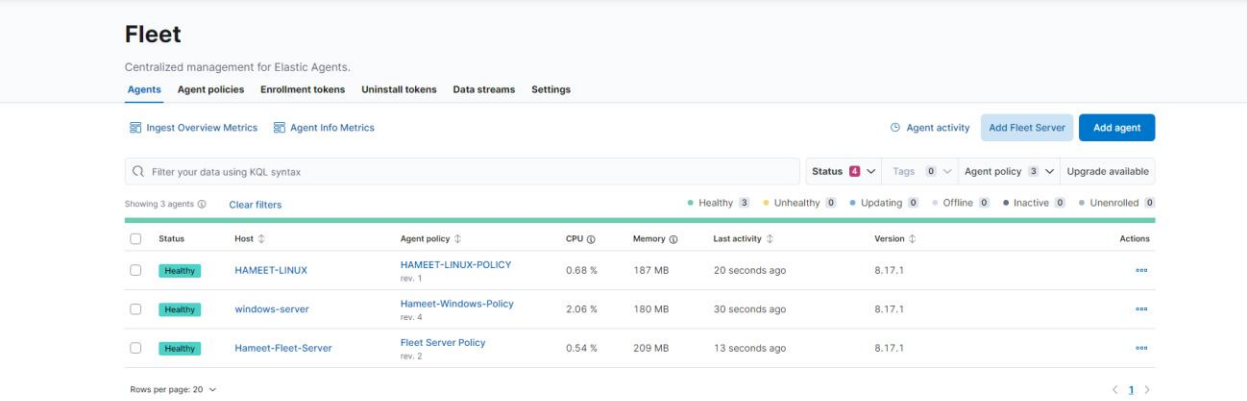
Linked Instances: 1

IPv4 Rules | IPv6 Rules | Linked Instances

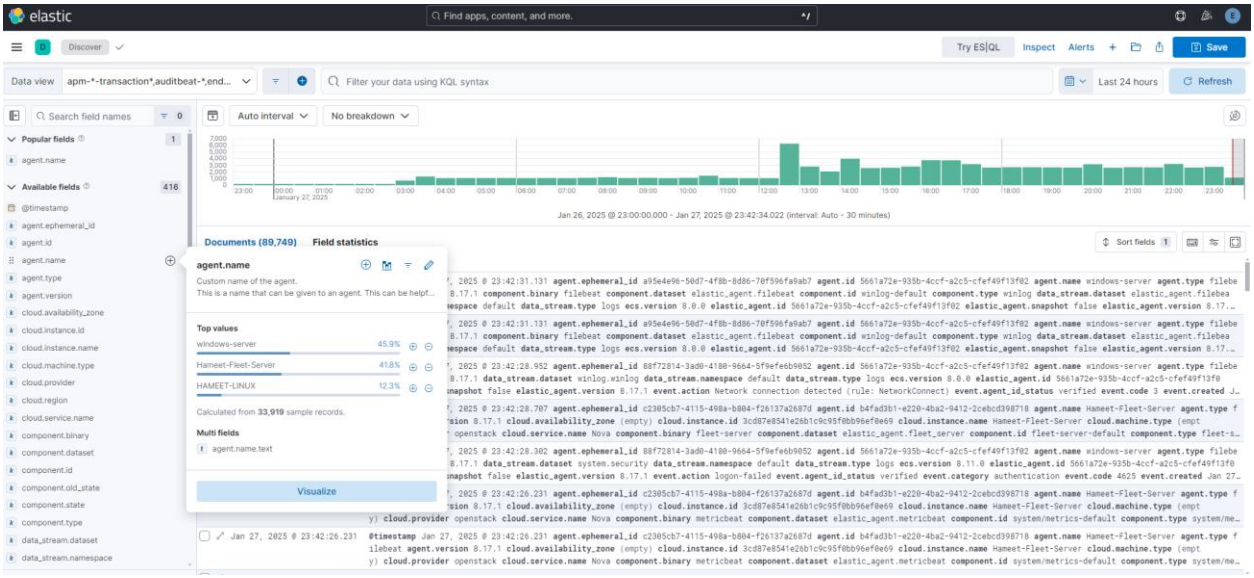
Inbound IPv4 Rules

Action	Protocol	Port (or range)	Source		Notes	Action
accept	SSH	22	Anywhere	0.0.0.0/0	Add note	+
accept	TCP	1 - 65535	99.240.179.192/32			
accept	TCP	1 - 65535	155.138.129.112/32			
accept	TCP	1 - 65535	216.128.179.107/32			
accept	TCP	1 - 65535	216.128.185.53/32			
accept	SSH	22	99.240.179.192/32			
drop	any	0 - 65535	0.0.0.0/0			(default)

Ref 1: Spin up 4 VMs and configure all accordingly. Install ELK on corresponding VM, windows server on another, Fleet Server on another and Sysmon on Windows server. Set up basic firewall to prevent all traffic, allow from my IP and allow other VMs to communicate.



Ref 2: Customize fleet policy to ingest logs from Windows Defender and Sysmon from both Linux machine and Windows Server.



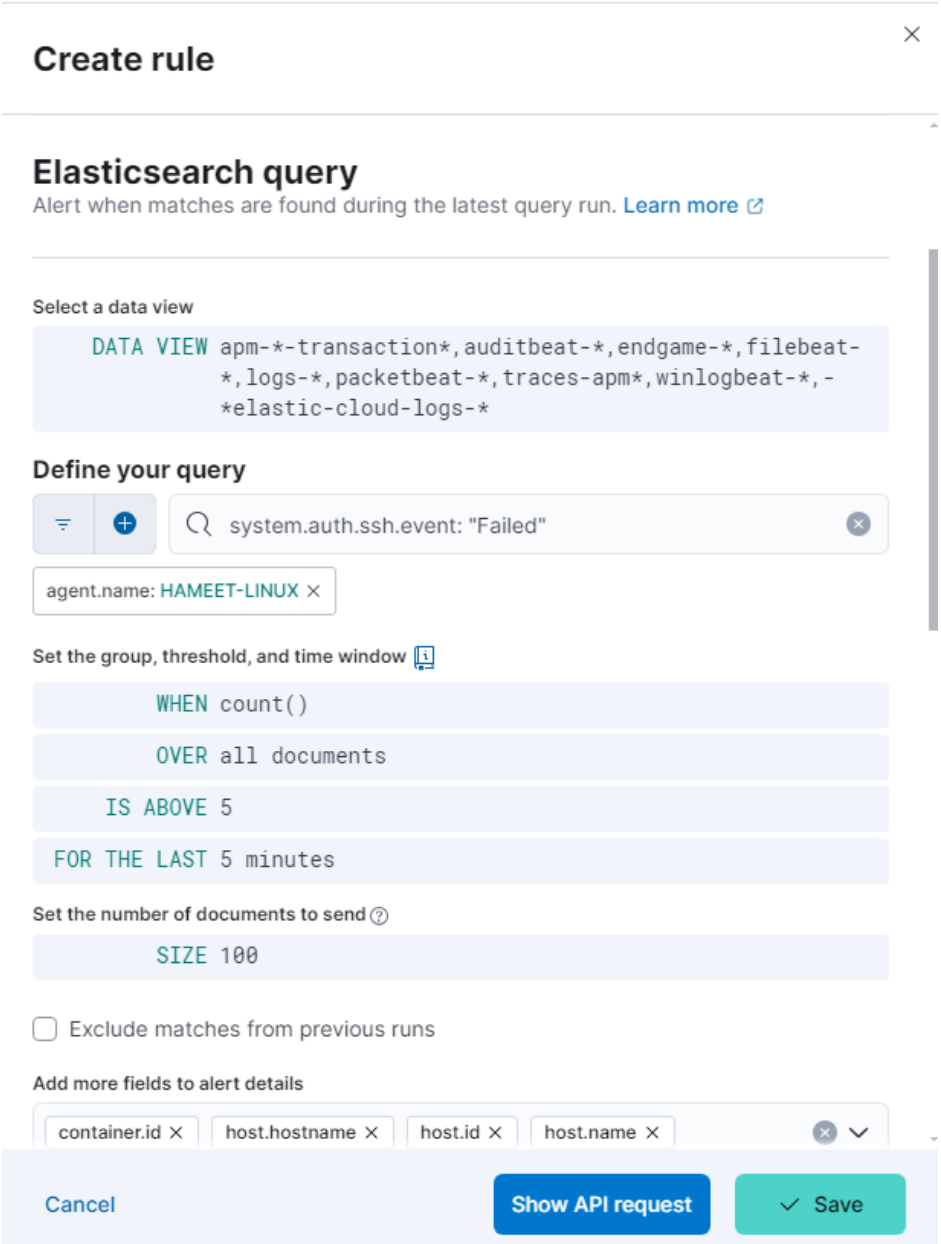
Ref 3: Confirm telemetry is being ingested from both machines by checking logs and seeing both agent names.

Documents (184) Field statistics

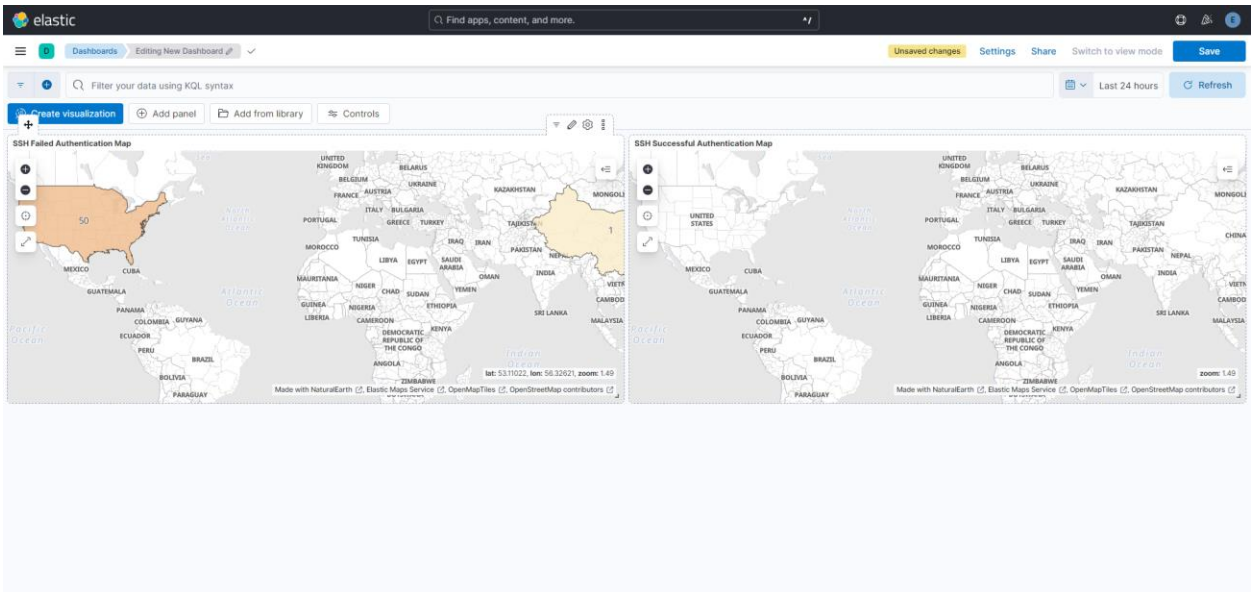
Columns 5 Sort fields 1

@timestamp	system.auth.ssh.event	user.name	source.ip	source.geo.country_name
Jan 27, 2025 @ 19:43:25.197	Failed	sync	92.118.39.74	United States
Jan 27, 2025 @ 19:42:30.491	Failed	ps	92.118.39.86	United States
Jan 27, 2025 @ 19:36:11.409	Failed	snark	92.118.39.74	United States
Jan 27, 2025 @ 19:30:41.582	Failed	v	92.118.39.86	United States
Jan 27, 2025 @ 19:28:53.845	Failed	snarkOS	92.118.39.74	United States
Jan 27, 2025 @ 19:21:35.566	Failed	grafana	92.118.39.74	United States
Jan 27, 2025 @ 19:18:47.437	Failed	solx	92.118.39.86	United States

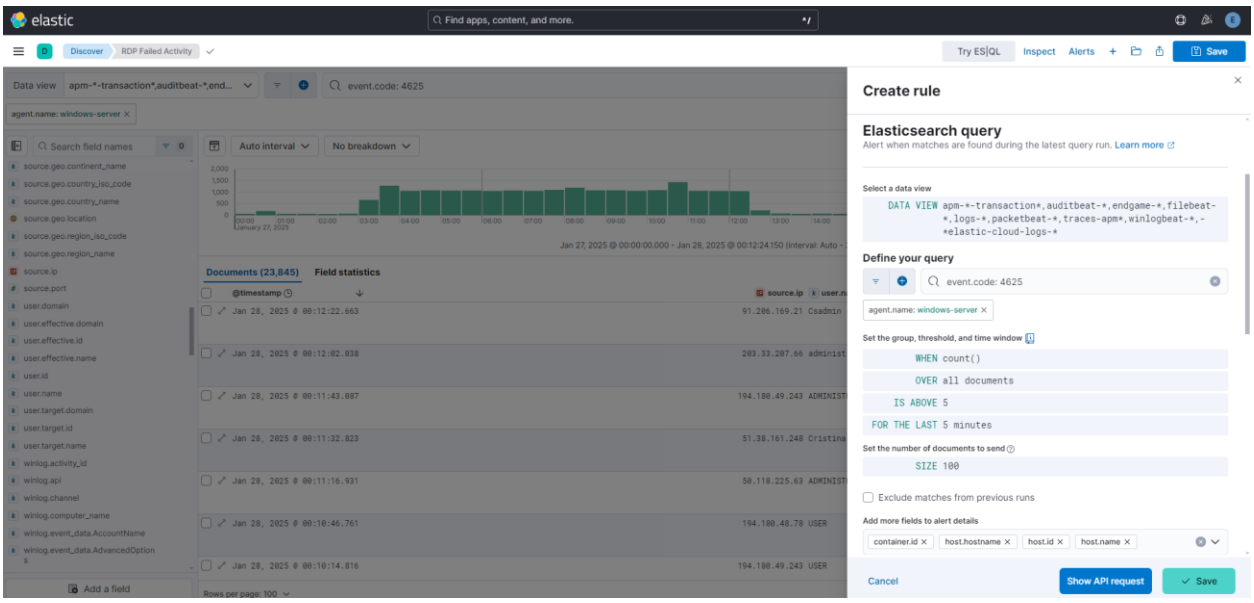
Ref 4: Apply appropriate filters and fields to filter results for brute force attack signs on Linux machine.



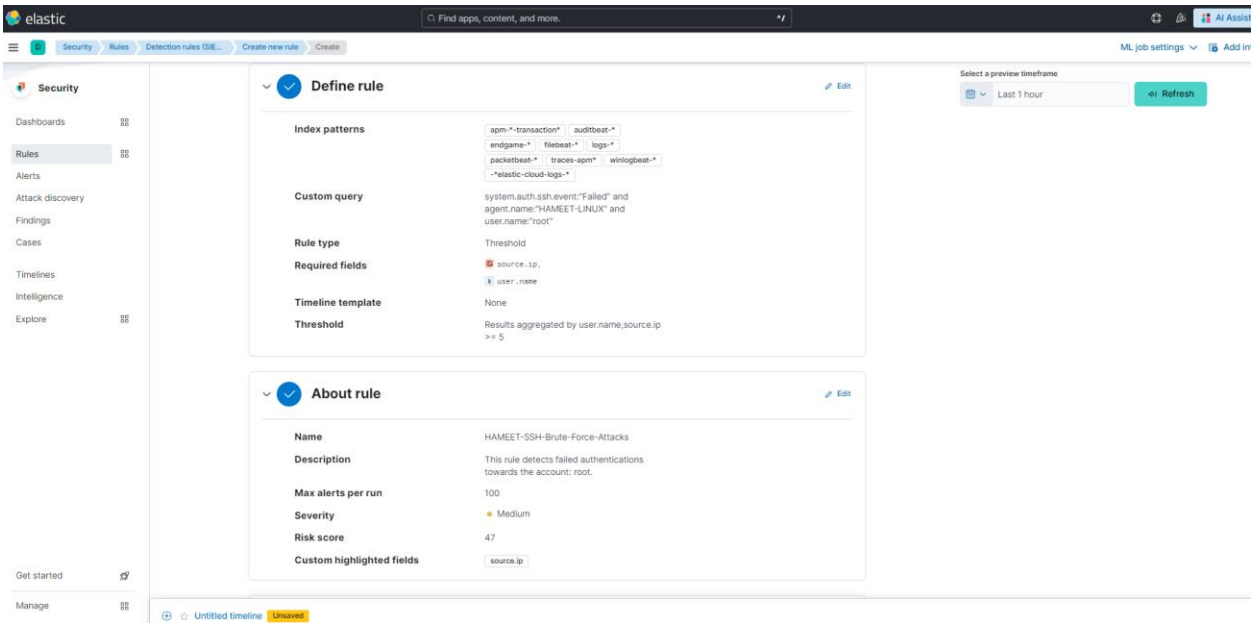
Ref 5: Create basic rule to send alert for brute force attack activity.



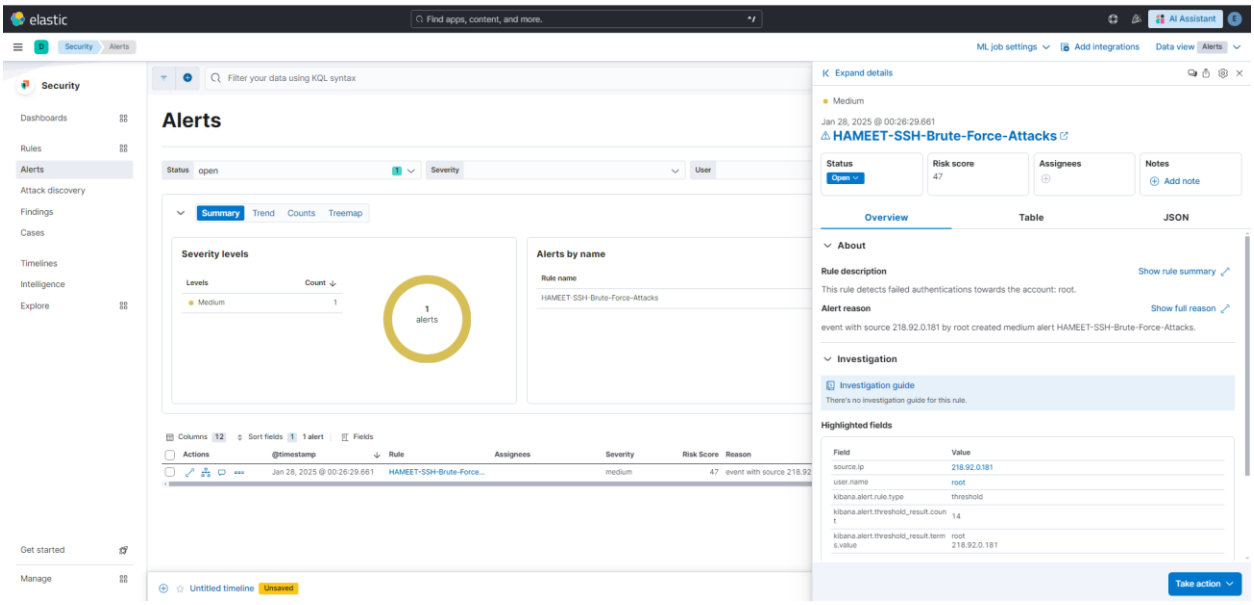
Ref 6: Create a simple dashboard to visualize the query above. Maps out attempts on a geolocation basis.



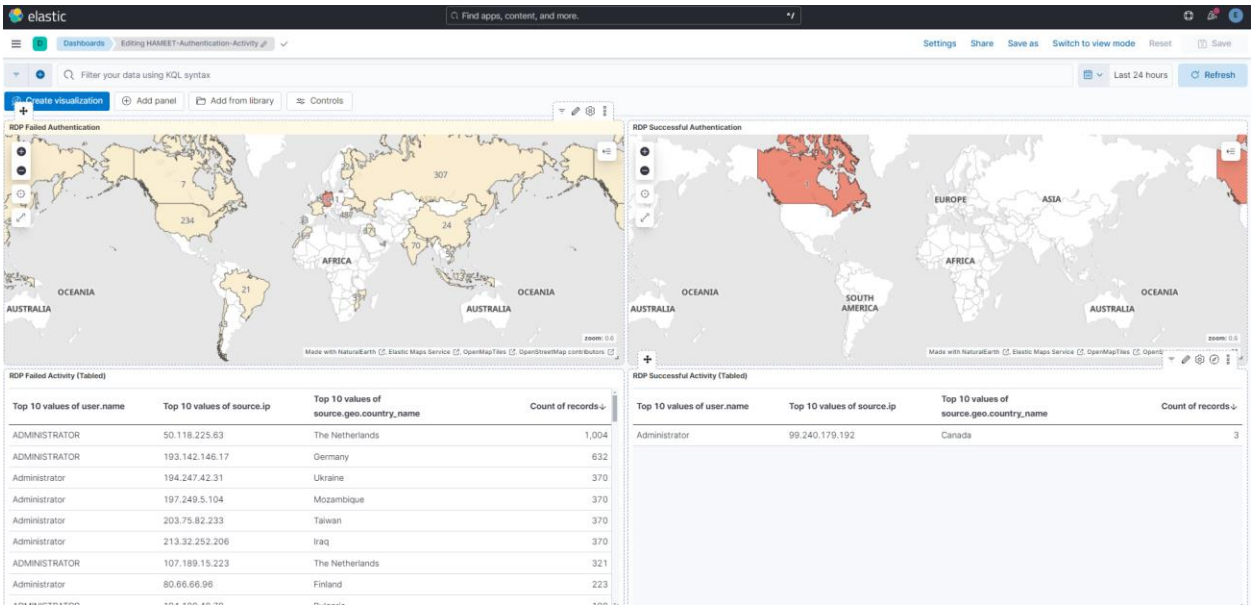
Ref 7: Create query and alert rule to filter for windows event code: 4625. This indicates a failed logon attempt. The alert and query is for RDP brute force attack warning.



Ref 8: That above rules were simple and provided little information. Creating a detection rule in elastic will allow us to provide more information with incoming alerts and are more customizable.



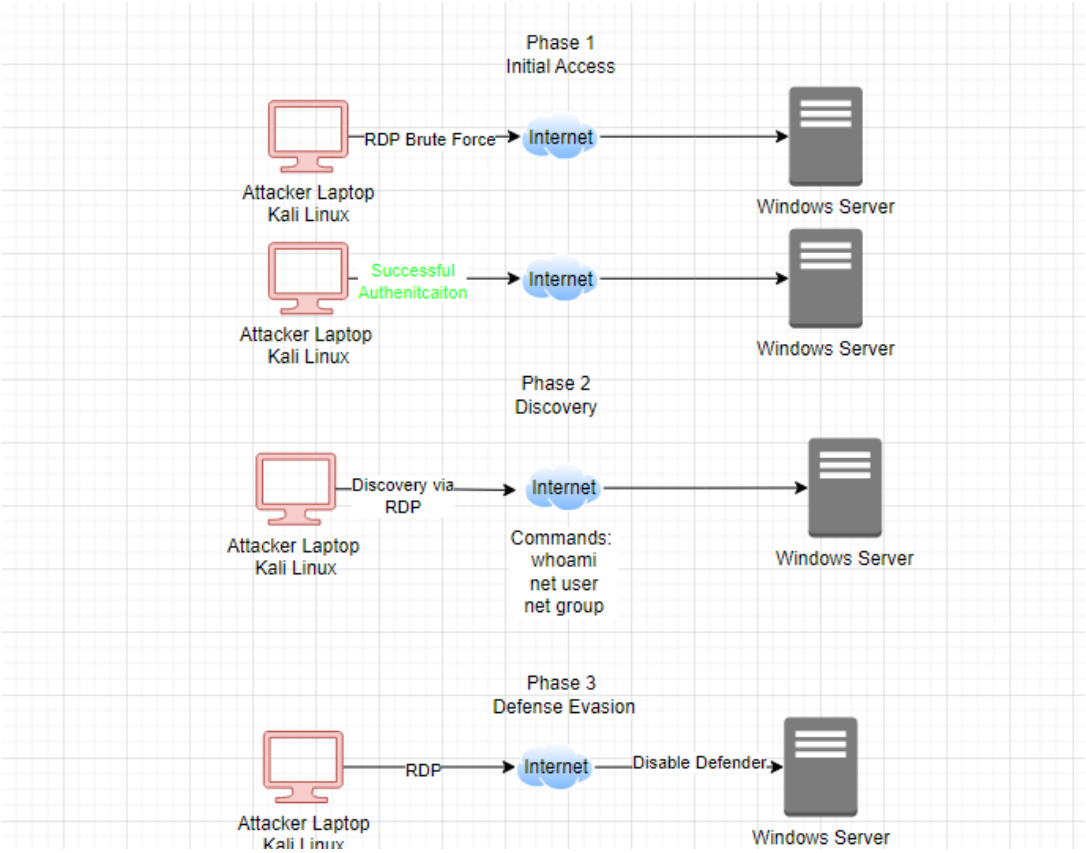
Ref 9: An example of an incoming alert for brute force attack.

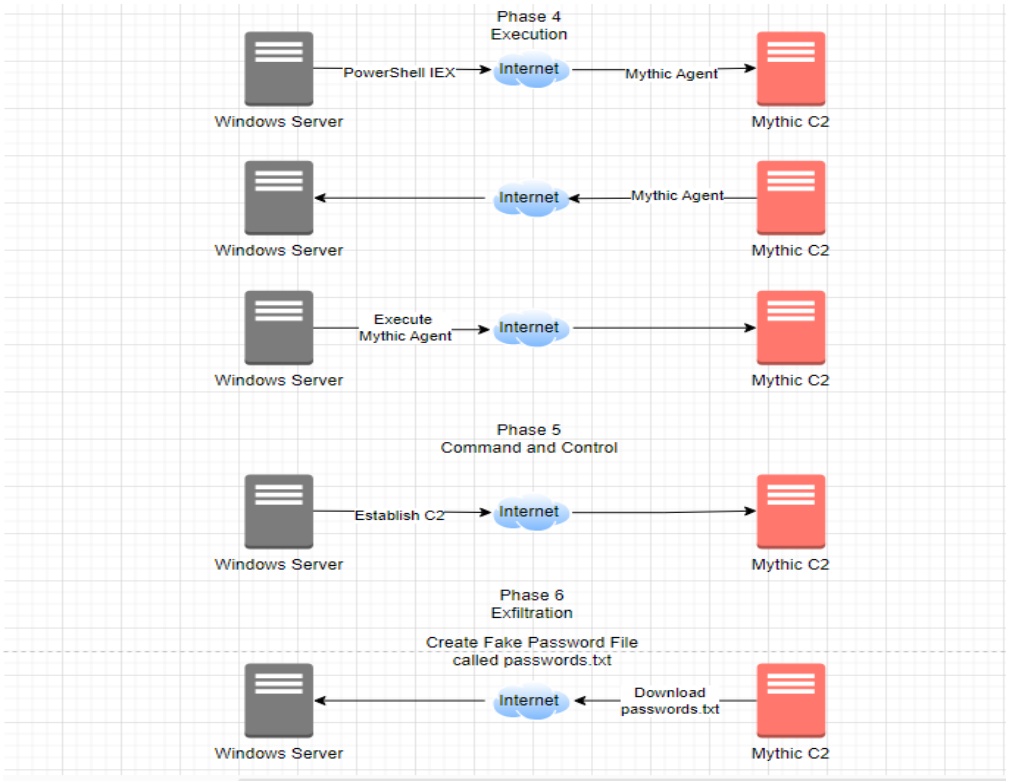


Ref 10: Created a visualization of the attempts on Elastic SIEM, both unsuccessful and successful for RDP and SSH brute force attempts. There was both a visual map and a table outlining all activity.

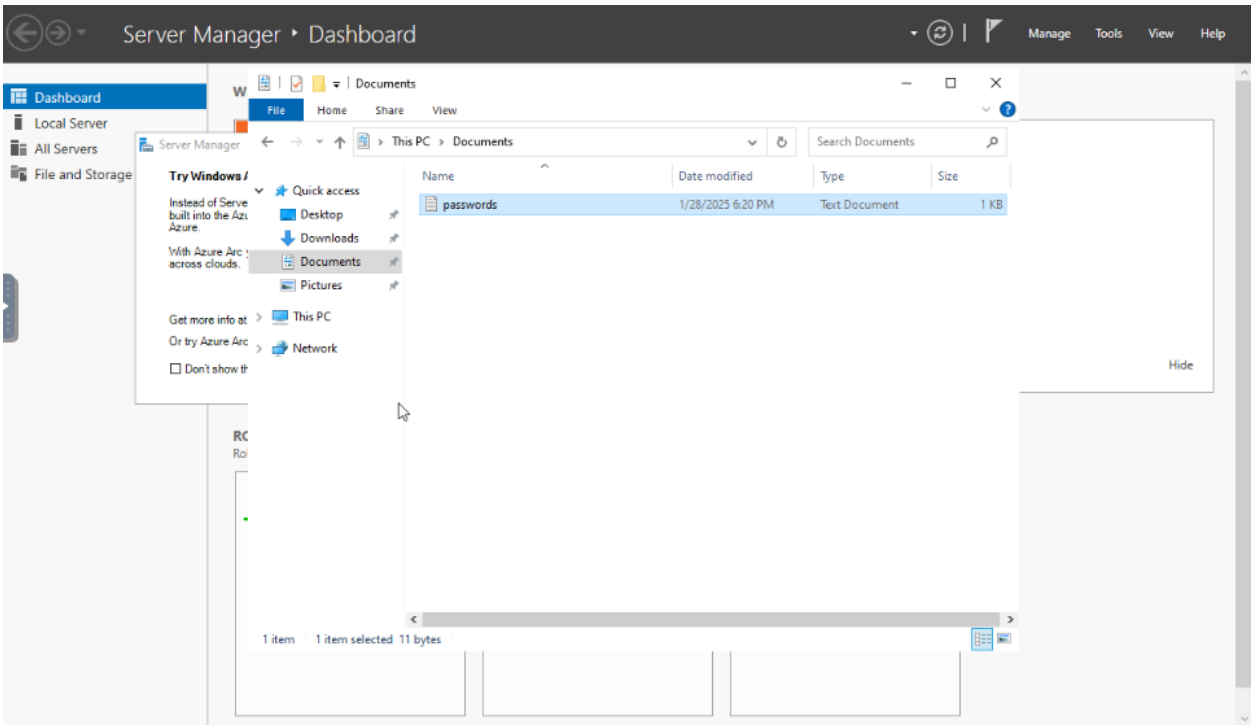
Now that our alerts and rules/policies are created. We are going to create an attack diagram to plan the command-and-control attack. Following that, the Mythic program was set up and loaded using an Ubuntu VM.

Command-and-Control Attack Phase



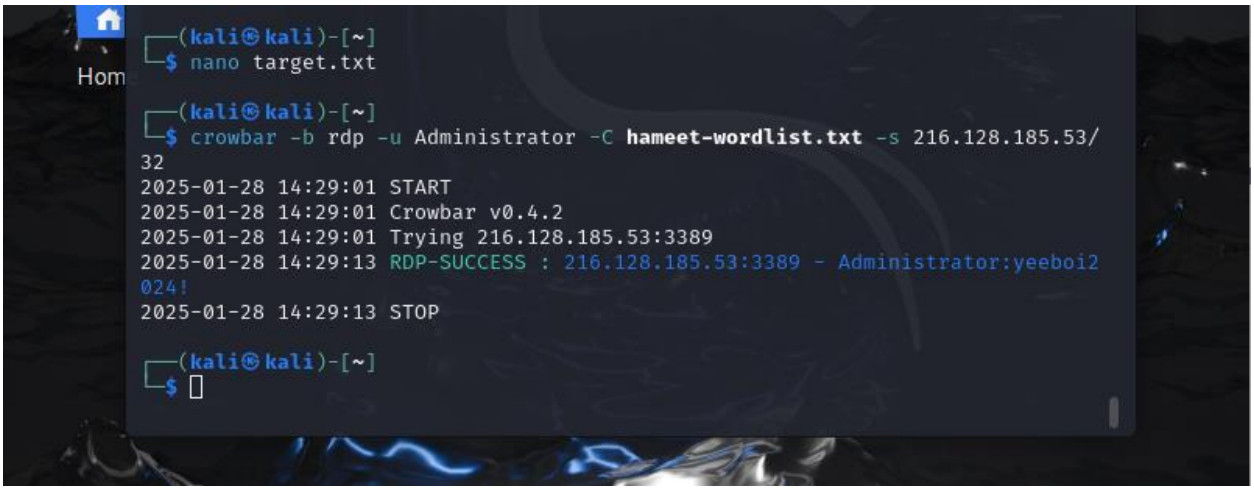


Ref 11: Attack diagram for Mythic command-and-control attack.



Ref 12: As we prepare to simulate the attack, a dummy text file called passwords was created. Additionally, the password for the windows logon for this machine was changed to the one in the text file.

A wordlist with our target password was created for use for the brute force attack on our Kali Linux system.

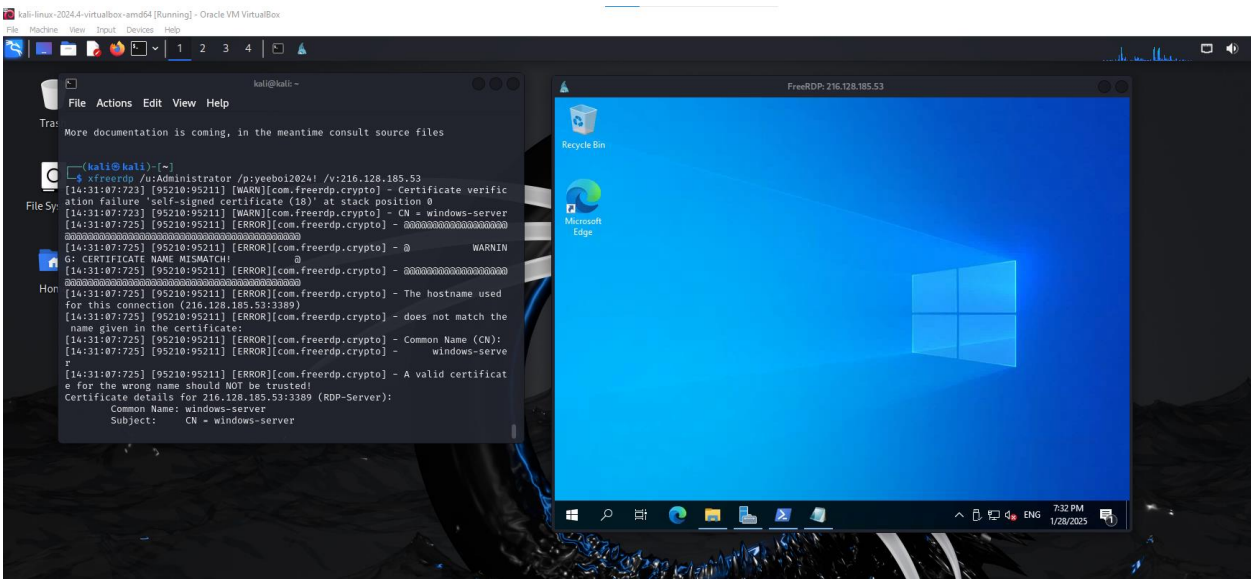


```
┌──(kali㉿kali)-[~]
└─$ nano target.txt

┌──(kali㉿kali)-[~]
└─$ crowbar -b rdp -u Administrator -C hameet-wordlist.txt -s 216.128.185.53/
32
2025-01-28 14:29:01 START
2025-01-28 14:29:01 Crowbar v0.4.2
2025-01-28 14:29:01 Trying 216.128.185.53:3389
2025-01-28 14:29:13 RDP-SUCCESS : 216.128.185.53:3389 - Administrator:yeeboi2
024!
2025-01-28 14:29:13 STOP

┌──(kali㉿kali)-[~]
└─$ ▯
```

Ref 13: Using crowbar in Kali Linux, we execute a brute force attack on our windows server. Using the appropriate IP, username, and wordlist.



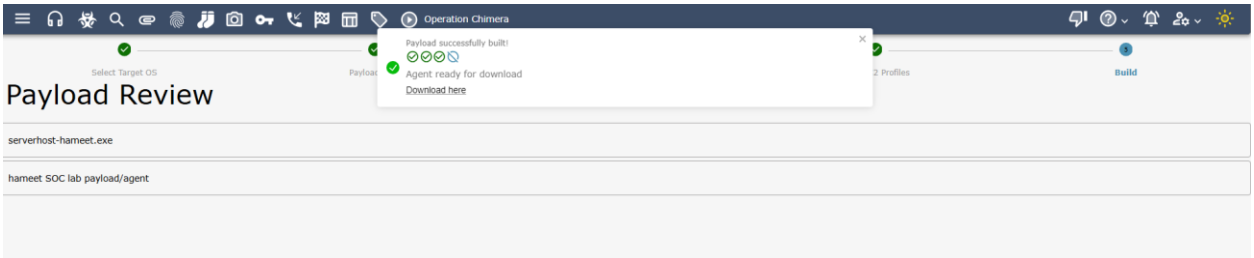
```
More documentation is coming, in the meantime consult source files


┌──(kali㉿kali)-[~]
└─$ xfreerdp /u:Administrator /p:yeeboi2024! /v:216.128.185.53
[14:31:07:723] [95210:95211] [WARN][com.freerdp.crypto] - Certificate verific
ation failure 'self-signed certificate (18)' at stack position 0
[14:31:07:723] [95210:95211] [WARN][com.freerdp.crypto] - CN = windows-server
[14:31:07:725] [95210:95211] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[14:31:07:725] [95210:95211] [ERROR][com.freerdp.crypto] - @           WARNIN
G: CERTIFICATE NAME MISMATCH!           @
[14:31:07:725] [95210:95211] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[14:31:07:725] [95210:95211] [ERROR][com.freerdp.crypto] - The hostname used
for this connection (216.128.185.53:3389)
[14:31:07:725] [95210:95211] [ERROR][com.freerdp.crypto] - does not match the
 name given in the certificate:
[14:31:07:725] [95210:95211] [ERROR][com.freerdp.crypto] - Common Name (CN):
[14:31:07:725] [95210:95211] [ERROR][com.freerdp.crypto] -      windows-serve
r
[14:31:07:725] [95210:95211] [ERROR][com.freerdp.crypto] - A valid certificat
e for the wrong name should NOT be trusted!
Certificate details for 216.128.185.53:3389 (RDP-Server):
        Common Name: windows-server
        Subject:     CN = windows-server
```

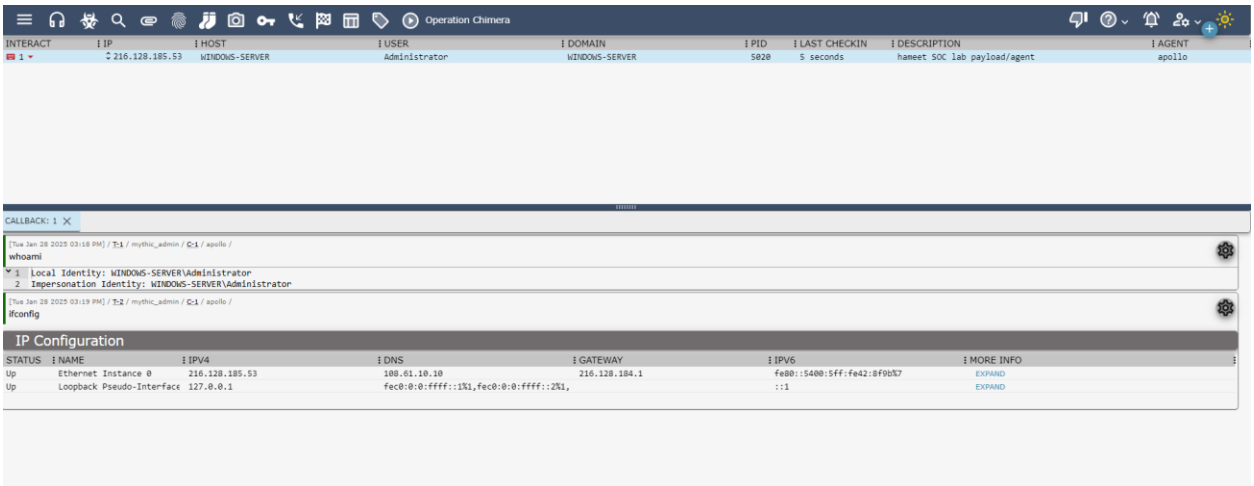
Ref 14: Now that we have the password from the brute force attack, we can establish an RDP connection using xfreerdp.

The next phase of the attack is now commenced. Using command prompt, we type in commands such as ipconfig, whoami, net user/ group, etc to gain knowledge and discovery. Then we proceed to disable windows defender to further compromise the machine.

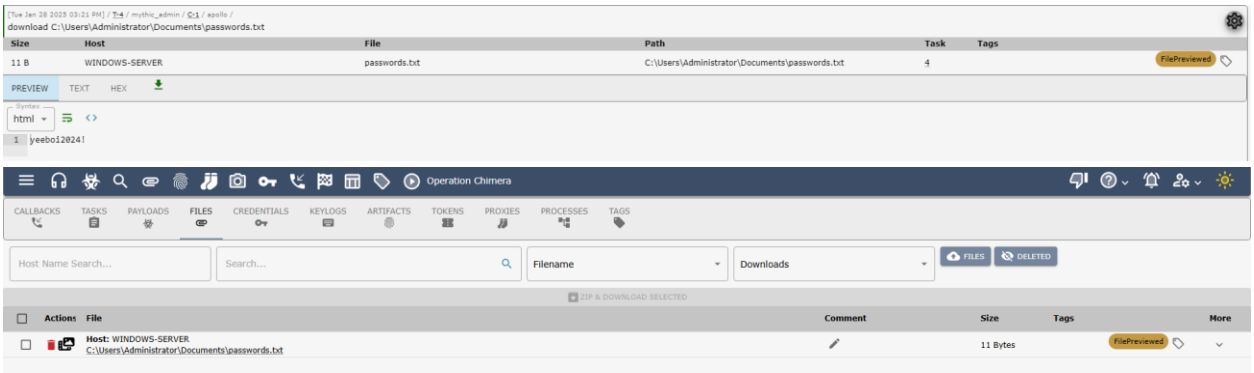
Ref 15: Install apollo agent and mythic C2 profile.



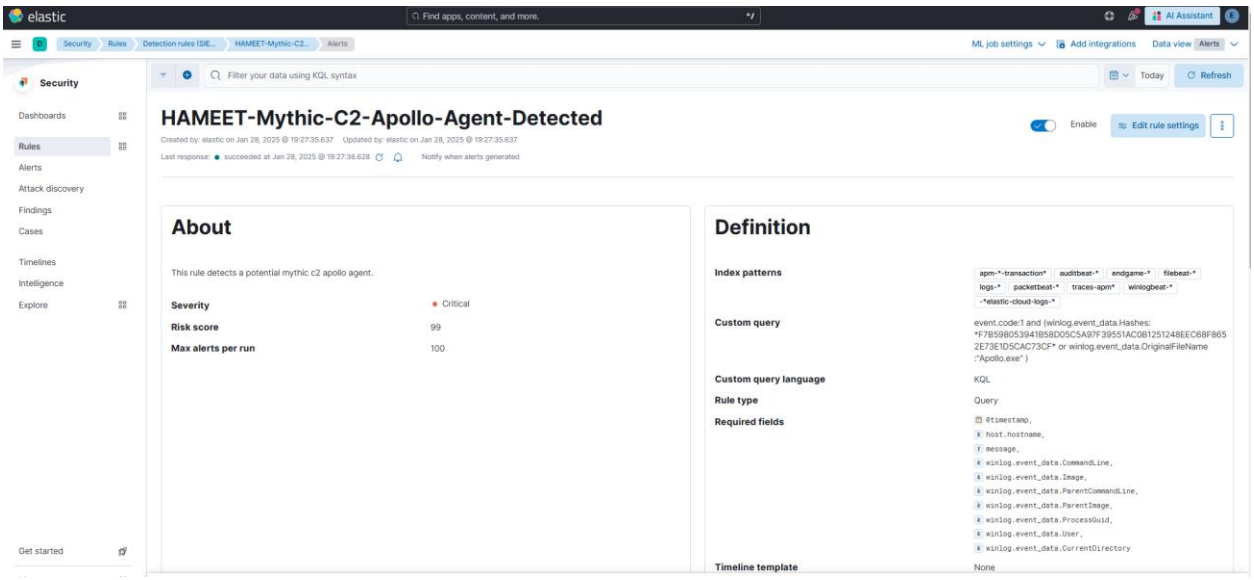
Ref 16: Create mythic payload.



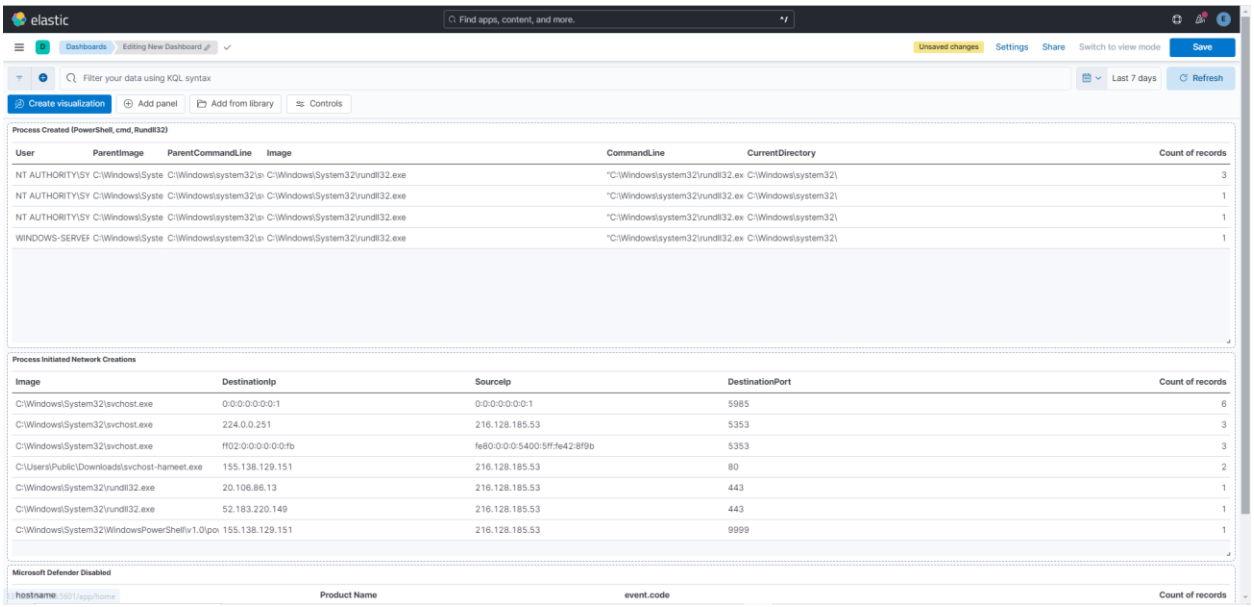
Ref 17: Service host was created in Mythic Server cli. Invoke Web Request was then used to get servicehost onto target machine via RDP. The success was ensured by callback showing up on Mythic GUI. Interacting with the callback, commands whoami and ifconfig were run to ensure functionality.



Ref 18: The dummy text file was downloaded and that completes all phases of the attack. The text file shows up in the attachments tab and has our dummy password inside of it.



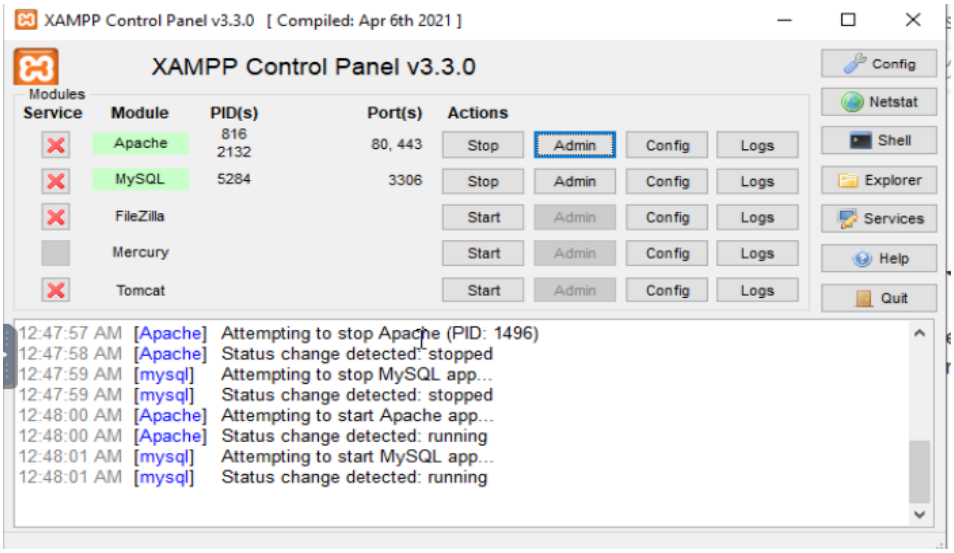
Ref 19: Create a rule in elastic in which if the SHA256 file hash matches Apollo.exe, or if original file name matches Apollo.exe then it will be detected. An alert with the above fields will be generated.



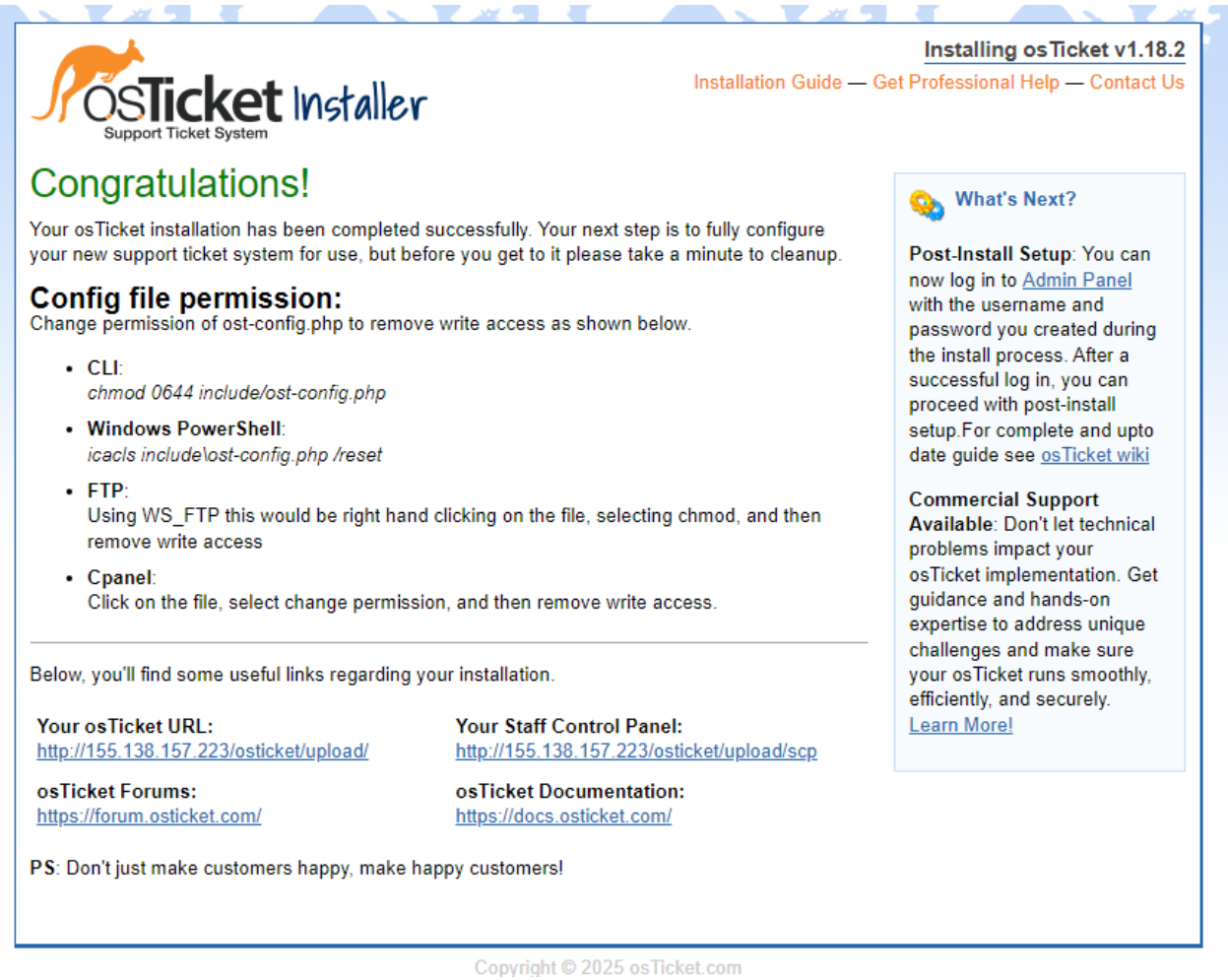
Ref 20: Now we are going to create a dashboard that looks for suspicious activity, including event id 3: Network Connection (external) - any processes creating a network connection outbound. Event id 1- process creates – powershell,cmd,etc. Event id 5001- windows defender disabled.

Investigation and Ticketing System

We will now create a ticketing system using OSTicket. This will simulate a SOC scenario and give me hands-on experience.

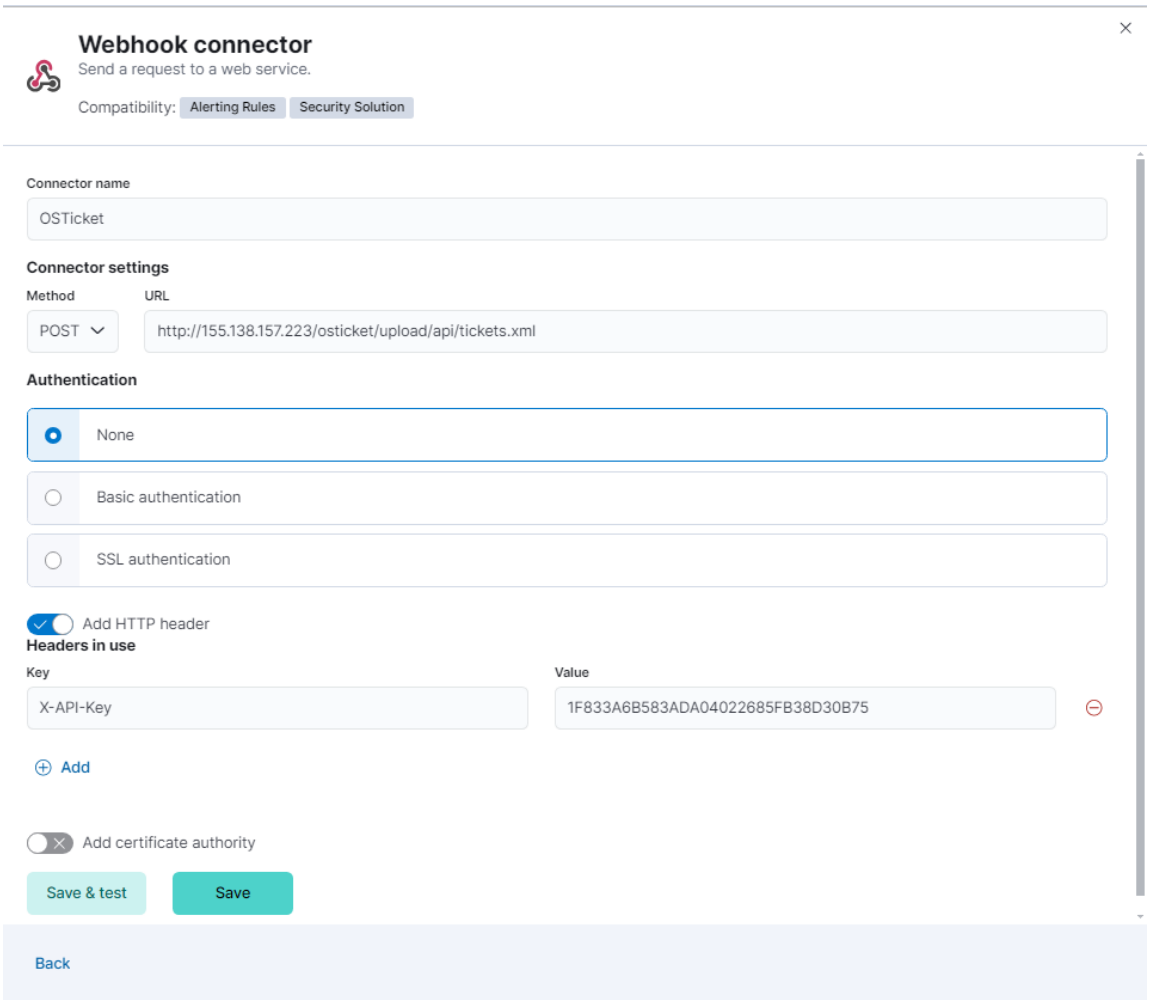


Ref 21: First a windows VM was spun up, and a XAMPP Apache server was installed and booted. Next OSTicket was installed on this VM.

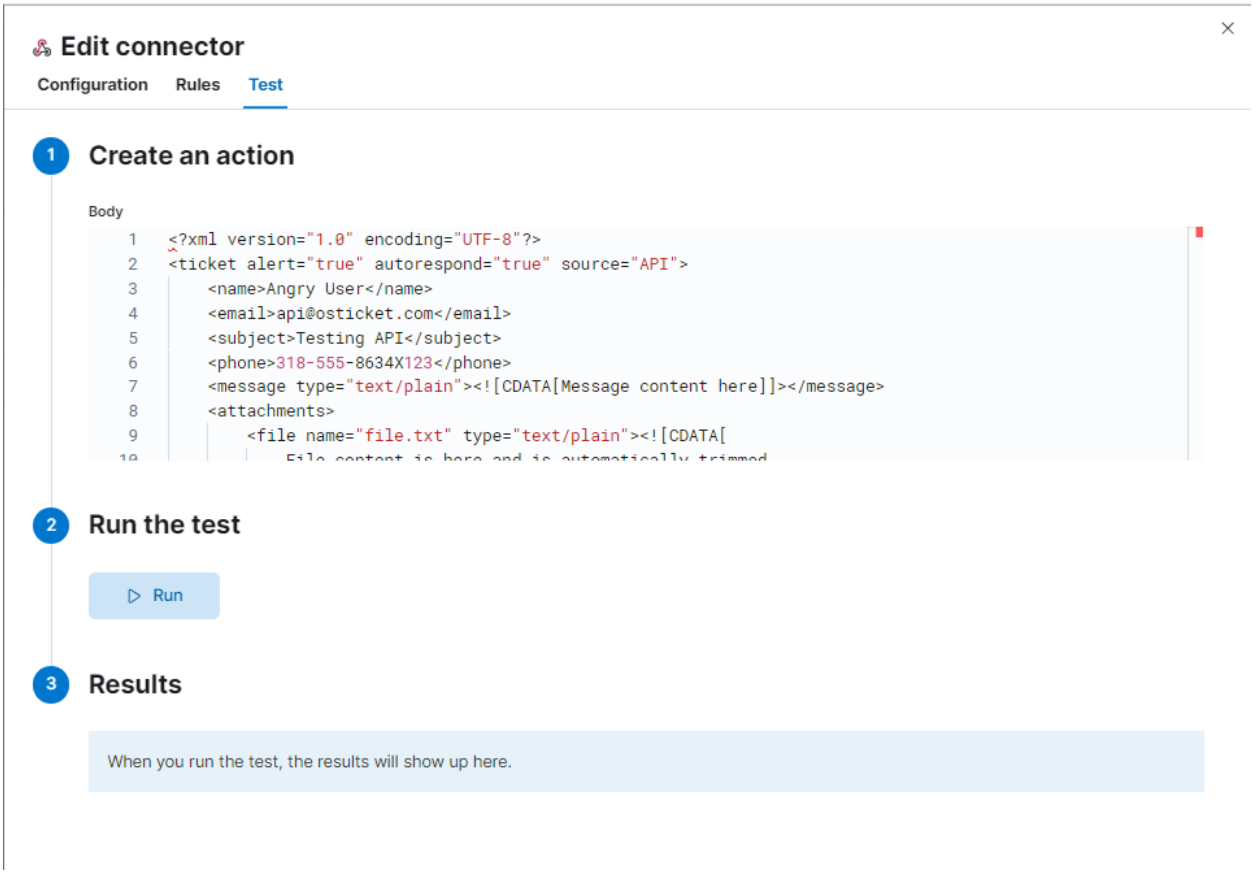


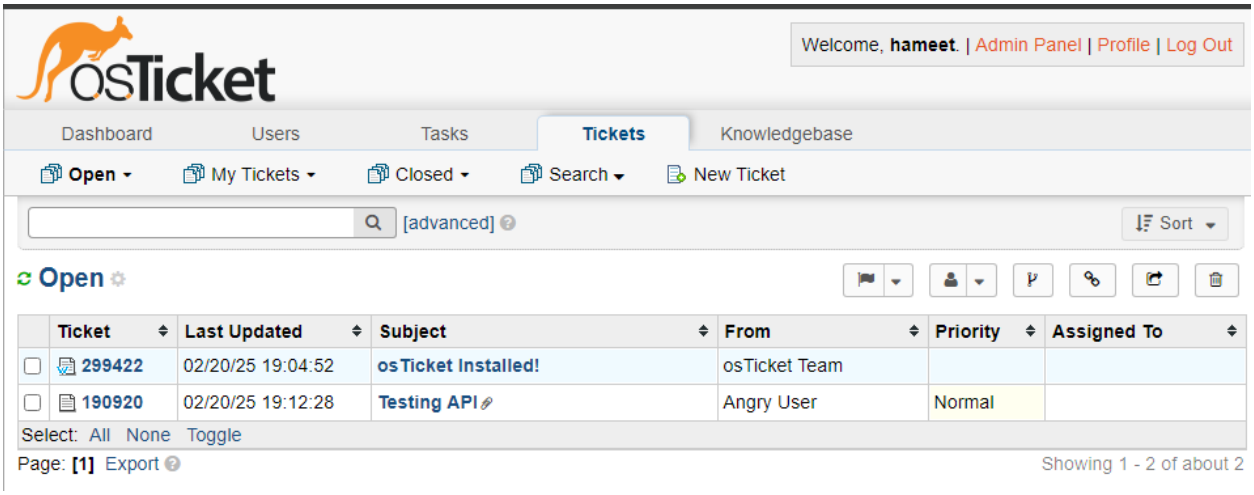
Ref 22: Our SQL database was created and OSTicket was successfully installed and configured.

Now the next step was to use an Elastic connector/webhook to connect our OSTicket to our ELK stack.

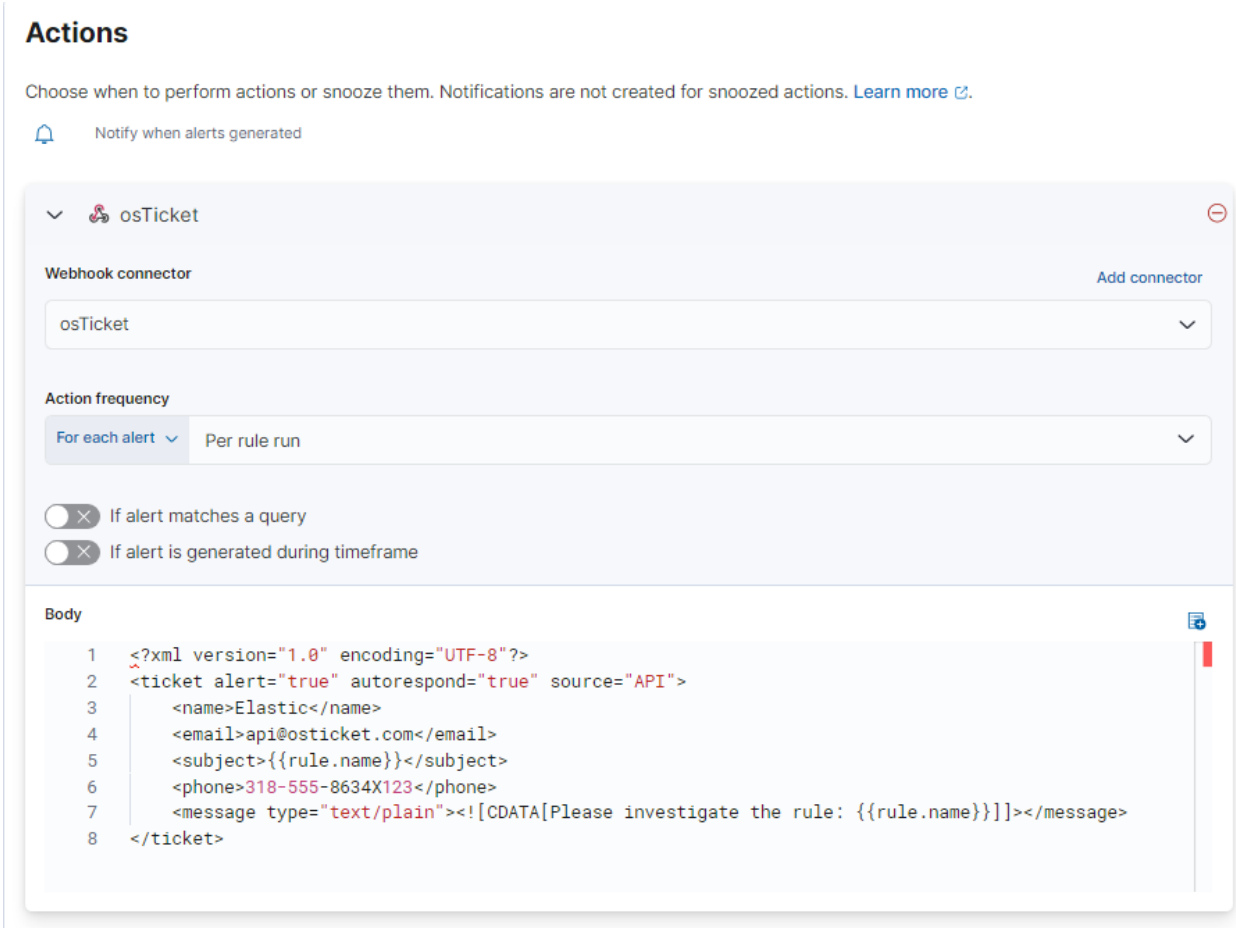


Ref 23: Create webook connector from Elastic to OSTicket.

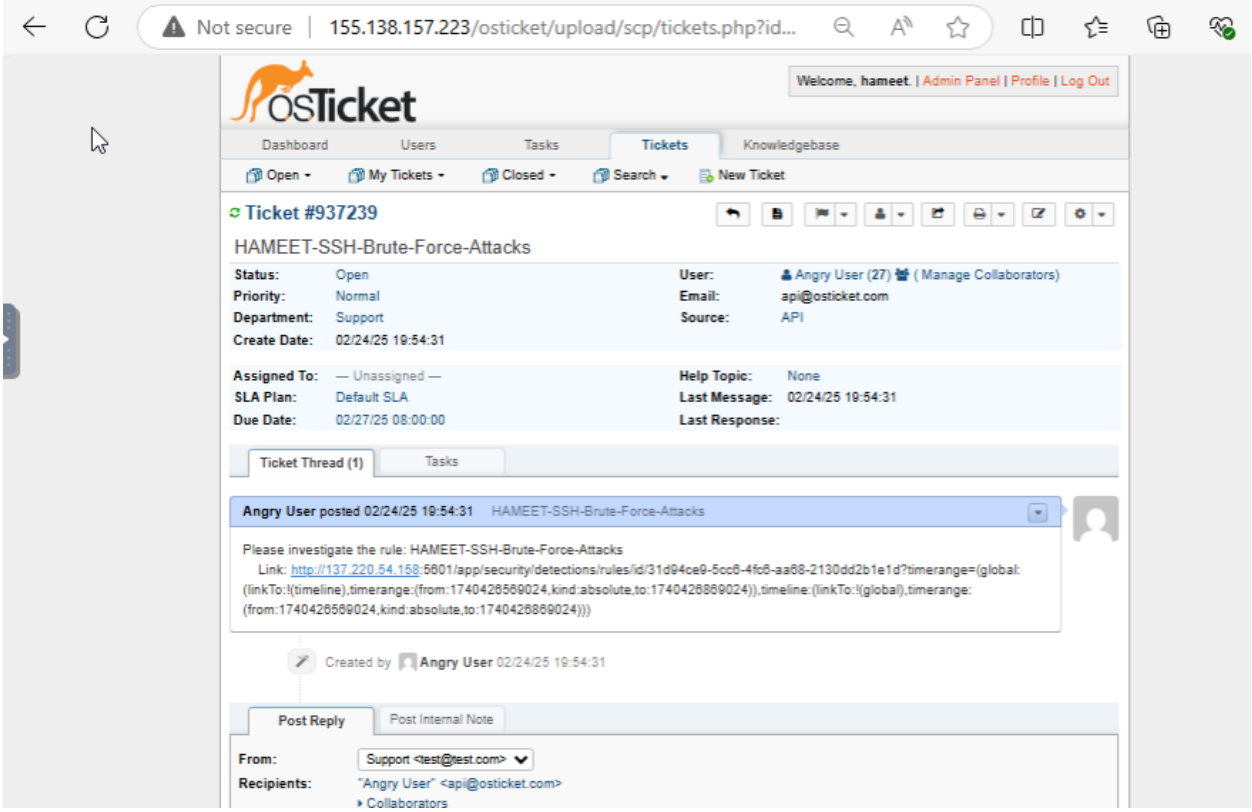




Ref 24: Once connected through the API key from osTicket, it was tested successfully. This was indicated by the agent panel in osTicket. Next, we are going to use our previously set-up alerts in Elastic (Brute force attacks) and use them to create tickets for the alerts.

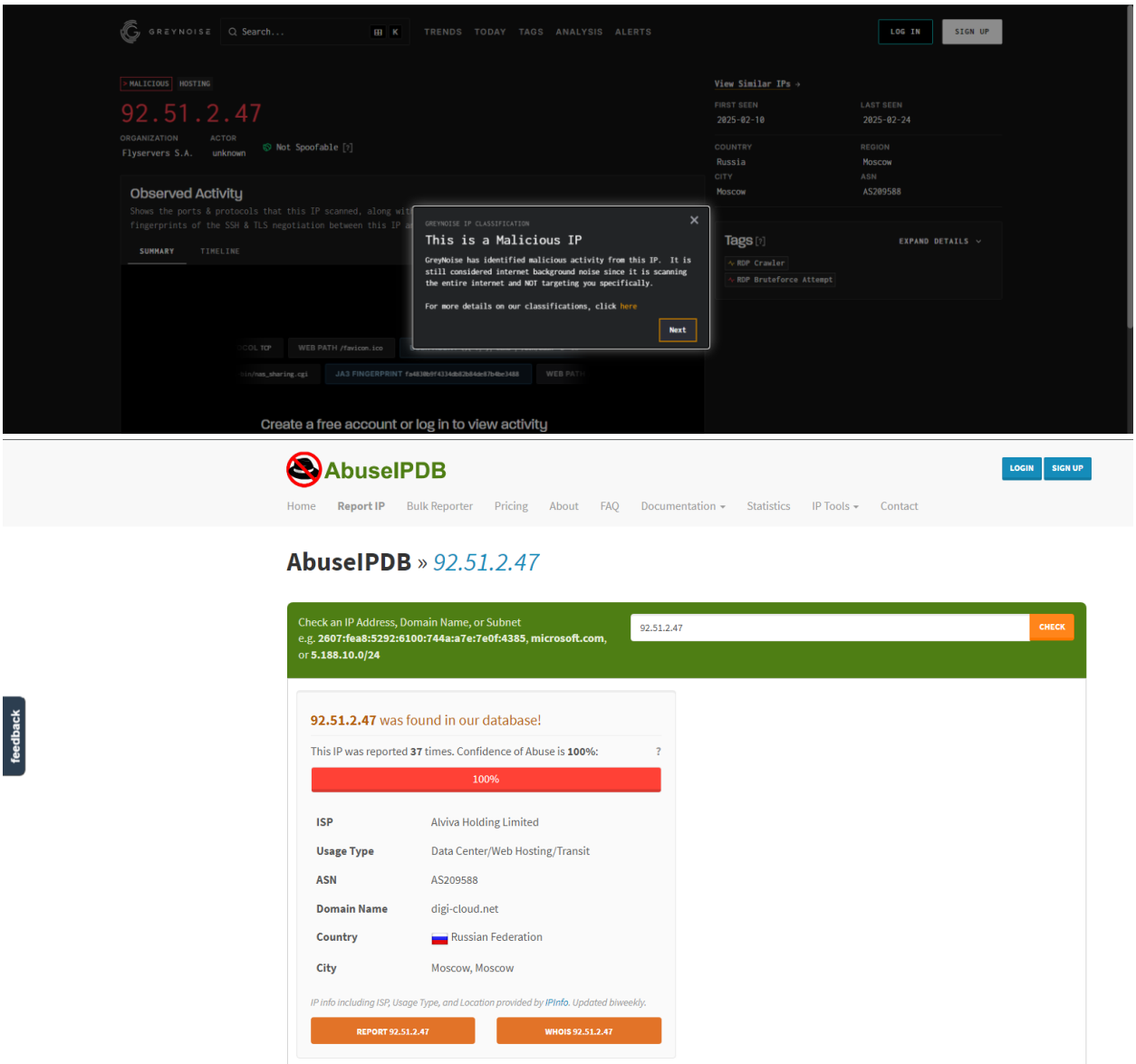


Ref 25: Create action to create ticket for alerts on Elastic.



Ref 26: As can be seen, the alerts in Kibana now open a ticket as well. This includes the rule name from Kibana as well as a link to investigate the incident further. The ticket can be assigned to someone, and comments can be made under the ticket. In a real-world scenario, once the proper investigation was done, the ticket can also be resolved and closed.

This step was repeated for the RDP brute force attack and the Mythic agent rules. After this, taking a slightly deeper dive into the investigation of the brute force attack we can check for the reputation of an IP address using intelligence websites like abuseipdb.com and greynoise.com.



Ref 27: Searching the source IP address of the latest RDP brute force attack shows that this is definitely a malicious IP and is a threat.