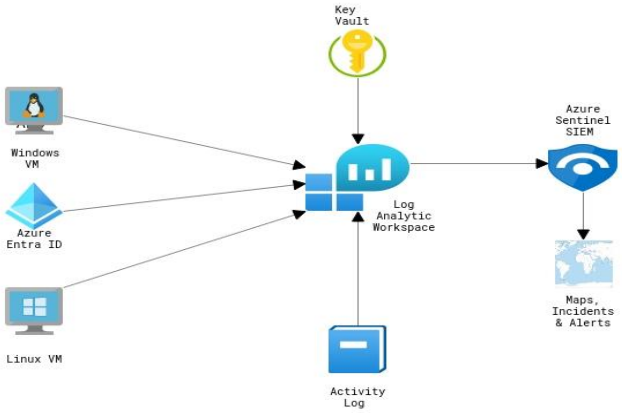
Report: Threat Detection and Monitoring with Sentinel

1. Introduction

With an emphasis on malware activity and brute force attacks, this assignment describes the deployment and efficiency of Azure Sentinel in identifying cybersecurity threats. The objective of this assignment was to evaluate Azure Sentinel's ability to collect endpoint logs, apply analytics rules, and identify potential security threats, such as repeated failed login attempts and malware activities.

These findings underscore the real-world applicability of Azure Sentinel in identifying and mitigating cybersecurity threats.



System Architecture

2. Key Objectives

- Set up Azure Sentinel to monitor both Windows and Linux endpoints.
- Identify brute force attacks by analysing repeated failed login attempts.
- Detect malware activities
- Automate incident generation for quick response.

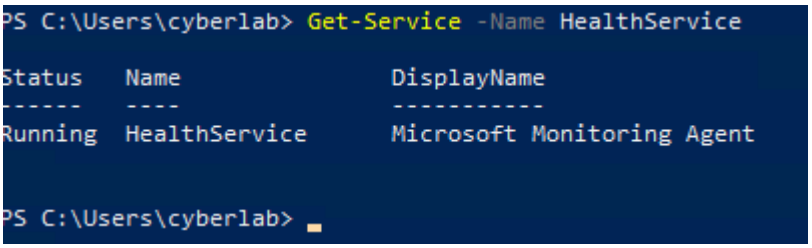
3. Environment Setup

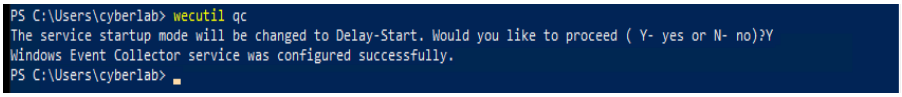
3.1 Azure Sentinel Configuration

- **Workspace Creation**: A new Log Analytics workspace was set up for Azure Sentinel

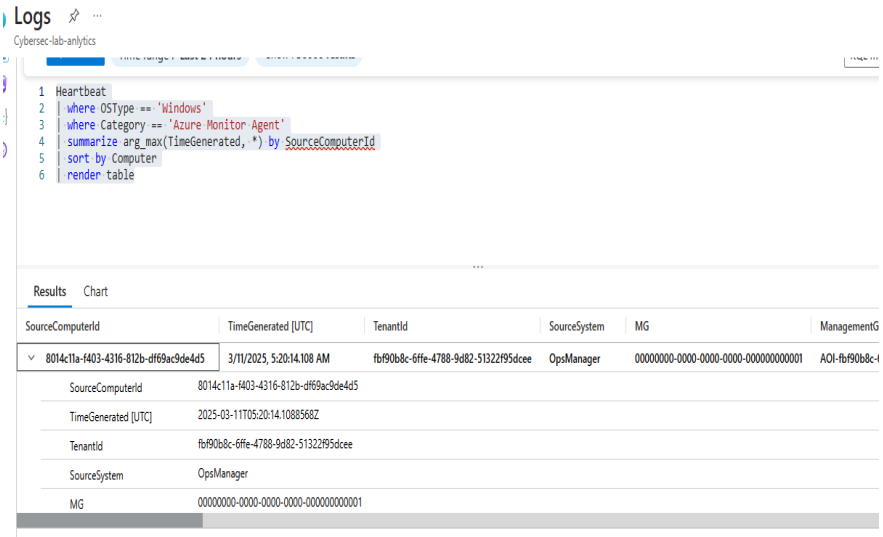
3.2 Data Connector Setup

Windows End Point: The Log Analytics Agent was installed on Windows endpoints to forward security logs to sentinel on Windows end point.





Screenshots showing the status of Azure Monitoring Agent(AMA)

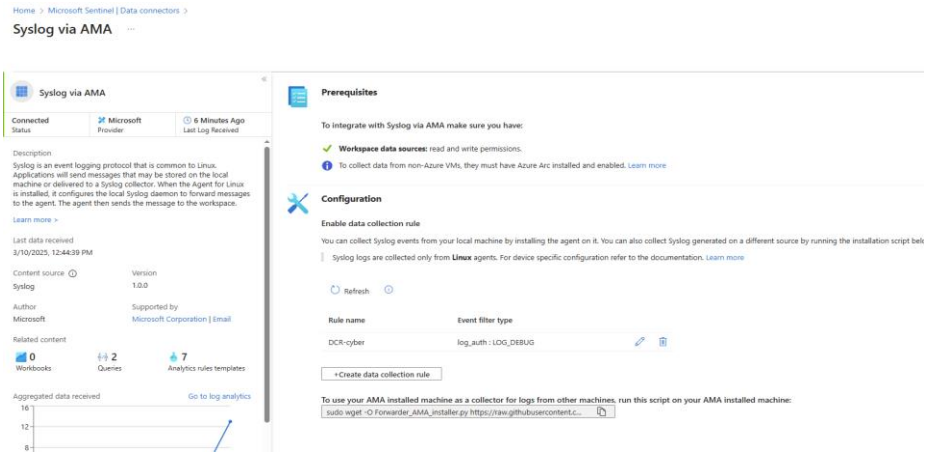


KQL query for showing status of AMA

Linux End point

Installed Syslog via AMA on the Linux machine.

created a Data Collection Rule and ran the provided script on the Linux machine to install AMA forwarder.

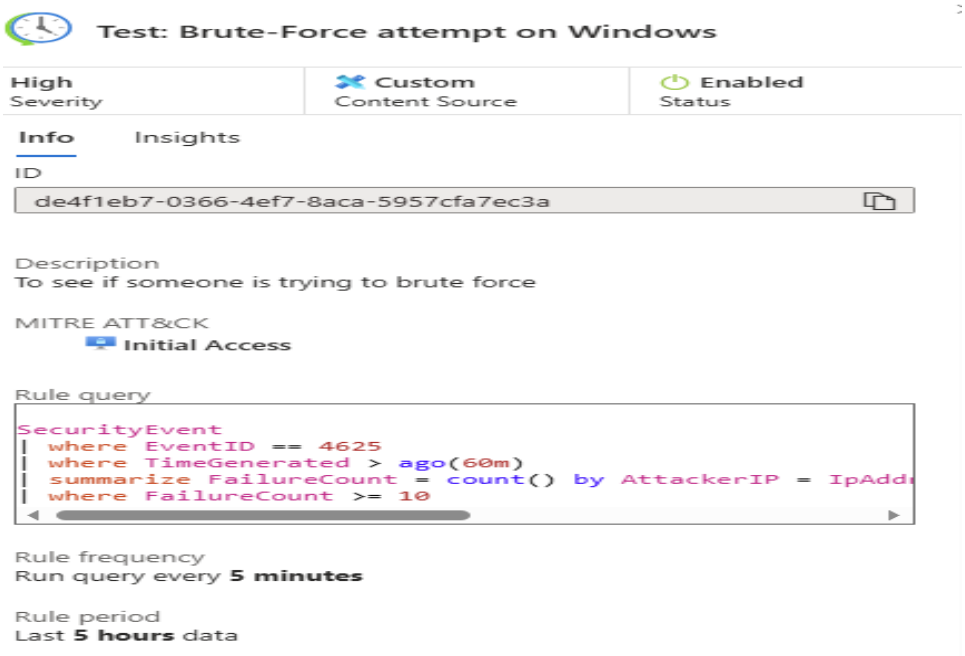



```
cyberlab@Linux-VM:~$ sudo wget -O Forwarder_AMA_installer.py https://raw.githubusercontent.com/Azure/Azure-Sentinel/master/DataConnectors/Syslog/Forwarder_AMA_installer
.py&&sudo python3 Forwarder_AMA_installer.py
--2025-03-10 02:56:08--  https://raw.githubusercontent.com/Azure/Azure-Sentinel/master/DataConnectors/Syslog/Forwarder_AMA_installer.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.110.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13604 (13K) [text/plain]
Saving to: 'Forwarder_AMA_installer.py'

Forwarder_AMA_installer.py              100%[=========================================================================>]  13.29K  --.-KB/s    in 0s

2025-03-10 02:56:08 (48.4 MB/s) - 'Forwarder_AMA_installer.py' saved [13604/13604]

Located rsyslog daemon running on the machine
Rsyslog.conf configuration was changed to fit required protocol - /etc/rsyslog.conf
Restarting rsyslog daemon.
sudo service rsyslog restart
Rsyslog daemon restarted successfully
Please note that the installation script opens port 514 to listen to incoming messages in both UDP and TCP protocols. To change this setting, refer to the Rsyslog confi
guration file located at '/etc/rsyslog.conf'.

Warning: please make sure your Syslog daemon configuration does not store unnecessary logs. This may cause a full disk on your machine, which will disrupt the function
of the oms agent installed. For more information:
https://www.rsyslog.com/doc/master/configuration/actions.html
Installation completed successfully
cyberlab@Linux-VM:~$
```

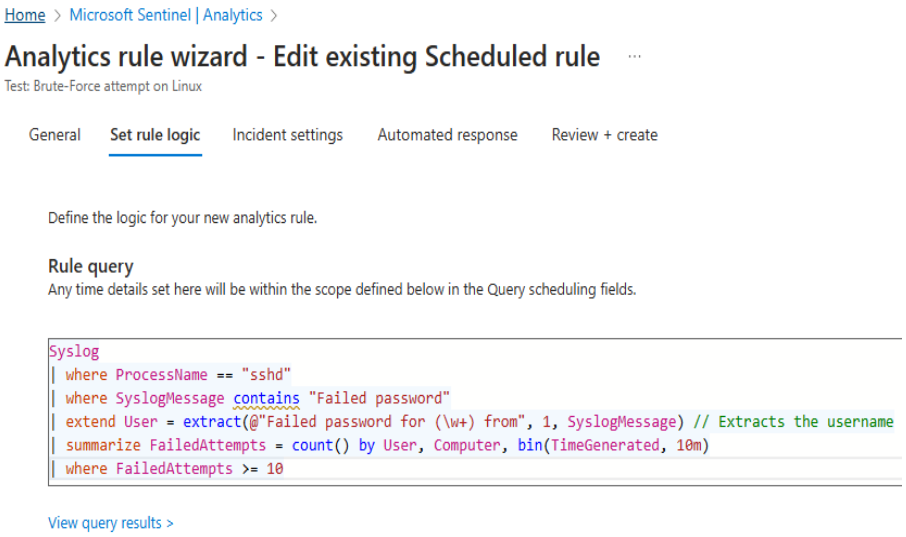
4. Analytics Rule Configuration

4.1 Scheduled Query Rules for Brute Force Detection

Windows Endpoint: The KQL query triggers an alert if more than 10 failed login attempts occur within 10 minutes.

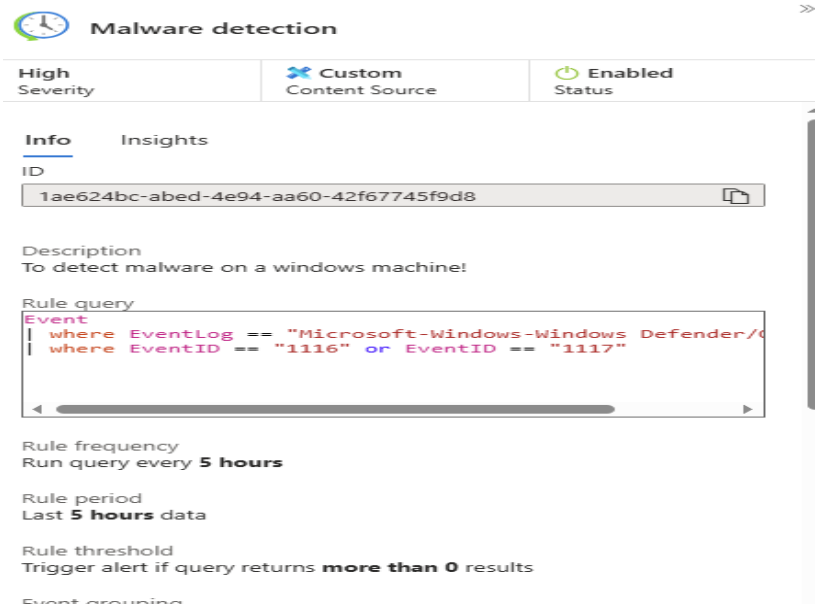


Linux Endpoint: The KQL query triggers an alert if more than 10 failed ssh login attempts occur within 10 minutes.

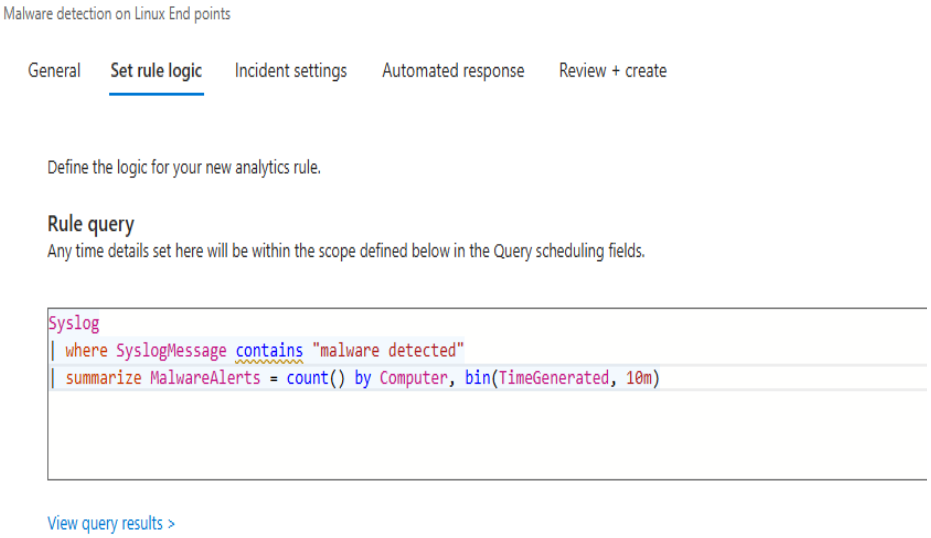


4.2 Scheduled Query Rules for Malware activity Detection

Windows Endpoint: The Rule detects windows event IDs 1116 and 1117 which are triggered when Windows Defender detects malware, potentially unwanted programs and when Windows Defender successfully removes or quarantines a detected threat respectively.



Linux End point: It triggers an alert when malware detection events are logged.

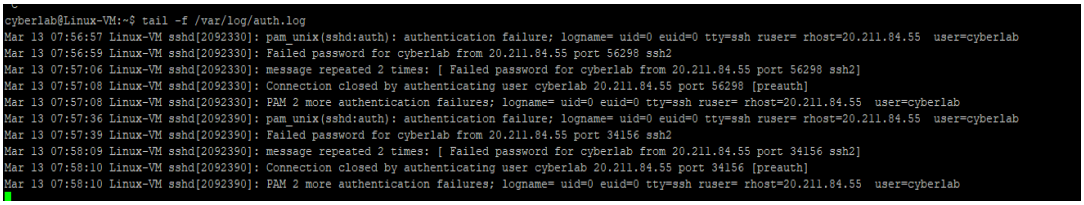


5. Results

5.1 Simulated Brute Force Attack

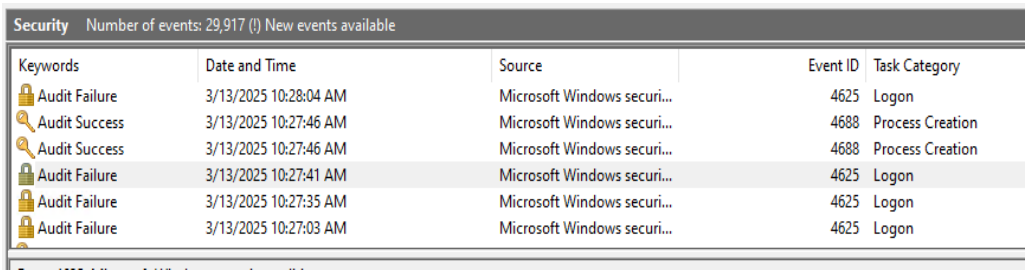
Linux Machine

Logs were generated by trying multiple failed login attempts on Linux machines, which can be seen in /var/log/auth.log.



Windows machine

Failed logon events logged in the event viewer.



5.2 Simulated Malware Activity

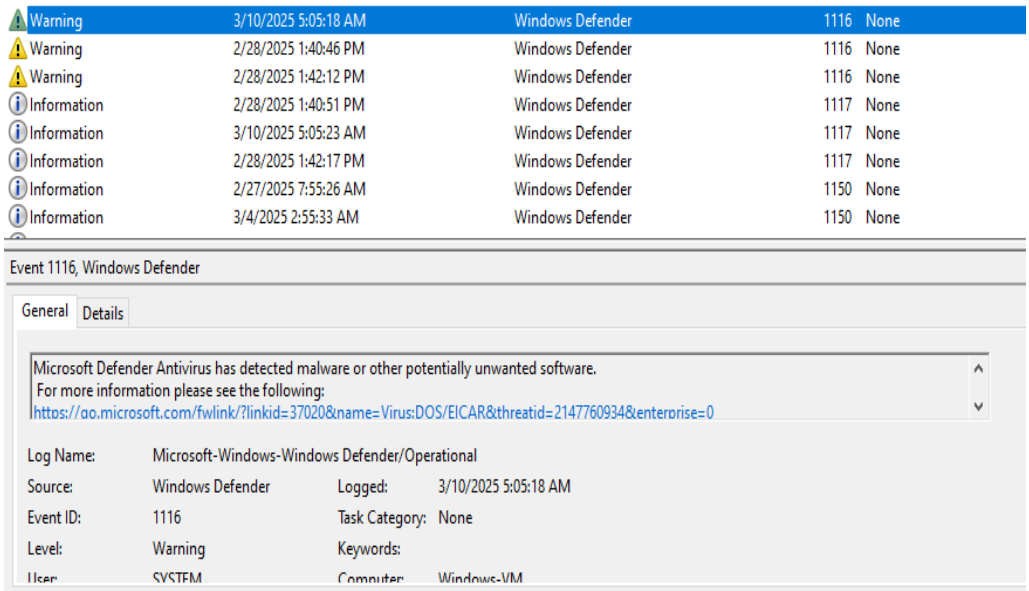
Linux Machine

The malware simulation tool EICAR test file was used to generate malware detection logs on Windows and Linux endpoints. The screenshot below shows the results of file scanning by Clamscan.

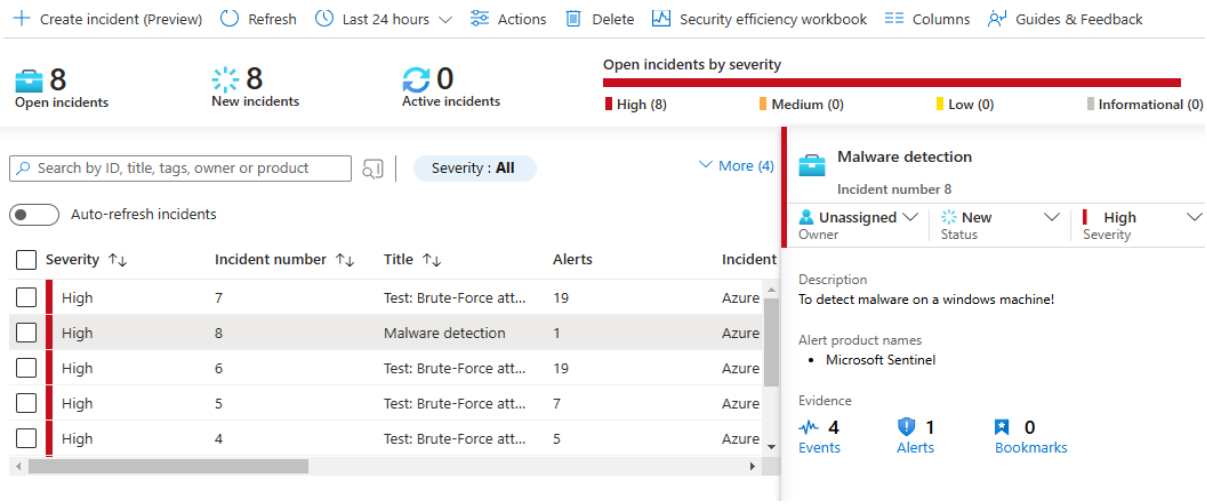
```
cyberlab@Linux-VM:~$ wget -O eicar.com.txt https://secure.eicar.org/eicar.com.txt
--2025-03-13 08:28:56--  https://secure.eicar.org/eicar.com.txt
Resolving secure.eicar.org (secure.eicar.org)... 89.238.73.97, 2a00:1828:1000:2497::2
Connecting to secure.eicar.org (secure.eicar.org)|89.238.73.97|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 68 [text/plain]
Saving to: 'eicar.com.txt'

eicar.com.txt                           100%[=========================================

2025-03-13 08:28:57 (69.6 MB/s) - 'eicar.com.txt' saved [68/68]

cyberlab@Linux-VM:~$ ls
 Forwarder_AMA_installer.py   eicar.txt                 'index.html?wpdmdl=8842'
 eicar.com.txt               'index.html?LinkId=828603'   omsagent-1.19.0-0.universal.x64.sh
cyberlab@Linux-VM:~$ clamscan eicar.com.txt
/home/cyberlab/eicar.com.txt: Win.Test.EICAR_HDB-1 FOUND

----------- SCAN SUMMARY -----------
Known viruses: 8704696
Engine version: 0.103.12
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 26.258 sec (0 m 26 s)
Start Date: 2025:03:13 08:29:11
End Date:   2025:03:13 08:29:37
cyberlab@Linux-VM:~$ cat eicar.comsudo nano /etc/clamav/clamd.conf
```

Windows Machine

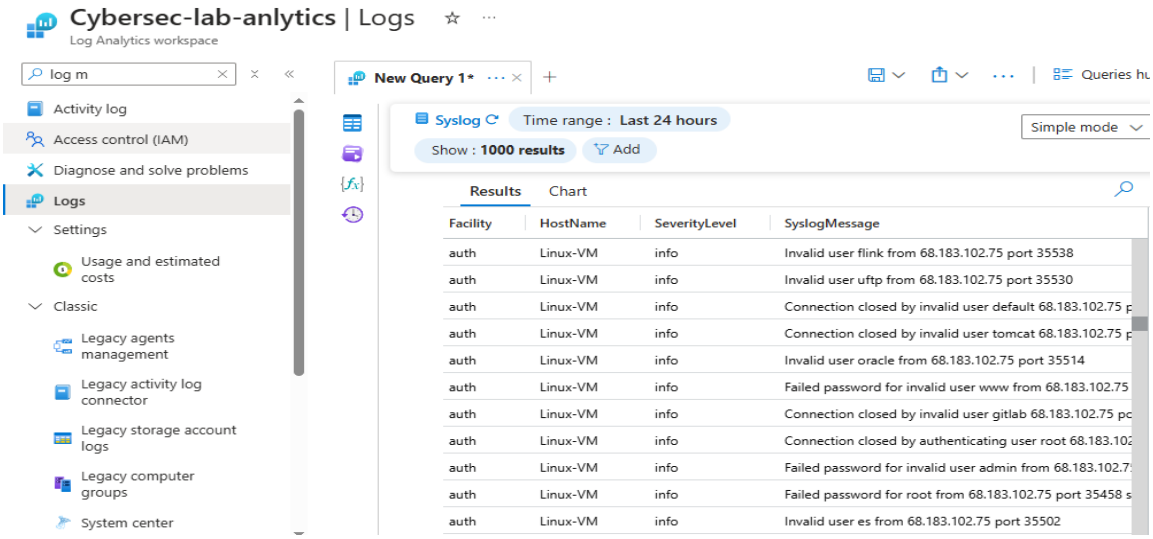
The below screenshot shows the event IDs 1116 and 1117 generated by Microsoft Defender on Windows machines.



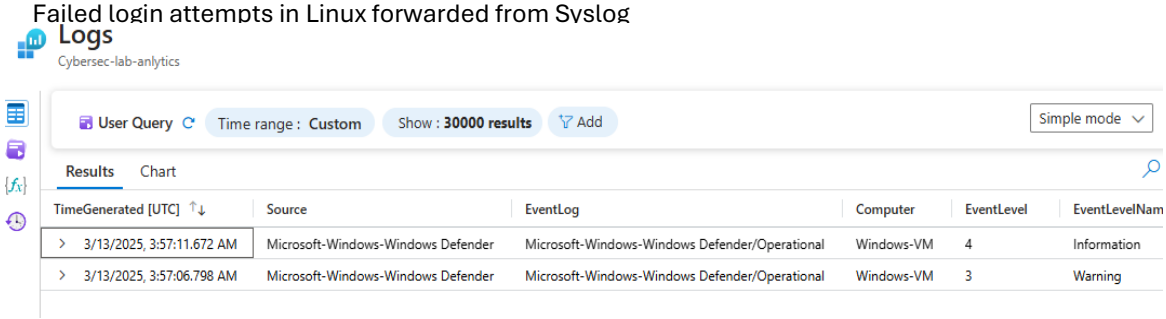
5.3 Incidents logged by Sentinel



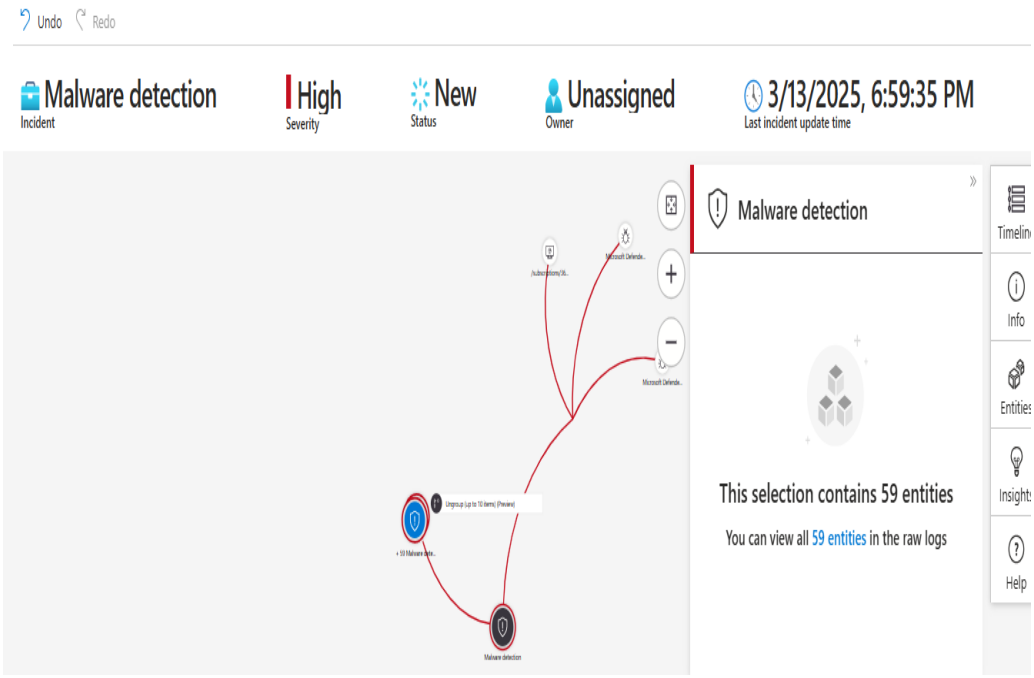
Brute force and Malware Incidents logged in Sentinel.



Failed login attempts in Linux forwarded from Syslog



Malware activity events 1116 and 1117 forwarded from Defender on Windows VM



Infographic diagram depicting malware detection

5.4 Malicious Activities Identified During Monitoring

Detection of Suspicious IP Addresses

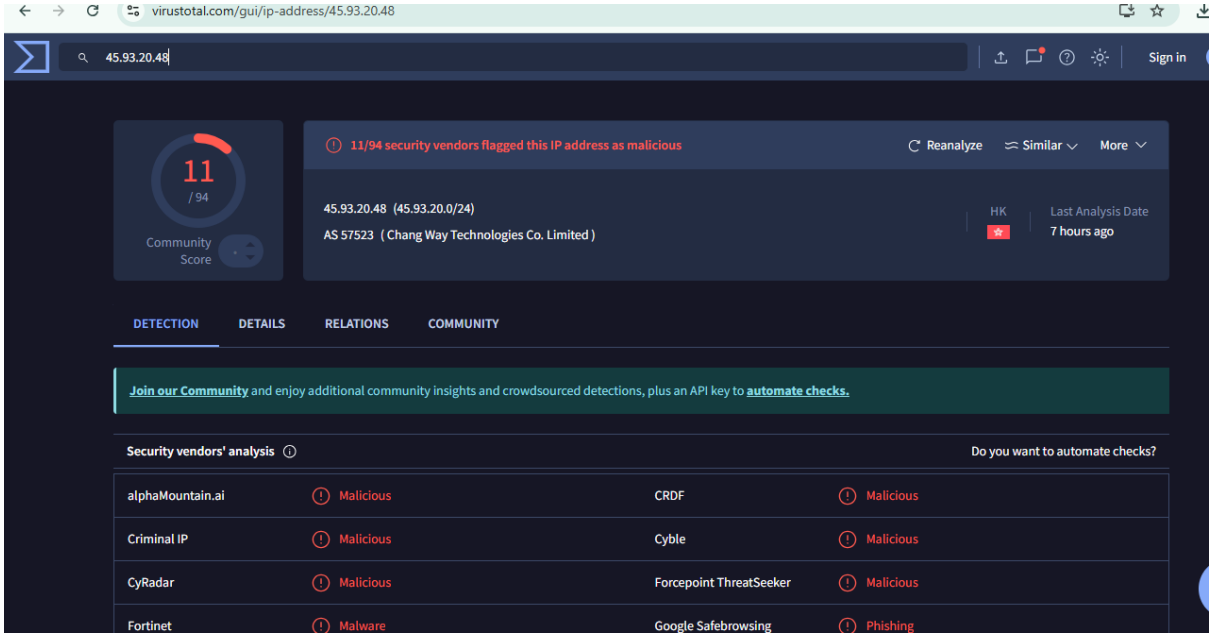
- During the monitoring phase, multiple source IP addresses involved in brute-force attempts were logged and analysed using VirusTotal. The analysis confirmed that these IPs were associated with known malicious actors.

Verification Process

- Further cross-referencing revealed links between these IP addresses, botnet activities, and large-scale brute-force campaigns.

Mitigation Actions

- To contain the threat, the identified IPs were promptly blocked at the firewall level. Additionally, alerts were escalated to the Security Operations Center (SOC) team for in-depth investigation and further response.



6. Key Insights

- Azure Sentinel's built-in analytics effectively detect brute force attacks and malware across multiple platforms.
- Integrating data ingestion from various endpoints enhances security visibility and threat detection.
- Automating response actions boosts operational efficiency and strengthens incident response.

7. Conclusion

The implementation of Azure Sentinel demonstrated its effectiveness in detecting and responding to cybersecurity threats, particularly brute force attacks and malware activities. By collecting and analysing endpoint logs, applying analytics rules, and identifying security threats, Azure Sentinel proved to be a valuable tool for proactive threat detection and mitigation. These findings highlight its real-world applicability in enhancing security operations, reinforcing its role as a critical component in modern cybersecurity defence strategies.

8. Next Steps

- Expand data integration to include more endpoints for better visibility.
- Refine analytics rules to improve detection accuracy and reduce false positives.
- Automate response actions using playbooks to enhance operational efficiency.
- Continuously monitor and tune detection and response mechanisms.
- Conduct penetration testing to evaluate security effectiveness.

- Improve reporting and dashboards for better insights.

9. References

- Microsoft Azure Sentinel Documentation: https://learn.microsoft.com/en-us/azure/sentinel/
- CloudSkew for Architecture diagrams: https://www.cloudskew.com/