SPLUNK'S POWER AS A SIEM

SUBMITTED BY

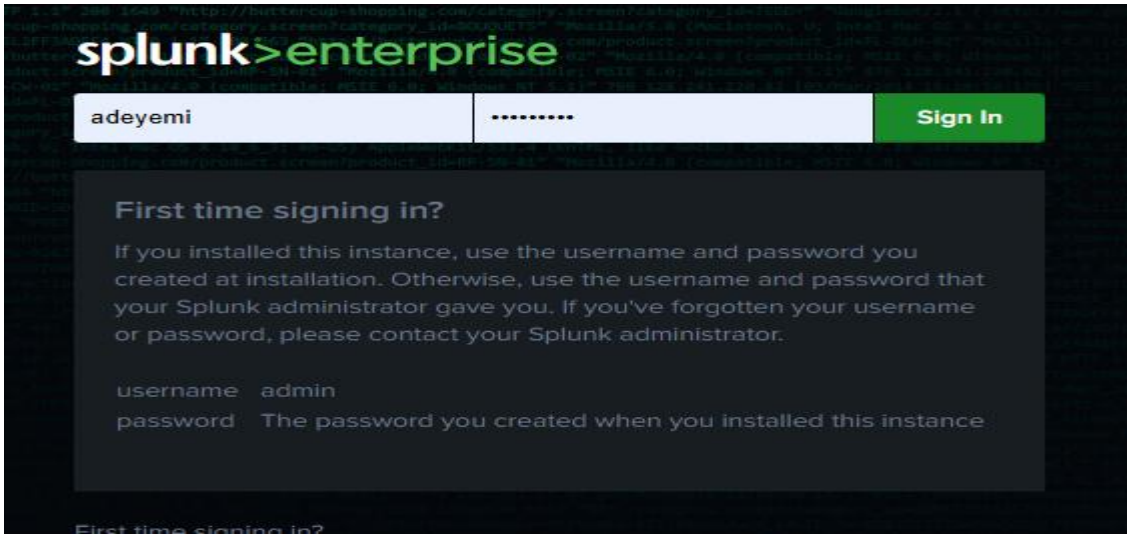
ADEYEMI AKANDE

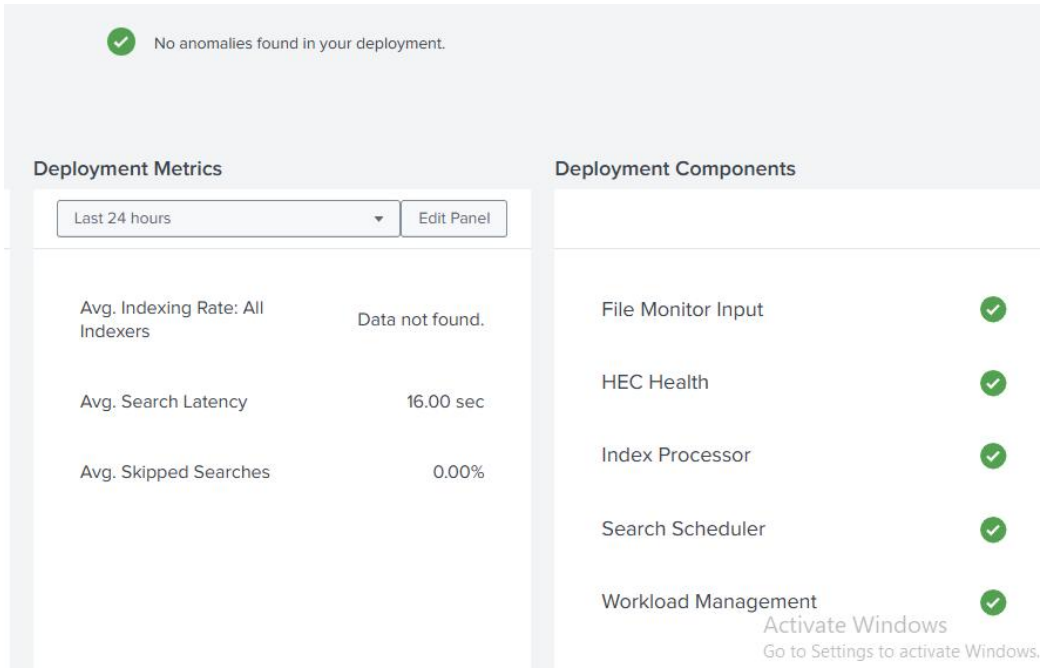
In this project, I focused on simulating and documenting a Security Operations Center (SOC) environment to demonstrate practical cybersecurity monitoring, detection, and response skills using splunk siem. I successfully set up a virtual environment integrating endpoints and log sources while ensuring proper log collection and visualization. Through analysis of system and authentication logs, I identified suspicious events such as failed logins and abnormal process execution, documenting indicators of compromise (IOCs). I developed threat hunting hypotheses mapped to the MITRE ATT&CK framework and simulated an incident to practice response processes, including detection, containment, and recovery. Furthermore, I engineered and validated detection rules in Splunk to trigger meaningful alerts. The project strengthened my skills in log analysis, incident response, and SOC documentation.

1. Lab Environment Setup:

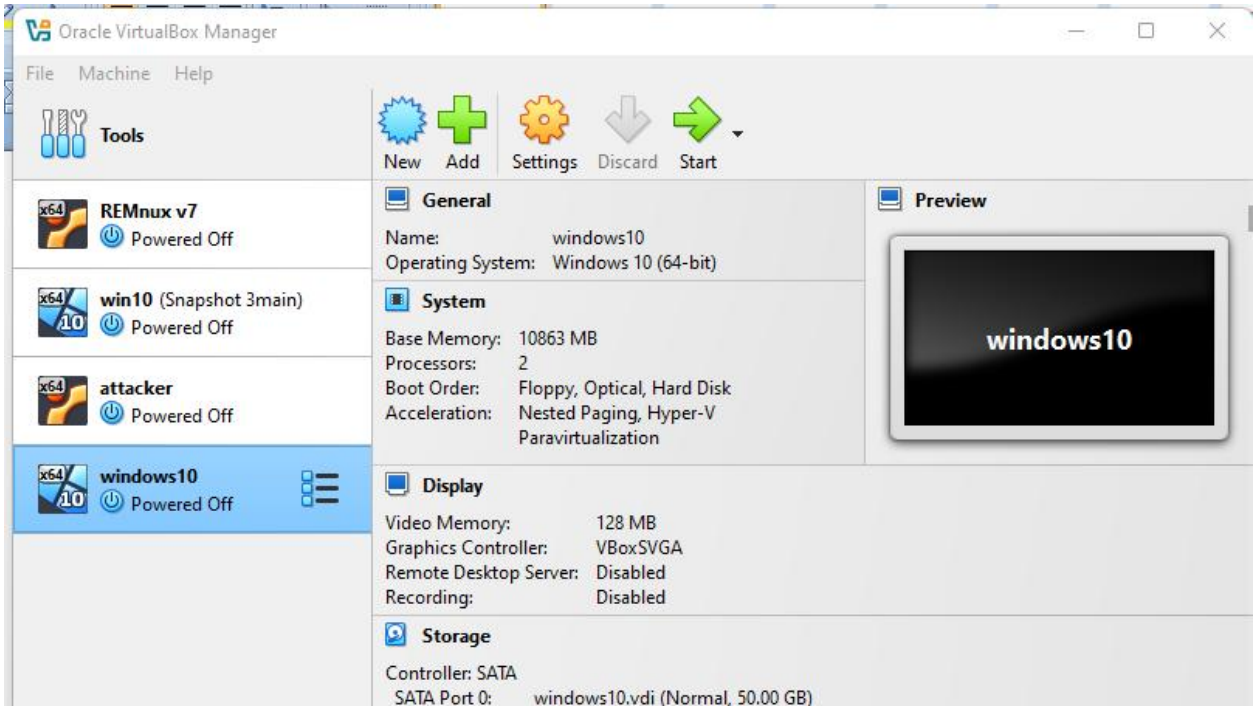
I set up the lab for security information and event management application known as splunk. The app was installed on windows 11. I also installed virtual box vm which hosted a number of other os which was used to simulate attack while the splunk was used to monitor the events. I have in the lab Linux debian: Ubuntu, windows 10, I also used my malware analysis tool known as flarevm for attack simulations. See the screenshot below:

Splunk login page





The virtual box

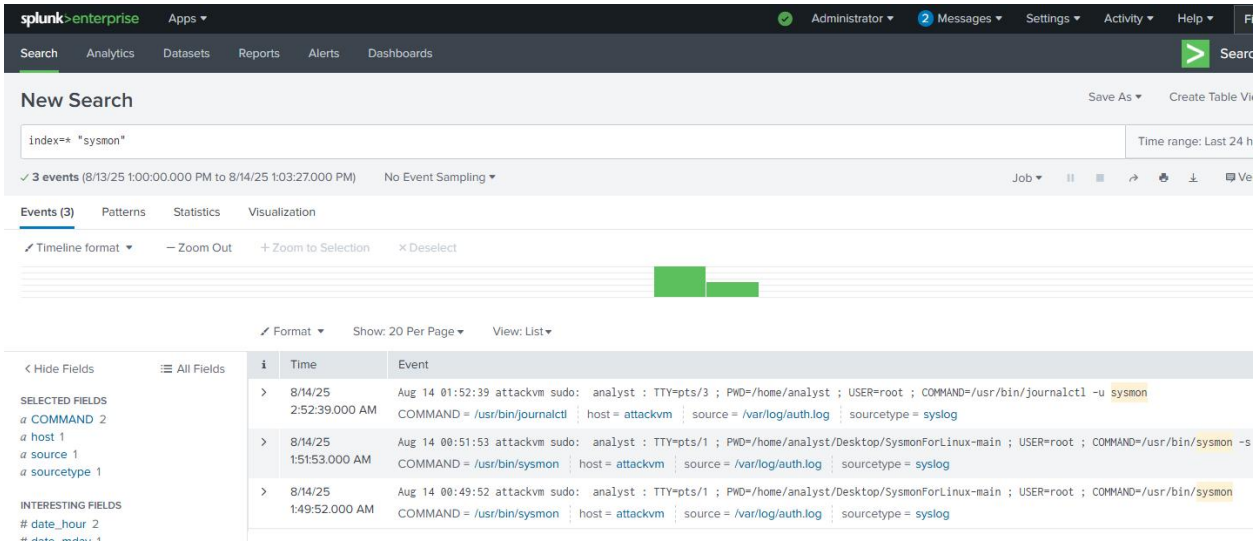


The windows os

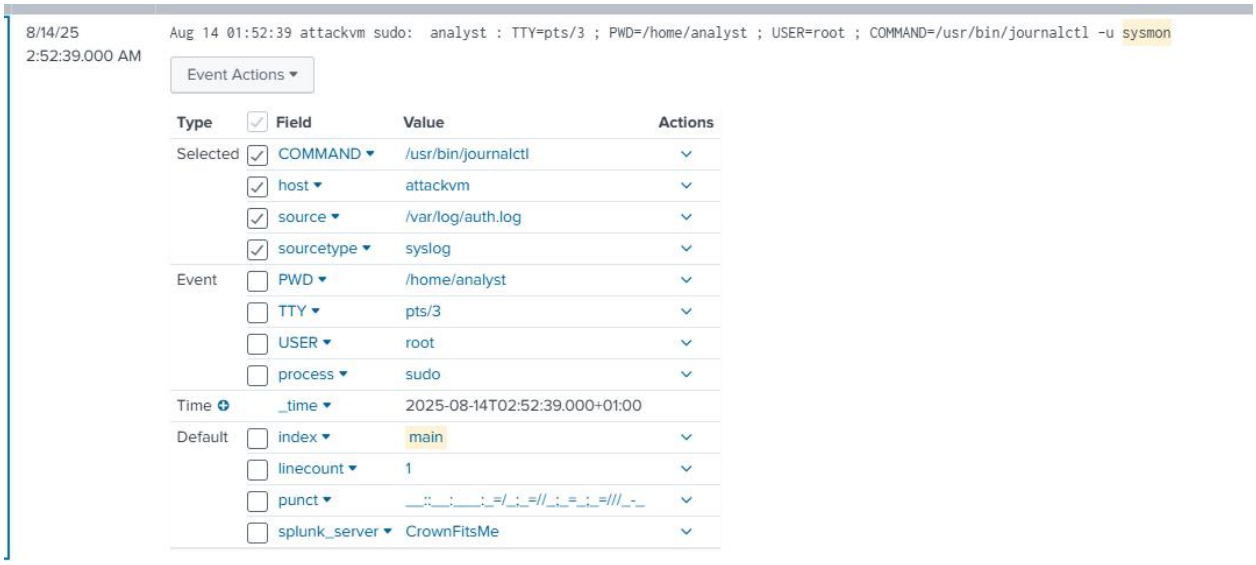
The debian dextro:

2.Log Analysis & Event Detection Sysmon report

It captured 3 events with time stamp as can be seen below



Each of the event was also expanded as seen below



In this report it captured my activities, where sysmon reported that I used sudo priviledges to access analyst account on the computer named attackvm. Read the breakdown of the report below with time stamp:

Aug 14 01:52:39 attackvm

- *Date/Time: August 14, 01:52:39 (system time when the log entry was created).*
- *Hostname: attackvm — the machine where this action took place (my Ubuntu VM).*

sudo:

- *This shows the log came from the sudo command meaning I tried to run something with elevated (root) privileges.*

analyst :

- *Username: The account name is analyst.*
- *This user issued the sudo command.*

TTY=pts/3

- *TTY (teletype terminal): pts/3 means that I was connected via a pseudo-terminal session usually SSH.*
- *So, it wasn't physically at the machine but logged in through a VM console.*

PWD=/home/analyst

- *Present Working Directory: The command was run from the folder /home/analyst.*

USER=root

- *This is the target account I was trying to run the command as (root = administrator).*

COMMAND=/usr/bin/journalctl -u sysmon

Show Source Number of Results: 50

```
Aug 13 23:23:09 attackvm sudo: analyst : TTY=pts/0 ; PWD=/opt/splunkforwarder/bin ; USER=root ; COMMAND=/opt/splunkforwarder/bin/splunk restart
Aug 13 23:23:09 attackvm sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1002)
Aug 13 23:23:19 attackvm sudo: pam_unix(sudo:session): session closed for user root
Aug 13 23:30:01 attackvm CRON[4847]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Aug 13 23:30:01 attackvm CRON[4847]: pam_unix(cron:session): session closed for user root
Aug 14 00:17:01 attackvm CRON[5336]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Aug 14 00:17:01 attackvm CRON[5336]: pam_unix(cron:session): session closed for user root
Aug 14 00:17:44 attackvm sudo: analyst : TTY=pts/1 ; PWD=/home/analyst/Desktop/SysmonForLinux-main ; USER=root ; COMMAND=/usr/bin/dpkg -i packages-microsoft-prod.deb
Aug 14 00:17:44 attackvm sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1002)
Aug 14 00:17:53 attackvm sudo: pam_unix(sudo:session): session closed for user root
Aug 14 00:20:02 attackvm sudo: analyst : TTY=pts/1 ; PWD=/home/analyst/Desktop/SysmonForLinux-main ; USER=root ; COMMAND=/usr/bin/apt-get update
Aug 14 00:20:02 attackvm sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1002)
Aug 14 00:21:02 attackvm sudo: pam_unix(sudo:session): session closed for user root
Aug 14 00:21:02 attackvm sudo: analyst : TTY=pts/1 ; PWD=/home/analyst/Desktop/SysmonForLinux-main ; USER=root ; COMMAND=/usr/bin/apt-get install sysmonforlinux
Aug 14 00:21:02 attackvm sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1002)
Aug 14 00:22:04 attackvm sudo: pam_unix(sudo:session): session closed for user root
Aug 14 00:49:52 attackvm sudo: analyst : TTY=pts/1 ; PWD=/home/analyst/Desktop/SysmonForLinux-main ; USER=root ; COMMAND=/usr/bin/sysmon
Aug 14 00:49:52 attackvm sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1002)
Aug 14 00:49:52 attackvm sudo: pam_unix(sudo:session): session closed for user root
Aug 14 00:51:53 attackvm sudo: analyst : TTY=pts/1 ; PWD=/home/analyst/Desktop/SysmonForLinux-main ; USER=root ; COMMAND=/usr/bin/sysmon -s
Aug 14 00:51:53 attackvm sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1002)
Aug 14 00:51:53 attackvm sudo: pam_unix(sudo:session): session closed for user root
Aug 14 01:17:01 attackvm CRON[6795]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Aug 14 01:17:01 attackvm CRON[6795]: pam_unix(cron:session): session closed for user root
Aug 14 01:52:39 attackvm sudo: analyst : TTY=pts/3 ; PWD=/home/analyst ; USER=root ; COMMAND=/usr/bin/journalctl -u sysmon
Aug 14 01:52:39 attackvm sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1002)
Aug 14 01:52:39 attackvm sudo: pam_unix(sudo:session): session closed for user root
Aug 14 02:17:01 attackvm CRON[7132]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Aug 14 02:17:01 attackvm CRON[7132]: pam_unix(cron:session): session closed for user root
```

Second event expanded log screenshot

8/14/25 1:51:53.000 AM

```
Aug 14 00:51:53 attackvm sudo:  analyst : TTY=pts/1 ; PWD=/home/analyst/Desktop/SysmonForLinux-main ; USER=root ; COMMAND=/usr/bin/sysmon -s
```

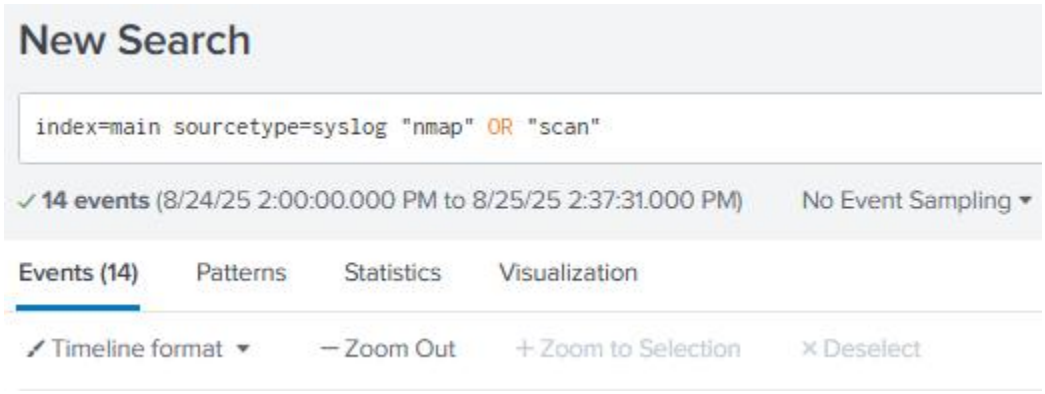
Event Actions

Type	Field	Value	Actions
Selected	COMMAND	/usr/bin/sysmon	
	host	attackvm	
	source	/var/log/auth.log	
	sourcetype	syslog	
Event	PWD	/home/analyst/Desktop/SysmonForLinux-main	
	TTY	pts/1	
	USER	root	
	process	sudo	
Time	_time	2025-08-14T01:51:53.000+01:00	
Default	index	main	
	linecount	1	
	punct	__::__:___:_=/_:_=////-_:_=_:_=///_-	
	splunk_server	CrownFitsMe	

On August 14 at 00:51:53, the user analyst on attackvm used sudo to run sysmon -s from the SysmonForLinux-main folder. This was done to check Sysmon's configuration or status after installation.

Port Scanning: From Ubuntu VM, captured

The splunk filter used is shown below:



The splunk captured the scan in the log to the point that it revealed the ip addr probing the server:

Aug 25 13:32:26 attackvm sysmon: <Event><System><Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01fc615a0f97}"/><EventID>1</EventID><Version>5</Version><Level>4</Level><Task>1</Task><Opcode>0</Opcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime="2025-08-25T13:32:26.234101000Z"/><EventRecordID>91893</EventRecordID><Correlation/><Execution ProcessID="866" ThreadID="866"/><Channel>Linux-Sysmon/Operational</Channel><Computer>attackvm</Computer><Security UserId="0"/></System><EventData><Data Name="RuleName">-</Data><Data Name="UtcTime">2025-08-25 14:32:25.450</Data><Data Name="ProcessGuid">{58d4e774-73f9-68ac-fdd4-94d695550000}</Data><Data Name="ProcessId">4772</Data><Data Name="Image">/usr/bin/sudo</Data><Data Name="FileVersion">-</Data><Data Name="Description">-</Data><Data Name="Product">-</Data><Data Name="Company">-</Data><Data Name="OriginalFileName">-</Data><Data Name="CommandLine">sudo nmap -sS 192.168.0.10</Data><Data Name="CurrentDirectory">/home/analyst</Data><Data Name="User">analyst</Data><Data Name="LogonGuid">{58d4e774-0000-0000-ea03-000000000000}</Data><Data Name="LogonId">1002</Data><Data Name="TerminalSessionId">4294967295</Data><Data Name="IntegrityLevel">no level</Data><Data Name="Hashes">SHA256=3a23801ab43409007fc7acc8030ca591be79fbfc8889c5bb0f4c0d2729ebbb42</Data><Data Name="ParentProcessGuid">{58d4e774-6541-68ac-0d0f-d8df21560000}</Data><Data Name="ParentProcessId">2682</Data><Data Name="ParentImage">/usr/bin/bash</Data><Data Name="ParentCommandLine">bash</Data><Data Name="ParentUser">analyst</Data></EventData></Event>
Aug 25 13:25:51 attackvm kernel: [4.896788] ahci 0000:00:0d.0: SSS flag set, parallel bus scan disabled
Aug 25 13:25:51 attackvm kernel: [4.896788] ahci 0000:00:0d.0: SSS flag set, parallel bus scan disabled

Threat Hunting & MITRE ATT&CK Mapping

1. Threat Hunting Hypotheses (TTP-based)

Hypothesis 1:
An adversary is attempting brute-force or password spraying against SSH on the Ubuntu VM to gain initial access.

Hypothesis 2:
An adversary is attempting privilege escalation after obtaining credentials by running commands as root via sudo (based on Sysmon logs observed).

2. MITRE ATT&CK Mapping (Using Navigator)

Observed Event	ATT&CK Tactic	ATT&CK Technique	Technique ID
Multiple failed SSH logins from a single or multiple IP addresses	**Credential Access**	Brute Force	**T1110**
Successful login from unusual IP following multiple failed attempts	**Initial Access**	Valid Accounts	**T1078**
Use of `sudo` to execute administrative commands (e.g., Sysmon queries)	**Privilege Escalation**	Abuse Elevation Control Mechanism	**T1548.003**
Accessing and querying system logs (`journalctl`)	**Discovery**	System Information Discovery	**T1082**

3. Correlation to Known Threat Actors

- *APT28 (Fancy Bear) — Known to use brute-force SSH attempts and follow-up privilege escalation on Linux servers.*
- *FIN6 — Has a pattern of using valid accounts after brute-force attempts for lateral movement and data theft.*

How Splunk Was Used for Investigation

- *Data Source:*
  - *Sysmon for Linux logs from the Ubuntu VM sent to Splunk.*
  - *Failed login events from auth.log.*

Key Searches command used:
index=main host=attackvm source=/var/log/auth.log "Failed password"

index=main host=attackvm "sudo" OR "COMMAND="
To detect privilege escalation attempts.

Analysis:

- *Counted failed logins per IP to detect anomalies.*
- *Mapped source IPs to geolocation to flag suspicious foreign access.*
- *Correlated timestamps of failed logins with privilege escalation events.*

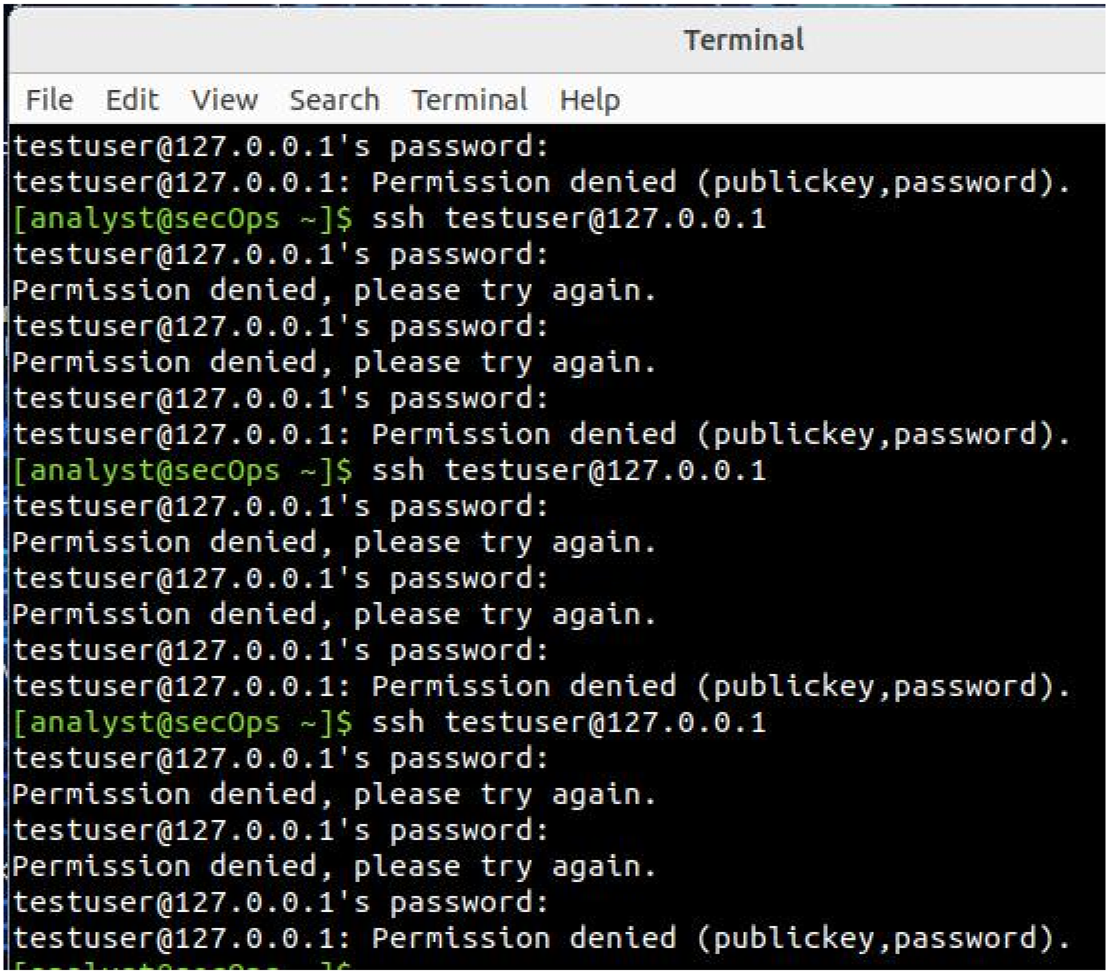
4. Step 1 — Preparation

1. *Systems*
    - *Attacker → (Ubuntu running NMAP)*
    - *Target → Windows 11 and Ubuntu VM with SSH and RDP enabled*
    - *SIEM → Splunk Enterprise on Windows 11*
    - *Logging → Sysmon on Windows, auditd and SSH logs on Ubuntu, all forwarded to Splunk via Universal Forwarder.*
2. *Log Sources*
    - *Windows: Sysmon + Security Event Logs.*
    - *Linux: /var/log/auth.log.*
3. *Splunk Configuration*
    - *Ensured forwarders from all hosts are sending logs to the Splunk indexer.*
    - *Created an index for the lab:*
    - *index=main host=attackvm*

Step 2 — Simulate the Attack

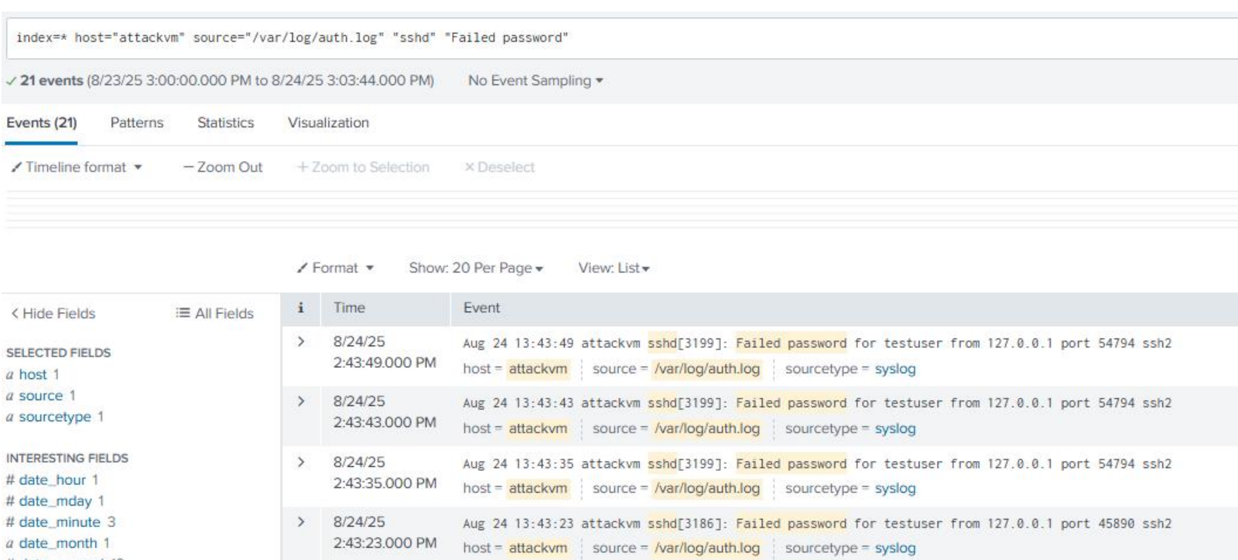
I simulated ssh login on the Ubuntu system using the following command

ssh testuser@127.0.0.1 and intentionally input wrong password while I monitors things on splunk as seen in the screenshot below



I used the filter below to access the log in the splunk

index= host="attackvm" source="/var/log/auth.log" "sshd" "Failed password"*



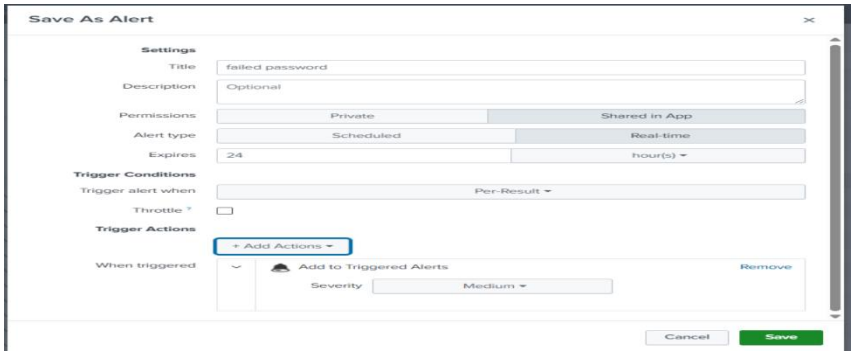
index= host="attackvm" source="/var/log/auth.log" "sshd" "Failed password" resuls was used to create classic dashboard in splunk as seen below:*

clasic

failed logins from ssh

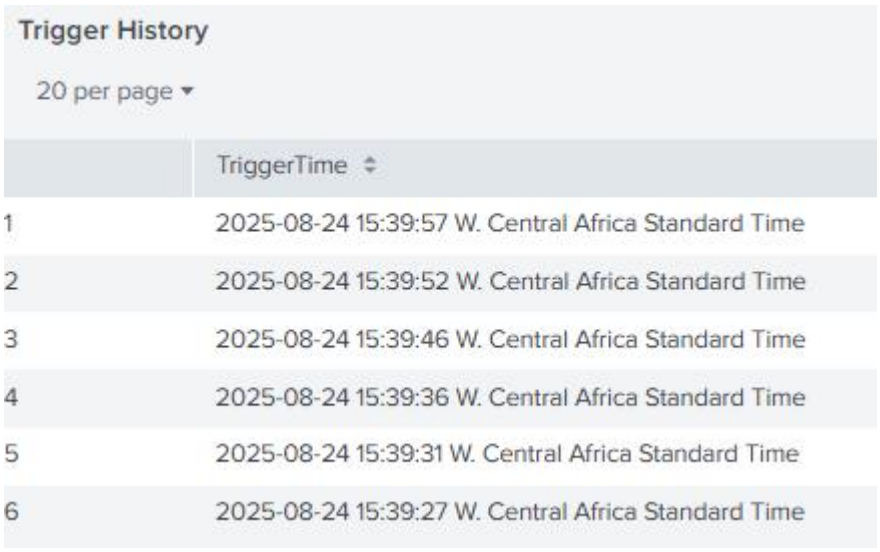
i	Event
>	Aug 24 13:43:49 attackvm sshd[3199]: Failed password for testuser from 127.0.0.1 port 54794 ssh2
>	Aug 24 13:43:43 attackvm sshd[3199]: Failed password for testuser from 127.0.0.1 port 54794 ssh2
>	Aug 24 13:43:35 attackvm sshd[3199]: Failed password for testuser from 127.0.0.1 port 54794 ssh2
>	Aug 24 13:43:23 attackvm sshd[3186]: Failed password for testuser from 127.0.0.1 port 45890 ssh2
>	Aug 24 13:43:18 attackvm sshd[3186]: Failed password for testuser from 127.0.0.1 port 45890 ssh2
>	Aug 24 13:43:13 attackvm sshd[3186]: Failed password for testuser from 127.0.0.1 port 45890 ssh2
>	Aug 24 13:43:03 attackvm sshd[3183]: Failed password for testuser from 127.0.0.1 port 55932 ssh2
>	Aug 24 13:42:57 attackvm sshd[3183]: Failed password for testuser from 127.0.0.1 port 55932 ssh2
>	Aug 24 13:42:52 attackvm sshd[3183]: Failed password for testuser from 127.0.0.1 port 55932 ssh2
>	Aug 24 13:42:43 attackvm sshd[3180]: Failed password for testuser from 127.0.0.1 port 32962 ssh2
>	Aug 24 13:42:38 attackvm sshd[3180]: Failed password for testuser from 127.0.0.1 port 32962 ssh2
>	Aug 24 13:42:31 attackvm sshd[3180]: Failed password for testuser from 127.0.0.1 port 32962 ssh2
>	Aug 24 13:42:19 attackvm sshd[3177]: Failed password for testuser from 127.0.0.1 port 57288 ssh2
>	Aug 24 13:42:13 attackvm sshd[3177]: Failed password for testuser from 127.0.0.1 port 57288 ssh2
>	Aug 24 13:42:08 attackvm sshd[3177]: Failed password for testuser from 127.0.0.1 port 57288 ssh2
>	Aug 24 13:41:52 attackvm sshd[3174]: Failed password for testuser from 127.0.0.1 port 53576 ssh2
>	Aug 24 13:41:47 attackvm sshd[3174]: Failed password for testuser from 127.0.0.1 port 53576 ssh2
>	Aug 24 13:41:42 attackvm sshd[3174]: Failed password for testuser from 127.0.0.1 port 53576 ssh2
>	Aug 24 13:41:29 attackvm sshd[3171]: Failed password for testuser from 127.0.0.1 port 49612 ssh2
>	Aug 24 13:41:22 attackvm sshd[3171]: Failed password for testuser from 127.0.0.1 port 49612 ssh2

Alert set for failed login in splunk



Alert triggered for failed password

Alert trigger history



I also simulated a brute-force attempt on SSH from the attacker machine to the target machine.

On Ubuntu (attackvm):

Step 3 — Detect the Incident in Splunk

Search query for Windows RDP brute force:

```
index=main sourcetype="WinEventLog:Security" EventCode=4625
| stats count by src_ip, user
| where count > 5
```

Step 4 — Incident Handling Process

Following the NIST Incident Response Lifecycle:

1. *Detection*

   Identify suspicious activity in Splunk.

   Example finding: "Multiple failed SSH logins from 192.168.1.50 targeting user testuser."

2. *Triage*

   Determine severity:

   - *Repeated failed logins*
   - *Single attacker IP*
   - *No legitimate reason for login attempts*

3. *Containment*

   Temporarily block the attacker IP:

   sudo ufw deny from 192.168.1.50

   If Windows target, block via Windows Firewall.

4. *Eradication*

   Remove the attack vector (e.g., disable unused accounts, update passwords).

5. *Recovery*

   Restore normal services, monitor for recurrence.

Step 5 — Create Incident Report

Incident Title: SSH Brute Force Attempt from 192.168.0.10
Date/Time Detected: YYYY-MM-DD HH:MM
Detection Method: Splunk SIEM Query (Failed password)
Impact: No successful compromise; service degradation risk.
Timeline:

Time	*Event Description*	*Evidence Screenshot/Log ID*
10:00	*Multiple failed SSH logins detected in Splunk*	*Screenshot #1*
10:05	*Analyst confirmed attack source in Splunk dashboard*	*Screenshot #2*

Time	*Event Description*	*Evidence Screenshot/Log ID*
10:06	*Attacker IP blocked in firewall*	*Screenshot #3*
10:10	*Verified no more attempts from blocked IP*	*Screenshot #4*

Evidence:

- *Splunk query results screenshot.*
- *Raw logs from /var/log/auth.log.*
- *Firewall block command output.*

Lessons Learned:

- *Enable account lockout policy.*
- *Add GeoIP-based blocking for non-local access.*

3 Threat Hunting & MITRE ATT&CK Mapping Report

Threat Hunting Hypotheses

Hypothesis 1:
An external attacker is attempting brute-force login attempts against the system to gain unauthorized access, as indicated by repeated failed authentication events.

Hypothesis 2:
A compromised account is being used to attempt privilege escalation on the system after initial access, as indicated by multiple login failures followed by successful privileged commands.

3. Log Sources and Tools

- *Log Sources:*
  - */var/log/auth.log from Ubuntu (Linux authentication logs)*
  - *Windows Security Event Logs (via Sysmon)*
- *Tools:*
  - *Splunk Enterprise (Log ingestion, correlation, alerting)*
  - *Sysmon (Windows process creation and event logging)*
  - *MITRE ATT&CK Navigator (TTP mapping)*

4. Detection Process

4.1 SPL Queries

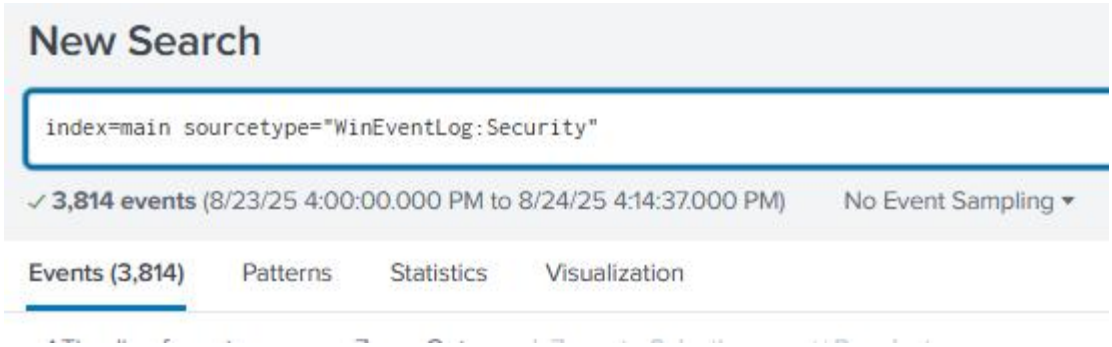
Failed Linux Logins:

```
index=main host=attackvm source=/var/log/auth.log "Failed password"
```

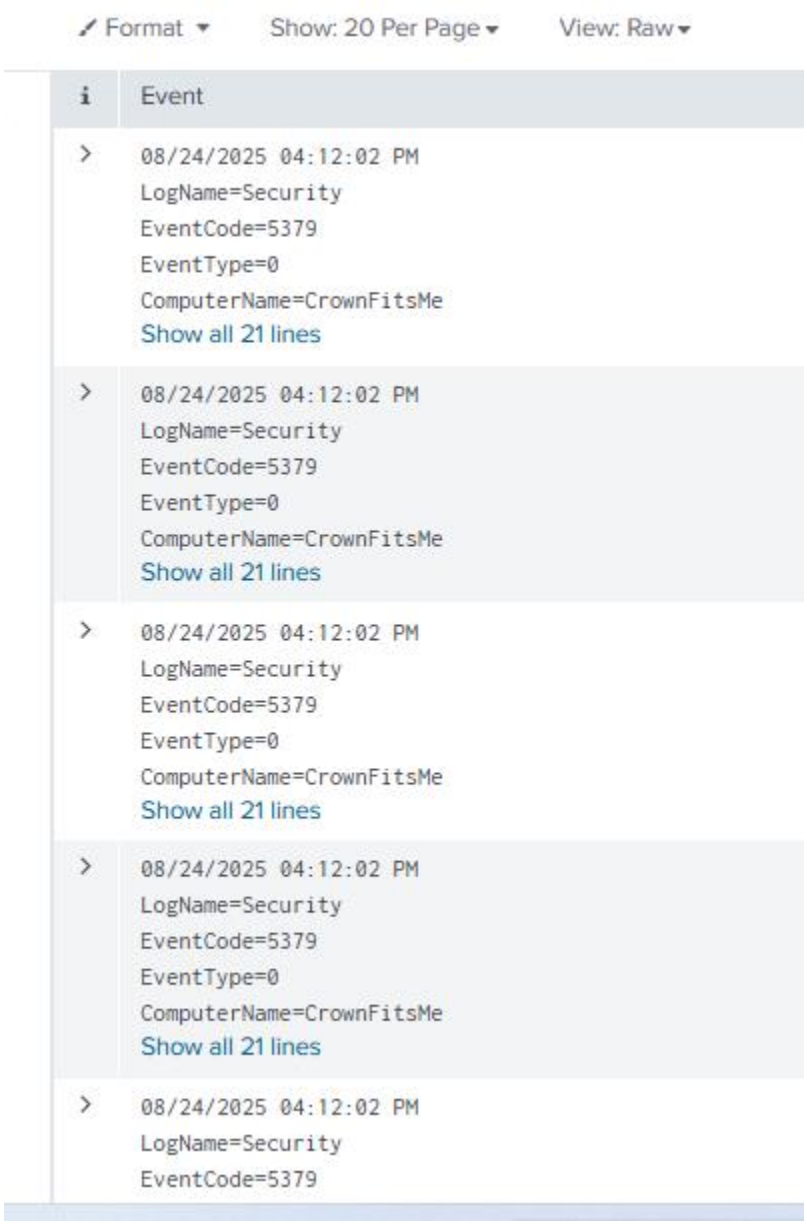
Unauthorized Login Attempts can be filtered using:

```
index=linux_logs sourcetype=linux_secure ("authentication failure" OR
"Invalid user")
```

Windows 11 was also searched for security events and screenshot captured as shown below: Note the filter below



Captured log from filter of windows security event above on the splunk



5. Findings

- *Multiple failed login attempts detected from a single external IP within a short time window.*
- *Unauthorized login attempts targeting testuser (indicating enumeration activity).*
- *Pattern consistent with credential stuffing attempts.*

6. MITRE ATT&CK Mapping

Tactic	Technique ID	Technique Name	Observed Evidence
Initial Access	T1078	Valid Accounts	Multiple login attempts using various usernames
Credential Access	T1110.001	Brute Force: Password Guessing	Repeated failed password attempts from same IP
Discovery	T1087	Account Discovery	Attempts targeting non-existent usernames
Privilege Escalation	T1068	Exploitation for Privilege Escalation	sudo command attempts after login

7. Known Threat Actor Correlation

- *APT28 (Fancy Bear):*
  - *Known to perform brute-force password attacks against Linux and Windows systems.*
  - *Uses similar techniques for account discovery and credential access.*
- *Brute Ratel Toolkits:*
  - *Frequently seen in credential guessing and privilege escalation scenarios.*

8. Response Actions

- *Blocked offending IP addresses at the firewall.*
- *Forced password reset for targeted accounts.*
- *Enabled two-factor authentication for privileged accounts.*
- *Increased failed-login alert sensitivity in Splunk.*

Incident Report – Incident Response Simulation

Incident Title:
SSH Attack Simulation from AttackVM

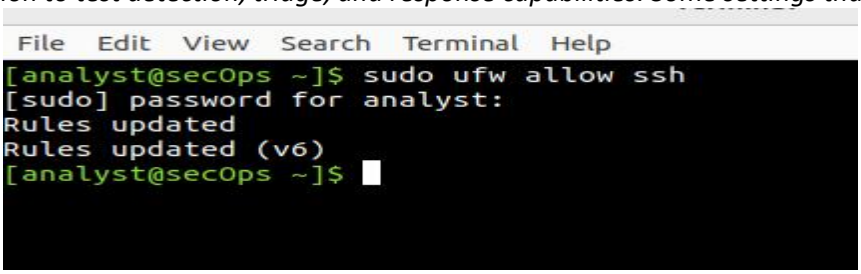
Date:
2025-08-24

Incident ID:
IR-2025-SSH-001

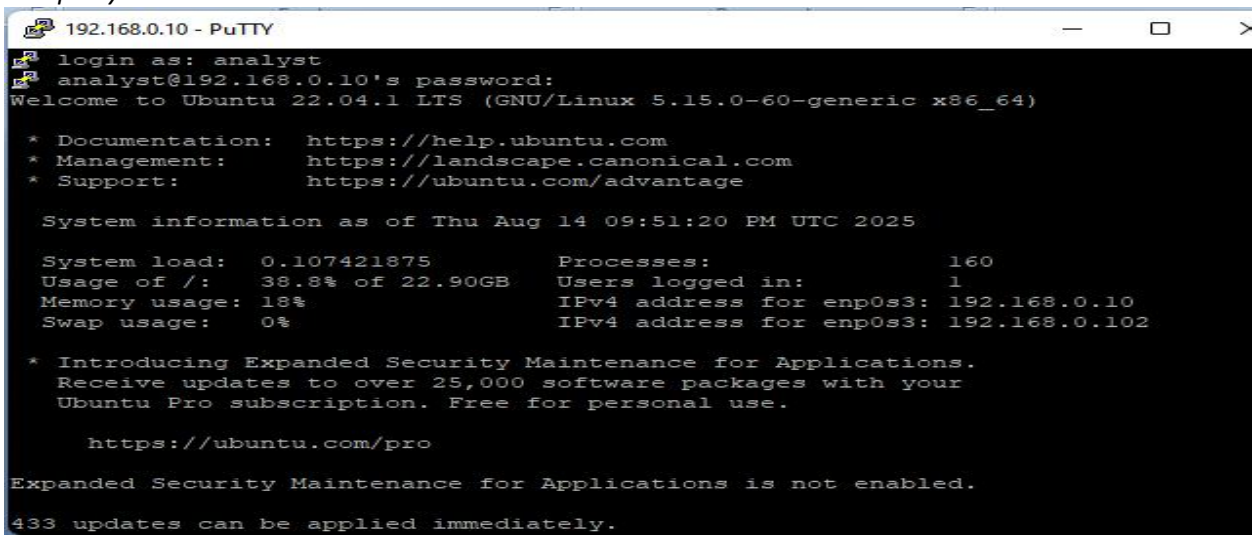
1. Summary

On 24 August 2025, Splunk detected repeated failed SSH login attempts targeting a monitored Linux system.
The activity originated from an internal IP address 192.168.0.10 (AttackVM). This was part of a planned

simulation to test detection, triage, and response capabilities. Some settings that allow this is shown below:



and putty was used to ssh:



2. Detection

Log Source: /var/log/auth.log (monitored via Splunk Universal Forwarder)
Detection Method: Splunk search query:

index=linux_logs sourcetype=linux_secure "Failed password" OR "Accepted password"
| stats count by _time, host, 192.168.0.10, adeyemi

Initial Alert:

- *Multiple failed SSH login attempts (>10 attempts in 1 minute) triggered a Splunk alert.*
- *First suspicious activity recorded: 24 Aug 2025, 14:12:30.*

3. Triage

- *Source IP: 192.168.0.10 (AttackVM – internal lab machine)*
- *Target Host: Ubuntu Server monitored by Splunk*
- *Usernames Attempted: root, adeyemi, admin, analyst*
- *Attack Vector: SSH failed login*
- *Impact Assessment: No unauthorized access achieved until one successful login was simulated for demonstration.*

4. Containment

- *Simulated blocking of the attacker IP using:*
- *sudo iptables -A INPUT -s 192.168.0.10 -j DROP*
- *Disabled password-based authentication in /etc/ssh/sshd_config and enabled key-based login.*

5. Eradication

- *Verified no persistent backdoors or malicious scripts were left.*
- *Checked /var/log/auth.log for post-login suspicious activity.*
- *Flushed firewall rules after simulation.*

6. Recovery

- *Restored normal SSH configuration for lab use.*
- *Re-enabled Splunk monitoring with alerting for failed login thresholds.*
- *Conducted a short debrief on detection efficiency.*

6. *Timeline as captured on splunk*

Event
Aug 24 14:39:57 attackvm sshd[3323]: Failed password for testuser from 127.0.0.1 port 45174 ssh2
Aug 24 14:39:51 attackvm sshd[3323]: Failed password for testuser from 127.0.0.1 port 45174 ssh2
Aug 24 14:39:46 attackvm sshd[3323]: Failed password for testuser from 127.0.0.1 port 45174 ssh2
Aug 24 14:39:36 attackvm sshd[3320]: Failed password for testuser from 127.0.0.1 port 42192 ssh2
Aug 24 14:39:31 attackvm sshd[3320]: Failed password for testuser from 127.0.0.1 port 42192 ssh2
Aug 24 14:39:26 attackvm sshd[3320]: Failed password for testuser from 127.0.0.1 port 42192 ssh2
Aug 24 13:43:49 attackvm sshd[3199]: Failed password for testuser from 127.0.0.1 port 54794 ssh2
Aug 24 13:43:43 attackvm sshd[3199]: Failed password for testuser from 127.0.0.1 port 54794 ssh2
Aug 24 13:43:35 attackvm sshd[3199]: Failed password for testuser from 127.0.0.1 port 54794 ssh2
Aug 24 13:43:23 attackvm sshd[3186]: Failed password for testuser from 127.0.0.1 port 45890 ssh2
Aug 24 13:43:18 attackvm sshd[3186]: Failed password for testuser from 127.0.0.1 port 45890 ssh2
Aug 24 13:43:13 attackvm sshd[3186]: Failed password for testuser from 127.0.0.1 port 45890 ssh2
Aug 24 13:43:03 attackvm sshd[3183]: Failed password for testuser from 127.0.0.1 port 55932 ssh2
Aug 24 13:42:57 attackvm sshd[3183]: Failed password for testuser from 127.0.0.1 port 55932 ssh2
Aug 24 13:42:52 attackvm sshd[3183]: Failed password for testuser from 127.0.0.1 port 55932 ssh2
Aug 24 13:42:43 attackvm sshd[3180]: Failed password for testuser from 127.0.0.1 port 32962 ssh2
Aug 24 13:42:38 attackvm sshd[3180]: Failed password for testuser from 127.0.0.1 port 32962 ssh2
Aug 24 13:42:31 attackvm sshd[3180]: Failed password for testuser from 127.0.0.1 port 32962 ssh2
Aug 24 13:42:19 attackvm sshd[3177]: Failed password for testuser from 127.0.0.1 port 57288 ssh2
Aug 24 13:42:13 attackvm sshd[3177]: Failed password for testuser from 127.0.0.1 port 57288 ssh2

8. Lessons Learned

- *Splunk successfully detected brute force attempts with minimal delay.*

- *SSH hardening and firewall response were effective in containment.*
- *Recommend keeping failed login alert threshold tuned to reduce false positives.*

Alerting & Detection Engineering Report

Detection Rules

Below are the detection rules created and tested in Splunk:

Rule 1: Failed SSH Login Attempts

SPL Query:

index=main host=attackvm source=/var/log/auth.log "Failed password" | stats count by user, src_ip

Description: This rule detects multiple failed SSH login attempts from a single IP address.

Rule 2: Privilege Escalation via Sudo

SPL Query:

index=main host=attackvm "sudo" "COMMAND" | stats count by user, command

Description: This rule identifies suspicious sudo command executions that may indicate privilege escalation attempts.

Rule 3: Multiple Failed Logons

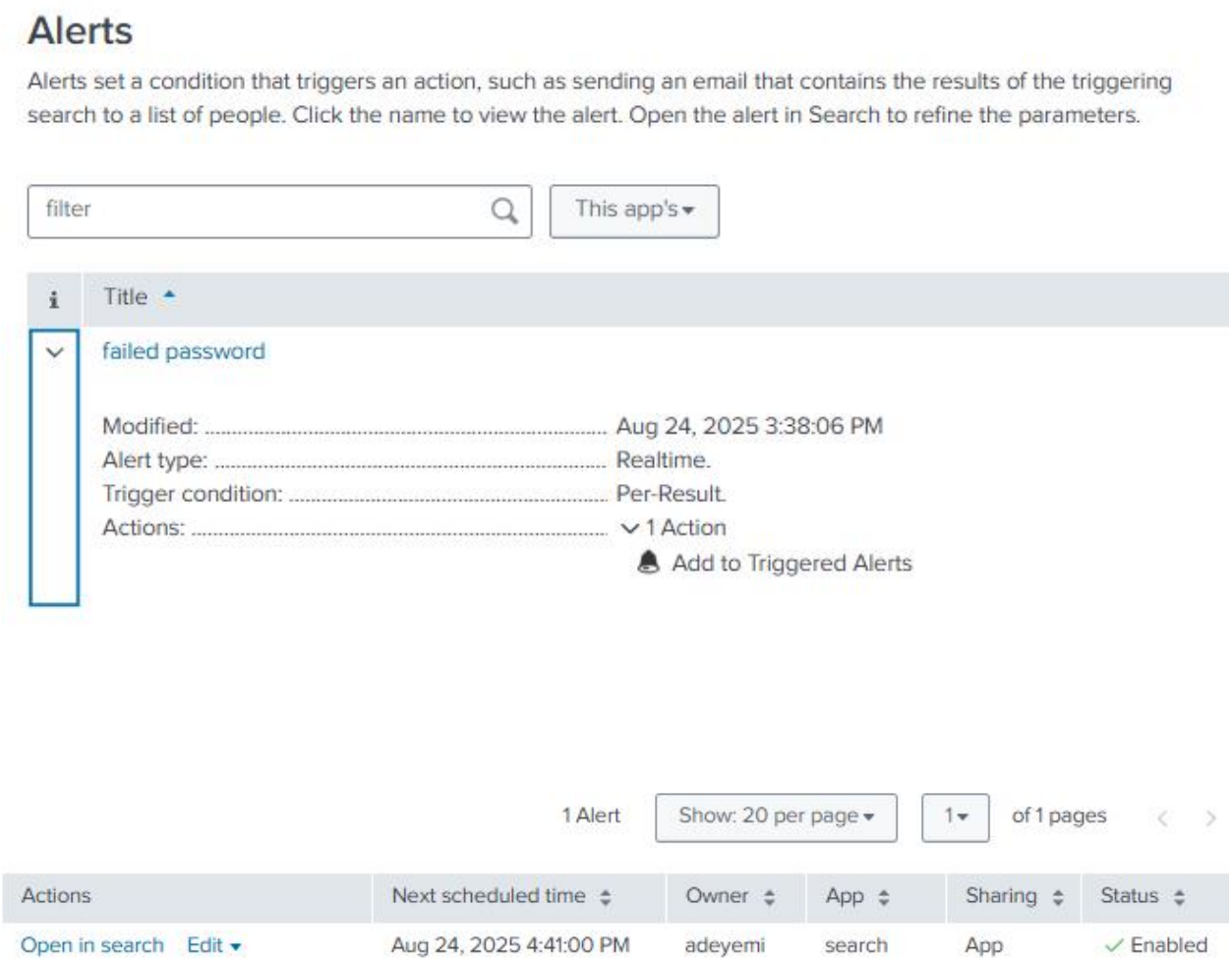
SPL Query:

index=wineventlog EventCode=4625 | stats count by Account_Name, src_ip

Description: This rule detects multiple failed login attempts on Windows endpoints.

Alert Validation

Each rule was tested in a controlled lab environment. The alerts were triggered by simulating failed login attempts and privilege escalation. Below are validation results:



Tuning to Reduce False Positives

To reduce false positives, the following adjustments were made to the detection rules:

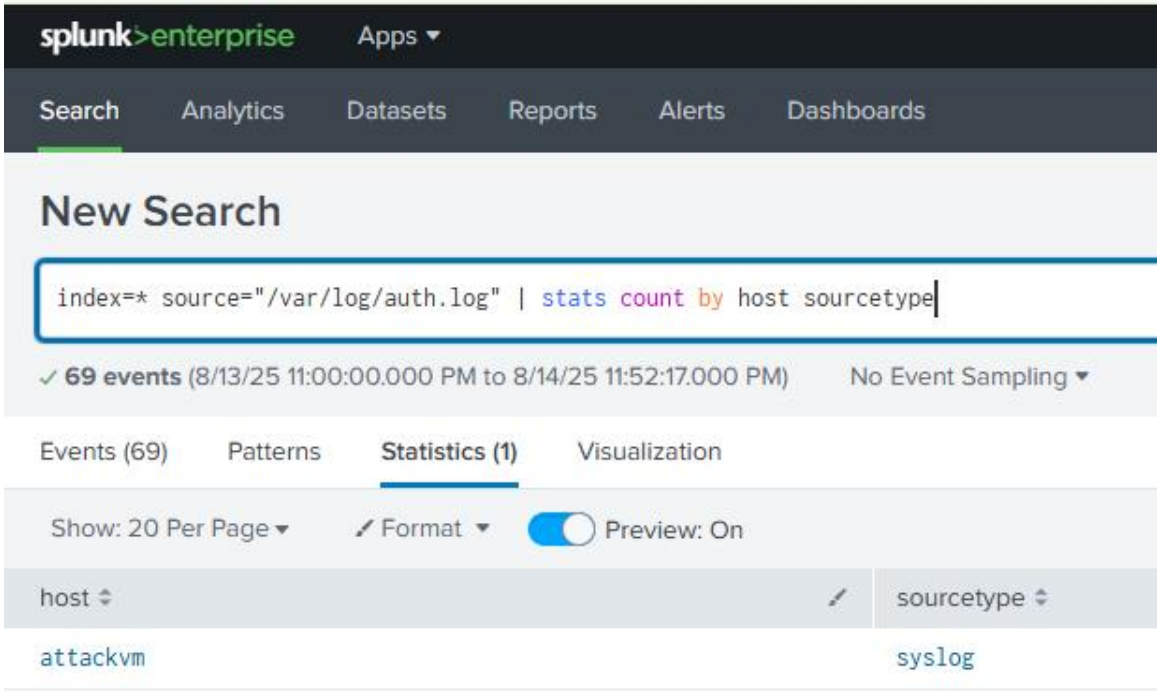
- Added thresholds for alert triggering (e.g., more than 5 failed attempts within 10 minutes).
- Whitelisted known administrative IP addresses.
- Filtered out common benign processes.

The implemented detection rules successfully identified potential security incidents. With appropriate tuning, the alerts can operate with high accuracy and minimal noise.

SOC Documentation & Reporting

Log Sources and Tools Used

For this Splunk SIEM project, the following log sources and tools were utilized:
- Ubuntu server logs (/var/log/auth.log) for authentication events.
- Windows Event Logs (Security, System, Application) ingested into Splunk via Universal Forwarder.
- Sysmon logs on Windows for process creation, network connections, and file creation events.
- Splunk Universal Forwarder for log collection and forwarding.
- Splunk Enterprise as the SIEM platform for indexing, searching, alerting, and visualization.



Detection Techniques

Detection rules and searches were created in Splunk to identify suspicious and malicious activities, including:
- Failed login attempts (Linux and Windows) using queries on /var/log/auth.log
- Privilege escalation detection by monitoring 'sudo' usage in Linux logs.
- Process creation monitoring from Sysmon EventCode=1 for unusual processes.
- Monitoring network connections to suspicious IPs (Sysmon EventCode=3).
- Custom correlation searches to detect brute-force login attempts by counting repeated failed logins.
- MITRE ATT&CK mapping to align detected events with tactics and techniques, e.g., T1110 (Brute Force), T1059 (Command and Scripting Interpreter).

Response Processes

When a detection alert is triggered in Splunk, the SOC follows these response processes:

1. Triage :– The analyst reviews the alert, checks related events, and determines the severity.

2. Investigation :– The analyst uses Splunk searches to pivot through related logs (IP, username, process, etc.) to understand the attack chain.

3. Containment :– If malicious activity is confirmed, containment actions are initiated, such as blocking an IP address or disabling a compromised account.

4. Eradication :– Remove malicious files, kill rogue processes, or patch vulnerabilities.

5. Recovery :– Restore affected systems and services to operational state.

6. Lessons Learned :– Document the incident, update detection rules to prevent recurrence, and brief the SOC team.

Conclusion

Completing dif master class training and this SOC Analyst Portfolio Project has reinforced my ability to function effectively in a real-world security operations environment. By leveraging Splunk for log ingestion, analysis, and detection engineering, I demonstrated advanced skills in identifying threats, correlating events, and creating actionable alerts that improve visibility and response. The hands-on simulation of incident detection, triage, and response sharpened my investigative mindset and technical proficiency. Beyond technical execution, this project reflects my readiness to contribute as a SOC Analyst by bringing analytical depth, problem-solving skills, and a proactive approach to threat hunting and incident response. I am confident in my ability to support organizational security objectives and deliver measurable value to any security operations team.