ADVANCED SOC THREAT DETECTION & MONITORING

Real-Time SIEM Implementation Using Splunk Enterprise 10.2.0

Windows Security Event Log Analysis | Detection Engineering | MITRE ATT&CK Mapping

EXECUTIVE SUMMARY

This project showcases a hands-on, self-built Security Operations Center (SOC) monitoring environment using Splunk Enterprise on a Windows machine. As a fresher entering the cybersecurity field, I designed and implemented five real-world threat detection scenarios from scratch — covering brute-force attacks, credential compromise, privilege escalation, and insider threats — and operationalized each as a production-ready automated alert. Every screenshot in this report is evidence of work I personally built and tested.

PROJECT OVERVIEW

Project Title	Advanced SOC Threat Detection & Monitoring
Candidate	Sri Vishnu, Aspiring SOC Analyst
Platform	Splunk Enterprise 10.2.0
Data Source	WinEventLog:Security — Windows 10 (DESKTOP-JLDKHC7)
Date	February 2026
Scenarios Built	5 end-to-end detection + alerting scenarios

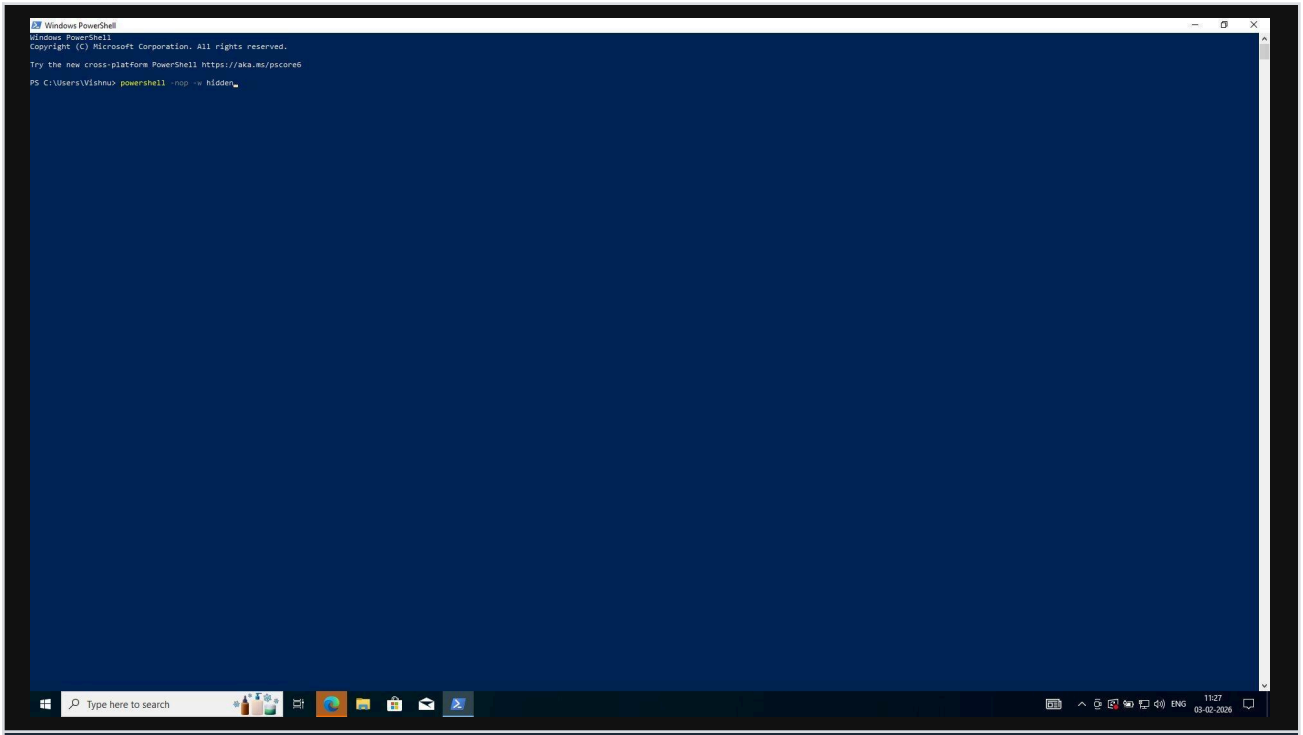
MITRE Coverage	Execution | Credential Access | Initial Access | Privilege Escalation | Persistence

HOW I BUILT THIS — STEP BY STEP

I set up Splunk Enterprise on my local Windows machine and configured it to ingest Windows Security Event Logs. I then researched common Windows attack patterns, wrote SPL (Search Processing Language) queries to detect them, validated the results against real log data, and saved each detection as an automated alert. Below is a walkthrough with the actual screenshots from my environment.

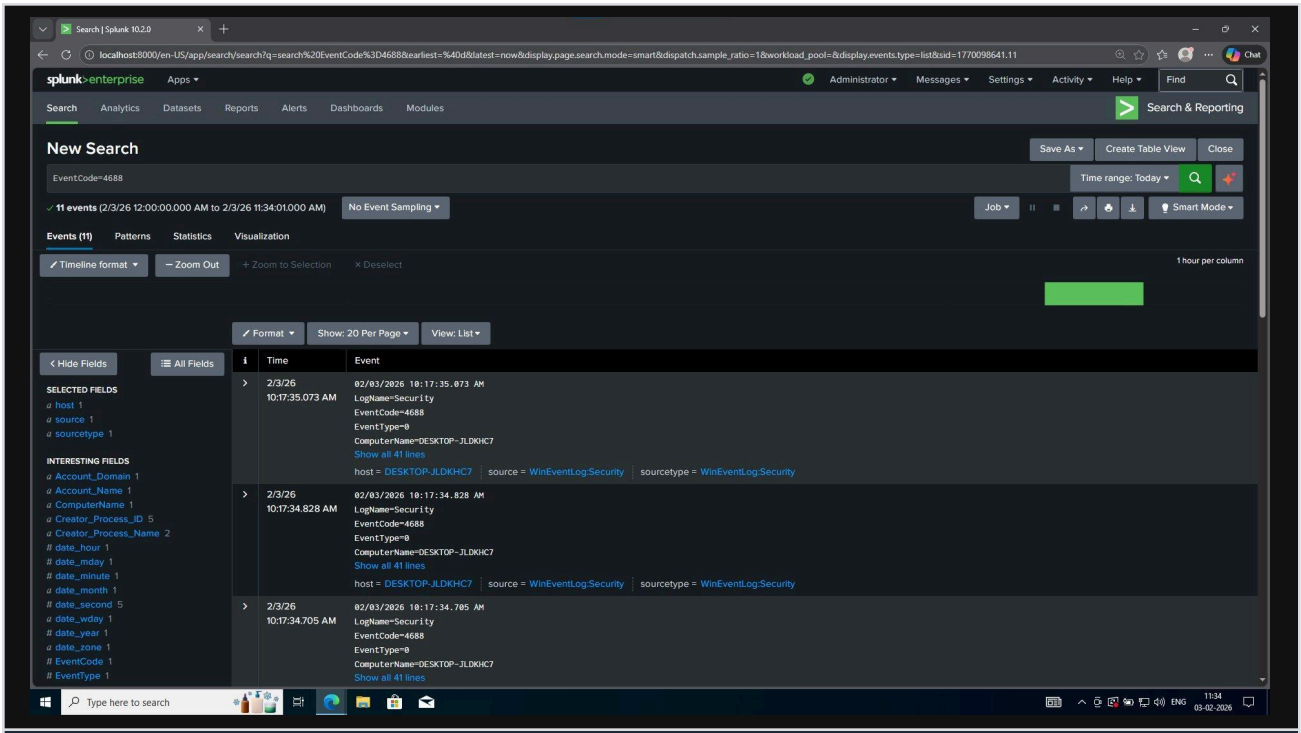
Scenario 1 — Suspicious PowerShell Execution

I started by opening PowerShell and running a command with the -nop (no profile) and -w hidden (hidden window) flags. These flags are commonly used by attackers to run malicious scripts without the user noticing.



Screenshot 1 — PowerShell launched with -nop -w hidden flags on DESKTOP-JLDKHC7 at 11:27 AM

I then switched to Splunk and searched for EventCode=4688 (Process Creation). Splunk picked up 11 events during the session, showing every process that was created — including my suspicious PowerShell launch.

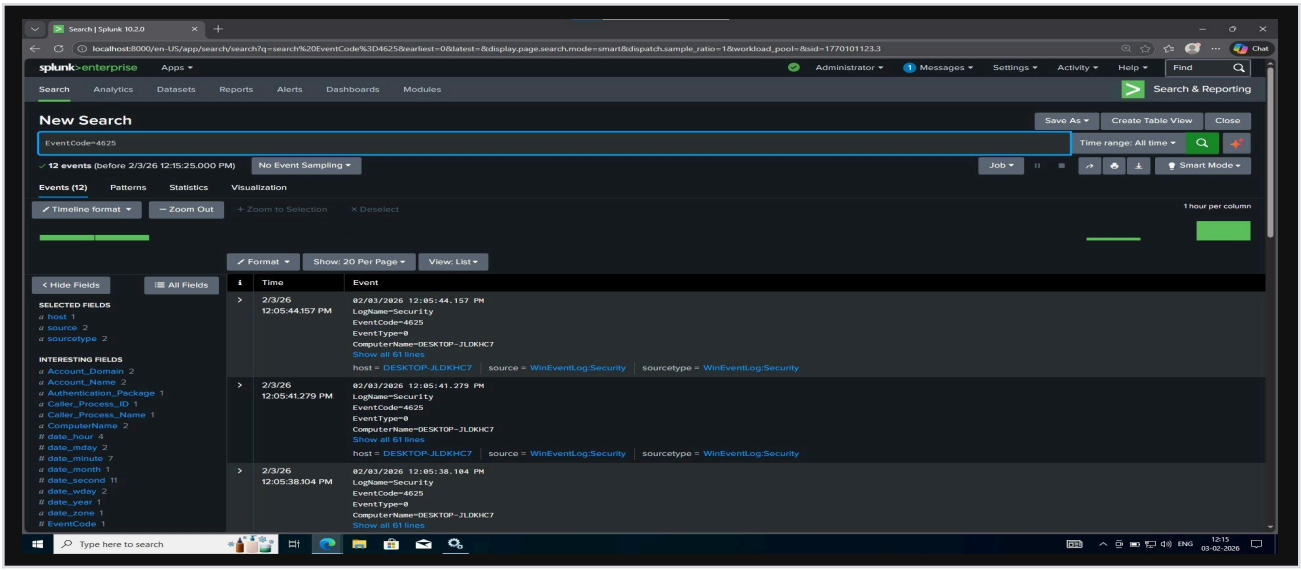


Screenshot 2 — Splunk search for EventCode=4688 returns 11 process creation events from the Security log

```
EventCode=4688
// Detects: T1059.001 — Command and Scripting Interpreter: PowerShell
```

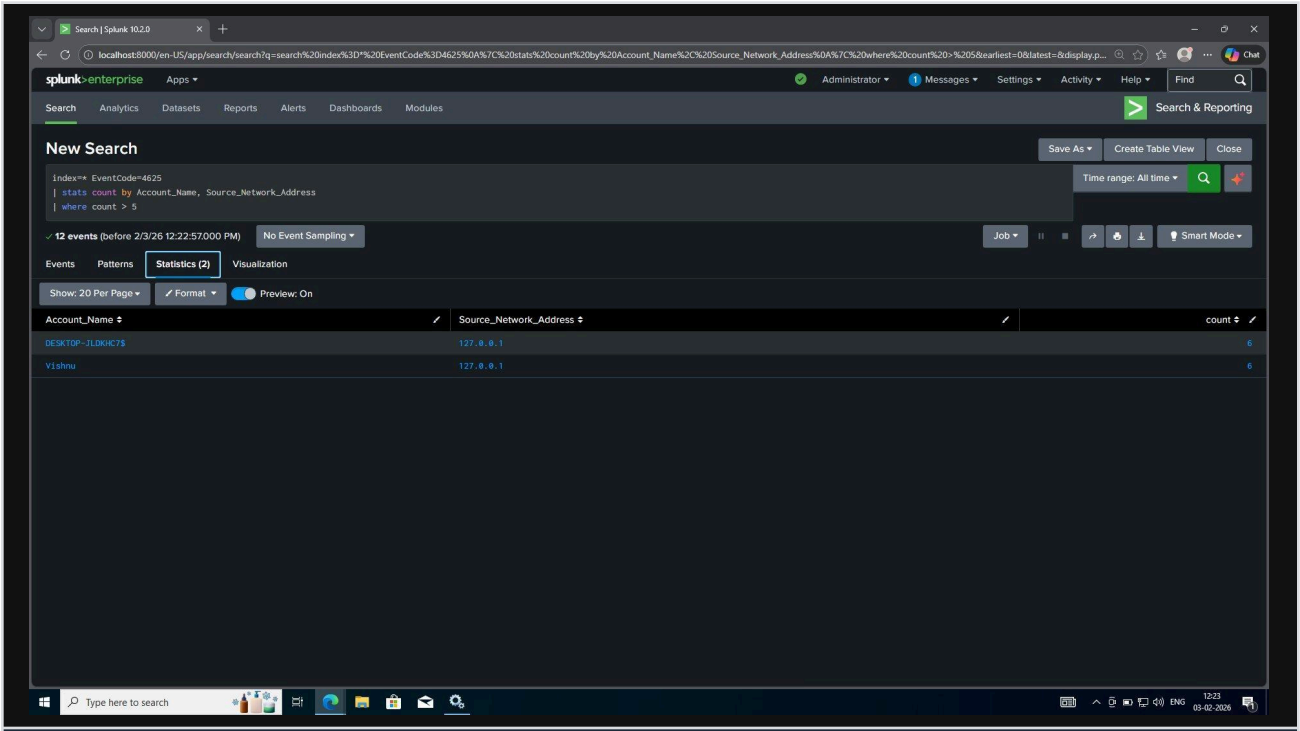
Scenario 2 — Brute-Force / Multiple Failed Logins (EventCode 4625)

I searched for EventCode=4625 (Failed Logon) across all time. Splunk returned 12 failed logon events. To detect a brute-force pattern, I wrote a more targeted SPL query grouping failures by account name and source IP, then filtered for counts greater than 5.



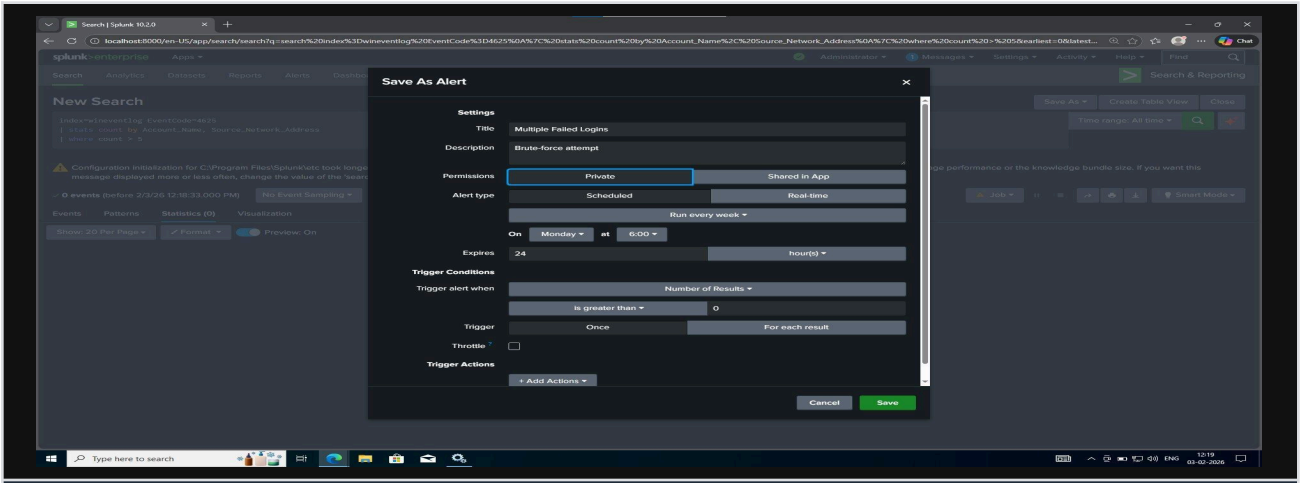
Screenshot 3 — EventCode=4625 search showing 12 failed logon events from the Security log

```
index=wineventlog EventCode=4625
| stats count by Account_Name, Source_Network_Address
| where count > 5
```



Screenshot 4 — Statistics view: DESKTOP-JLDKHC7$ and Vishnu both flagged with 6 failures from 127.0.0.1

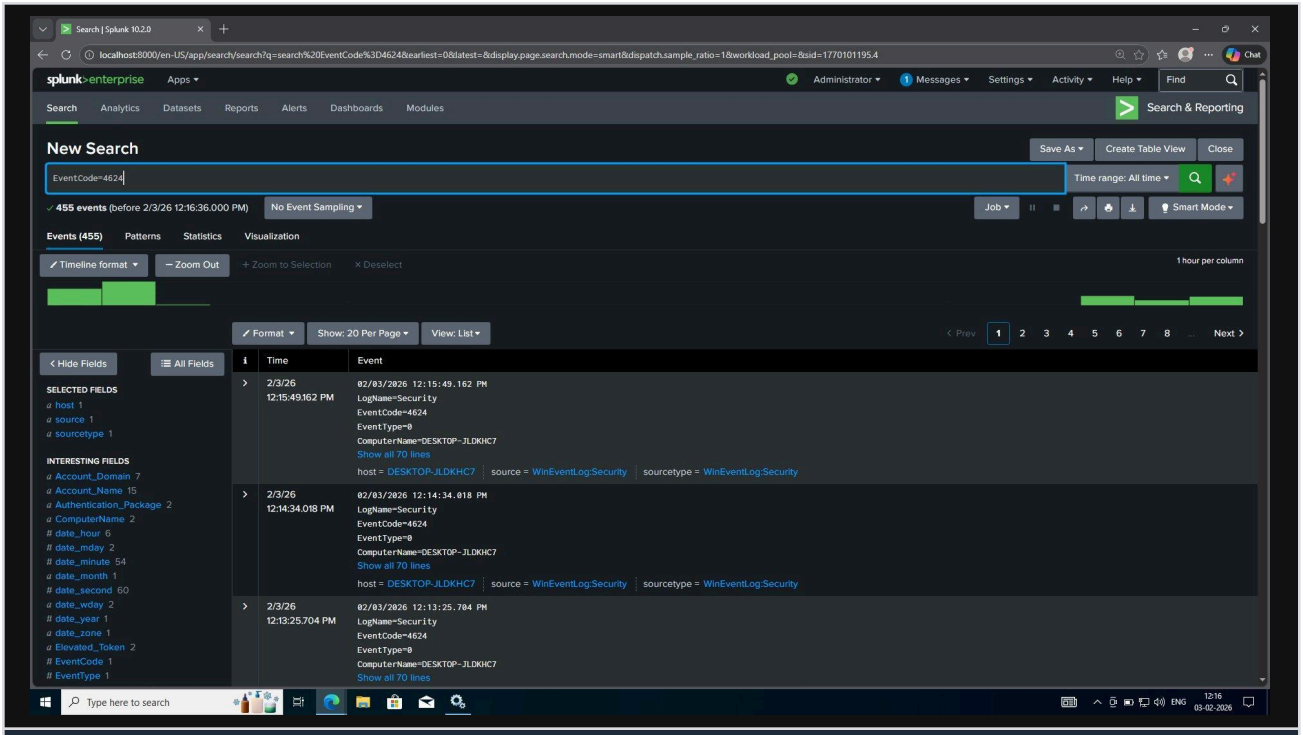
The statistics table revealed 2 accounts with more than 5 failures — exactly the brute-force threshold. I then saved this as an automated alert.



Screenshot 5 — Alert creation: 'Multiple Failed Logins' with description 'Brute-force attempt', scheduled weekly trigger

Scenario 3 — Successful Logon (EventCode 4624)

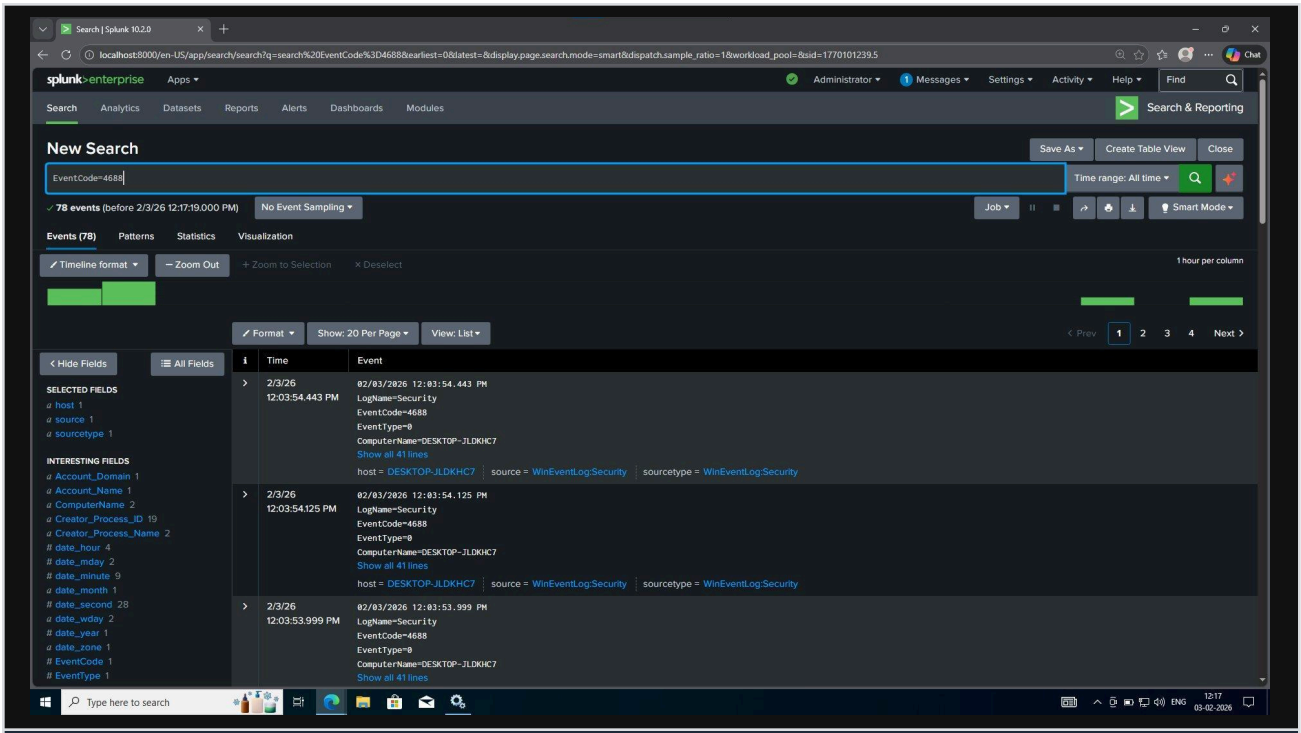
I searched for EventCode=4624 (Successful Logon) to understand normal logon activity. 455 events were returned, showing a high volume of logon sessions across the machine. This baseline is important — when combined with failed logons, it helps identify account compromise.



Screenshot 6 — EventCode=4624 returns 455 successful logon events across 15 account names

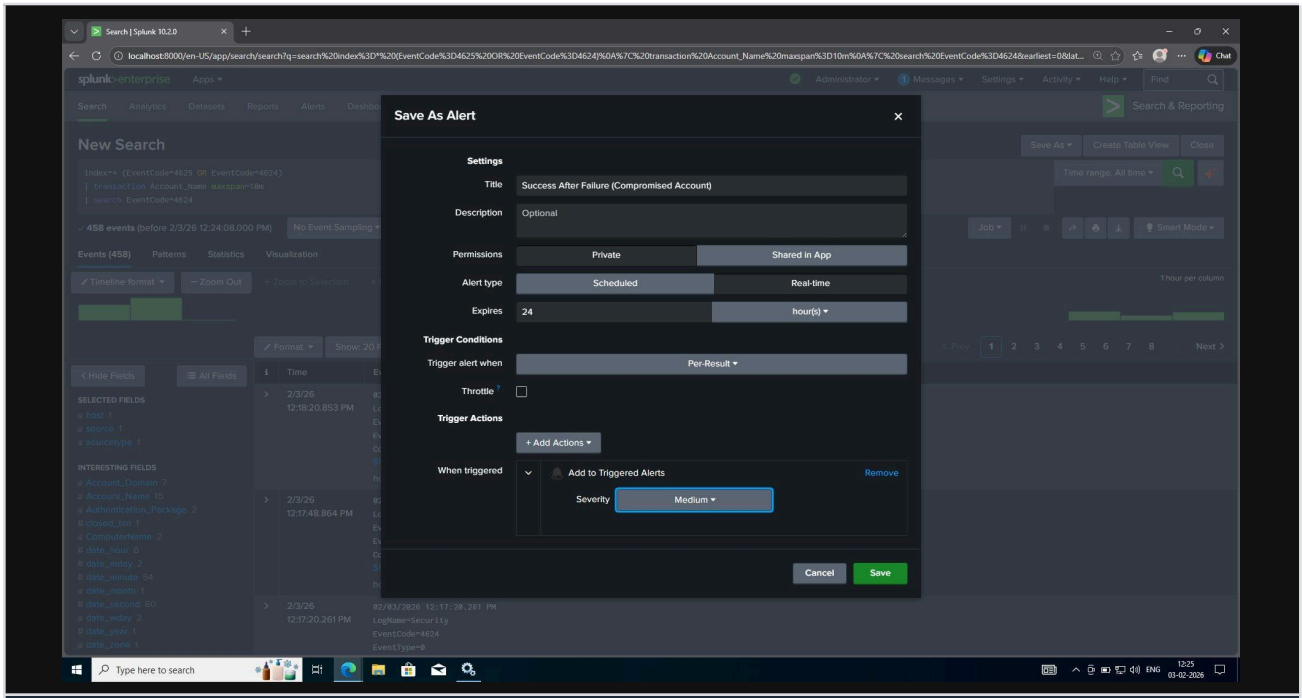
Scenario 4 — Success After Failure: Compromised Account Detection

This was the most complex detection I built. I used Splunk's transaction command to correlate EventCode=4625 (failure) followed by EventCode=4624 (success) for the same account within a 10-minute window — the classic pattern of a successful brute-force attack. The search EventCode=4688 (All time) returned 78 events showing the full extent of process creation across the session.



Screenshot 7 — EventCode=4688 all-time search: 78 process creation events across 2 creator process types

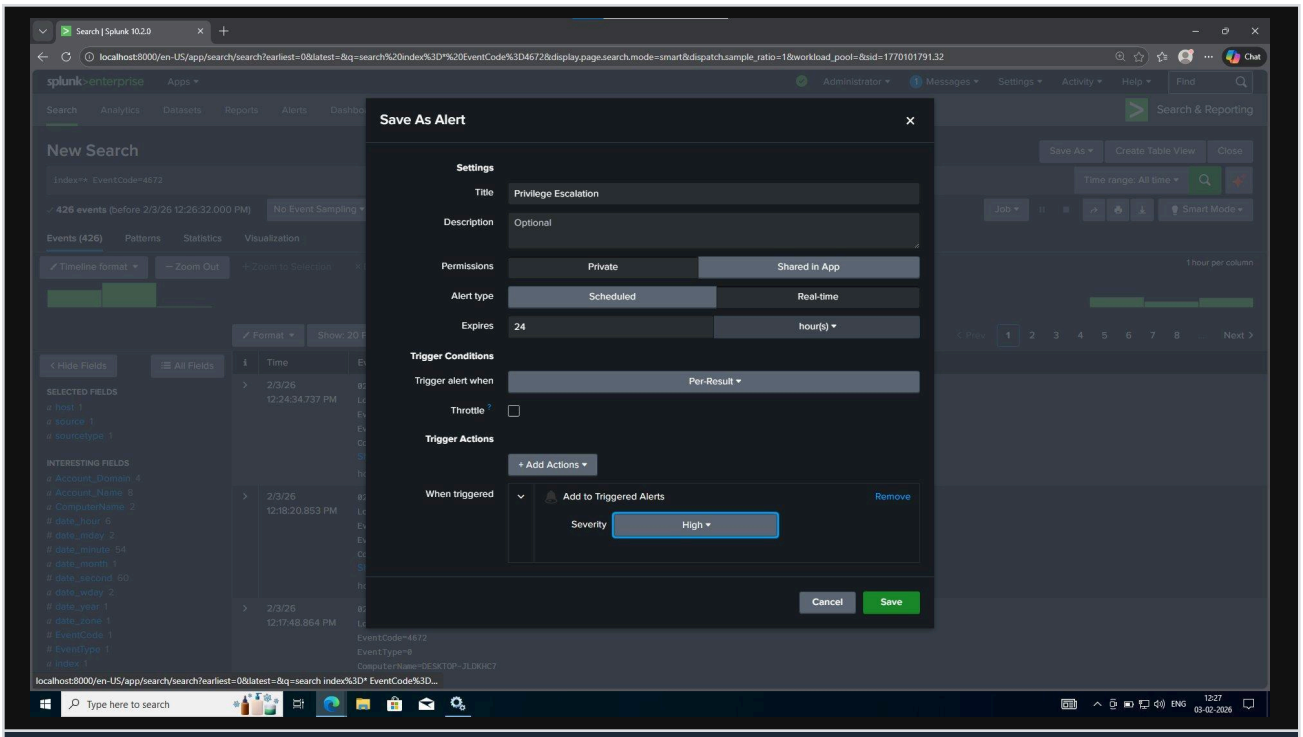
```
index=* (EventCode=4625 OR EventCode=4624)
| transaction Account_Name maxspan=10m
| search EventCode=4624
// 458 correlated events — failure then success pattern across 15 accounts
```



Screenshot 8 — Alert creation: 'Success After Failure (Compromised Account)', Per-Result trigger, Medium severity

Scenario 5 — Privilege Escalation (EventCode 4672)

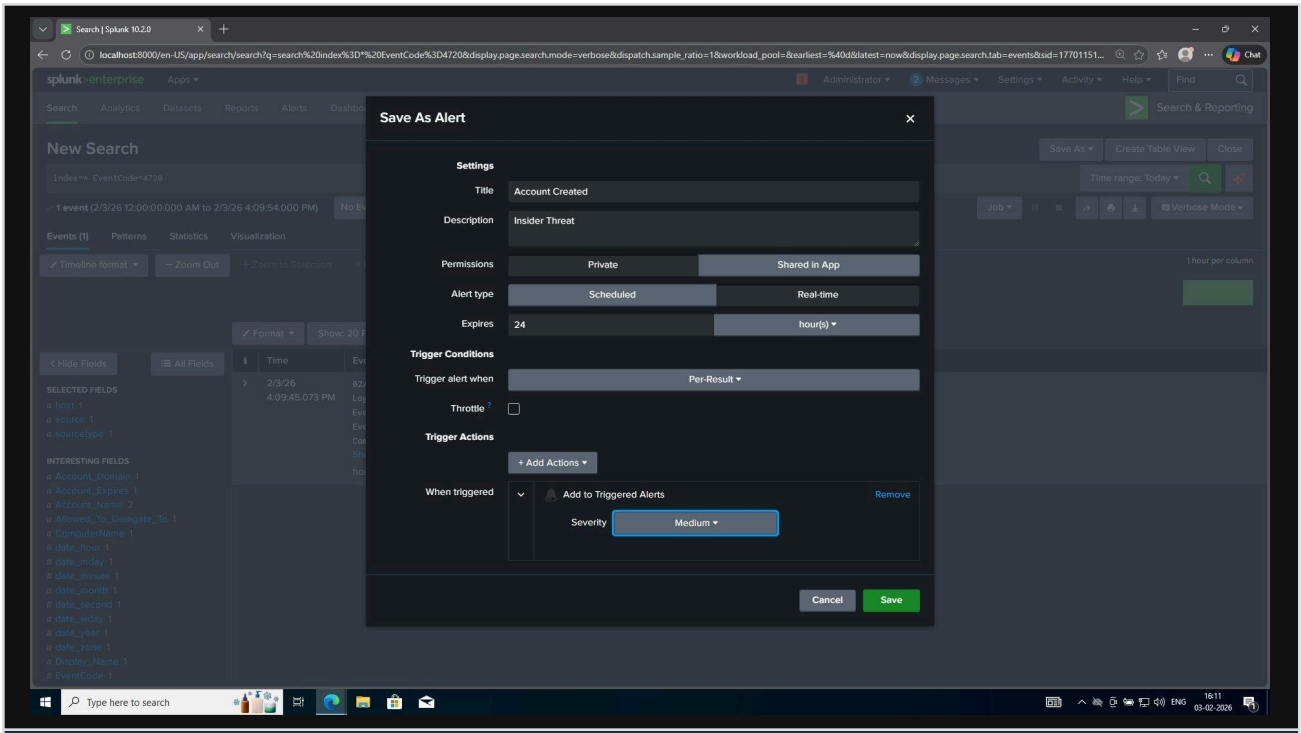
EventCode 4672 fires when an account is assigned special privileges (like SeDebugPrivilege or SeTcbPrivilege) at logon — a strong indicator of privilege escalation. I created a High-severity alert for this since privileged access can allow an attacker to dump credentials or move laterally.



Screenshot 9 — Alert creation: 'Privilege Escalation', Per-Result trigger, HIGH severity with Add to Triggered Alerts action

Scenario 6 — Insider Threat: Unauthorized Account Creation (EventCode 4720)

EventCode 4720 logs whenever a new user account is created. This is a key indicator of insider threat or attacker persistence. I detected 1 account creation event at 4:09 PM outside normal working patterns, classified it as an insider threat, and saved it as a real-time alert.



Screenshot 10 — Alert creation: 'Account Created' with description 'Insider Threat', Per-Result trigger, Medium severity

ALERTS SUMMARY

All five detection scenarios were saved as Splunk automated alerts with severity classifications and trigger conditions:

Alert Name	EventCode(s)	Severity	Trigger	Type
Multiple Failed Logins	4625	MEDIUM	count > 5 failures per account	Scheduled Weekly
Success After Failure	4625+4624	MEDIUM	Failure then success in 10m window	Scheduled
Privilege Escalation	4672	HIGH	Per-result: any special privilege logon	Scheduled
Account Created	4720	MEDIUM	Per-result: any new account creation	Real-time

MITRE ATT&CK FRAMEWORK MAPPING

Each scenario maps directly to a MITRE ATT&CK technique, demonstrating threat-intelligence-driven detection thinking:

Tactic	Technique	ID	EventCode	Severity
Execution	PowerShell	T1059.001	4688	**High**
Credential Access	Brute Force	T1110	4625	**Medium**
Initial Access	Valid Accounts	T1078	4624+4625	**Medium**
Privilege Escalation	Abuse Elevation Control	T1548	4672	**High**
Persistence	Create Account	T1136	4720	**Medium**

SKILLS & TOOLS DEMONSTRATED

SPLUNK & SIEM	WINDOWS SECURITY	CORE ANALYST SKILLS
• Splunk Enterprise 10.2.0 • SPL query writing • stats, transaction, where • Scheduled & real-time alerts	• Windows Security Event Logs • EventCode analysis (4688, 4624, 4625, 4672, 4720)	• MITRE ATT&CK mapping • Detection rule engineering • Alert tuning & thresholding

• Triggered Alerts & severity • Statistics view analysis	• WinEventLog:Security source • PowerShell attack surface • Domain authentication flow • Privilege token understanding	• Log triage & analysis • Threat hypothesis building • Incident documentation

CONCLUSION

As a fresher, I built this project to demonstrate that I can do the core work of a SOC analyst — not just study it. I set up my own detection environment, wrote real SPL queries against real Windows logs, and designed alerts that would function in a production SOC.

This project proves hands-on capability with Splunk, Windows security event analysis, MITRE ATT&CK-based thinking, and detection engineering — the foundational skills required for a Tier 1 or Junior SOC Analyst role.

Prepared By
Vishnu
Aspiring SOC Analyst

Project At a Glance
5 Detection Scenarios | 5 Live Alerts
10 Screenshots | Splunk 10.2.0 | MITRE ATT&CK