Project Title: SIEM Threat Detection Lab Using Splunk & Atomic Red Team

Author: Pasham Shashank Reddy

Date: June 2025

Tools: Splunk Enterprise, Splunk Universal Forwarder, Sysmon, Atomic Red Team, Windows 10 VM, Ubuntu Server VM, Virtual Box

Summary

This project focused on building a simulated Security Operations Center (SOC) environment to practice real-world threat detection using Splunk as a SIEM tool. A Windows 10 virtual machine was configured to generate event logs and simulate adversary behavior using Atomic Red Team, while an Ubuntu Server hosted Splunk Enterprise to ingest and analyze the data. The project emphasized detection of key attack techniques aligned with the MITRE ATT&CK framework, including suspicious PowerShell execution, brute force login attempts, and registry-based persistence.

Log forwarding was implemented using Splunk Universal Forwarder, and system-level monitoring was enhanced using Sysmon to capture detailed Windows telemetry. Multiple Splunk Reports, dashboard and alert rules were developed to monitor these activities in real time. Advanced SPL queries were written to detect anomalies and simulate threat hunting use cases, including correlation logic such as brute-force attacks followed by successful logins and PowerShell execution followed by registry modifications.

DAY 1 Setting Up the Virtual Machines and Splunk

**** Install Windows 10 ISO image:**
By downloading MediaCreationTool_22H2 from
URL: https://www.microsoft.com/en-ca/software-download/windows10
and creating an iso file.

**** Setting up the Windows 10 Victim VM:**

- Name: Windows10-SOC
- RAM - 4GB
- Processors – 2
- Hard Disk - 50GB(not pre-allocated)
- Created 2 network adapters - one is Host-only and the other is NAT

**** Windows Setup**

- Creating a Local Account as I don't need a live Account for Security Testing Purposes
- Name - victim
- Password - WindowsVictimPassword
- Security Questions & Answer - name of city born - name of city parents met - name of first school attended – Give Any Answer

**** Enable PowerShell Execution Policy**

- By default, scripts are blocked because Restricted is the default ExecutionPolicy.
- Enable them by executing the following command in PowerShell as Administrator:
  > Set-ExecutionPolicy RemoteSigned
- When prompted, type Y to confirm.
- This allows locally created scripts to run but blocks unsigned remote scripts (safer than Unrestricted).

**** Install Sysmon (from Sysinternals)**

- Download Sysinternals Suite from
  URL : https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon
- Recommended config file: SwiftOnSecurity's Sysmon config (Download sysmonconfig-export.xml from GitHub)
  URL: https://github.com/SwiftOnSecurity/sysmon-config
- Extract it and run in PowerShell as Admin:
  >.\Sysmon64.exe -accepteula -i sysmonconfig-export.xml

****Install Ubuntu Server 22.04 LTS ISO Image:**
By downloading it from
URL: https://ubuntu.com/download/server

****Setting up the Ubuntu Server VM:**

- Name: Splunk-SIEM
- RAM: 4 GB
- CPUs: 2
- HDD: 30GB
- OS: Ubuntu Server 22.04 LTS ISO
- Network Adapters: One adapter is Host-only and another one is NAT
- Specify the name, server's name, username, password and other details and functions.

****Disabling cloud-init in Ubuntu Server:**

Reason	Benefit
This is not on a cloud platform	It's unnecessary overhead
It slows down boot slightly	Speeds up your boot time
It can interfere with manual configs	Avoid surprises during network/user setup
Cleaner logs	Fewer irrelevant services in your logs

To do it:
Create a file to disable it permanently:
$sudo touch /etc/cloud/cloud-init.disabled

Optionally, to purge it completely (No need for my setup):
$sudo apt purge cloud-init
$sudo rm -rf /etc/cloud/ /var/lib/cloud/

Note: It should only be purged completely if tight on space on your system. So, I am not going to purge it completely and only disabled it. I only mentioned the purge option to know for future references.

****Installing Splunk Enterprise on Ubuntu Server(splunk-siem):**
Run the following commands to download and setup splunk enterprise in ubuntu server

$ wget -O splunk-9.4.3-237ebbd22314-linux-amd64.tgz
"https://download.splunk.com/products/splunk/releases/9.4.3/linux/splunk-9.4.3-237ebbd22314-linux-amd64.tgz"
$ tar -xvzf splunk-9.4.3-237ebbd22314-linux-amd64.tgz
$ sudo mv splunk /opt/
$ cd /opt/splunk/bin
$ sudo ./splunk start --accept-license
(Set up admin username and password)
$ sudo /opt/splunk/bin/splunk enable boot-start **(To start splunk on server startup)**

To access it via host computer's browser use link:
http://<ubuntu-server-ip>:8000

Enable Receiving on Port 9997 of Ubuntu Server to receive log from the forwarder

- In Splunk Web UI:
    - Go to Settings > Forwarding and Receiving > Configure Receiving
    - Add port 9997

****Installing & setting up Splunk Universal Forwarder in Windows-SOC:**
Download Splunk Universal Forwarder Windows 64-bit MSI installer from
URL: https://www.splunk.com/en_us/download/universal-forwarder.html

Setup Process:

- Run the Splunk Universal Forwarder Windows 64-bit MSI file as administrator.
- Accept License Agreement.
- Select on-premises Splunk Enterprise in the bottom.
- Click on Customize options.
- Let the location path be the same.
- No need to select the SSL certificate.
- Choose Local System Account.
- Select the Logs you want to send to splunk server. (I chose all Windows Event Logs and Performance Monitor Logs)
- Create a username and password.
- Give the deployment server hostname or IP and port (optional).
- Give the Receiving Index hostname or IP and post, for me that is my **Ubuntu Server Host-only Adapter IP address and Port 9997.** This is the destination where the logs are forwarded to.
- Then click install and click Yes to any User Account Control Prompts.
- To check whether the splunk server is receiving the logs you can go to:
    - On host computer go to http://<ubuntu-server-ip>:8000
    - Login with your credentials.
    - Go to Search & Reporting.
    - In New Search, Enter ” host=”WindowsHostName” “ or any other your machine specific filters.
    - Select the time to today and search.
    - If any logs are shown for your machine, then the setup is successful.

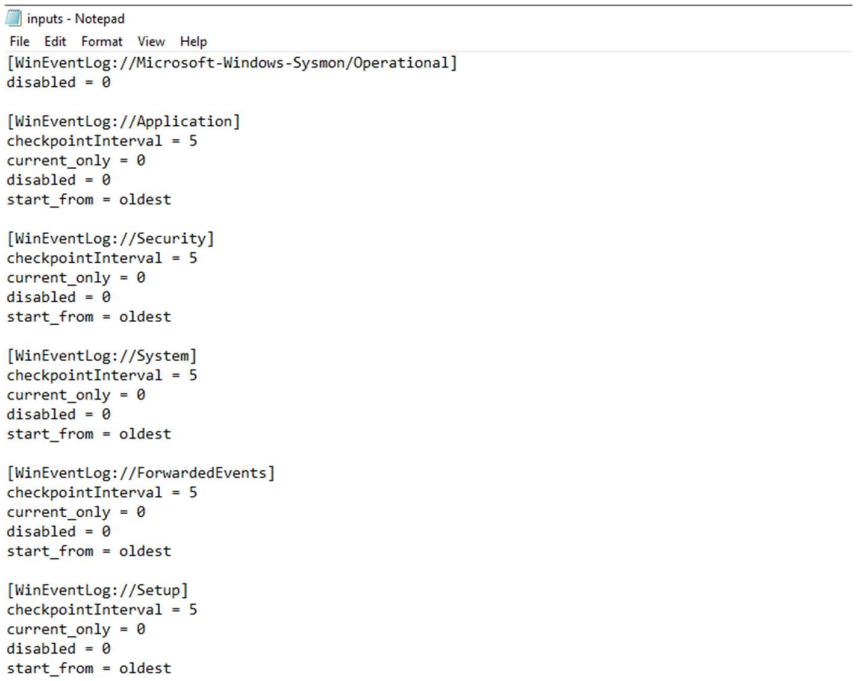
Setup Reference Video:
URL: https://www.youtube.com/watch?v=mYuEQRX3Rtg

Note: The Splunk Universal Forwarder automatically starts up at boot up always without any need for specification during setup.

DAY 2 Ingesting Sysmon Logs and Building Threat Hunting Dashboard in Splunk

****Setting up Sysmon logs of Windows-SOC VM to be ingested into Splunk Server:**

- Go to C:\Program Files\SplunkUniversalForwarder folder and Search for inputs.conf file.
- It will contain the type of logs that are being sent to Splunk Server.
- Copy the contents of that file.
- Go to C:\Program Files\SplunkUniversalForwarder\etc\system\local and create an inputs.conf file if not already present.
- Paste the content you copied in this file.
- To ingest Sysmon logs paste the following content in the inputs.conf file at the beginning.
  “
  [WinEventLog://Microsoft-Windows-Sysmon/Operational]
  disabled = 0
  “
- You can also add another line of index like “index=wineventlog” after disabled line to save it under a specified index in splunk server.
- Save the file.



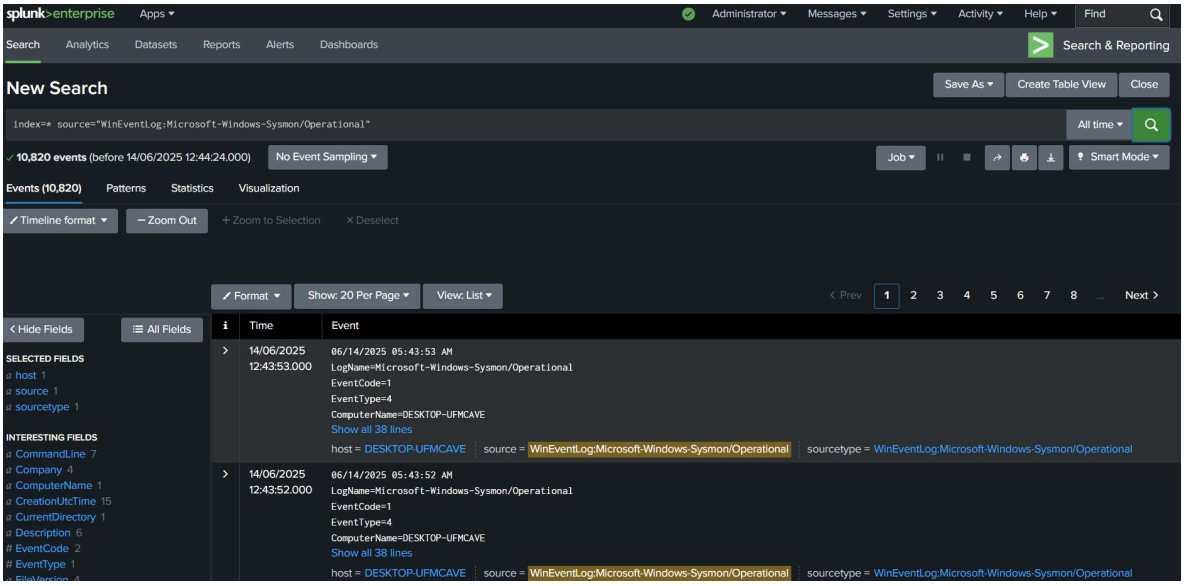
Inputs.conf File content

- Run PowerShell as Administrator.

- Use the following commands to restart splunk universal forwarder to make changes effective.

  > cd “C:\Program Files\SplunkUniversalForwarder\bin”
  > .\splunk.exe restart.

- To check whether the Sysmon logs are ingested go to Splunk Web UI -> Login -> Search & Reporting -> in search enter
  “
  index=* source="WinEventLog:Microsoft-Windows-Sysmon/Operational"
  “
  and the Sysmon logs will be shown.



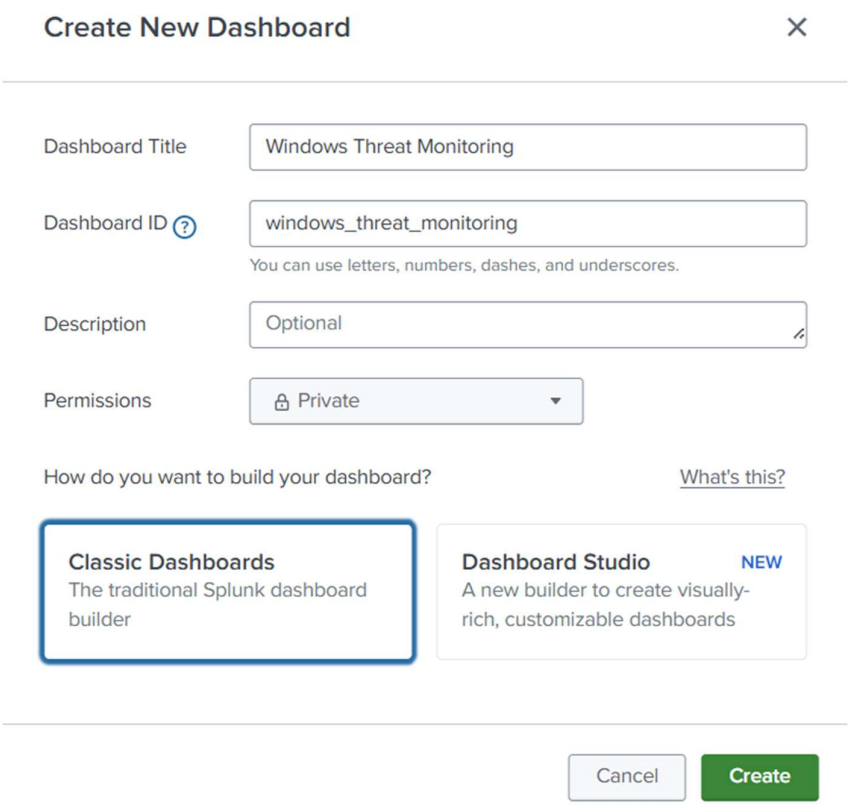
Sysmon Events

****Creating a Custom App in Splunk (Optional but Recommended):**

1. Go to Splunk web → Manage Apps
2. Click Create App
3. Enter:
   - App Name: SOC Threat Dashboard
   - Folder Name: soc_dashboard
   - Version: 1.0.0
   - Author: Your name
   - Leave the rest as default
4. Submit and restart Splunk if required
5. Navigate to the app from the Splunk home screen

**Creating the Dashboard:

- In the app, click on Dashboards > Create New Dashboard
- Set:
    - Title: Windows Threat Monitoring
    - ID: auto-fills, or keep it readable.
    - Permissions: Keep as private for now
- Choose Classic Dashboard for easier panel design



Create New Dashboard Panel

**Adding Panels in Dashboard:

Panel 1: Failed Logon Attempts

- Tracks brute-force or suspicious login behavior.
- Visualization: Bar Chart (X = Account_Name, Y = count)
- Time Picker: “Shared Time Picker”
- Click "Save to Dashboard"

SPL Query:

```
index=main sourcetype="WinEventLog:Security" EventCode=4625
| stats count by Account_Name
| sort – count
```

Panel 2: PowerShell Command Usage

- Detects PowerShell usage and potential misuse based on a suspicion score calculated by the components of the PowerShell Command executed.
- Visualization: Table or Pie Chart
- Time Picker: “Shared Time Picker”

SPL Query:

```
index=main sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational"
EventCode=1
| eval LowerCmd=lower(CommandLine)
| eval suspicion_score=0
| eval suspicion_score=suspicion_score +
  if(like(LowerCmd, "%encodedcommand%"), 1, 0) +
  if(like(LowerCmd, "%iex%"), 1, 0) +
  if(like(LowerCmd, "%invoke-expression%"), 1, 0) +
  if(like(LowerCmd, "%bypass%"), 1, 0) +
  if(like(LowerCmd, "%downloadstring%"), 1, 0) +
  if(like(LowerCmd, "%.ps1%"), 1, 0) +
  if(like(LowerCmd, "%new-object%"), 1, 0) +
  if(like(LowerCmd, "%frombase64string%"), 1, 0) +
  if(like(LowerCmd, "%start-bitstransfer%"), 1, 0) +
  if(like(LowerCmd, "%hidden%"), 1, 0) +
  if(like(LowerCmd, "%nop%"), 1, 0) +
  if(like(LowerCmd, "%windowstyle%"), 1, 0)
| where suspicion_score > 0
| table _time, ComputerName, User, ParentImage, CommandLine, suspicion_score
| sort -suspicion_score -_time
```

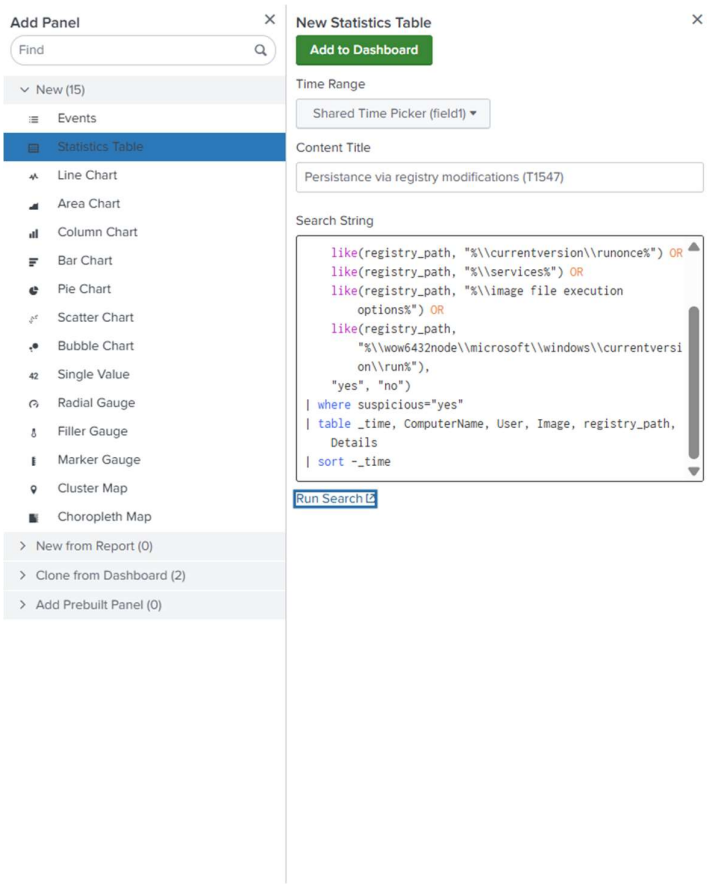
Panel 3: Registry Persistence Attempts

- Detects persistence via registry modifications based on whether the registry path that is modified is suspicious or not.
- Visualization: Table
- Time Picker: Shared Time Picker

SPL Query:

```
index=main sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational"
(EventCode=13 OR EventCode=4657)
| eval registry_path=lower(TargetObject)
| eval suspicious=if(
  like(registry_path, "%\\currentversion\\run%") OR
  like(registry_path, "%\\currentversion\\runonce%") OR
  like(registry_path, "%\\services%") OR
  like(registry_path, "%\\image file execution options%") OR
```

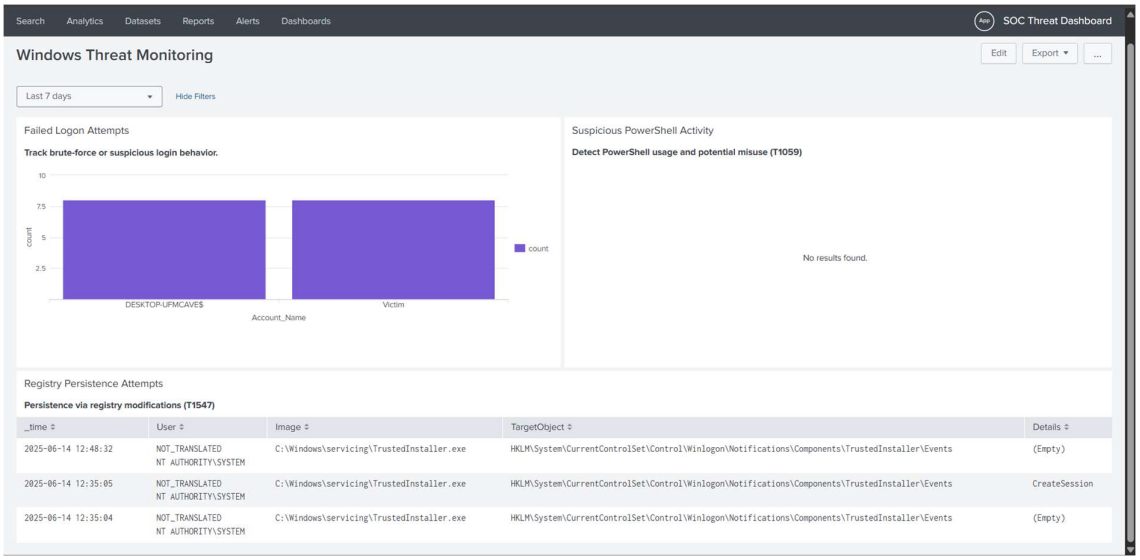
```
  like(registry_path, "%\\wow6432node\\microsoft\\windows\\currentversion\\run%"),
  "yes", "no")
| where suspicious="yes"
| table _time, ComputerName, User, Image, registry_path, Details
| sort -_time
```



Add Panel Pane

****Customize the Dashboard**

- Add a Time Picker to the top of the dashboard:
  - Click Edit → Add Input → Time
  - Link it to your panels so queries reflect time ranges (e.g., last 24h)
- Set default time to Last 24 Hours or Last 7 Days
- Arrange panels in 2-column layout.



Windows Threat Monitoring Dashboard

DAY 3 Installing and Setting Up Atomic Red Team in Windows-SOC VM

****Installing atomic red team:**

What is Atomic Red Team?
Atomic Red Team (ART) is a library of small, independent tests mapped to MITRE ATT&CK techniques. We run these to simulate attacker behavior (e.g., PowerShell obfuscation, credential dumping) in a safe, controlled way to validate detections.

Step 1:
Temporarily Disable Real-time Protection (Recommended in VM)
Since the Windows-SOC VM is isolated and temporary:

1. Open: Windows Security → Virus & Threat Protection
2. Click "Manage settings" under Virus & threat protection settings.
3. Toggle Real-time protection → OFF

OR

Exclude Folder from Defender Scans
Safer if we want Defender active but don't want alerts for atomic files:

1. Open Virus & Threat Protection → Manage Settings
2. Scroll to Exclusions → Click Add or remove exclusions
3. Click Add an exclusion → Choose Folder
4. Select your atomic-red-team directory

Step 2: Download Atomic Red Team zip file

- Go to URL: https://github.com/redcanaryco/atomic-red-team.git

- Code -> Download ZIP file.
- Extract the zip file contents.
- Open PowerShell as Administrator and navigate to atomics folder in atomic-red-team folder you extracted from the zip file.
- Copy its path using pwd to view the path and CTRL+C.
- Save this path somewhere because it is needed to run tests. For me it was : “C:\Users\Victim\Downloads\atomic-red-team-master\atomic-red-team-master\atomics”

Step 3: Install Prerequisites (Invoke-AtomicTest Framework)

- Install Atomic Red Team Execution Framework using the following commands and Type “Y” if prompted to trust the repository and press enter.
  > Install-Module -Name invoke-atomicredteam,powershell-yaml
- Now explicitly import the module:
  > Import-Module Invoke-AtomicRedTeam
- You should now be able to run any ART cmdlets like:
  > Get-Command -Module Invoke-AtomicRedTeam
- Test if it works, try listing available tests for a technique:
  > Invoke-AtomicTest T1059.001 -PathToAtomicsFolder "C:\Users\Victim\Downloads\atomic-red-team-master\atomic-red-team-master\atomics" -ShowDetailsBrief

If any problems occur go to the GitHub link and see the installation guide.

DAY 4 Running Atomic Red Team Simulation Tests in Windows-SOC VM

****Running Atomic Red Team Simulation Tests:**

1. Brute Force / Failed Logon Attempts

Atomic Technique: T1110.001 – Password Guessing

Command to Run:

> Invoke-AtomicTest T1110.001 -TestNumbers 1 -PathToAtomicsFolder "C:\Users\Victim\Downloads\atomic-red-team-master\atomic-red-team-master\atomics"

What it does: Tries multiple logins using different usernames/passwords.

Splunk Dashboard Panel:

Failed Logon Attempts – We can see new entries with EventCode=4625 showing attempted usernames and IPs.

2. Malicious PowerShell Execution

Atomic Technique: T1059.001 – PowerShell

Command to Run:

> Invoke-AtomicTest T1059.001 -TestNumbers 1,2,3 -PathToAtomicsFolder "C:\Users\Victim\Downloads\atomic-red-team-master\atomic-red-team-master\atomics"

These simulate:

- PowerShell base64 encoded commands
- Execution of obfuscated PowerShell
- Suspicious download behavior

Splunk Dashboard Panel:

PowerShell Command Usage – We can now see the command lines pop up in the CommandLine column with high suspicion_score.

3. Registry Persistence

Atomic Technique: T1547.001 – Registry Run Keys / Startup Folder

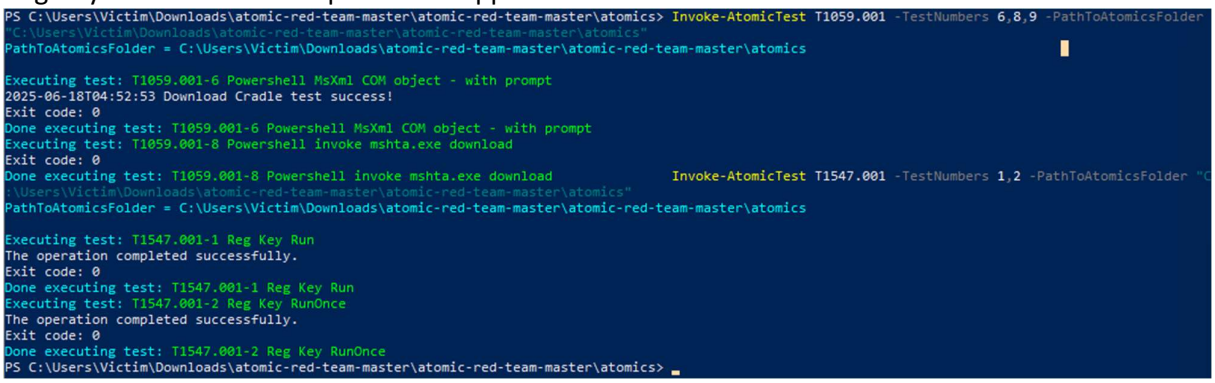
Command to Run:

> Invoke-AtomicTest T1547.001 -TestNumbers 1,2 -PathToAtomicsFolder "C:\Users\Victim\Downloads\atomic-red-team-master\atomic-red-team-master\atomics"

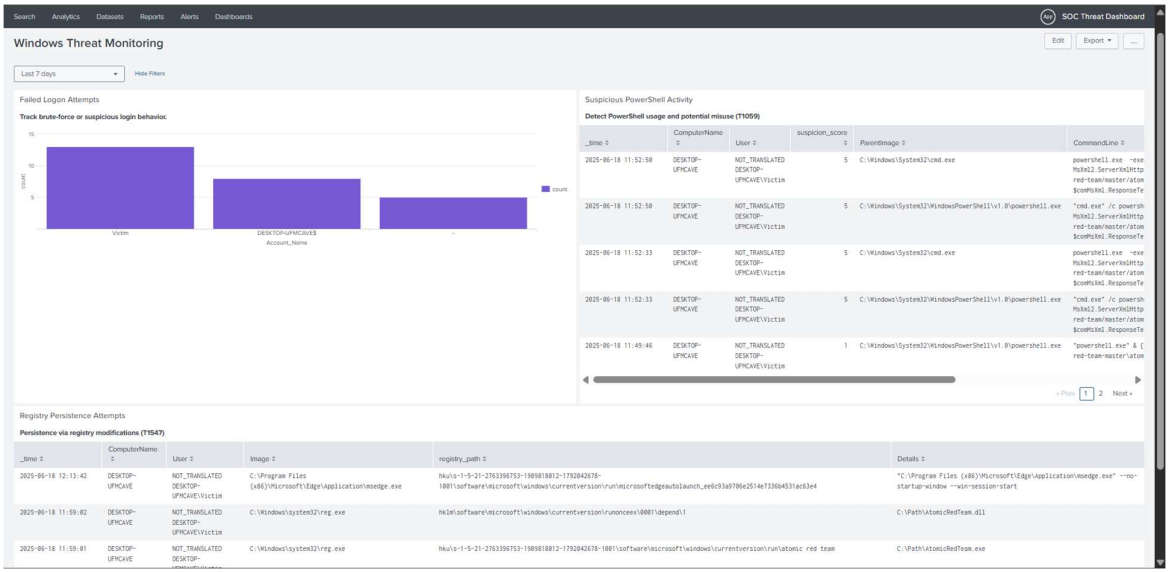
What it does: Adds keys in common persistence paths (e.g., HKCU\Software\Microsoft\Windows\CurrentVersion\Run).

Splunk Dashboard Panel:

Registry Persistence Attempts should appear under EventCode=13 OR EventCode=4657



Running Tests using Atomic Red Team in Windows-SOC VM



Windows Threat Monitoring Dashboard after the 3 Simulation Tests

DAY 5 Creating and Configuring Detection Alerts in Splunk

**** Creating and Configuring Detection Alerts in Splunk:**

Step 1: Identify Key Detection Use Cases to Alert On
From the existing panels, pick detections that:

- Are high-confidence indicators of attack
- Don't generate excessive false positives
- Can be triggered with a clear logic

Selected Use Cases:

1. Suspicious PowerShell usage
2. Multiple failed logon attempts from the same source
3. Registry modification in persistence paths

Step 2: Turn Detection Queries into Alerts

Use the SPL Queries in Search Tab in the SOC Threat Dashboard App and Select Save As Alert

Alert 1: **Suspicious PowerShell Execution**
This alert triggers when obfuscated or suspicious PowerShell commands are detected using keywords like frombase64string, encodedcommand, iex, etc..

MITRE ATT&CK technique:

- **Technique ID**: T1059
- **Name**: Command and Scripting Interpreter
- **Sub-technique**:
  - **T1059.001** – PowerShell
- **Description**: Adversaries abuse PowerShell to execute malicious scripts, download payloads, or interact with the OS while avoiding detection.

SPL Query:

```
index=main sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational"
EventCode=1
| eval LowerCmd=lower(CommandLine)
| eval suspicion_score=0
| eval suspicion_score=suspicion_score +
   if(like(LowerCmd, "%encodedcommand%"), 1, 0) +
   if(like(LowerCmd, "%iex%"), 1, 0) +
   if(like(LowerCmd, "%invoke-expression%"), 1, 0) +
   if(like(LowerCmd, "%bypass%"), 1, 0) +
   if(like(LowerCmd, "%downloadstring%"), 1, 0) +
```

```
  if(like(LowerCmd, "%.ps1%"), 1, 0) +
  if(like(LowerCmd, "%new-object%"), 1, 0) +
  if(like(LowerCmd, "%frombase64string%"), 1, 0) +
  if(like(LowerCmd, "%start-bitstransfer%"), 1, 0) +
  if(like(LowerCmd, "%hidden%"), 1, 0) +
  if(like(LowerCmd, "%nop%"), 1, 0) +
  if(like(LowerCmd, "%windowstyle%"), 1, 0)
| where suspicion_score >= 2
| table _time, ComputerName, User, CommandLine, suspicion_score| where
suspicion_score >= 2
| table _time, ComputerName, User, CommandLine, suspicion_score
```

Settings:

- Alert: Suspicious PowerShell Activity
- Description: Detects potentially malicious PowerShell usage based on keyword scoring
- Alert Type: Scheduled, Run on Cron Schedule
- Time Range: Last 5 minutes (Use advanced to set it)
- Cron Expression: */5 * * * *
- Expires 24 hours

Trigger Conditions:

- Trigger alert when: Number of Results > 0
- Trigger: For each result
- Throttle: yes
- Suppress results containing field value: User
- Suppress triggering for: 5 min

Trigger Action: Add to Triggered Alerts

Alert 2: Brute Force Login Attempt
Triggers when 5 or more failed login attempts are seen from the same user or source within one search window.

MITRE ATT&CK technique:

- **Technique ID**: T1110
- **Name**: Brute Force
- **Description**: The adversary attempts to gain access to accounts by guessing passwords or using a dictionary of common credentials.

If you're specifically targeting **local logon failures** (EventCode 4625):

- **Sub-technique:**
  - **T1110.001** – Password Guessing

SPL Query:

```
index=main sourcetype="WinEventLog:Security" EventCode=4625
| stats count by Account_Name, Source_Network_Address, host
| where count >= 5
```

Settings:

- Alert: Brute Force Login Attempt
- Description: Detects repeated login failures (EventCode 4625) from a single user or IP
- Alert Type: Scheduled, Run on Cron Schedule
- Time Range: Last 10 minutes (Use advanced to set it)
- Cron Expression: */10 * * * *
- Expires 24 hours

Trigger Conditions:

- Trigger alert when: Number of Results > 0
- Trigger: Once
- Throttle: yes
- Suppress triggering for: 10 min

Trigger Action: Add to Triggered Alerts

Alert 3: Registry Persistence Detected

Detects changes to common persistence registry keys

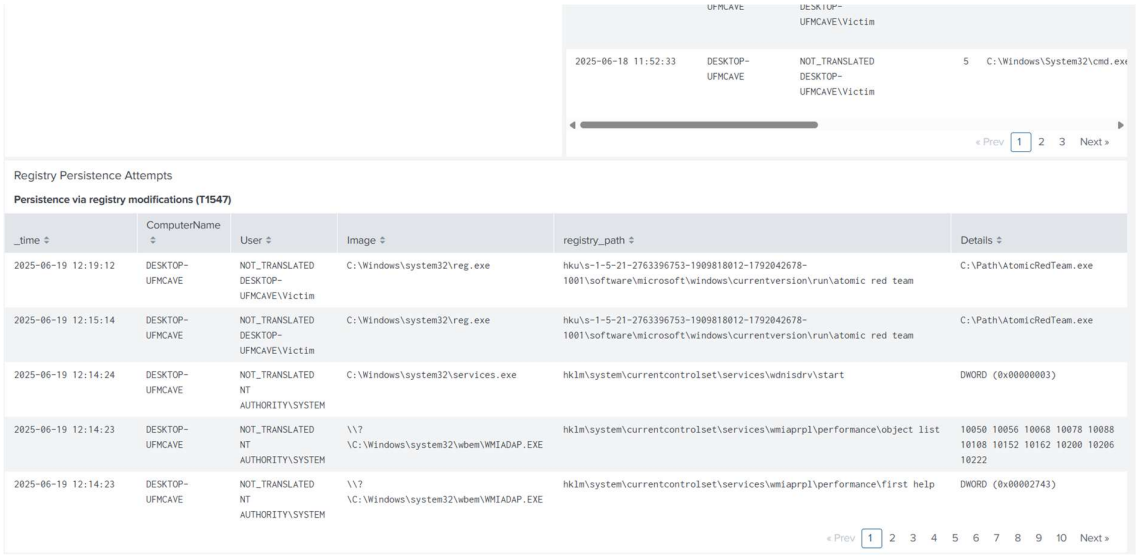
MITRE ATT&CK technique:

- **Technique ID**: T1547
- **Name**: Boot or Logon AutoStart Execution
- **Sub-technique**:
  - **T1547.001** – Registry Run Keys / Startup Folder
- **Description**: Adversaries can set a registry key to execute a program during system boot or user logon, allowing for persistence.

SPL Query:

```
index=main sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational"
(EventCode=13 OR EventCode=4657)
| eval registry_path=lower(TargetObject)
| where (like(registry_path, "%\\currentversion\\run%") OR like(registry_path,
"%\\runonce%"))
| where like(Image, "%powershell.exe") OR like(Image, "%cmd.exe") OR like(Image,
"%wscript.exe") OR like(Image, "%reg.exe")
| table _time, User, Image, TargetObject, Details
```

** We can change the Search query for the panel corresponding to this type to the above SPL Query as it is more refined. This can make the dashboard more precise and remove the noisy events(normal operation events).



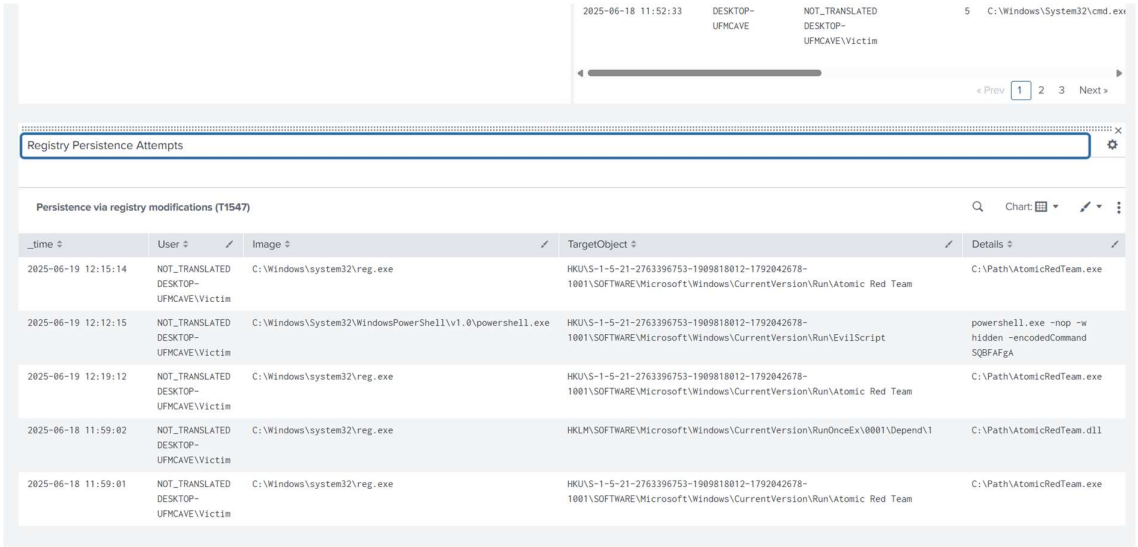
Registry Persistence Attempts

Persistence via registry modifications (T1547)

_time	ComputerName	User	Image	registry_path	Details
2025-06-19 12:19:12	DESKTOP-UFMCAVE	NOT_TRANSLATED DESKTOP-UFMCAVE\Victim	C:\Windows\system32\reg.exe	hku\s-1-5-21-2763396753-1909818012-1792042678-1001\software\microsoft\windows\currentversion\run\atomic red team	C:\Path\AtomicRedTeam.exe
2025-06-19 12:15:14	DESKTOP-UFMCAVE	NOT_TRANSLATED DESKTOP-UFMCAVE\Victim	C:\Windows\system32\reg.exe	hku\s-1-5-21-2763396753-1909818012-1792042678-1001\software\microsoft\windows\currentversion\run\atomic red team	C:\Path\AtomicRedTeam.exe
2025-06-19 12:14:24	DESKTOP-UFMCAVE	NOT_TRANSLATED NT AUTHORITY\SYSTEM	C:\Windows\system32\services.exe	hklm\system\currentcontrolset\services\wdnisdrv\start	DWORD (0x00000003)
2025-06-19 12:14:23	DESKTOP-UFMCAVE	NOT_TRANSLATED NT AUTHORITY\SYSTEM	\\?\C:\Windows\system32\wbem\WMIADAP.EXE	hklm\system\currentcontrolset\services\wmiaprpl\performance\object list	10050 10056 10068 10078 10088 10108 10152 10162 10200 10206 10222
2025-06-19 12:14:23	DESKTOP-UFMCAVE	NOT_TRANSLATED NT AUTHORITY\SYSTEM	\\?\C:\Windows\system32\wbem\WMIADAP.EXE	hklm\system\currentcontrolset\services\wmiaprpl\performance\first help	DWORD (0x00002743)

« Prev 1 2 3 4 5 6 7 8 9 10 Next »

Registry Persistence Attempts Panel Before Modification



Registry Persistence Attempts

Persistence via registry modifications (T1547)

Chart:

_time	User	Image	TargetObject	Details
2025-06-19 12:15:14	NOT_TRANSLATED DESKTOP-UFMCAVE\Victim	C:\Windows\system32\reg.exe	HKU\S-1-5-21-2763396753-1909818012-1792042678-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Atomic Red Team	C:\Path\AtomicRedTeam.exe
2025-06-19 12:12:15	NOT_TRANSLATED DESKTOP-UFMCAVE\Victim	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	HKU\S-1-5-21-2763396753-1909818012-1792042678-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\EvilScript	powershell.exe -nop -w hidden -encodedCommand SQBFAFgA
2025-06-19 12:19:12	NOT_TRANSLATED DESKTOP-UFMCAVE\Victim	C:\Windows\system32\reg.exe	HKU\S-1-5-21-2763396753-1909818012-1792042678-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Atomic Red Team	C:\Path\AtomicRedTeam.exe
2025-06-18 11:59:02	NOT_TRANSLATED DESKTOP-UFMCAVE\Victim	C:\Windows\system32\reg.exe	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx\0001\Depend\1	C:\Path\AtomicRedTeam.dll
2025-06-18 11:59:01	NOT_TRANSLATED DESKTOP-UFMCAVE\Victim	C:\Windows\system32\reg.exe	HKU\S-1-5-21-2763396753-1909818012-1792042678-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Atomic Red Team	C:\Path\AtomicRedTeam.exe

Registry Persistence Attempts Panel After Modification

Settings:

- Alert: Registry Persistence Detected
- Description: Detects changes to common persistence registry keys
- Alert Type: Scheduled, Run on Cron Schedule
- Time Range: Last 10 minutes (Use advanced to set it)
- Cron Expression: */10 * * * *
- Expires 24 hours

Trigger Conditions:

- Trigger alert when: Number of Results > 0
- Trigger: For each result
- Throttle: yes
- Suppress results containing field value: User
- Suppress triggering for: 10 min

Trigger Action: Add to Triggered Alerts

****Executing commands or Atomic Red Team Tests to check whether the above alerts are being generated:**

1. Suspicious PowerShell Activity
Best Method: Manual PowerShell Command (no dependencies)
Run in PowerShell (as admin or regular):
> powershell -exec bypass -windowstyle hidden -encodedcommand SQBFAFgA
This uses obfuscation, bypass, hidden, and encoded command.
Atomic Test:
> Invoke-AtomicTest T1059.001 -TestNumbers 5
Test 5 is lightweight and executes a suspicious but harmless PowerShell command.

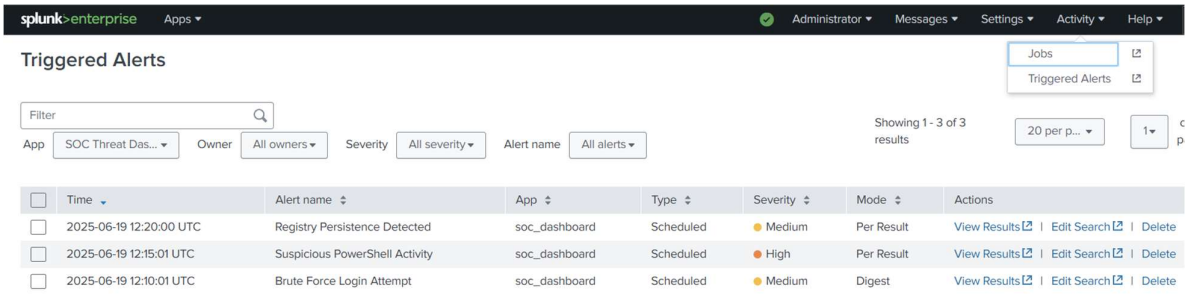
2. Brute Force Login Attempt
Manual Method: Lockout-style login failures

1. Lock the screen (Win + L)
2. Enter wrong password 6–8 times quickly.

Atomic Test:
> Invoke-AtomicTest T1110.001 -TestNumbers 1
Simulates multiple login attempts to trigger EventCode 4625

3. Registry Persistence Attempts
Manual Method: Create a registry Run key
Run this in PowerShell (admin):
> New-ItemProperty -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Run" -Name "EvilScript" -Value "powershell.exe -nop -w hidden -encodedCommand SQBFAFgA"
Triggers EventCode 13/4657 + hits the registry path filter in your query.
Atomic Test:
Invoke-AtomicTest T1547.001 -TestNumbers 1

We can check whether alerts were triggered by going to Splunk Web UI -> Activity -> Triggered Alerts



Triggered Alerts

DAY 6 Using Correlation Rules and Creating Reports

****Using Correlation Rules and Creating Reports:**

Brute Force Attack Followed by Successful Login:
The below SPL Query goes through the events with EventCode 4625(Failed Login Attempt) and 4624(Successful Login Attempt) and aggregates them to calculate the number of failed attempts and the time difference between the latest failed attempt and latest successful Login. If the time difference is between the threshold, then it is reported in the report.

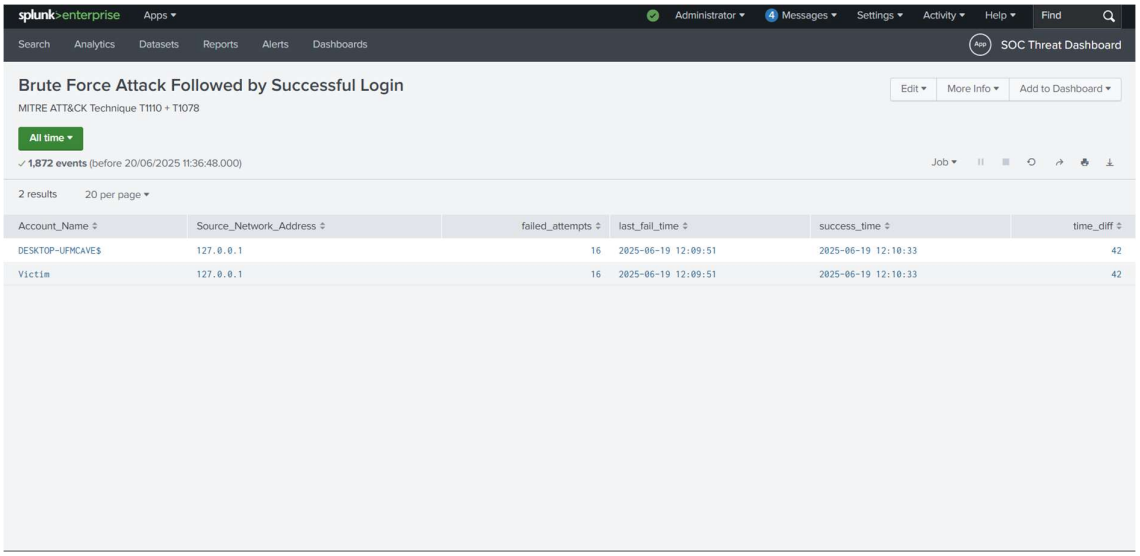
SPL Query:

```
index=main sourcetype="WinEventLog:Security" (EventCode=4625 OR EventCode=4624)
| eval event_type=if(EventCode==4625, "fail", "success")
| eval login_time=_time
| eval fail_time=if(event_type=="fail", login_time, null())
| eval success_login_time=if(event_type=="success", login_time, null())
| stats
   count(eval(event_type=="fail")) as failed_attempts
   latest(fail_time) as last_fail_time
   latest(success_login_time) as success_time
   by Account_Name, Source_Network_Address
| where failed_attempts >= 5 AND isnotnull(success_time)
| eval time_diff = success_time - last_fail_time
| where time_diff >= 0 AND time_diff <= 600
| eval last_fail_time=strftime(last_fail_time, "%Y-%m-%d %H:%M:%S")
| eval success_time=strftime(success_time, "%Y-%m-%d %H:%M:%S")
| table Account_Name, Source_Network_Address, failed_attempts, last_fail_time,
success_time, time_diff
```

Steps to save it as a report:

- Open Search Tab in SOC Threat Dashboard App.

- Paste the above SPL query and run it.
- Check the results and see if they are satisfactory.
- If they are satisfactory, Click Save As -> Report.
- Name: Brute Force Attack Followed by Successful Login
- Description: MITRE ATT&CK Technique T1110 + T1078
- Allow Time range picker and Click Save.
- You can view the report from the Reports Tab.



Brute Force Attack Followed by Success Login Report

PowerShell Execution Followed by Registry Persistence

The below SPL query identifies suspicious activity where PowerShell is used with potentially malicious flags (e.g., -encodedCommand, -bypass, -hidden) and correlates it with subsequent registry modifications (EventCode 13 or 4657) by the same user. It calculates the time difference between the PowerShell execution and the registry change. If the registry activity occurs within a short window (e.g., 5 minutes) of the suspicious PowerShell command, it flags it as a potential persistence attempt by an attacker.

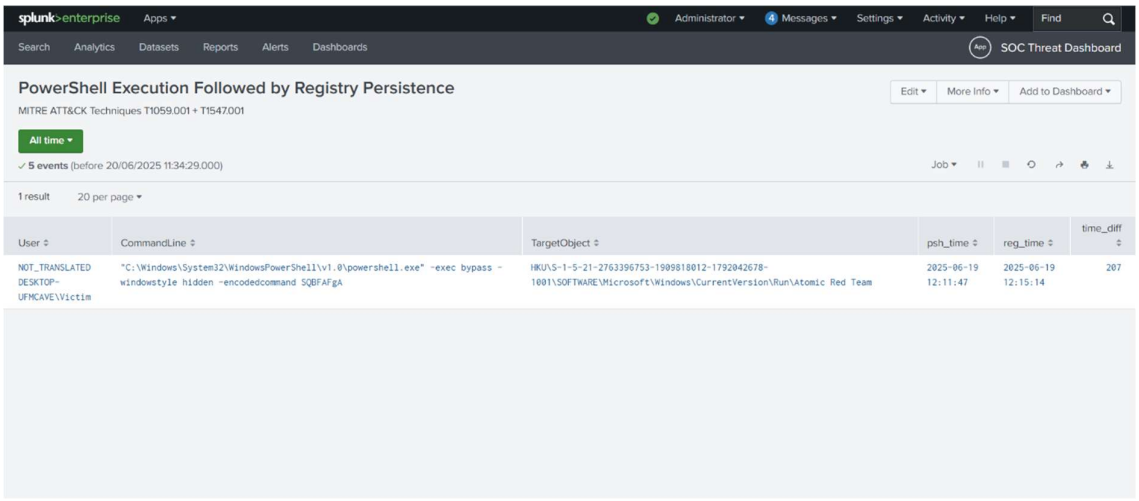
SPL Query:

```
index=main sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational"
EventCode=1
| eval suspicious=if(like(CommandLine, "%encodedcommand%") OR like(CommandLine,
"%bypass%") OR like(CommandLine, "%hidden%"), 1, 0)
| where suspicious=1
| rename _time as psh_time
| table psh_time, User, CommandLine
| join type=inner User [
  search index=main sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational"
(EventCode=13 OR EventCode=4657)
  | rename _time as reg_time
```

```
    | table reg_time, User, TargetObject
]
| eval time_diff = reg_time - psh_time
| where time_diff >= 0 AND time_diff < 300
| eval reg_time = strftime(reg_time, "%Y-%m-%d %H:%M:%S")
| eval psh_time = strftime(psh_time, "%Y-%m-%d %H:%M:%S")
| table User, CommandLine, TargetObject, psh_time, reg_time, time_diff
```

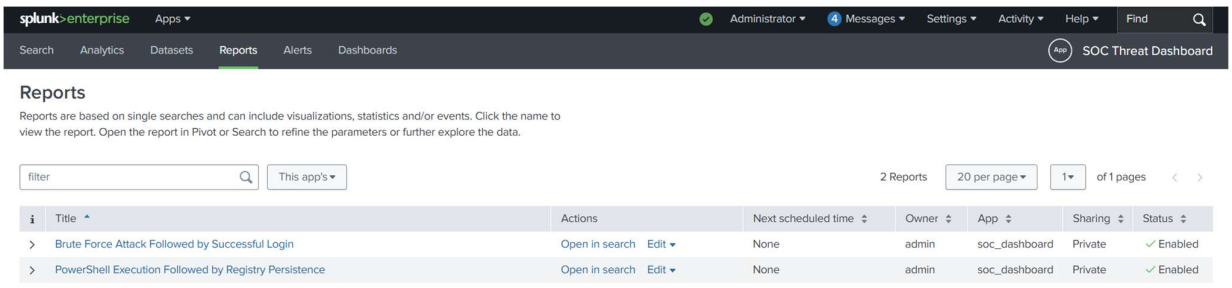
Steps to save it as a report:

- Open Search Tab in SOC Threat Dashboard App.
- Paste the above SPL query and run it.
- Check the results and see if they are satisfactory.
- If they are satisfactory, Click Save As -> Report.
- Name: PowerShell Execution Followed by Registry Persistence
- Description: MITRE ATT&CK Techniques T1059.001 + T1547.001
- Allow Time range picker and Click Save.
- You can view the report from the Reports Tab.



PowerShell Execution Followed by Registry Persistence Report

We can view all the reports available to you or created by you in the Reports tab.



Reports Tab

MITRE ATT&CK Techniques Used

Use Case	Technique ID	Technique Name
Suspicious PowerShell Execution	T1059.001	Command & Scripting Interpreter: PowerShell
Brute Force Login	T1110.001	Brute Force: Password Guessing
Registry Key Persistence	T1547.001	Boot/Logon AutoStart Execution: Registry Keys
Correlation: Brute Force → Successful Login	T1110 + T1078	Brute Force & Valid Accounts
Correlation: PowerShell Execution → Registry Persistence	T1059.001 + T1547.001	Command & Scripting Interpreter: PowerShell and Boot/Logon AutoStart Execution: Registry Keys

Conclusion

This hands-on project strengthened my skills in log analysis, detection engineering, and threat hunting using Splunk in a controlled lab environment. It demonstrated my ability to design and implement a detection pipeline from log ingestion and dashboard creation to alert generation and multi-event correlation using industry tools and threat simulation frameworks. The approach also emphasized the importance of MITRE ATT&CK mapping in real-world blue team operations.

Through this experience, I developed practical knowledge in identifying attack patterns, writing effective SPL queries, and simulating attacker behavior safely using Atomic Red Team. The project serves as a strong foundation for future work in SOC analysis, detection engineering, and blue team cybersecurity roles.