#90-Days-SOC-Challenge #Hax Security

#Challenge 1

Task #1: Explore and Collect Logs from Windows Event Viewer using PowerShell

Objective:

The goal of this task is to introduce PowerShell-based attack detection using Windows Event Viewer. Students will simulate a suspicious PowerShell command execution, retrieve logs using PowerShell, and analyze the security events that get recorded in the system.

Preparation:

By default, Powershell logs are not enabled. We need to enable both script blok logging and module execution logs.

Enable Logging for PowerShell Execution

- Press Win + R to open the Run dialog.
- Type gpedit.msc and press Enter to open the Group Policy Editor.
- Navigate to the following path: Computer Configuration > Administrative Templates > Windows Components > Windows PowerShell
- Turn on Module Logging
- Turn on Powershell Script Block Logging
- Turn on Script Execution
- Turn on Powershell Transcription
- Apply

Attack Simulation & Detection Using PowerShell

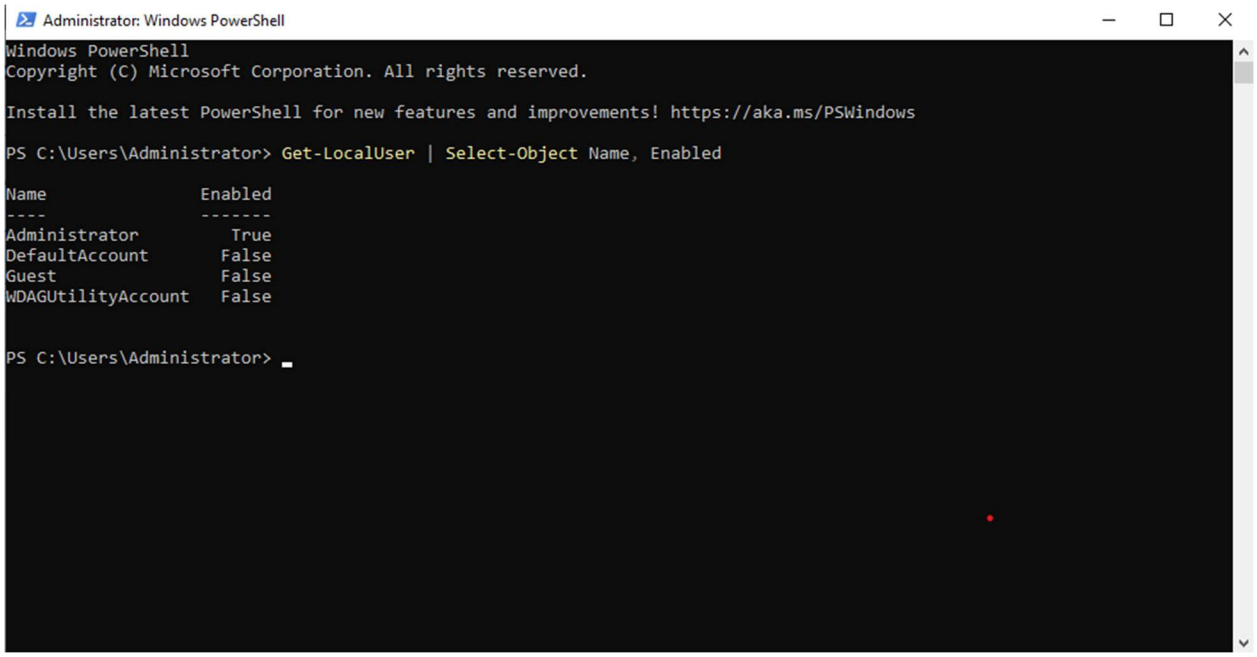
We will now simulate an attacker's reconnaissance technique by executing a PowerShell command that retrieves local user accounts.

Step 1: Execute a Suspicious PowerShell Command

Run the following command in an elevated PowerShell session:

Get-LocalUser | Select-Object Name, Enabled

This command lists all local user accounts on the system along with their status (enabled/disabled). Attackers use similar commands post-exploitation to enumerate users before escalating privileges.

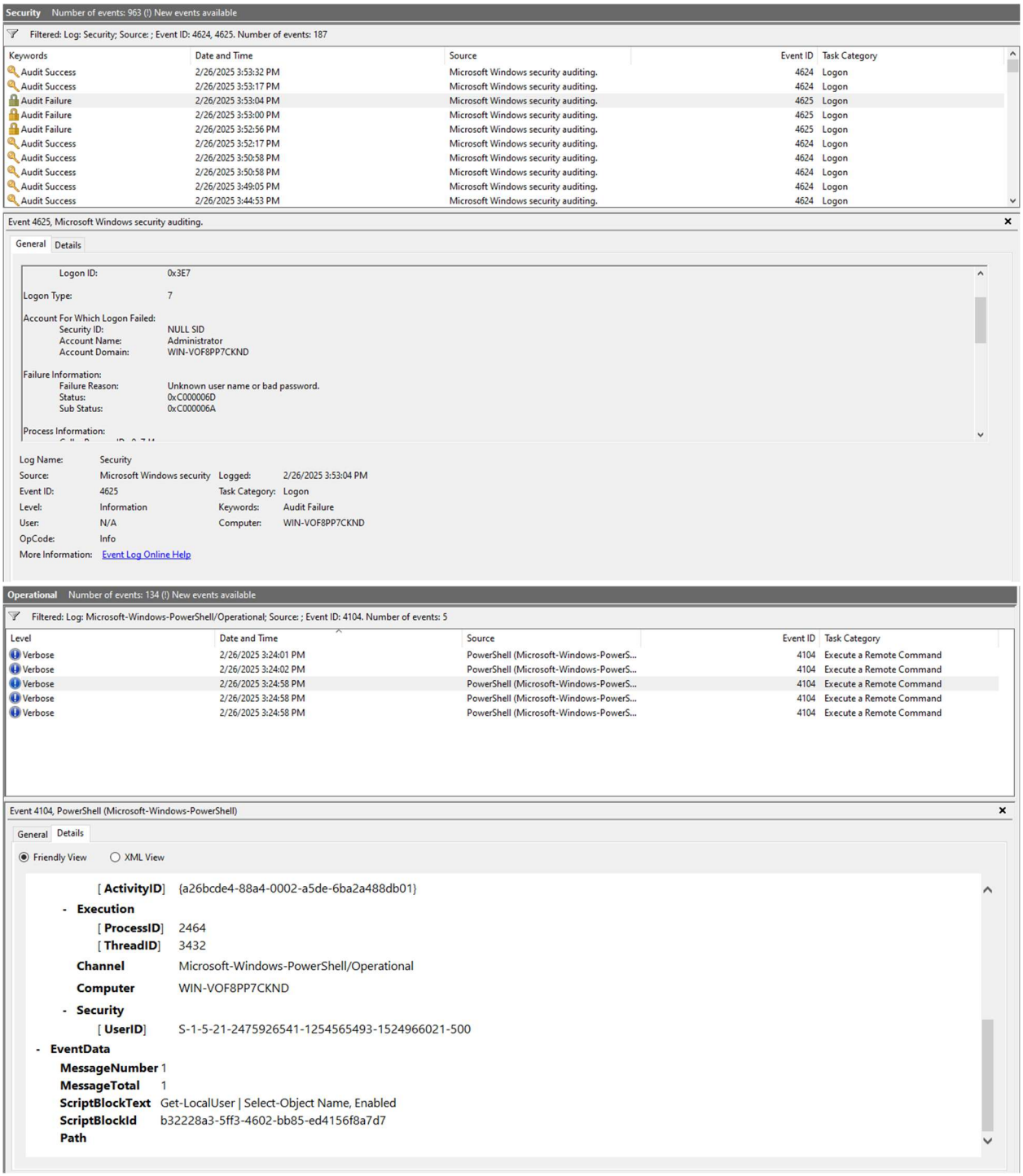


Step 2: Detect the Attack using Windows Event Viewer

1. Open Event Viewer (Win + R, type eventvwr.msc, press Enter).
2. Navigate to:

Applications and Services Logs → Microsoft → Windows → PowerShell → Operational

3. Click Filter Current Log and enter Event ID 4104 (Execute a Remote Command).
4. Locate the entry showing the execution of the Get-LocalUser command.
5. Take a screenshot of the event details.

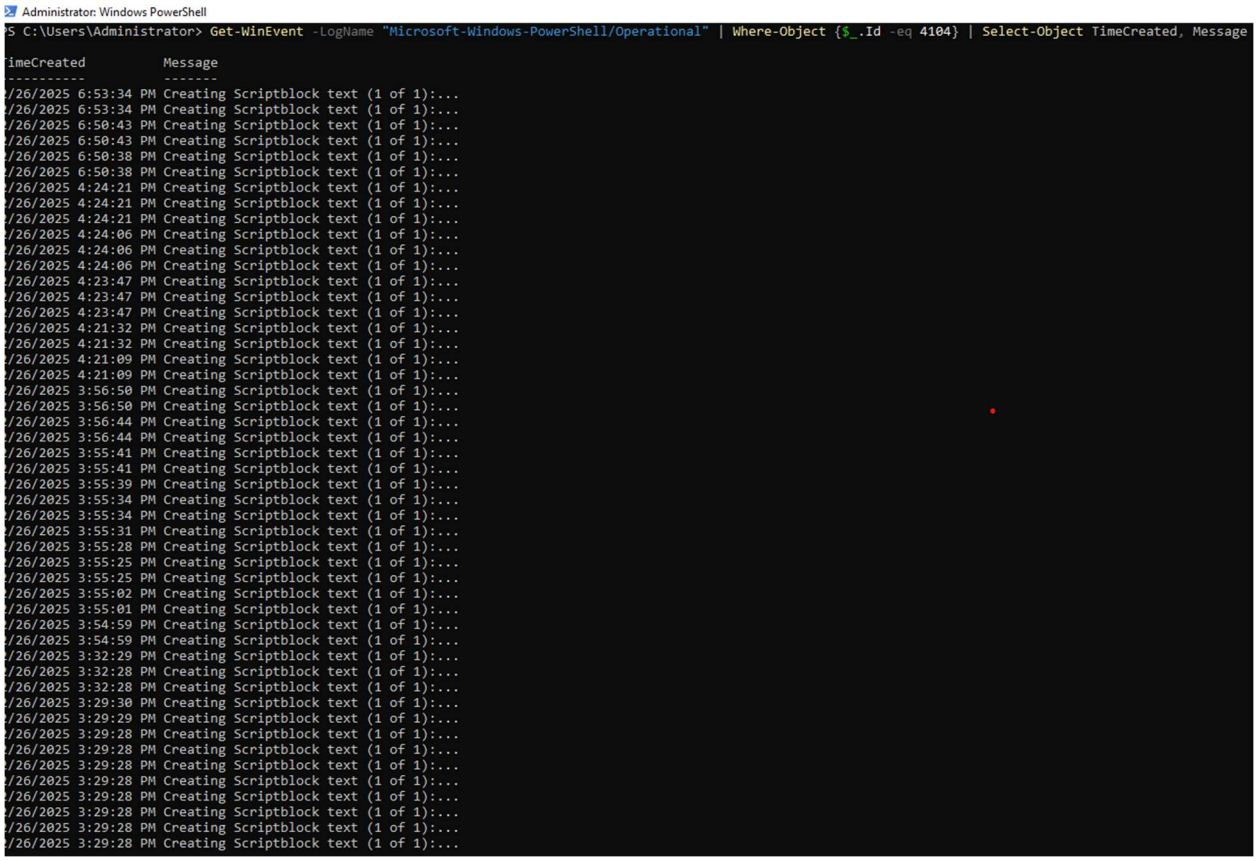


Step 3: Retrieve Logs using PowerShell (Alternative Detection Method)

Instead of using Event Viewer, use PowerShell to directly extract the event:

Get-WinEvent -LogName "Microsoft-Windows-PowerShell/Operational" | Where-Object {$_.Id -eq 4104} | Select-Object TimeCreated, Message

- This command fetches all script block executions from PowerShell logs and filters them by Event ID 4104.
- Look for the command Get-LocalUser in the output.



Lesson Learn:

- I have learnt PowerShell is a very powerful tool for not just automating and managing, but also for monitoring and security auditing.
- By enabling features like module logging, script block logging, and transcription, I can track what is happening on a system in a detailed way which is very useful for incident detection.