

# SOC Use Case Report

## UC-008: PowerShell Execution

<b>Author</b>	Abishek V
<b>Date</b>	22 January 2026
<b>Environment</b>	Home SOC Lab (VirtualBox)
<b>Primary Logs</b>	Windows Security Event Logs / PowerShell Logs
<b>Target OS</b>	Windows 10

## 1. Use Case Summary

<b>Use Case ID</b>	UC-008
<b>Use Case Name</b>	PowerShell Execution
<b>Category</b>	Execution / Living-off-the-Land
<b>SOC Tier</b>	L1 (Monitoring + Investigation)
<b>Severity Guideline</b>	Low → High (based on context)

PowerShell execution is common in Windows environments. This use case focuses on separating expected administrative usage from suspicious or malicious execution.

## 2. Scenario

- Endpoint: Windows 10 (192.168.56.110)
- User account: `clair`
- Execution context: Interactive PowerShell session
- Source system: Local workstation

**SOC risk point:** PowerShell execution alone is not malicious. Context, arguments, and follow-on behavior determine risk.

## 3. Telemetry and Evidence

### Primary logs

- Windows Security Event Logs
- PowerShell Operational Logs

### Key Event IDs

<b>4688</b>	Process creation ( <code>powershell.exe</code> )
<b>4103</b>	PowerShell module logging
<b>4104</b>	PowerShell script block logging
<b>4624</b>	Associated logon session

## 4. Detection Logic

Trigger when:

- PowerShell execution occurs on endpoints where usage is uncommon
- PowerShell launched by unusual parent process
- PowerShell executed by non-admin user in sensitive context

Risk amplifiers:

- execution outside business hours
- spawned child processes
- network connections initiated post-execution

## 5. SOC L1 Playbook

### Phase A: Triage

1. Confirm PowerShell execution event
2. Identify user, host, and parent process
3. Determine initial risk based on context

### Phase B: Investigation

1. Review command line and script content (if available)
2. Check parent-child process relationship
3. Validate user intent (admin task vs unexpected)
4. Review post-execution activity (processes, network)
5. Scope similar PowerShell executions across hosts

## 6. Evidence Timeline

Time	Event ID	Entity	Observation
16:41:12	4688	powershell.exe	PowerShell launched interactively
16:41:15	4103	ModuleLoad	PowerShell module loaded
16:41:22	4688	notepad.exe	Child process spawned
16:42:10	4624	clair	Active user session confirmed

**Outcome:** PowerShell execution observed. No immediate malicious indicators identified. Activity monitored.

## 7. False Positive Checks

- routine administrative scripts
- user troubleshooting activity
- developer or automation tasks

## 8. Verdict Criteria

### Benign if:

- command content is expected
- no suspicious child processes or network activity
- usage aligns with user role

### Escalate if:

- PowerShell launched by unusual parent
- suspicious follow-on behavior observed
- repeated execution across endpoints

## 9. SOC Response Actions

- continue monitoring for follow-on activity
- alert L2 if risk indicators appear
- review PowerShell usage baseline

## 10. Ticket Notes

**Ticket:** UC-008 PowerShell Execution

**Severity:** Low (monitoring)

**Verdict:** Benign (this instance)

### Analyst Notes

- Observed interactive PowerShell execution under user `clair`.
- No encoded commands, downloads, or suspicious follow-on activity detected.
- Activity aligns with expected usage. No escalation required at this time.