Incident Response Case Study Report 1

Case Type 1: Suspicious PowerShell Execution
SOC Analyst: Parvathy Krishnan
Environment: SOC Home Lab
SIEM: Splunk

---

1. Executive Summary

During routine SOC monitoring, a **high-severity alert** was generated by the SIEM platform indicating **suspicious PowerShell activity** on a Windows endpoint. The alert was triggered due to the execution of PowerShell using the `-EncodedCommand` parameter, a technique frequently leveraged by threat actors to **obfuscate commands and evade detection**.

The objective of this investigation was to:

- Determine whether the activity was **malicious or benign**
- Identify any **indicators of compromise (IOCs)**
- Assess the **scope and impact** of the activity
- Recommend appropriate **remediation and preventative controls**

Following a detailed investigation, the activity was determined to be **non-malicious in outcome**, but **high-risk in technique**. The alert was therefore classified as a **True Positive**, as the detection correctly identified behaviour commonly associated with attacker tradecraft.

2. Detection & Alert Context

Alert Information

- **Alert Name:** Suspicious PowerShell Execution
- **Severity:** High
- **Detection Source:** Splunk SIEM
- **Trigger Condition:** Execution of PowerShell with `-EncodedCommand` flag
- **Affected Host:** WIN10-ENDPOINT-01
- **User Account:** standard_user

Why This Alert Matters

Encoded PowerShell execution is a **well-documented attack technique** used by adversaries to:

- Conceal malicious scripts
- Execute payloads in memory
- Bypass traditional signature-based security controls
- Because legitimate administrative use of encoded PowerShell is limited, this behaviour is treated as **high risk** and requires immediate SOC investigation to rule out malware execution or post-exploitation activity.

Splunk Query – Alert Trigger Validation

```
index=windows_logs process_name="powershell.exe" CommandLine="*-EncodedCommand*"
| table _time host user process_name CommandLine
```

3. Initial Triage & Validation

Actions Taken

- Verified that the alert originated from legitimate SIEM telemetry
- Confirmed the timestamp and host alignment
- Checked for similar alerts across other endpoints
- Identified the privilege level of the executing user account

Splunk Query – Check for Similar Alerts

```
index=windows_logs process_name="powershell.exe" CommandLine="*-EncodedCommand*"
| stats count by host
```

Initial Findings

- The activity was **isolated to a single endpoint**
- No related alerts were detected on other systems
- The user account did **not have administrative privileges**
- No immediate indicators of lateral movement were observed

Based on triage results, the alert was deemed **valid** and escalated for a full investigation.

4. Data Collection & Log Analysis

Log Sources Reviewed

- Windows Security Event Logs
- Sysmon Process Creation Logs
- PowerShell Operational Logs
- Network Traffic Logs

Splunk Query – Process Creation Analysis

```
index=sysmon EventCode=1 Image="*powershell.exe*"
| table _time host User ParentImage Image CommandLine
```

Key Observations

- PowerShell was executed interactively by the logged-in user
- The command included the `-EncodedCommand` argument
- The parent process was identified as `explorer.exe`
- No suspicious child processes were spawned
- No persistence mechanisms (registry keys, scheduled tasks) were observed
- Network logs showed **no outbound connections** related to the execution

These observations suggested **local execution without follow-on malicious activity**.

5. Command & Behaviour Analysis

Decoded Command Findings

The encoded PowerShell command was decoded and manually reviewed:

- The command performed **basic system enumeration**
- No external payloads were downloaded
- No registry modifications or scheduled tasks were created
- No credential access or privilege escalation attempts were detected

Splunk Query – Extract Encoded Command

```
index=sysmon Image="*powershell.exe*" CommandLine="*-EncodedCommand*"
| rex field=CommandLine "(?<encoded_cmd>[A-Za-z0-9+/=]{20,})"
| table _time host encoded_cmd
```

Behavioural Assessment

While the **technique mirrored common attacker behaviour**, the actual execution **did not result in malicious impact**. This distinction between **technique-based risk** and **execution outcome** was critical in determining the final classification.

6. Timeline Reconstruction

Time (UTC)	Event
14:28	User logged into Windows endpoint
14:30	PowerShell process initiated
14:31	Encoded PowerShell command executed
14:32	SIEM alert generated
14:35	SOC investigation initiated
14:55	Encoded command decoded and analyzed
15:10	Incident classified and documented

7. Indicator of Compromise (IOC) Analysis

IOCs Identified

- **Process:** powershell.exe
- **Command Flag:** -EncodedCommand
- **Host:** WIN10-ENDPOINT-01
- **User:** standard_user

Splunk Query – IOC Pivoting

```
index=windows_logs host="WIN10-ENDPOINT-01"
| search process_name="powershell.exe"
| table _time user process_name CommandLine
```

IOC Enrichment

- No malicious IP addresses identified
- No suspicious domains contacted
- No known malware hashes detected
- No matches found in threat intelligence sources

The absence of external IOCs supported a **non-compromised outcome**.

8. Scope & Impact Assessment

Scope

- **Affected Systems:** 1
- **Affected Users:** 1
- **Network Impact:** None
- **Data Impact:** None

Impact Conclusion

Although the technique used presented potential risk, the investigation confirmed **no system compromise, no data loss, and no network impact**.

9. Root Cause Analysis

Root Cause

The activity originated from a **user-initiated PowerShell execution** conducted during testing activities. While the intent was benign, the execution method closely resembled **malicious attacker techniques**.

Contributing Factors

- Unrestricted PowerShell execution policies
- Encoded command usage not explicitly blocked
- Limited contextual information available at execution time
-

10.MITRE ATT&CK Mapping

Tactic	Technique	ID
Execution	PowerShell	T1059.001

11. Final Verdict

True Positive – Suspicious Activity (Non-Malicious Outcome)

The detection successfully identified high-risk behaviour consistent with attacker tradecraft. No compromise was confirmed.

12. Remediation & Recommendations

- Enforce PowerShell Constrained Language Mode
- Enable enhanced PowerShell logging (Script Block Logging)

- Restrict or monitor encoded PowerShell execution
- Improve SIEM alert enrichment with user context
- Continue monitoring the endpoint for repeated behaviour

13. Lessons Learned

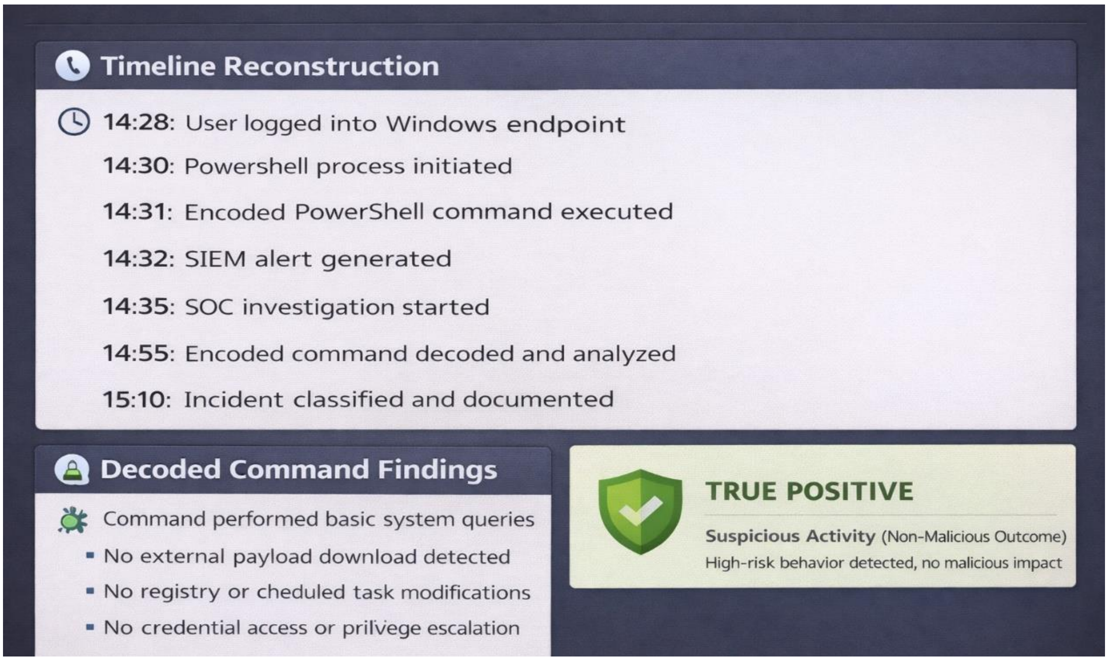
- Technique-based detections remain critical even when outcomes are benign
- Encoded PowerShell execution should always be investigated
- Contextual analysis prevents false escalation
- Comprehensive logging significantly improves SOC visibility

14. Analyst Notes

This case reinforced the importance of:

- Separating **detection accuracy** from **malicious outcome**
- Applying structured investigation methodologies
- Maintaining consistent and thorough SOC documentation

15. Summary



Timeline Reconstruction

14:28: User logged into Windows endpoint
14:30: Powershell process initiated
14:31: Encoded PowerShell command executed
14:32: SIEM alert generated
14:35: SOC investigation started
14:55: Encoded command decoded and analyzed
15:10: Incident classified and documented

Decoded Command Findings

Command performed basic system queries
- No external payload download detected
- No registry or cheduled task modifications
- No credential access or prilvege escalation

TRUE POSITIVE

Suspicious Activity (Non-Malicious Outcome)
High-risk behavior detected, no malicious impact