




# Supercharge into Roota

Uncoder AI

In production

From Splunk Alert (SPL) 

SUPERCHARGE



```
1 [PowerShell Base64 Encoded Invoke Keyword]
2 alert.severity = 4
3 description = Detects UTF-8 and UTF-16 Base64 encoded powershell 'Invoke-' calls (Rule ID: 6385697e-9f1b-40bd-8817-f4a91f40508e) Reference: https://tdm.socprime.com/tdm/info/
4 cron_schedule = 0 * * * *
5 disabled = 1
6 is_scheduled = 1
7 is_visible = 1
8 dispatch.earliest_time = -60m@m
9 dispatch.latest_time = now
10 search = index=* source="WinEventlog:Microsoft-Windows-Sysmon/Operational" AND (((NewProcessName="*\\powershell.exe" OR NewProcessName="*\\pwsh.exe") OR (OriginalFileName="PowerShell.EXE" OR OriginalFileName="pwsh.dll")) AND
    Command line="* -e*" AND (Command line="*SQRuAHVhWReAGll!l0*" OR Command line="*kAhR2dCRAsuRlAFCA*" OR Command line="*7dC4AdnRvAFcA70nA*" OR Command line="*SWC3h2!l!*" OR Command line="*ludm0r7S*" OR Command line="*7bn7uz2!l!*
```

To  Roota

Enhance With More Queries

Save As 

```
20 logsource: {}
21 timeline: ''
22 - false-positives: |-
23   Possible False-Positives or Benign Activities for PowerShell Base64 Encoded Invoke Keyword
24   1. Legitimate administrative scripts that utilize Base64 encoding for obfuscation.
25   2. Third-party applications that use PowerShell for automation and may encode commands in Base64.
26   3. Security tools that leverage PowerShell for legitimate purposes, such as endpoint protection or monitoring.
27   4. User-initiated PowerShell scripts that are encoded for ease of transfer or storage.
28   5. Scheduled tasks or cron jobs that execute Base64 encoded PowerShell commands for routine maintenance.
29
30   Recommendations to Avoid False-Positives or Benign Activities
31   1. Implement a whitelist of known legitimate scripts and applications that use Base64 encoding.
32   2. Monitor the context in which the Base64 encoded commands are executed, including user identity and execution time.
33   3. Analyze the content of the decoded Base64 strings to determine if they align with known benign activities.
34   4. Correlate PowerShell execution events with other logs (e.g., user activity, file access) to establish a clearer picture of intent.
35   5. Educate users on the risks of using Base64 encoding in scripts and encourage the use of clear, readable code.
36 - triage_recommendations: |-
37   Possible Actions for Validating and Investigating Malicious Activity
38
39   1. Review logs for PowerShell execution events to identify any suspicious 'Invoke-' calls.
40   2. Analyze the Base64 encoded strings to determine their decoded content and intent.
41   3. Check for unusual patterns or anomalies in the PowerShell command execution history.
42   4. Correlate the identified PowerShell activity with known threat intelligence to assess potential risks.
43   5. Investigate the source of the PowerShell execution to determine if it originated from a legitimate user or process.
```

**Turn a platform-specific rule or query into a Roota rule and enrich it with metadata using SOC Prime's proprietary algorithms and AI.**

- AI input on possible false positives and triage recommendations (only metadata is used for prompting)
- Adding possible log sources if they were not specified in the original content. Additionally, the audit section is filled that specifies what logging service should be enabled to have the logs required and how to enable it
- Prediction of relevant MITRE ATT&CK techniques and sub-techniques with a machine learning model