SOC Playbook: Suspicious Script Execution Detection (T1059)

1. Objective

Detect and respond to unauthorized, suspicious, or potentially malicious script execution on endpoints across Windows, Linux, and macOS environments.

2. Scope

- Windows PowerShell scripts, batch files, WMI scripts
- Linux/macOS Bash, Zsh, Python, Ruby, Perl scripts
- Execution from unusual locations or with suspicious parameters
- Obfuscated or encoded script content

3. Log Sources

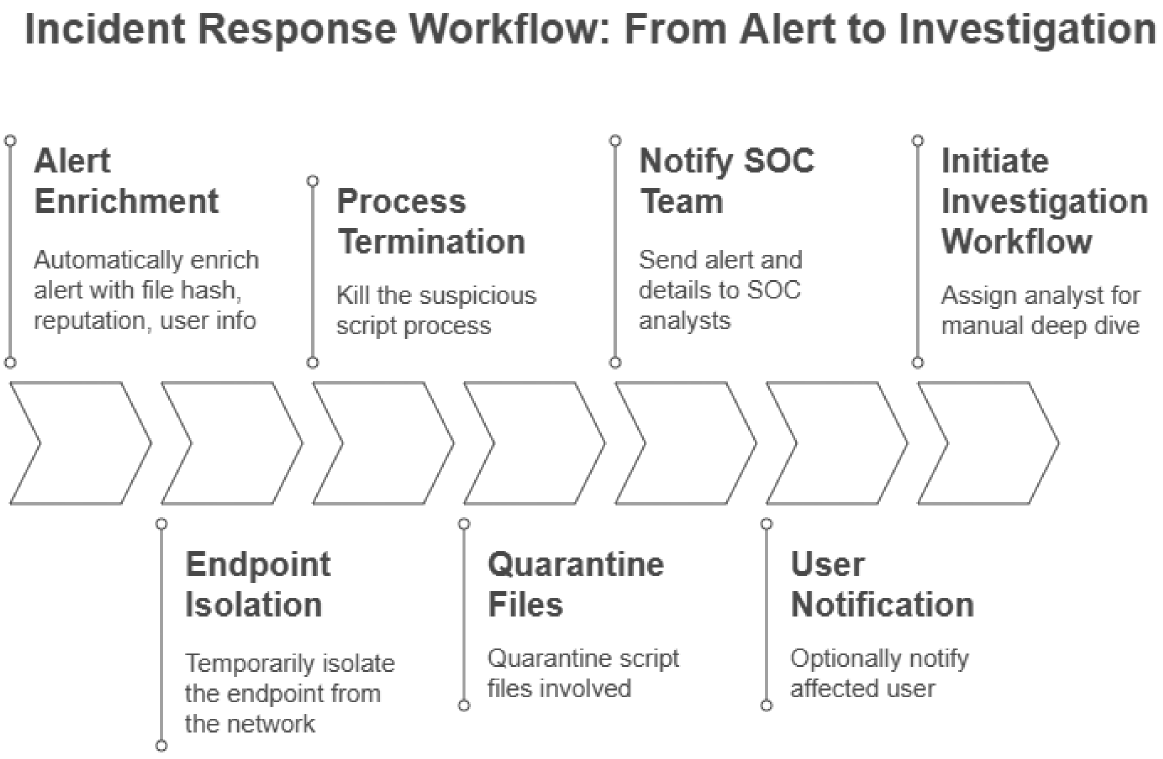
Platform	Log Source	Description
Windows	PowerShell Logs (Event ID 4104), Security Logs (4688), Sysmon (1)	Script execution, process creation
Linux/mac OS	Syslog, Auditd (execve syscall), Shell histories (.bash_history, .zsh_history), OSQuery	Script execution and command history
All	File Integrity Monitoring (Tripwire, OSQuery)	Changes to script files

4. Detection Rules / Alerts

Alert Name	Description	Conditions / Triggers
Suspicious Script Execution	Script run from unusual locations or user context	Script launched from temp directories, downloads folder, or unknown user folders
PowerShell Script Block Logging Alert	PowerShell script blocks with suspicious content	Event ID 4104 with obfuscated or encoded commands
Process Creation of Scripting Hosts	Unexpected invocation of scripting hosts	New process creation of powershell.exe, bash, python from unusual parents or paths
Obfuscated Script Detection	Scripts with suspicious encoding or packing	Base64 or encoded payload in script content

5. Automated Response Play

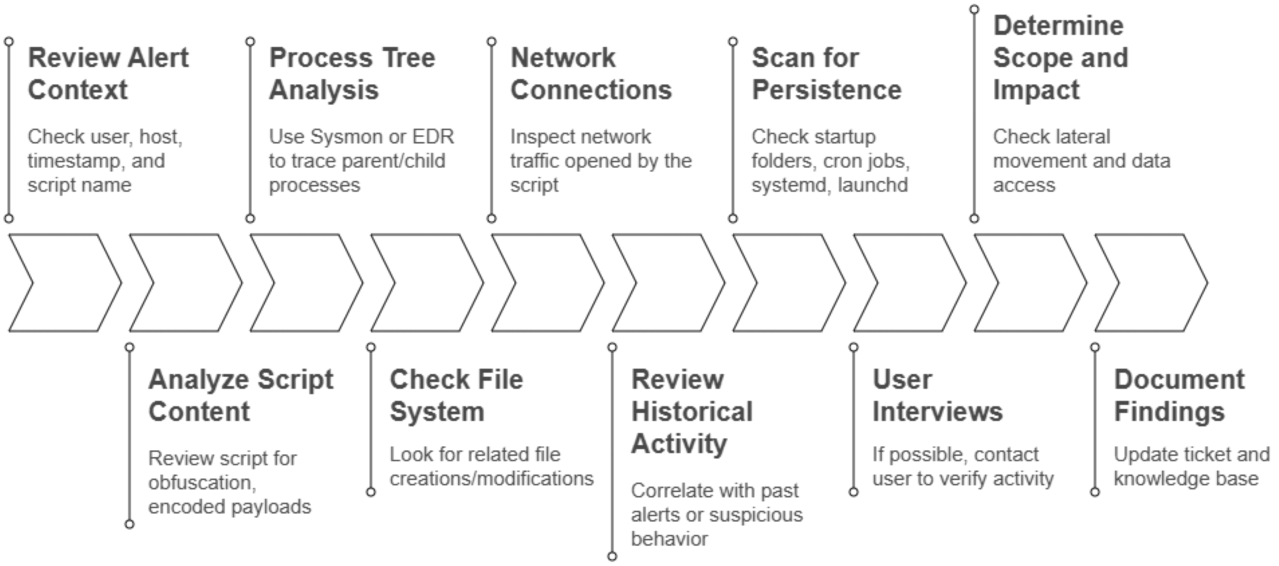
Step	Action	Notes
1. Alert Enrichment	Automatically enrich alert with file hash, reputation, user info	Integrate VirusTotal, internal reputation DB
2. Endpoint Isolation	Temporarily isolate the endpoint from the network	Prevent lateral movement
3. Process Termination	Kill the suspicious script process	Use EDR or remote execution tools
4. Quarantine Files	Quarantine script files involved	Prevent further execution
5. Notify SOC Team	Send alert and details to SOC analysts	Use Slack, email, or ticketing system
6. User Notification	Optionally notify affected user	To gather context or inform
7. Initiate Investigation Workflow	Assign analyst for manual deep dive	Involve threat intel, review logs, and system state


Incident Response Workflow: From Alert to Investigation
Alert Enrichment
Automatically enrich alert with file hash, reputation, user info
Endpoint Isolation
Temporarily isolate the endpoint from the network
Process Termination
Kill the suspicious script process
Quarantine Files
Quarantine script files involved
Notify SOC Team
Send alert and details to SOC analysts
User Notification
Optionally notify affected user
Initiate Investigation Workflow
Assign analyst for manual deep dive

6. Investigation Checklist

Step	Description / Tools
1. Review Alert Context	Check user, host, timestamp, and script name
2. Analyze Script Content	Review script for obfuscation, encoded payloads
3. Process Tree Analysis	Use Sysmon or EDR to trace parent/child processes
4. Check File System	Look for related file creations/modifications
5. Network Connections	Inspect network traffic opened by the script
6. Review Historical Activity	Correlate with past alerts or suspicious behavior
7. Scan for Persistence	Check startup folders, cron jobs, systemd, launchd
8. User Interviews	If possible, contact user to verify activity
9. Determine Scope and Impact	Check lateral movement and data access
10. Document Findings	Update ticket and knowledge base

Comprehensive Alert Analysis Process


Review Alert Context
Check user, host, timestamp, and script name
Analyze Script Content
Review script for obfuscation, encoded payloads
Process Tree Analysis
Use Sysmon or EDR to trace parent/child processes
Check File System
Look for related file creations/modifications
Network Connections
Inspect network traffic opened by the script
Review Historical Activity
Correlate with past alerts or suspicious behavior
Scan for Persistence
Check startup folders, cron jobs, systemd, launchd
User Interviews
If possible, contact user to verify activity
Determine Scope and Impact
Check lateral movement and data access
Document Findings
Update ticket and knowledge base

7. Playbook Notes

- Tune detection rules to reduce false positives from legitimate admin scripts.
- Enable script block logging for PowerShell on Windows endpoints.
- Regularly update known good script hashes for allowlisting.
- Integrate threat intelligence for reputation-based blocking.
- Incorporate endpoint detection and response (EDR) tools for automated containment.