Windows Defender Exclusions Added via PowerShell

edit

Identifies modifications to the Windows Defender configuration settings using PowerShell to add exclusions at the folder directory or process level.

Rule type: eql

Rule indices:

- winlogbeat-*
- logs-endpoint.events.process-*
- logs-windows.forwarded*
- logs-windows.sysmon_operational-*
- endgame-*
- logs-system.security*
- logs-m365_defender.event-*
- logs-sentinel_one_cloud_funnel.*

Severity: medium

Risk score: 47

Runs every: 5m

Searches indices from: now-9m (Date Math format, see also `Additional look-back time`)

Maximum alerts per execution: 100

References:

- https://www.bitdefender.com/files/News/CaseStudies/study/400/Bitdefender-PR-Whitepaper-MosaicLoader-creat5540-en-EN.pdf
- https://www.elastic.co/security-labs/elastic-security-uncovers-blister-malware-campaign
- https://www.elastic.co/security-labs/operation-bleeding-bear
- https://www.elastic.co/security-labs/invisible-miners-unveiling-ghostengine

Tags:

- Domain: Endpoint
- OS: Windows
- Use Case: Threat Detection
- Tactic: Defense Evasion
- Resources: Investigation Guide
- Data Source: Elastic Endgame
- Data Source: Elastic Defend
- Data Source: System

Rule license: Elastic License v2

Investigation guide

edit

Triage and analysis

Investigating Windows Defender Exclusions Added via PowerShell

Microsoft Windows Defender is an antivirus product built into Microsoft Windows. Since this software product is used to prevent and stop malware, it’s important to monitor what specific exclusions are made to the product’s configuration settings. These can often be signs of an adversary or malware trying to bypass Windows Defender’s capabilities. One of the more notable examples was observed in 2018 where Trickbot incorporated mechanisms to disable Windows Defender to avoid detection.

Possible investigation steps

- Investigate the process execution chain (parent process tree) for unknown processes. Examine their executable files for prevalence, whether they are located in expected locations, and if they are signed with valid digital signatures.
- Identify the user account that performed the action and whether it should perform this kind of action.
- Contact the account owner and confirm whether they are aware of this activity.
- Examine the exclusion in order to determine the intent behind it.
- Assess whether this behavior is prevalent in the environment by looking for similar occurrences across hosts.
- If the exclusion specifies a suspicious file or path, retrieve the file(s) and determine if malicious:
- Use a private sandboxed malware analysis system to perform analysis.
- Observe and collect information about the following activities:
- Attempts to contact external domains and addresses.
- File and registry access, modification, and creation activities.
- Service creation and launch activities.
- Scheduled task creation.
- Use the PowerShell Get-FileHash cmdlet to get the files' SHA-256 hash values.
- Search for the existence and reputation of the hashes in resources like VirusTotal, Hybrid-Analysis, CISCO Talos, Any.run, etc.

False positive analysis

- This rule has a high chance to produce false positives due to how often network administrators legitimately configure exclusions. In order to validate the activity further, review the specific exclusion and its intent. There are many legitimate reasons for exclusions, so it’s important to gain context.

Related rules

- Isolate the involved host to prevent further post-compromise behavior.
- If the triage identified malware, search the environment for additional compromised hosts.
- Implement temporary network rules, procedures, and segmentation to contain the malware.
- Stop suspicious processes.
- Immediately block the identified indicators of compromise (IoCs).
- Inspect the affected systems for additional malware backdoors like reverse shells, reverse proxies, or droppers that attackers could use to reinfect the system.
- Remove and block malicious artifacts identified during triage.
- Run a full antimalware scan. This may reveal additional artifacts left in the system, persistence mechanisms, and malware components.
- Exclusion lists for antimalware capabilities should always be routinely monitored for review.
- Determine the initial vector abused by the attacker and take action to prevent reinfection through the same vector.
- Using the incident response data, update logging and audit policies to improve the mean time to detect (MTTD) and the mean time to respond (MTTR).

Rule query

edit

```
process where host.os.type == "windows" and event.type == "star
 (process.name : ("powershell.exe", "pwsh.exe", "powershell_ise
  process.args : ("*Add-MpPreference*", "*Set-MpPreference*") a
  process.args : ("*-Exclusion*")
```

Framework: MITRE ATT&CK^{TM}

- Tactic:
  - Name: Defense Evasion
  - ID: TA0005
  - Reference URL: https://attack.mitre.org/tactics/TA0005/
- Technique:
  - Name: Impair Defenses
  - ID: T1562
  - Reference URL: https://attack.mitre.org/techniques/T1562/
- Sub-technique:
  - Name: Disable or Modify Tools
  - ID: T1562.001
  - Reference URL: https://attack.mitre.org/techniques/T1562/001/
- Sub-technique:

- Technique:
  - Name: Command and Scripting Interpreter
  - ID: T1059
  - Reference URL: https://attack.mitre.org/techniques/T1059/
- Sub-technique:
  - Name: PowerShell
  - ID: T1059.001
  - Reference URL: https://attack.mitre.org/techniques/T1059/001/

« Windows Defender Disabled via Registry Modification

Windows Event Logs Cleared »