Security Control Validation

EDR-Freeze: Security Agent Freezing via Windows Error Reporting (WerFaultSecure)

Purple Team Testing Plan — v2.0
Prepared: February 18, 2026
Classification: CONFIDENTIAL
MITRE ATT&CK: T1562.001

PCSIRT — Purple Team Operations

Table of Contents

1. Executive Summary

This document presents a comprehensive testing plan to validate a detection use case targeting **EDR-Freeze**, a publicly available proof-of-concept tool that abuses **WerFaultSecure.exe** (Windows Error Reporting) to suspend endpoint security agents. The tool operates entirely in user mode — requiring no vulnerable drivers — and leverages Protected Process Light (PPL) plumbing and the MiniDumpWriteDump API to freeze EDR/AV processes for an attacker-controlled duration.

The plan includes seven test scenarios, updated simulation scripts aligned to the actual EDR-Freeze command-line signatures, and a detailed gap analysis of the current detection queries based on threat intelligence from the tool's source code, Binary Defense ARC Labs analysis, and community Sigma rules.

Scope

Tool: EDR-Freeze (github.com/TwoSevenOneT/EDR-Freeze)

MITRE: T1562.001 — Impair Defenses: Disable or Modify Tools

Detection Source: Event ID 4688, Sysmon Events 1/10/11/23

Key Indicators: WerFaultSecure.exe + /h /pid /tid /type 268310 /encfile /cancel

Test Scenarios: 7 (3 true-positive, 4 true-negative)

Scripts: Simulate-EDRFreeze.ps1, validate_edr_freeze.py

2. Threat Intelligence Background

EDR-Freeze was published by researcher TwoSevenOneT and has been analyzed by multiple security teams. The tool demonstrates a realistic attack primitive that is attractive to both criminal and advanced adversary operations due to its user-mode execution and reliance on legitimate OS components.

2.1 Why This Matters

Unlike BYOVD (Bring Your Own Vulnerable Driver) techniques, EDR-Freeze does not require deploying malicious drivers or exploiting kernel vulnerabilities. It leverages the Windows Error Reporting service — a signed, trusted OS component — to achieve process suspension. As noted in the Binary Defense ARC Labs analysis, this technique fits well within ransomware operations during credential harvesting phases that rely on Mimikatz and LSASS dumps, against which EDR vendors have developed countermeasures.

2.2 Operational Impact

If an attacker can suspend an EDR process for even 1–3 seconds, they can perform credential dumps, persistence installation, file staging, or lateral movement with greatly reduced detection probability. The tool's author recommends short freeze durations with immediate follow-on execution, or embedding attack payloads directly into the EDR-Freeze source code.

References

EDR-Freeze Repository — github.com/TwoSevenOneT/EDR-Freeze

Author Write-up — zerosalarium.com/2025/09/EDR-Freeze-Puts-EDRs-Antivirus-Into-Coma.html

ARC Labs Analysis — binarydefense.com/resources/blog/dont-freeze-me-out-bro...

Hunt Guide — blog.axelarator.net/hunting-for-edr-freeze/

Sigma Rule — SigmaHQ proc_creation_win_werfaultsecure_process_freeze

ColdWer (Cobalt Strike BOF variant) — github.com/0xsh3llf1r3/ColdWer

3. EDR-Freeze Technical Analysis

Based on source code review and public analyses, EDR-Freeze executes the following steps:

Step	Action	Detail
1	SeDebugPrivilege	Enables SeDebugPrivilege via AdjustTokenPrivileges to interact with system processes
2	Create dump handle	Creates an inheritable file handle for t.txt (the dump encfile) and an inheritable cancel event
3	Launch WerFaultSecure as PPL	Uses CreateProcessW with EXTENDED_STARTUPINFO_PRESENT | CREATE_PROTECTED_PROCESS and PROC_THREAD_ATTRIBUTE_PROTECTION_LEVEL to spawn WerFaultSecure.exe at WinTCB PPL level
4	Pass attack parameters	Command line: /h /pid <TargetPID> /tid <TargetTID> /type 268310 /encfile <handle> /cancel <handle>
5	MiniDumpWriteDump suspends target	WerFaultSecure calls MiniDumpWriteDump (from dbgcore.dll) which suspends all threads in the target process
6	Suspend WerFaultSecure	Monitor thread detects target suspension, then calls NtSuspendProcess on WerFaultSecure itself — target remains frozen
7	Sleep & cleanup	After SleepTime expires, terminates WerFaultSecure (target resumes) and deletes t.txt

3.1 Hardcoded Indicators

Indicator	Description
/type 268310	Full dump mode flag — hardcoded in source as '// dump full'. High-fidelity pivot.
/h	Headless mode — always present. Prevents WER GUI from appearing.
/encfile <handle>	Inheritable dump file handle passed as decimal integer.
/cancel <handle>	Inheritable cancel event handle.
/pid <PID>	Target process PID (the EDR/AV to freeze).
/tid <TID>	Target main thread ID.
t.txt	Default temp dump file created in the working directory, deleted at cleanup.
EDR-Freeze.exe <PID> <SleepMs>	Tool invocation with two numeric arguments.

4. Detection Logic Analysis

4.1 Current Detection (Under Test)

The current use case employs two queries against Event ID 4688. **Query 1** looks for WerFaultSecure.exe where the command line matches wildcards `*type*` and `*encfile*`. **Query 2** looks for any 4688 event whose command line matches regex `.*\d+\s+\d+`. A correlation join checks that the ProcessId from Query 1 equals the NewProcessId from Query 2.

Gap Analysis: Current Queries vs. Actual EDR-Freeze Behavior

Issue 1 — Broad wildcards: `*type*` matches any command line containing 'type' (e.g., filetype, prototype). Should target `/type 268310` specifically.

Issue 2 — Missing high-fidelity flags: The detection does not check for /h, /pid, /tid, or /cancel, which are always present in EDR-Freeze invocations and form the basis of the Sigma rule.

Issue 3 — Overly permissive regex: `.*\d+\s+\d+` matches any command line with two numbers separated by whitespace (very high false-positive potential).

Issue 4 — No file artifact correlation: The t.txt create/delete pattern is a strong supplementary signal not currently leveraged.

Issue 5 — No Sysmon enrichment: ProcessAccess (EventID 10) with PROCESS_SUSPEND_RESUME (0x0800) and CallTrace containing dbgcore.dll are high-confidence signals not in scope.

4.2 Recommended Enhanced Detection

Based on the Sigma rule and ARC Labs analysis, the following enhanced query logic would significantly improve fidelity:

Signal	Detection Logic
WerFaultSecure image	Image ends with \WerFaultSecure.exe OR OriginalFileName = WerFaultSecure.exe
Required flags (all)	CommandLine contains ALL of: /h, /pid, /encfile, /cancel, /type
Dump type literal	CommandLine contains '268310'
Supplementary: t.txt	FileCreate targeting 't.txt' within 5s of WerFaultSecure creation
Supplementary: ProcessAccess	Sysmon EventID 10 with GrantedAccess 0x0800 against EDR process + dbgcore.dll in CallTrace
Supplementary: Heartbeat	EDR agent heartbeat/telemetry gap correlated with above

5. Test Environment Prerequisites

Requirement	Detail
Event ID 4688 Auditing	Audit Process Creation = Success + command-line logging enabled
Sysmon (recommended)	EventIDs 1 (ProcessCreate), 10 (ProcessAccess), 11 (FileCreate), 23 (FileDelete)
SIEM Ingestion	Windows Security + Sysmon logs flowing to detection pipeline
Test Host	Windows 10/11 endpoint (non-production), administrator access
SOC Coordination	Maintenance window agreed; test hostname/operator/times documented
Scripts Deployed	Simulate-EDRFreeze.ps1 and validate_edr_freeze.py on test host

6. Test Scenarios

6.1 True Positive: Full EDR-Freeze Chain

Script: `-TestMode TruePositive`

Confirm the use case fires when the exact EDR-Freeze command-line pattern executes: WerFaultSecure.exe /h /pid /tid /type 268310 /encfile /cancel, plus the EDR-Freeze parent process with numeric PID/SleepTime arguments.

Expected: Alert SHOULD fire with correct hostname grouping and PID correlation.

6.2 True Positive: Variant Command Lines

Script: `-TestMode VariantCmdLine`

Test flag reordering, mixed case (/H /PID vs /h /pid), and extra whitespace to assess whether wildcard/regex matching is robust to cosmetic variations.

Expected: Alert SHOULD fire for all variants.

6.3 True Positive: Full Chain with Artifacts

Script: `-TestMode FullChainWithArtifacts`

Complete attack simulation including t.txt file creation, EDR-Freeze parent process, WerFaultSecure child process with all flags, and t.txt cleanup. Validates all detection signals fire together.

Expected: Primary alert + supplementary artifact detections SHOULD fire.

6.4 True Negative: Query 1 Only

Script: `-TestMode Query1Only`

Only WerFaultSecure with attack flags — no correlated EDR-Freeze parent process. PID correlation should fail.

Expected: Alert should NOT fire.

6.5 True Negative: Query 2 Only

Script: `-TestMode Query2Only`

Numeric-argument process (PID + SleepTime) spawned from PowerShell — no WerFaultSecure child. PID correlation should fail.

Expected: Alert should NOT fire.

6.6 True Negative: Legitimate WER Crash

Script: `-TestMode LegitWER`

Intentional application crash triggering legitimate WER. Command line will NOT contain /type 268310, /encfile, or /cancel.

Expected: Alert should NOT fire.

6.7 True Negative: Artifact Only

Script: `-TestMode ArtifactOnly`

Only t.txt file creation and deletion — no process events. Tests whether supplementary file rules fire independently (expected) vs. the main 4688 correlation (not expected).

Expected: Main alert should NOT fire. File artifact rule MAY fire.

7. Simulation Scripts

7.1 Simulate-EDRFreeze.ps1 (PowerShell)

Primary simulation script generating Event ID 4688 artifacts matching EDR-Freeze's exact command-line signatures. Includes t.txt artifact creation/deletion, pre-flight auditing checks, and timestamped logging.

Scenario	Command
TruePositive	.\Simulate-EDRFreeze.ps1 -TestMode TruePositive
VariantCmdLine	.\Simulate-EDRFreeze.ps1 -TestMode VariantCmdLine
FullChainWithArtifacts	.\Simulate-EDRFreeze.ps1 -TestMode FullChainWithArtifacts
Query1Only	.\Simulate-EDRFreeze.ps1 -TestMode Query1Only
Query2Only	.\Simulate-EDRFreeze.ps1 -TestMode Query2Only
LegitWER	.\Simulate-EDRFreeze.ps1 -TestMode LegitWER
ArtifactOnly	.\Simulate-EDRFreeze.ps1 -TestMode ArtifactOnly

7.2 validate_edr_freeze.py (Python)

Post-test validation with enhanced detection logic comparison. Runs both the current (basic) and recommended (enhanced) correlation checks side-by-side.

Mode	Description
check-logs	Query Windows Event Log and simulate both basic and enhanced correlation
parse-log	Parse PowerShell simulation log and summarize scenario outcomes
checklist	Generate printable validation checklist for manual recording

8. Validation Checklist

#	Validation Check	Expect	Pass/Fail
1	True Positive — full EDR-Freeze chain fires alert	YES	
2	Variant command lines fire alert	YES	
3	Full chain with artifacts — all signals present	YES	
4	Query 1 only (no parent) does NOT fire	NO	
5	Query 2 only (no child) does NOT fire	NO	
6	Legitimate WER crash does NOT fire	NO	
7	Artifact-only (t.txt) does NOT fire main alert	NO	
8	Hostname grouping correct	YES	
9	Modified_object populated	YES	
10	PID correlation accurate	YES	
11	Alert latency within SLA	YES	
12	/type 268310 detected in command line	YES	
13	/h /pid /tid /encfile /cancel all present	YES	
14	t.txt file artifact detected (if Sysmon enabled)	YES	

Operator: _____ Date: _____

Reviewer: _____ Date: _____

9. Detection Gap Analysis & Recommendations

9.1 Tighten Query 1 Wildcards

Replace `*type*` and `*encfile*` with checks for all five Sigma-rule flags: `/h`, `/pid`, `/encfile`, `/cancel`, `/type`. Add a literal match for `268310` (the hardcoded dump-full value) for maximum fidelity.

9.2 Tighten Query 2 Regex

The regex `.*\d+\s+\d+` is very broad. Consider anchoring to match the EDR-Freeze invocation pattern specifically: `^[\w:\\/.-]+\.exe\s+\d{3,5}\s+\d+$` (executable name followed by a 3-5 digit PID and a numeric sleep time).

9.3 Add Sysmon Enrichment

If Sysmon is deployed, correlate with EventID 10 (ProcessAccess) where GrantedAccess includes 0x0800 (PROCESS_SUSPEND_RESUME) against EDR/AV processes, and where CallTrace contains `dbgcore.dll` or `dbghelp.dll`. Also monitor EventID 11/23 for t.txt file creation and deletion patterns.

9.4 Monitor PPL Process Creation

EDR-Freeze uses PROC_THREAD_ATTRIBUTE_PROTECTION_LEVEL and CREATE_PROTECTED_PROCESS to launch WerFaultSecure as a PPL. If your EDR exposes protection level metadata, alert on unexpected processes created with PPL attributes that are not vendor-signed binaries.

9.5 Detect SeDebugPrivilege Enablement

EDR-Freeze programmatically enables SeDebugPrivilege. Detecting privilege token modifications by untrusted binaries is a useful supplementary signal. Event ID 4703 (token right adjusted) can capture this when enabled.

9.6 Agent Heartbeat Correlation

If the EDR agent process is suspended, its heartbeat and telemetry stream will pause. Correlating heartbeat gaps with WerFaultSecure process creation events provides a high-confidence compound detection signal.

9.7 Time-Proximity Constraint for PID Join

Windows recycles PIDs. The correlation join should include a time-proximity filter (events within 5–10 seconds) to prevent false joins from PID reuse on busy systems.

10. Appendix: Indicator Reference

10.1 MITRE ATT&CK Mapping

Technique	Tactic	Description
T1562.001	Defense Evasion	Impair Defenses: Disable or Modify Tools
T1003.001	Credential Access	OS Credential Dumping: LSASS Memory (follow-on)
T1134.002	Defense Evasion	Access Token Manipulation: Create Process with Token (PPL)

10.2 Command-Line Signature

```
WerFaultSecure.exe /h /pid <PID> /tid <TID> /type 268310 /encfile <handle> /cancel <handle>
```

```
EDR-Freeze.exe <TargetPID> <SleepTimeMs>
```

10.3 File Artifacts

Artifact	Location	Notes
t.txt	Temp dump file in CWD	Created before WerFaultSecure launch, deleted at cleanup
dbgcore.dll	Loaded by WerFaultSecure	Visible in Sysmon CallTrace field

10.4 Related Tools

Tool	Author	Notes
EDR-Freeze	TwoSevenOneT	Original PoC — C++, user-mode
ColdWer	0xsh3llf1r3	Cobalt Strike BOF variant with freeze + LSASS dump
CreateProcessAsPPL	TwoSevenOneT	Helper to launch processes with PPL protection

End of Document