INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-16 19:20 UTC

Recursive NTFS Junctions Create Unscannable Directories, Blind EDR and Windows Defender

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0215
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Windows (NTFS filesystem), Windows Defender, EDR products broadly, all versions supporting NTFS junction points; no specific Windows version exclusions noted
Published	2026-06-16T10:17:27
Discovery Source	Rss

Executive Summary

Varonis researchers disclosed GhostTree, a technique allowing unprivileged Windows users to create recursive NTFS junction point loops that cause antivirus and EDR scanners to fail, time out, or hang indefinitely when traversing affected directories. Because the technique exploits a legitimate Windows filesystem feature and requires only two shell commands with no elevated privileges, the barrier to adoption by threat actors is exceptionally low. Microsoft initially declined to patch the Windows Defender bypass before reversing course, a response pattern that signals systemic underinvestment in filesystem-layer defensive coverage across the industry.

Technical Analysis

GhostTree exploits NTFS junction points, a native Windows filesystem mechanism that redirects directory traversal from one location to another. The technique requires no elevated privileges: any user with write access to a target directory can create a junction. Two shell commands are sufficient to construct a self-referential loop in which directory A points to directory B, and directory B points back to directory A, producing a mathematically infinite traversal path.

When a scanner, whether Windows Defender or a third-party EDR, attempts to recursively enumerate a directory tree containing this loop, it encounters an unbounded traversal problem with no reachable exit condition. Scanners handle this differently: some time out, some hang the scanning process, and some silently fail. In each case, the parent directory and any malware staged within it becomes effectively invisible to endpoint protection. The mechanism maps to CWE-835 (Loop with Unreachable Exit Condition) and parallels classic

UNIX symbolic link following vulnerabilities (CWE-61), though it operates entirely within Windows-native junction semantics.

The MITRE ATT&CK mapping reflects the technique's operational utility across multiple stages. T1564.001 (Hidden Files and Directories) and T1564 (Hide Artifacts) capture the concealment objective. T1562.001 (Impair Defenses: Disable or Modify Tools) applies because scanner failure is the intended outcome, not a side effect. T1059.003 (Windows Command Shell) covers the two-command execution path. T1083 (File and Directory Discovery) is relevant for adversaries mapping the environment before staging payloads in the blind spot.

Microsoft's initial classification of the Defender bypass as outside its security servicing boundary is significant beyond this specific finding. It reflects a documented tendency to treat filesystem-layer scanner failures as the scanner's problem rather than an OS-level responsibility. The eventual patch, issued after Varonis public disclosure, suggests that public disclosure pressure rather than internal risk assessment drove remediation. Security teams should not treat the patch as a signal that the broader class of scanner exhaustion techniques has been addressed; junction-based and symlink-based traversal loops represent a category of evasion, not a single bug.

From a source quality standpoint, the primary analytical basis is the Varonis research (reported via BleepingComputer, T3). The Microsoft Defender for Endpoint documentation (T1) provides product context but does not address GhostTree directly. The CSA Defender zero-day research note (T3) offers adjacent context on Defender attack surface but is not the originating source for this finding. The GitLab-hosted Hive Security blog post (T3) provides supplementary analysis on trusted Windows process abuse. No CVE identifier has been publicly assigned to this finding in the available source data.

Action Checklist

1. Step 1: Assess exposure, confirm your endpoint protection stack includes Windows Defender or any EDR product performing recursive filesystem scanning on Windows NTFS volumes; virtually all Windows-based organizations are in scope regardless of EDR vendor
2. Step 2: Verify patch status, confirm Microsoft's GhostTree-related Defender patch has been applied across your Windows fleet; cross-reference with your patch management telemetry and review against NIST CM-6 (Configuration Settings) to ensure patched configurations are baselined
3. Step 3: Audit EDR scan-failure logging, query your EDR console and SIEM for scan timeout events, process hang alerts, or silent scan failures on filesystem traversal; unmonitored failures are the intended outcome of this technique; align with NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs)
4. Step 4: Test your EDR against junction loop conditions, work with your EDR vendor to confirm whether their product handles infinite junction traversal gracefully or silently fails; if vendor documentation is unavailable, request this capability validation directly; this is a control validation exercise under NIST IR-3 (Incident Response Testing)
5. Step 5: Harden directory write permissions, review which users and processes hold write access to directories outside their operational need; apply least-privilege principles per NIST CM-7 (Least Functionality) and CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts) to limit who can create junctions in sensitive paths
6. Step 6: Update threat model and hunt rules, add junction-based scan evasion to your threat register under T1562.001 and T1564; develop hunting queries for unexpected junction creation events in monitored directories; reference NIST SI-4 (System Monitoring) as the governing control for detection

coverage

7. Step 7: Brief leadership, frame this as a scanner blind-spot class of risk, not a single patched bug; the patch addresses Microsoft's implementation, not third-party EDR products that may remain vulnerable; quantify the proportion of your endpoints with unverified EDR junction-loop handling

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately if Defender scan-failure Event IDs 1008, 1009, or 5008 are observed on hosts storing PII, PHI, or regulated financial data — confirming active scanner blindness in a regulated data environment triggers breach-notification assessment obligations and requires senior IR leadership engagement within 24 hours.
Recovery Notes	After applying the Microsoft GhostTree-related Defender patch and remediating ACLs, perform a full on-demand scan of previously affected directory trees using `Start-MpScan -ScanType 3 -ScanPath ` and verify the scan completes without Event ID 1008/1009/5008 errors before returning hosts to normal operations. Run the junction enumeration query (`Get-ChildItem -Attributes ReparsePoint -Recurse`) across all remediated hosts for a minimum of 30 days post-remediation to detect any re-creation of loop structures. Monitor EDR vendor release notes for junction-loop handling updates specific to your non-Microsoft EDR products, as the Microsoft patch does not extend protection to third-party scanner implementations.
Forensic Artifacts	Microsoft-Windows-Windows Defender/Operational event log entries with Event IDs 1008 (scan failed), 1009 (scan canceled), and 5008 (scan engine failure) — these are the direct forensic signatures of GhostTree causing Defender to fail on junction traversal, exportable via `wevtutil epl` to preserve native .evtx format | NTFS reparse point enumeration output from `Get-ChildItem -Recurse -Attributes ReparsePoint | Select-Object FullName, Target` — junction points with self-referential or circular targets (where the junction target resolves back to an ancestor directory) are the on-disk artifact of a GhostTree structure | Directory creation timestamps from the MFT ($MFT) for the junction-containing directories — forensic parsing with a tool such as MFTECmd (Eric Zimmerman) will reveal the precise creation time and the SID of the user who created the junction, supporting attribution to a specific user account | Windows Security Event Log Event ID 4663 (An attempt was made to access an object) with Object Type 'File' and Access Mask 0x40 (CreateDirectory) in the path containing the junction — when object access auditing is enabled on sensitive directories, junction creation generates this event identifying the creating process and user | Sysmon Event ID 11 (FileCreate) log entries where the TargetFilename attribute resolves to a path with a reparse point attribute — these entries record the process name, PID, and user context responsible for junction creation, providing process lineage to identify whether the junction was created by a shell, script interpreter, or malware dropper

Per-Action IR Details

Step 1: Assess exposure — confirm your endpoint protection stack includes Windows Defender or any EDR product performing recursive filesystem scanning on Windows NTFS volumes; virtually all Windows-based organizations are in scope regardless of EDR vendor

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: establishing IR capability and understanding the environment before an incident occurs

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run `Get-MpComputerStatus | Select-Object AMProductVersion, RealTimeProtectionEnabled, AntivirusEnabled` via PowerShell remoting across the fleet to enumerate Defender state. Cross-reference with `wmic product get name,version` to identify installed third-party EDR agents. A 2-person team can script this across a domain using a GPO-deployed scheduled task writing results to a central share.

Evidence: This is a pre-incident scoping step that does not alter live state. No volatile capture is required before execution. Document baseline scan configurations — capture `Get-MpPreference` output (exclusion lists, scan type settings) and EDR agent version strings as a configuration snapshot for later comparison if GhostTree artifacts are discovered.

Step 2: Verify patch status — confirm Microsoft's GhostTree-related Defender patch has been applied across your Windows fleet; cross-reference with your patch management telemetry and review against NIST CM-6 (Configuration Settings) to ensure patched configurations are baselined

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: ensuring tools and systems are maintained and hardened before exploitation occurs

Controls: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Query Windows Update history with `Get-HotFix | Where-Object {$_.InstalledOn -gt (Get-Date).AddDays(-90)} | Select-Object HotFixID, InstalledOn` and cross-reference the returned KB numbers against Microsoft's published Defender update history for the GhostTree-specific fix. For offline hosts, use `Get-MpComputerStatus | Select-Object AMEngineVersion, AMProductVersion` and validate against the minimum safe version documented in Microsoft's advisory.

Evidence: Before remediating any unpatched host that may already be under active exploitation, capture volatile state: run `Get-ChildItem -Path C:\ -Recurse -Depth 3 -Attributes ReparsePoint 2>$null` to enumerate existing junction points prior to patching, as the patch does not remove attacker-created junctions already present on disk. Preserve this output; it establishes a pre-patch junction inventory for forensic comparison.

Step 3: Audit EDR scan-failure logging — query your EDR console and SIEM for scan timeout events, process hang alerts, or silent scan failures on filesystem traversal; unmonitored failures are the intended outcome of this technique; align with NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs)

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: reviewing logs and telemetry to identify indicators of adverse events

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, enable Windows Event Tracing for antimalware via `wevtutil qe Microsoft-Windows-Windows Defender/Operational /q:"*[System[EventID=1008 or EventID=1009 or EventID=5008]]" /f:text` — Event IDs 1008 (scan failed), 1009 (scan canceled), and 5008 (scan engine failure) are the specific Defender events produced when junction-loop traversal causes engine hang or timeout. Pipe results to a timestamped log file and schedule via Task Scheduler for daily review.

Evidence: Query the Microsoft-Windows-Windows Defender/Operational event log for Event IDs 1008, 1009, and 5008 before any remediation action. Capture the full event detail including the target path reported in the scan failure — the path will reference the junction-containing directory if GhostTree is active. Export with `wevtutil epl Microsoft-Windows-Windows Defender/Operational C:\IR\defender_ops_export.evtx` to preserve the original log in native format before any log rotation occurs.

Step 4: Test your EDR against junction loop conditions — work with your EDR vendor to confirm whether their product handles infinite junction traversal gracefully or silently fails; request vendor documentation on depth limits and loop detection; this is a control validation exercise under NIST IR-3 (Incident Response

Testing)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: validating that IR tools and controls function as expected under adversarial conditions

Compensating: On an isolated test VM (never production), reproduce the GhostTree condition using the two-command technique: `mkdir C:\testjunc\a` followed by `mklink /J C:\testjunc\a\b C:\testjunc\a` — this creates a self-referential junction loop requiring no elevated privileges, as documented in the Varonis GhostTree disclosure. Trigger a Defender on-demand scan of the parent directory (`Start-MpScan -ScanType 3 -ScanPath C:\testjunc`) and observe whether the scan hangs, times out with an error, or completes — recording the outcome and elapsed time. Clean up with `rmdir /s /q C:\testjunc` after testing.

Evidence: This step is performed in an isolated test environment and does not touch production systems, so no volatile pre-capture is required. Document the test results: capture `Get-MpComputerStatus` before and after the scan attempt to detect any state changes, and preserve the Defender Operational event log segment covering the test window to establish vendor-comparison baseline for post-remediation validation.

Step 5: Harden directory write permissions — review which users and processes hold write access to directories outside their operational need; apply least-privilege principles per NIST CM-7 (Least Functionality) and CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts) to limit who can create junctions in sensitive paths

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment: implementing measures to limit further damage and prevent the technique from being exercised in sensitive paths

Controls: NIST AC-6 (Least Privilege), NIST AC-3 (Access Enforcement), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Use `icacls C:\SensitivePath /inheritance:d` to remove inherited permissions, then audit current ACLs with `icacls C:\SensitivePath` and remove non-essential write and modify permissions with `icacls C:\SensitivePath /remove:g "BUILTIN\Users"`. Because GhostTree requires only CreateDirectory and CreateSymbolicLink (or junction) rights — not SeCreateSymbolicLinkPrivilege for junctions specifically — focus on removing Write and Modify ACEs from user-writable directories in high-value paths such as `%ProgramData%`, `%TEMP%`, and application staging directories.

Evidence: Before modifying any ACLs, capture a full ACL snapshot of the targeted directory tree: `icacls C:\SensitivePath /save C:\IR\acl_baseline.txt /T` preserves the pre-change state. If any junction points already exist in these paths, capture them with `Get-ChildItem -Path C:\SensitivePath -Recurse -Attributes ReparsePoint | Select-Object FullName, Target` before removal, as existing junctions are forensic evidence of prior exploitation and must be preserved for analysis.

Step 6: Update threat model and hunt rules — add junction-based scan evasion to your threat register under T1562.001 and T1564; develop hunting queries for unexpected junction creation events in monitored directories; reference NIST SI-4 (System Monitoring) as the governing control for detection coverage

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: improving detection capability and updating threat models based on lessons learned

Controls: NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation)

Compensating: Deploy a Sysmon configuration that enables Event ID 11 (FileCreate) and Event ID 23 (FileDelete) with path filters on high-value directories; junction creation is logged as a FileCreate event where the FileName attribute contains a reparse point target path. Write a Sigma rule matching Sysmon Event ID 11 where `TargetFilename` resolves to a directory within `%ProgramData%`, `C:\Windows\Temp`, or user-writable application paths AND the file attributes include `REPARSE_POINT`. Schedule a daily osquery query: `SELECT path, type FROM file WHERE path LIKE 'C:\%' AND type = 'directory' AND symlink = 1` to enumerate new junction points across monitored paths.

Evidence: Junction creation events do not alter volatile memory state, so no pre-capture is required for hunting rule deployment. However, before deploying Sysmon changes on a potentially compromised host, capture a current junction inventory via `Get-ChildItem -Path C:\ -Recurse -Depth 5 -Attributes ReparsePoint 2>$null | Select-Object FullName` as a timestamped baseline — newly appearing junctions after Sysmon deployment will be detectable by differential comparison against this snapshot.

Step 7: Brief leadership — frame this as a scanner blind-spot class of risk, not a single patched bug; the patch addresses Microsoft's implementation, not third-party EDR products that may remain vulnerable; quantify the proportion of your endpoints with unverified EDR junction-loop handling

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: communicating findings, residual risk, and recommended improvements to organizational leadership

Compensating: Produce a one-page risk summary using data from Steps 1–4: total endpoint count from asset inventory, percentage with confirmed patched Defender, percentage with EDR vendor confirmation of junction-loop handling, and the number of scan-failure events found in Step 3. For unverified EDR products, frame residual risk quantitatively — if 40% of endpoints have an EDR with no confirmed loop-detection behavior, that 40% represents a sustained scanner blind-spot independent of the Defender patch status.

Evidence: No volatile state is altered by a leadership briefing; no pre-capture is required. Attach supporting artifacts from earlier steps: the junction inventory from Step 2, scan-failure event exports from Step 3, and vendor test results from Step 4 as documented evidence of current exposure, ensuring the briefing is grounded in environment-specific findings rather than generic advisory language.

Detection Guidance

Detection for GhostTree-style abuse concentrates on three signal categories: junction creation events, scan failure telemetry, and filesystem anomalies in monitored directories.

Junction creation: Windows does not generate a default audit event for junction point creation, which is a coverage gap in most environments. Enable object access auditing on sensitive directories (NIST AU-2, AU-12) and configure Sysmon Event ID 11 (FileCreate) or equivalent EDR file-operation telemetry to capture reparse point creation. A junction created by a non-administrative user process in a temp, staging, or application data path warrants investigation. The MITRE technique T1564.001 should anchor your hunting hypothesis.

Scan failure telemetry: Query your EDR management console and SIEM for scan timeout events, incomplete scan completions, or scanning process restarts on specific directories. Correlate these failures with the directories involved. A directory that consistently fails to scan, or causes the scanner process to restart, is a behavioral indicator even without a specific IOC. Align log collection with CIS 8.2 (Collect Audit Logs) and retention with NIST AU-11 (Audit Record Retention).

Filesystem anomaly hunting: Use MITRE D3FEND D3-SFA (System File Analysis) principles to baseline directory structures in high-value paths. Alert on directories whose reported depth exceeds a configurable threshold during enumeration, or on directories that return errors during recursive listing commands. PowerShell's Get-Item with -Force and directory depth counters can surface unexpected reparse points during threat hunting sweeps.

Privilege context: Because junction creation requires only write access, not administrative rights, do not filter hunting queries to administrator accounts. Low-privileged user accounts creating reparse points in shared or writable directories are a meaningful signal. Review access control lists on writable directories as a preventive audit step per CIS 3.3 (Configure Data Access Control Lists).

EDR vendor coordination: Contact your EDR vendor directly to ask whether their scanner implements junction loop detection and at what depth limit. Absence of a clear answer is itself a risk indicator. D3FEND countermeasure D3-UAP (User Account Permissions) applies to restricting junction creation rights as a preventive layer.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`cmd.exe (mklink /j)`	cmd.exe (mklink /j) leveraged via standard user write access to create recursive NTFS junction point loops that cause EDR and Windows Defender filesystem scanners to timeout or hang, rendering staged malware invisible to endpoint protection.	HIGH
TOOL	`Pending — refer to Varonis GhostTree research for published indicators`	Varonis published the originating research; specific file hashes, directory path patterns, or proof-of-concept tool artifacts may be available in the full Varonis disclosure not fully reproduced in available source text	LOW

Framework Mappings

MITRE-ATTACK

- **T1564.001** — Hidden Files and Directories
- **T1562.001** — Disable or Modify Tools
- **T1083** — File and Directory Discovery
- **T1564** — Hide Artifacts
- **T1059.003** — Windows Command Shell

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **AT-2** — Literacy Training and Awareness

CIS-V8

- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1564.001	Hidden Files and Directories	Defense-Evasion
T1562.001	Disable or Modify Tools	Defense-Evasion
T1083	File and Directory Discovery	Discovery
T1564	Hide Artifacts	Defense-Evasion
T1059.003	Windows Command Shell	Execution

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/ghosttree-attack-abu...	T3
When Your Defender Becomes the Attacker: How Trusted Windows ...	https://hivesecurity.gitlab.io/blog/windows-trusted-process-abuse-d...	T3
Microsoft Defender for Endpoint on Windows	https://learn.microsoft.com/en-us/defender-endpoint/microsoft-defen...	T1
Both Windows Defender + EDR on endpoints, worth it? - Reddit	https://www.reddit.com/r/cybersecurity/comments/uxgk68/both_windows...	T3
Defender Triple Zero-Day: BlueHammer, RedSun, and UnDefend	https://labs.cloudsecurityalliance.org/research/csa-research-note-d...	T3

Generated 2026-06-16 19:20 UTC by TJS Security Command Center