INTELLIGENCE BRIEFING
Security Command Center

TLP:CLEAR
2026-03-29 18:36 UTC

HwAudKiller Campaign: Tax-Season Malvertising Deploys BYOVD Kernel-Mode EDR Killer via Signed Huawei Driver

THREAT CAMPAIGN | **HIGH** | CVSS 9.5

SCC Item ID	SCC-CAM-2026-0098
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	9.5
Affected Products	ConnectWise ScreenConnect (rogue installer), Microsoft Defender, Kaspersky, SentinelOne (EDR targets), Huawei HWAuidoOs2Ec.sys driver (BYOVD vector), FleetDeck Agent, NetExec, Google Ads (delivery mechanism)
Published	2026-03-24
Discovery Source	Rss

Executive Summary

An active malvertising campaign, running since January 2026, targets employees searching for tax-related software by serving fraudulent Google Ads that deliver trojanized ConnectWise ScreenConnect installers. Once installed, attackers disable endpoint security tools at the kernel level and establish persistent remote access, with post-compromise behavior consistent with ransomware staging or credential sale to access brokers. Organizations using ConnectWise ScreenConnect, particularly those downloading it via search ads, are at elevated risk of full environment compromise and potential ransomware deployment.

Technical Analysis

Campaign active since January 2026. Delivery vector: Google Ads targeting tax-season search queries, using dual commercial cloaking services to bypass ad-platform policy enforcement (MITRE T1583.008, T1036). Lure: trojanized ConnectWise ScreenConnect installer (T1195, T1036.005). Post-installation, the threat actor loads HWAuidoOs2Ec.sys, a signed but vulnerable Huawei audio driver, via Bring Your Own Vulnerable Driver (BYOVD) technique (T1068, T1562.001, CWE-693, CWE-114, CWE-269). The driver achieves kernel-mode execution to terminate EDR processes including Microsoft Defender, Kaspersky, and SentinelOne. No CVE has been assigned to this driver vulnerability as of the report date (2026-03-04); the vulnerability appears to represent a previously unknown issue in the Huawei driver ecosystem, with CVE assignment pending if formally reported to MITRE. With EDR neutralized, actors perform LSASS credential dumping (T1003.001), may inject payloads into legitimate processes for evasion (T1055), perform lateral movement and network reconnaissance via NetExec (T1021), and deploy stacked RMM tools including FleetDeck Agent for persistent access (T1219, T1543). Post-compromise behavior consistent with pre-ransomware staging or initial access brokering. At least 60 victim instances reported by initial sources; independent victim count verification pending. Attribution: Unknown; no confirmed threat group attribution at this time. Confidence: medium, primary source is The Hacker News (2026-03-04, T3); independent corroboration from CISA, NVD, or a second authoritative source has not been confirmed. MITRE techniques mapped: T1566, T1219, T1566.002, T1036, T1003.001, T1055, T1078, T1195, T1583.001, T1021, T1070.004, T1068, T1583.008, T1543, T1562.001, T1036.005.

Action Checklist

1. Step 1, Immediate: Verify any ConnectWise ScreenConnect installation sourced from search ads; compare installer hash against official ConnectWise release hashes available at https://docs.connectwise.com/ConnectWise_Control_Documentation/On-premises/Installation,_update,_and_licensing/Secure_installer_verification or from your ConnectWise account portal; isolate any host where the installer cannot be verified.
2. Step 2, Immediate: Block HWAuidoOs2Ec.sys from loading via Windows Defender Application Control (WDAC) or equivalent driver blocklist policy; verify whether the driver appears in your environment using EDR telemetry or 'sc query' and kernel driver enumeration.
3. Step 3, Detection: Hunt for FleetDeck Agent installations not authorized by IT; audit all active RMM tools on endpoints for unauthorized or stacked instances; review NetExec execution artifacts in process logs.
4. Step 4, Detection: Search LSASS access events in EDR and Windows Security Event Log (Event ID 10 in Sysmon, Event ID 4656/4663 for LSASS handle requests); treat any LSASS access from an unrecognized process as a confirmed indicator requiring immediate escalation.
5. Step 5, Assessment: Inventory all ConnectWise ScreenConnect deployments; confirm installation sources for all instances; cross-reference against official ConnectWise release hashes; identify any hosts where EDR was restarted unexpectedly or shows a gap in telemetry coverage.
6. Step 6, Communication: Notify security leadership and IT asset owners of the BYOVD driver risk; if pre-ransomware staging is suspected on any host, escalate to incident response and consider engaging external IR support before broader notification.
7. Step 7, Long-term: Add HWAuidoOs2Ec.sys to your organization's driver blocklist; review and tighten WDAC or AppLocker policy to prevent unsigned or newly identified vulnerable drivers; evaluate whether your vulnerable driver blocklist is current against the Microsoft Recommended Driver Block Rules and supplement with threat-intelligence-sourced additions.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to senior leadership and external IR firm immediately if any host shows LSASS access attempts, EDR kill events, lateral movement to file servers, or disabled backup systems; otherwise escalate to CISO if more than 5 hosts affected or any trojanized ScreenConnect installer confirmed.

<table>
<tr><td><b>Recovery Notes</b></td><td>After eradication: (1) Rebuild or restore affected hosts from clean backups dated before January 2026 (campaign start); verify backup integrity and isolation during recovery. (2) Re-baseline EDR/antivirus deployment and confirm telemetry flow for 72 hours post-recovery. (3) Conduct threat hunt on high-value targets (domain controllers, file servers) for lateral movement artifacts (credential dumping, unusual logons, file access) using Windows Event Log 4769/4770 (Kerberos), 4624/4625 (logon success/failure), and 5140 (network share access).</td></tr>
<tr><td><b>Forensic Artifacts</b></td><td>Windows Event Log Security 4688 (process creation, 30+ days historical for NetExec/FleetDeck/ScreenConnect execution) | Windows Event Log System 7045 (service installation, HWAuidoOs2Ec.sys driver load events) | Sysmon Event Log 10 (ProcessAccess to LSASS, 30+ days) | Windows Event Log Security 4656/4663 (LSASS handle requests and file object access) | File system artifacts: %ProgramFiles%/ConnectWise/ScreenConnect/, %SystemRoot%/System32/drivers/HWAuid*.sys, %ProgramFiles%/*FleetDeck*, %ProgramFiles%/*NetExec*; capture file hashes, creation/modification timestamps, and digital signatures | Registry hives: HKLM\Software\ConnectWise, HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall, HKLM\System\CurrentControlSet\Services (driver configuration) | Browser download history and cache: %USERPROFILE%/AppData/Local/Google/Chrome/User Data/Default/History (malvertising link tracking) | Network connections: netstat -ano output and Get-NetTCPConnection (identify command-and-control callbacks from ScreenConnect/RMM/NetExec) | LSASS memory dump and loaded module list (for credential dumping and further malware analysis)</td></tr>
</table>

Per-Action IR Details

Step 1 — Immediate: Verify any ConnectWise ScreenConnect installation sourced from search ads; compare installer hash against ConnectWise's official distribution hashes; isolate any host where the installer cannot be verified.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (preparation, malware prevention)

Controls: NIST IR-4(1) — Automated incident handling, CIS 2.4 — Software inventory, CIS 7.1 — Address unauthorized software

Compensating: Without EDR: (1) Export all installed ScreenConnect versions via PowerShell: Get-WmiObject Win32_Product | Where-Object {$_.Name -like '*ScreenConnect*'} | Select-Object Name, Version, InstallDate; (2) Download official ScreenConnect release hashes from ConnectWise release notes; (3) Compute SHA-256 of each installer in %ProgramFiles% using: certutil -hashfile '' SHA256; (4) Compare against official list; (5) Isolate non-matching hosts to VLAN without internet access pending verification.

Evidence: Before isolation: capture (a) file system hash of ScreenConnect installation directory (%ProgramFiles%/ConnectWise/ScreenConnect/); (b) registry keys HKLM\Software\ConnectWise\ScreenConnect and HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall\*ScreenConnect* (export via reg export); (c) Process execution history via Windows Event Log 4688 (filtered for ScreenConnect exe launch); (d) file metadata and timestamps via Get-ChildItem -Recurse | Select-Object FullName, CreationTime, LastWriteTime; (e) browser download history from %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\History (if Chrome present).

Step 2 — Immediate: Block HWAuidoOs2Ec.sys from loading via Windows Defender Application Control (WDAC) or equivalent driver blocklist policy; verify whether the driver appears in your environment using EDR telemetry or 'sc query' and kernel driver enumeration.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.2 (containment, malware eradication)

Controls: NIST SI-7(15) — Information System Monitoring (driver integrity), CIS 6.2 — Application and Kernel Driver Control, NIST IR-4(2) — Malware incident handling

Compensating: Without WDAC or enterprise driver blocklist: (1) Enumerate loaded drivers via: driverquery /v or Get-WindowsDriver -Online | Where-Object {$_.ProviderName -like '*Huawei*'} or $_.FileName -like '*HWAuid*'}; (2) Query Services Control Manager for Huawei-signed drivers: sc query | findstr /i 'hwaud'; (3) Check kernel memory via WinDbg or OSChaos.exe (free tool) for HWAuidoOs2Ec.sys in loaded module list; (4) Disable via: sc stop HWAudioDecode (where HWAudioDecode is the service name) and sc config HWAudioDecode start= disabled; (5) Rename driver file in %SystemRoot%\System32\drivers\ to .sys.blocked and reboot to prevent load on next startup.

Evidence: Before blocking: capture (a) full kernel module list via Get-Process | Select-Object -ExpandProperty Modules (memory snapshot); (b) driver signing information via sigcheck.exe (Sysinternals tool); (c) driver file properties: Get-ChildItem -Path 'C:\Windows\System32\drivers\HWAuid*' -Force | Select-Object FullName, CreationTime, LastWriteTime; (d) Service Control Manager configuration: reg export HKLM\System\CurrentControlSet\Services ; (e) Windows Event Log 7045 (service installation events, 48-72 hours back) to identify when driver was installed.

Step 3 — Detection: Hunt for FleetDeck Agent installations not authorized by IT; audit all active RMM tools on endpoints for unauthorized or stacked instances; review NetExec execution artifacts in process logs.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.1 (detection and analysis, malware detection)

Controls: NIST SI-4(4) — System Monitoring (unauthorized software), CIS 7.1 — Address unauthorized software, CIS 7.9 — Deny or restrict remote access

Compensating: Without EDR: (1) Query all installed software via: Get-WmiObject Win32_Product | Select-Object Name, InstallDate, Version and cross-reference against IT-approved RMM list (create baseline); (2) Hunt for FleetDeck: Get-ChildItem -Path 'C:\Program Files*' -Recurse -Filter '*FleetDeck*' -ErrorAction SilentlyContinue; (3) Check running processes: Get-Process | Where-Object {$_.ProcessName -like '*FleetDeck*' -or $_.ProcessName -like '*NetExec*'}; (4) Audit Scheduled Tasks for suspicious RMM startup: Get-ScheduledTask | Where-Object {$_.TaskName -like '*Fleet*' -or $_.TaskName -like '*remote*'} | Get-ScheduledTaskInfo; (5) Search for NetExec in running processes and command-line history via Get-WinEvent -FilterHashtable @{LogName='Security'; ID=4688} | Where-Object {$_.Message -like '*NetExec*'}.

Evidence: Before remediation: capture (a) full Windows Event Log 4688 (process creation) for 7 days: wevtutil export-log Security .evtx; (b) registry hive for uninstall keys: reg export HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall ; (c) list of running processes and their command lines: Get-CimInstance Win32_Process | Select-Object ProcessId, Name, CommandLine > processes.csv; (d) Scheduled Tasks: Get-ScheduledTask | Get-ScheduledTaskInfo | Export-Csv tasks.csv; (e) network connections from RMM processes: Get-NetTCPConnection -State Established | Select-Object LocalAddress, LocalPort, RemoteAddress, RemotePort, OwningProcess | ConvertTo-Csv > netconns.csv.

Step 4 — Detection: Search LSASS access events in EDR and Windows Security Event Log (Event ID 10 in Sysmon, Event ID 4656/4663 for LSASS handle requests); treat any LSASS access from an unrecognized process as a confirmed indicator requiring immediate escalation.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.1 (detection and analysis, indicator identification)

Controls: NIST AC-2(7) — Access Control (LSASS credential access prevention), CIS 6.1 — Audit detailed process monitoring, CIS 5.3 — Configure logging for authentication

Compensating: Without EDR: (1) Enable Sysmon (free tool by Sysinternals) with focus on Event 10 (ProcessAccess) and deploy config file targeting LSASS (download SwiftOnSecurity Sysmon config); (2) Export Sysmon logs: wevtutil export-log 'Microsoft-Windows-Sysmon/Operational' sysmon.evtx and query: Get-WinEvent -FilterHashtable @{Path='sysmon.evtx'; ID=10} | Where-Object {$_.Message -like '*LSASS*'}; (3) Alternatively, enable Windows Security Event Log 4656/4663 via Group Policy: gpedit.msc > Computer Configuration > Windows Settings > Security Settings > Advanced Audit Policy Configuration > Object Access > Audit File System; (4) Query: Get-WinEvent -FilterHashtable @{LogName='Security'; ID=4656} | Where-Object {$_.Message -like '*lsass.exe*'} | Select-Object

TimeCreated, Message; (5) Correlate source process (SubjectProcessId in event) with suspicious process list from Step 3.

Evidence: Before investigation: capture (a) Sysmon Event Log 10 (ProcessAccess) for 30 days minimum: wevtutil export-log 'Microsoft-Windows-Sysmon/Operational' .evtx; (b) Windows Security Event Logs 4656/4663 (file object access): wevtutil export-log Security .evtx; (c) LSASS memory dump (use volatility or local memory acquisition): rundll32 C:\Windows\System32\comsvcs.dll MiniDump .dmp (note PID is 4 or query Get-Process lsass | Select-Object Id); (d) loaded DLLs in LSASS: Get-Process lsass | Select-Object -ExpandProperty Modules | Export-Csv lsass_modules.csv; (e) registry hives for credential storage artifacts: reg save HKLM\Security security.hive and reg save HKLM\SAM sam.hive.

Step 5 — Assessment: Inventory all ConnectWise ScreenConnect deployments; confirm installation sources for all instances; cross-reference against official ConnectWise release hashes; identify any hosts where EDR was restarted unexpectedly or shows a gap in telemetry coverage.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.1 (detection and analysis, scope identification)

Controls: NIST IR-4(6) — Incident handling (insider threat and supply chain), CIS 2.4 — Address unauthorized software, CIS 13.5 — Maintain an up-to-date inventory of services

Compensating: Without EDR: (1) Query all endpoints via Active Directory or SCCM for installed ScreenConnect: Get-ADComputer -Filter * | ForEach-Object {Get-WmiObject -ComputerName $_.Name -Query "select * from Win32_Product where Name like '%ScreenConnect%'" -ErrorAction SilentlyContinue}; (2) Create deployment inventory with: hostname, install date, version, installation path, and installer source (from Control Panel uninstall metadata); (3) Check EDR/antivirus service restart logs: Get-WinEvent -FilterHashtable @{LogName='System'; ID=7040} (service state change) | Where-Object {$_.Message -like '*Defender*' -or $_.Message -like '*Kaspersky*' -or $_.Message -like '*SentinelOne*'}; (4) Identify telemetry gaps by checking EDR agent status and last report time; (5) Cross-reference hash inventory from Step 1 against official ConnectWise release notes via https://docs.connectwise.com/ConnectWise_Control_Documentation/On-Premise/Release_notes (verify against 3+ recent releases).

Evidence: Before assessment: capture (a) software inventory report from SCCM or manual scan (WMI query output above); (b) installation event logs from all endpoints: Get-WinEvent -FilterHashtable @{LogName='System'; ID=7045} (Service Control Manager service installation, 90-day history); (c) uninstall registry entries: Export from HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall on all hosts; (d) EDR agent service status and version: Get-Service -Name '*Defender*' -ErrorAction SilentlyContinue | Select-Object Name, Status, DisplayName and Get-MpComputerStatus; (e) EDR telemetry gaps (no events in 24+ hours): correlate with EDR agent restart logs and System Event Log 7034 (service crashed) and 7045 (service installed/modified).

Step 6 — Communication: Notify security leadership and IT asset owners of the BYOVD driver risk; if pre-ransomware staging is suspected on any host, escalate to incident response and consider engaging external IR support before broader notification.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.2.2 (incident response notification and communication)

Controls: NIST IR-2(2) — Incident reporting (senior management notification), CIS 17.1 — Communicate incident response plan

Compensating: Without formal IR program: (1) Identify escalation contacts: Chief Information Security Officer (CISO), VP of IT, IT Security Manager, and legal/compliance lead; (2) Prepare concise escalation brief with: attack vector (malvertising + trojanized installer), indicators present on this environment (HWAuidoOs2Ec.sys confirmed or suspected), number of affected hosts, and risk to ransomware (EDR kill capability); (3) If ransomware staging suspected (indicators: lateral movement to file servers, disabled backups, mass file enumeration, persistence mechanisms), immediately engage: (a) external Incident Response firm (IR retainer or on-demand), (b) FBI Cyber Division (via IC3.gov or local FBI field office), (c) cyber insurance carrier (notify within policy timeframe); (4) If only driver + RMM confirmed (no staging), notify CISO and proceed with containment; broader endpoint notification can wait until Step 7 mitigation is deployed.

Evidence: Before escalation: compile (a) confirmed indicator count: number of hosts with HWAuidoOs2Ec.sys, unauthorized RMM instances, EDR kill events, LSASS access attempts (from Steps 2–4); (b) timeline: installation date of trojanized ScreenConnect, EDR restart events, first RMM execution; (c) affected user count and departments; (d) backup status check: confirm backup system availability and test restore capability (may be compromised); (e) network segmentation status: whether affected hosts have access to high-value targets (domain controller, file servers, backup systems).

Step 7 — Long-term: Add HWAuidoOs2Ec.sys to your organization's driver blocklist; review and tighten WDAC or AppLocker policy to prevent unsigned or newly identified vulnerable drivers; evaluate whether your vulnerable driver blocklist is current against the Microsoft Recommended Driver Block Rules and supplement with threat-intelligence-sourced additions.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.4 (post-incident activity, process improvements)

Controls: NIST SI-7(15) — Information System Monitoring (driver integrity), CIS 6.2 — Application and Kernel Driver Control, NIST CA-7(1) — Continuous Monitoring

Compensating: Without enterprise device management: (1) Download Microsoft Recommended Driver Block Rules from Microsoft: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/microsoft-recommended-driver-block-rules (verify URL and download latest version); (2) Create WDAC policy via: New-CIPolicy -DriverFilesPath 'C:\path\to\blocked\drivers' -Level FileName -Fallback Hash; deploy via Group Policy or direct binary (convert to WDAC XML); (3) If WDAC not available, use AppLocker fallback: Create AppLocker rule to block driver execution by file path or hash (gpedit.msc > Computer Configuration > Windows Settings > Security Settings > Application Control Policies > AppLocker > Rules for DLLs and Executables); (4) Integrate threat intel: subscribe to CISA alerts (https://www.cisa.gov/news-and-updates) and KnownDrivers.com curated blocklist; (5) Update quarterly: monthly review of new BYOVD exploits and add to blocklist; audit enforcement logs monthly via Get-AppLockerFileInformation.

Evidence: Post-recovery (ongoing): maintain (a) driver blocklist version and deployment date; (b) WDAC policy version and audit logs (Microsoft-Windows-CodeIntegrity/Operational); (c) monthly AppLocker enforcement reports (Group Policy event logs); (d) threat intelligence feed subscription records (dates of alerts processed); (e) policy test results (verify blocked driver cannot load on non-production test host).

Detection Guidance

Driver load events: Monitor for HWAuidoOs2Ec.sys in Sysmon Event ID 6 (driver loaded) and Windows Event ID 7045 (new service installed). Any appearance of this driver on a production host is a high-confidence indicator of BYOVD activity. EDR process termination: Look for unexpected termination or service-stop events targeting security tool processes (MsMpEng.exe, SentinelAgent.exe, avp.exe). A gap in EDR telemetry on a host that was previously reporting is itself a detection signal. LSASS access: Sysmon Event ID 10 targeting lsass.exe from non-system processes; Windows Security Event IDs 4656 and 4663 for LSASS object access; review for mimikatz-style access rights (0x1010, 0x1038). RMM tool presence: Audit installed services and running processes for FleetDeck Agent and ConnectWise ScreenConnect; flag any RMM tool not present in your authorized software inventory. Network reconnaissance: Hunt for NetExec (nxc.exe; successor to legacy CrackMapExec) execution in process creation logs (Sysmon Event ID 1, Windows Event ID 4688); correlate with SMB enumeration traffic (Event ID 4624 type 3 lateral logons). Ad-delivered installer lure: If proxy or DNS logs are available, look for download events from domains mimicking ConnectWise branding that are not connectwise.com or screenconnect.com. Behavioral chain: The sequence of (1) new ScreenConnect install, (2) driver load of HWAuidoOs2Ec.sys, (3) EDR process termination, (4) LSASS access is a high-confidence kill-chain indicator; any two of these four in sequence on the same host warrants immediate investigation.

Indicators of Compromise

Type	Value	Context	Confidence
DRIVER	`HWAuidoOs2Ec.sys`	Signed but vulnerable Huawei audio driver used as BYOVD vector to achieve kernel-mode execution and terminate EDR processes. Newly identified; not previously documented in vulnerable driver lists as of 2026-03-04.	MEDIUM
PROCESS	`FleetDeck Agent`	RMM tool deployed post-EDR neutralization for persistent access. Presence outside authorized IT deployments is a high-suspicion indicator in this campaign context.	MEDIUM
PROCESS	`nxc.exe / crackmapexec.exe (NetExec)`	Used for network reconnaissance and lateral movement following credential dumping. Execution in non-pentest environments is anomalous.	MEDIUM
SOFTWARE	`Trojanized ConnectWise ScreenConnect installer`	Delivered via fraudulent Google Ads targeting tax-season search queries. Specific installer hashes not published in source as of report date; verify all ScreenConnect installs against official ConnectWise distribution hashes.	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1566** — Phishing
- **T1219** — Remote Access Tools
- **T1566.002** — Spearphishing Link
- **T1036** — Masquerading
- **T1003.001** — LSASS Memory
- **T1055** — Process Injection
- **T1078** — Valid Accounts
- **T1195** — Supply Chain Compromise
- **T1583.001** — Domains
- **T1021** — Remote Services
- **T1070.004** — File Deletion
- **T1068** — Exploitation for Privilege Escalation
- **T1583.008** — Malvertising
- **T1543** — Create or Modify System Process

- **T1562.001** — Disable or Modify Tools
- **T1036.005** — Match Legitimate Resource Name or Location

NIST-800-53R5

- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection
- **AC-6** — Least Privilege
- **AC-2** — Account Management
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **SA-9** — External System Services
- **SR-2** — Supply Chain Risk Management Plan
- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **AC-17** — Remote Access
- **AC-3** — Access Enforcement
- **CM-7** — Least Functionality
- **SI-2** — Flaw Remediation
- **CP-9** — System Backup
- **IR-4** — Incident Handling

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

CIS-V8

- **5.4**
- **6.8**
- **6.3** — Require MFA for Externally-Exposed Applications
- **8.2** — Collect Audit Logs

NIST-CSF-2

- **RS.MI-01** — Incidents are contained
- **DE.CM-01** — Networks and network services are monitored

HIPAA-SECURITY

- **164.308(a)(7)(ii)(A)** — Data Backup Plan
- **164.312(d)** — Person or Entity Authentication

ISO-27001-2022

- **A.5.29** — Information security during disruption
- **A.8.8** — Management of technical vulnerabilities

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1566	Phishing	Initial-Access
T1219	Remote Access Tools	Command-And-Control
T1566.002	Spearphishing Link	Initial-Access
T1036	Masquerading	Defense-Evasion
T1003.001	LSASS Memory	Credential-Access
T1055	Process Injection	Defense-Evasion
T1078	Valid Accounts	Defense-Evasion
T1195	Supply Chain Compromise	Initial-Access
T1583.001	Domains	Resource-Development
T1021	Remote Services	Lateral-Movement
T1070.004	File Deletion	Defense-Evasion
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1583.008	Malvertising	Resource-Development
T1543	Create or Modify System Process	Persistence
T1562.001	Disable or Modify Tools	Defense-Evasion
T1036.005	Match Legitimate Resource Name or Location	Defense-Evasion

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/03/tax-search-ads-deliver-screenconn...	T3

Source	URL	Tier
CISA warns of active exploitation of ConnectWise ScreenConnect ...	https://www.linkedin.com/posts/v2-systems-inc-_cisa-warns-of-connec...	T3
ConnectWise ScreenConnect vulnerability raises concerns of ...	https://www.theinsurer.com/cyber-risk/news/connectwise-screenconnec...	T3
Critical Flaws Found in ConnectWise ScreenConnect Software	https://social.cyware.com/news/critical-flaws-found-in-connectwise-...	T3
Latest Connectwise Vulnerabilities - Feedly	https://feedly.com/cve/vendors/connectwise	T3

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-03-29 18:36 UTC by TJS Security Command Center