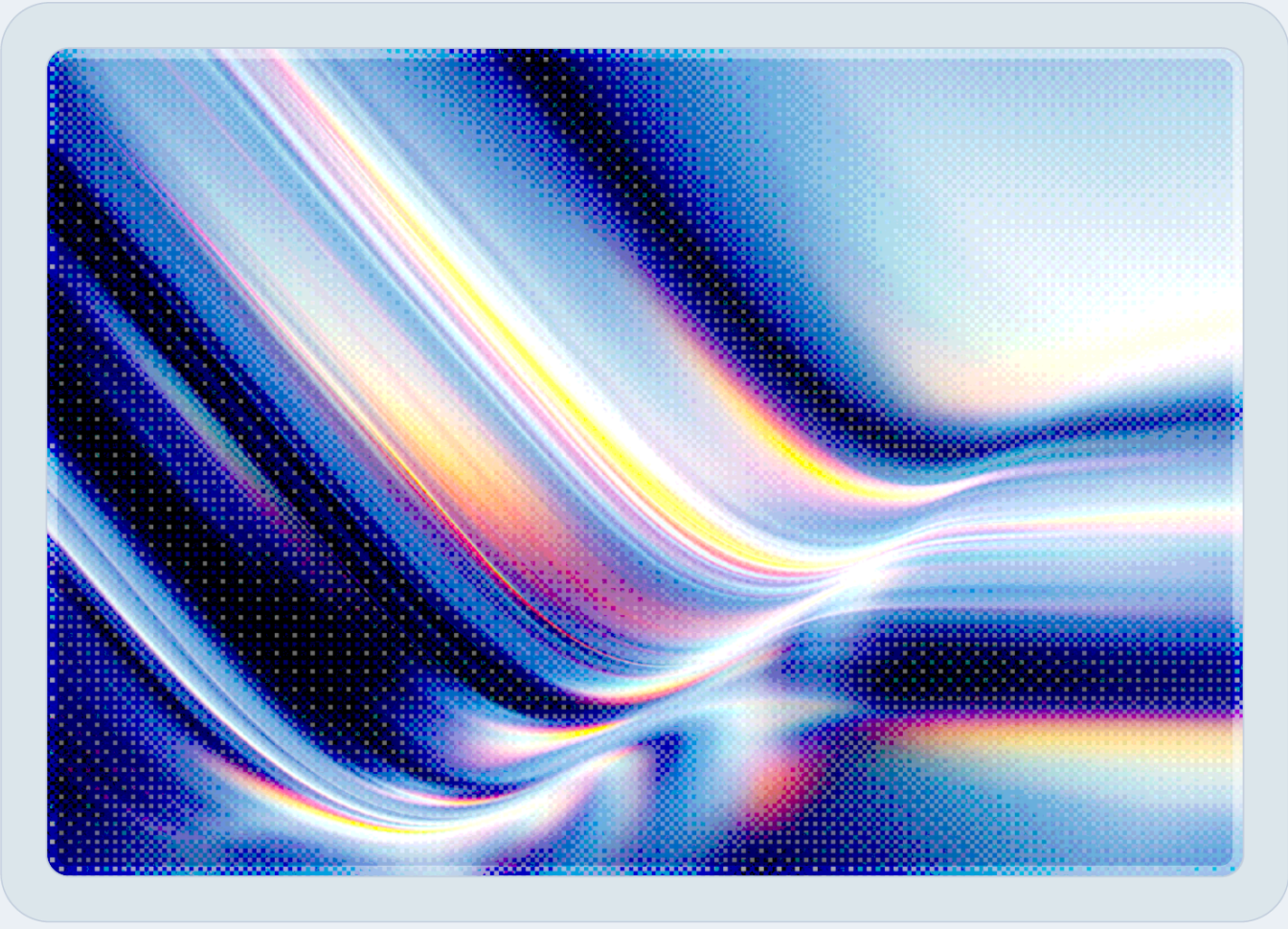
CSAI Foundation | Cloud Security Alliance

SCKit and Rogue ScreenConnect Clients: MSP Remote Support as Attack Path

Security Implications and Guidance for MSPs and Their Customers

2026-11-13

AI-assisted Rapid Research



CSAI

cloud security alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

On November 11, 2026, an unidentified operator used ConnectWise ScreenConnect's Run Command feature to push attacker-controlled ScreenConnect clients and a toolkit called SCKit to at least 288 endpoints across 48 organizations, with every affected host reached within roughly 90 minutes [1][2]. Huntress attributes initial access to RMM+, a third-party ScreenConnect extension whose versions 1.0.31 and 1.0.32 accepted a blank shared access key when administrators had never set one, allowing commands to be issued without ScreenConnect login, role checks, or multi-factor authentication [1][2]. ConnectWise had not independently confirmed the initial access vector as of Huntress's latest update, and has stated that the presence of the affected extensions does not by itself indicate compromise [1][2]. The affected environments were running current ScreenConnect releases, which suggests that patching the core product would not have prevented this intrusion [1].

SCKit is designed to go beyond persistence. Its RelayTrap component replaces the legitimate ScreenConnect guest process and waits for a technician to connect, then attempts to steal the technician's panel session cookie and push a payload to every session the technician can reach through the Shared Toolbox [1][2]. Huntress and Blackpoint Cyber report that this technician-to-fleet propagation has not succeeded in any observed case, and that the file-transfer exploit SCKit relies on (CVE-2026-84869) is blocked by ScreenConnect client 26.6.5 and later [1][2]. The design nonetheless shows a deliberate attempt to turn one MSP's support relationship into a path across all of its customers.

The immediate guidance runs counter to normal practice: organizations should not investigate suspect endpoints by connecting to them over ScreenConnect, because an interactive session is exactly the trigger the toolkit waits for [1][2]. MSPs and their customers should disable RMM+ and related extensions, hunt for second ScreenConnect clients and the RelayTrap and TrapWatch artifacts, and revoke technician web sessions from clean devices.

Background

ScreenConnect, owned by ConnectWise since 2015, is a remote support and access tool commonly used by managed service providers (MSPs) and internal IT help desks [3][4]. An MSP typically runs a single ScreenConnect instance, either cloud-hosted by ConnectWise or on-premises, from which its technicians reach endpoints belonging to many customer organizations. That architecture is what makes the product

valuable, and it is also what makes it a recurring target. A single console session can carry SYSTEM-level command execution across hundreds or thousands of machines that belong to unrelated businesses, and the remote support traffic it generates is expected, trusted, and often subject to less scrutiny from endpoint security tools [4].

The abuse of remote monitoring and management (RMM) tools is not new. In January 2023, CISA, NSA, and MS-ISAC warned that threat actors were using legitimate RMM software, ScreenConnect among them, as backdoors and noted that RMM use generally does not trigger antivirus defenses [4]. In February 2024, a ScreenConnect authentication bypass (CVE-2024-1709, CVSS 10.0) allowed attackers with network access to create administrator accounts on on-premises servers and was added to CISA's Known Exploited Vulnerabilities catalog [5]. Earlier joint guidance from the Five Eyes agencies specifically warned that compromising an MSP gives an adversary a route into the MSP's customers, and urged both parties to treat that relationship as a shared risk [6].

The November activity follows a closely related campaign earlier in the autumn. In early September 2026, Huntress described rogue, modified ScreenConnect clients that pushed a four-stage VBScript chain to each new machine they connected to, spreading in a worm-like fashion [7]. ConnectWise published an advisory on September 3 recommending that partners disable technician file transfers, and by September 11 had fixed, in client version 26.6.5, a file-transfer flaw now tracked as CVE-2026-84869, which allowed files to be transferred and executed in an active session without authorization or host confirmation [7]. Huntress later characterized the vulnerability as critical with a CVSS score of 9.9 [1]. SCKit appears to build on the same idea of using the guest side of a remote support session to reach the technician, although public reporting has not established a link between the two operations' actors.

Security Analysis

Initial Access Through a Third-Party Extension

The central finding of the Huntress investigation is that the attackers, by Huntress's assessment, did not need to defeat ScreenConnect's own authentication at all. RMM+ is an extension that lets external management tools issue commands into ScreenConnect sessions without a technician using the console, and it authenticates those requests with a shared access key rather than through ScreenConnect user accounts, roles, and MFA [1][2]. In versions 1.0.31 and 1.0.32 the `RMMPlus.AccessKey` setting defaulted to empty, and the extension accepted an empty key as valid. Huntress observed commands reaching endpoints across several instances within minutes with no corresponding technician logins, and found log entries tying those commands to RMM+ [1]. Most of the RMM+ installations Huntress examined appear to have been legitimate deployments added for a specific integration project and then left in

place [1]. This attribution remains Huntress's assessment: ConnectWise had not yet identified an initial access vector as of Huntress's latest update, and has cautioned that the presence of the extensions "does not mean those environments were compromised" [1][2].

This is a familiar pattern in a new location. The extension created a second, parallel authorization path into the most privileged function of the platform, and that path inherited none of the controls the operator had configured for the primary one. Because the observed environments ran current ScreenConnect builds (26.6.6.9747 and 26.7.6.9783), routine patch hygiene on the core product would not, by itself, have prevented this access path [1]. ConnectWise responded on November 11 by disabling five potentially related extensions (Labtech Plus, RMM Plus, RMM+, RMM Plus Password Link, and RMM+ Password Link) across more than 400 partner environments as a precaution and later extended that action to all hosted environments [1][2]. The company stated that it had "found no evidence of a compromise of ConnectWise infrastructure" and cautioned that "disabling extensions and resetting credentials alone may not remove unauthorized software" [3].

Deployment and Persistence

With command access in hand, the operator used Run Command to execute a script as SYSTEM on many endpoints simultaneously. The script checked for an existing ScreenConnect client, silently installed an attacker-controlled ScreenConnect installer, and dropped a base64-encoded batch file that retrieved a decoder and a host-specific encrypted copy of SCKit [1]. The kit consists of 16 PowerShell, batch, and configuration components, and the delivery server releases the decryption key only when the request includes the target machine's UUID, which complicates retrospective analysis of captured samples [1][2]. Huntress notes that scripts are rebuilt with different values per host, making file hashes weak indicators [1].

The persistence design is aimed specifically at the remote support channel. The following table summarizes the principal components described by Huntress and BleepingComputer.

Component	Function	Defender relevance
Rogue ScreenConnect clients	Two attacker-controlled Access clients connected to attacker relays; legitimate service stopped and set to manual	Access persists after the MSP's session ends and after extension removal

Component	Function	Defender relevance
RelayTrap (`C:\ProgramData\RelayTrap\`)	Locally compiled C# replacement for the guest process that reuses legitimate client libraries so the host still appears normal on the MSP relay	Endpoint looks healthy in the console while under attacker control
TrapWatch scheduled task	Runs as SYSTEM every 10 seconds, stops legitimate clients, and takes their connection slot	Reasserts control after cleanup attempts
`install.ps1`	Adds `RelayTrapLiteAlert` Run key and SRP/AppLocker rules that block files pushed through ScreenConnect temp folders	Blocks the MSP from pushing remediation tools over ScreenConnect
`Installer.ps1`	Adds Microsoft Defender exclusions and installs the second rogue client	Weakens endpoint protection
`Lite-Alert.ps1`	Logs ScreenConnect file-run events with on-screen alerts disabled	Lets the operator see when a technician acts on the host

Sources: Huntress [1]; BleepingComputer [2].

The AppLocker rules deserve particular attention. By blocking files delivered through ScreenConnect's own transfer folders, the toolkit attempts to prevent the MSP from using its normal tool to remediate the endpoint. Huntress also reports that hosts became reinfected after network isolation was lifted, which indicates that isolation contains the threat but does not remove it [1].

The Technician Pivot

SCKit's most far-reaching capability targets the person on the other end of the connection. When a technician connects to a compromised host, RelayTrap initiates a file transfer roughly 2.5 seconds later in an attempt to deliver a batch file to the technician's workstation by exploiting CVE-2026-84869 [1][2]. If that succeeded, the `Panel-Lateral.ps1` component would read browser cookie databases on the technician's machine, extract the ScreenConnect `.aspxauth` session cookie, and replay it against the MSP's panel to enumerate every reachable session. A companion uploader would then place a script in the panel's Shared Toolbox, execute it as SYSTEM on all reachable sessions, and delete the toolbox item about 60 seconds later [1]. A replayed session cookie would typically bypass both credentials and MFA, so this chain would let a single infected customer endpoint compromise every other customer the technician could reach.

Public evidence indicates that this chain has not worked in practice. Huntress reports that the console takeover attempt failed in every recovered host log it had examined as of its latest update, and Blackpoint Cyber reproduced the exploit against a current client and found that "the Guest side queued the transfer, but the current client never requested or ran the payload" [1][2]. That outcome depends on technicians running patched client software, however, and organizations whose technician clients predate the September 26.6.5 update may not have that protection, although no successful exploitation has been publicly reported. Huntress also notes that the RelayTrap source appears to have been generated with a large language model, based on extensive comments explaining functionality [1][3]. CSA considers this an indication, not a finding, that AI-assisted development may be lowering the effort required to build tooling tailored to a specific remote support product. No threat actor has been publicly named, and no ransomware deployment has been publicly tied to this wave as of this writing [3].

Why the MSP Model Amplifies the Impact

The incident illustrates a structural property of MSP remote support that individual customers rarely control. In many MSP arrangements, customer organizations do not see the MSP's ScreenConnect configuration, the extensions installed on it, or the audit logs of commands run against their endpoints. The 48 organizations Huntress counted are only those protected by its own product, so the true scope across the ScreenConnect partner base may be larger [1]. The trust relationship also runs in both directions: customers trust the MSP's tooling, and technicians trust the endpoints they connect to. SCKit attacks the second assumption, which receives less attention in public MSP security guidance than the first. Any endpoint a technician touches becomes, in effect, a potential source of input to the technician's workstation and session, and therefore to every other tenant on the same panel.

Recommendations

Immediate Actions

MSPs operating ScreenConnect should confirm that RMM+ and the related extensions named by ConnectWise are disabled on every instance, including on-premises servers, since ConnectWise's precautionary action covered hosted environments and cannot reach on-premises installations [1][2]. They should then review Run Command and Shared Toolbox history and instance audit logs for unexpected mass execution beginning November 11, 2026, and for commands that lack a corresponding technician login [1]. Every managed endpoint should be checked for a second ScreenConnect client, the `TrapWatch` scheduled task, the `C:\ProgramData\RelayTrap\` directory, `sckit-*` files under `%TEMP%`, `C:\Windows\Temp\sc.msi`, and the `RelayTrapLiteAlert` Run value [1]. Huntress has published network indicators and rogue client identifiers that should be loaded into detection tooling, with the caveat that file hashes vary per host [1].

Investigation and remediation of suspect hosts should be performed through EDR live response, WinRM, or another out-of-band channel. Technicians should not open interactive ScreenConnect sessions to potentially infected endpoints, and any technician workstation that has already connected to a suspect host should be treated as potentially exposed [1][2]. From a clean device, MSPs should terminate ScreenConnect web sessions, rotate credentials, and revoke browser and identity-provider sessions for affected technicians, since the toolkit is built to steal session tokens [1]. All ScreenConnect clients, including technician host clients, should be at version 26.6.5 or later to close CVE-2026-84869 [7]. Customers of MSPs should ask their providers directly whether these steps have been completed and request evidence.

Short-Term Mitigations

Over the following weeks, MSPs should inventory every extension installed on their remote support platforms and remove any that are not actively required. For those that remain, operators should verify that each one routes through the platform's native authentication, or that any separate credential it uses is strong, unique, and rotated. Extensions should be treated as privileged code with their own change management, not as convenience add-ons. Role permissions for Run Command, Shared Toolbox, and file transfer should be restricted to the smallest set of technicians and session groups that require them, following the role-editing approach ConnectWise documented for the September file-transfer issue [7].

© 2026 Cloud Security Alliance. AI-assisted Rapid Research.

Customer organizations and MSPs together should also apply the application control guidance from CISA's 2023 RMM advisory: allowlist the specific remote access tools and instance identifiers that are authorized, and alert on any RMM client connecting to an unapproved relay [4]. A second ScreenConnect client pointing at an unknown server is a strong indicator in both campaigns, though co-managed and MSP-transition environments may legitimately carry more than one client and should be baselined. Finally, technician workstations should be treated as privileged access workstations, isolated from general browsing and email, with browser session storage protected and with short session lifetimes for the remote support console.

Strategic Considerations

The SCKit incident suggests that remote support tooling should be governed as tier-zero infrastructure alongside identity providers and domain controllers. Organizations relying on MSPs should build contractual requirements for extension inventories, MFA on every privileged path, audit log retention, log sharing with customers, and timely incident notification into their provider agreements. MSPs should consider segmenting customers across separate instances or session groups so that one compromised technician session cannot reach every tenant, accepting some operational cost in exchange for a smaller blast radius.

Threat models for remote support should also account for the reverse direction of trust. A managed endpoint can be a source of hostile input to the technician, through file transfers, clipboard sharing, or protocol-level flaws such as CVE-2026-84869. Designing controls on the assumption that customer endpoints may be compromised, and that the technician's console session is the asset under attack, is a more accurate starting point than treating the MSP as the sole source of risk to the customer.

CSA Resource Alignment

CSA's *Roles and Responsibilities of Third-Party Security Services* addresses how responsibility should be divided between organizations and the external service providers that operate security and IT functions on their behalf [8]. The SCKit incident tests that division directly: customers bore the impact of a configuration weakness in an extension they could neither see nor control. The guidance on clearly defining provider obligations, visibility, and accountability in service agreements provides a basis for the contractual requirements described under Strategic Considerations, including extension inventories, audit log sharing, and incident notification.

The CSA AI Safety Initiative's rapid research note *Helpdesk Hijack: How Agentic AI Support Systems Enable Account Takeover* examines how support channels, designed to help legitimate users, become a privileged path for attackers [9]. Although that paper concerns AI-driven support systems and this incident concerns human-operated remote support, both show that support tooling concentrates privilege and trust in ways that are often less scrutinized than primary authentication paths. The parallel between an unauthenticated extension API and an over-trusting support agent is instructive for teams reviewing every route into their support infrastructure.

CSA's *Zero Trust Guidance for Achieving Operational Resilience* is relevant to the architectural response [10]. The core failure in this incident was implicit trust: an extension's requests were trusted without user authentication, endpoints were trusted by technicians, and a session cookie would have been trusted in place of a fresh MFA challenge. Zero Trust principles of continuous verification, least privilege, and segmentation map directly to the recommendations above on extension authentication, role restriction, and per-tenant segmentation of remote support instances.

Finally, the AI Controls Matrix (AICM) v1.1 provides control objectives for supply chain management, identity and access management, and threat and vulnerability management that organizations can use to evaluate MSP and remote support risk [11]. Its supply chain domain supports assessing third-party extensions and providers, while its identity domain applies to privileged access paths such as RMM+ that operate outside primary authentication. Separately, the indication that RelayTrap may have been written with LLM assistance is relevant to CSA's broader AI Safety Initiative work on AI-enabled offensive capability, though it does not change the control analysis above.

References

[1] Huntress. "SCKit Worm Installs Rogue ScreenConnect Clients and Session Cookie Relays Across Hundreds of Endpoints." Huntress Blog, November 11, 2026 (updated).

[2] Abrams, Lawrence. "New worm-like 'SCKit' malware targets MSPs in ScreenConnect attacks." BleepingComputer, November 12, 2026.

[3] The Register. "Hackers exploiting ScreenConnect plugin flaw to spread malware across hundreds of PCs." The Register, November 12, 2026.

[4] CISA, NSA, and MS-ISAC. "Protecting Against Malicious Use of Remote Monitoring and Management Software (AA23-025A)." CISA, January 25, 2023.

[5] NIST. "CVE-2024-1709 Detail." National Vulnerability Database, February 2024.

[6] CISA, NSA, FBI, and international partners. "Protecting Against Cyber Threats to Managed Service Providers and their Customers (AA22-131A)." CISA, May 11, 2022.

[7] Markovic, Sinisa. "Attackers use rogue ScreenConnect clients to spread malware." Help Net Security, September 7, 2026 (updated).

[8] Cloud Security Alliance. "Roles and Responsibilities of Third-Party Security Services." CSA, 2021.

[9] Cloud Security Alliance. "Helpdesk Hijack: How Agentic AI Support Systems Enable Account Takeover." CSA, June 2026.

[10] Cloud Security Alliance. "Zero Trust Guidance for Achieving Operational Resilience." CSA, 2026.

[11] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." CSA.