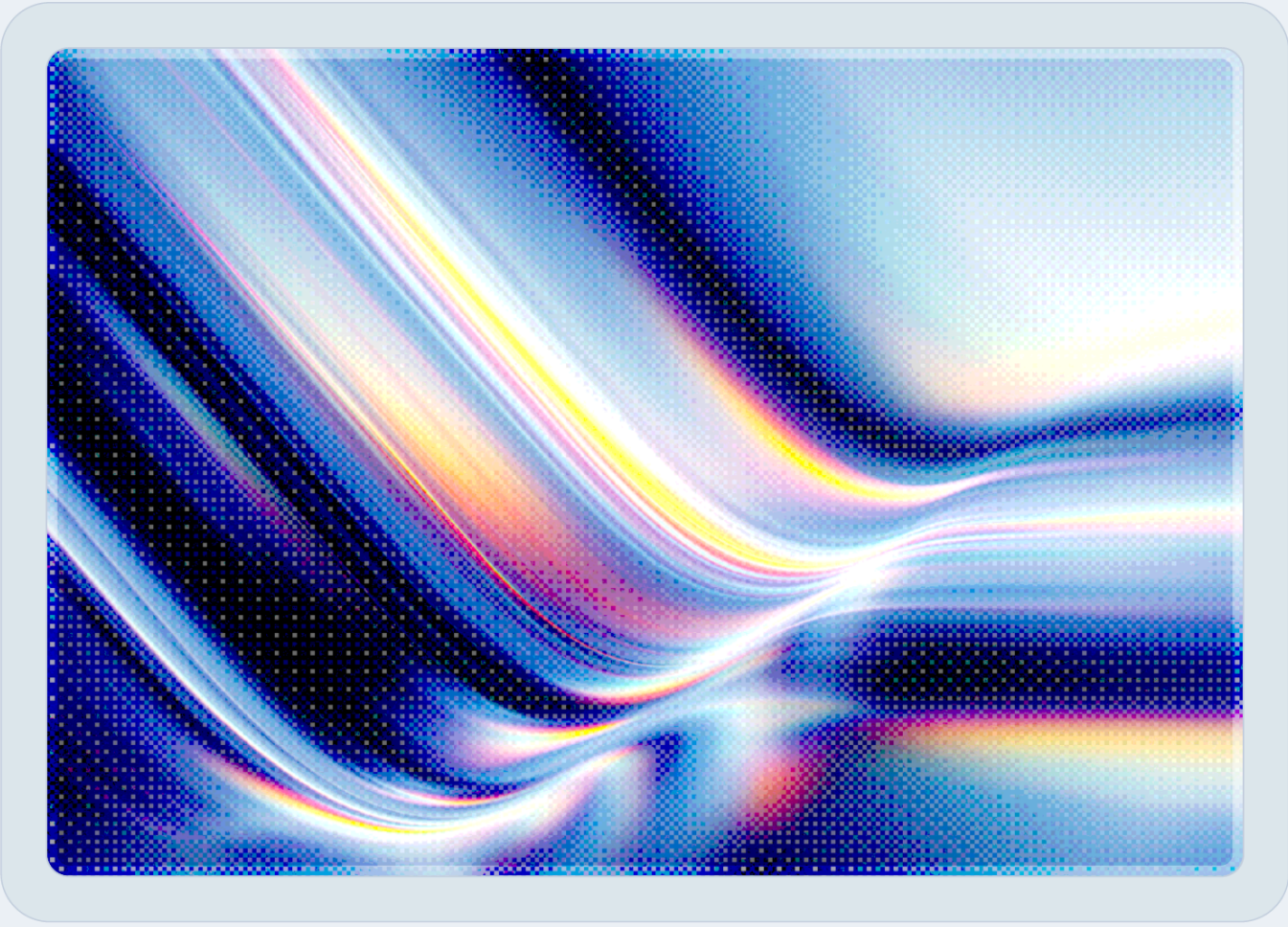
CSAI Foundation | Cloud Security Alliance

SimpleHelp Auth Bypass Deploys AI-Targeting Djinn Stealer

CVE-2026-48558 OIDC Exploit Exposes Developer AI Credentials and MCP Configurations

2026-06-30

AI-assisted Rapid Research



CSAI | cloud security alliance®

Key Takeaways

- CVE-2026-48558, a CVSS 10.0 critical authentication bypass [1] in SimpleHelp's OpenID Connect (OIDC) implementation, is being actively exploited to deliver Djinn Stealer, a previously undocumented cross-platform credential harvester with an unusual focus on AI development tool configurations.
- Attackers who successfully exploit the vulnerability obtain a fully authenticated technician session, enabling them to reach every managed endpoint connected to the compromised SimpleHelp server—transforming a single authentication bypass into broad downstream access across organizations.
- Djinn Stealer is specifically engineered to extract configuration and session data for AI coding assistants including Claude, Gemini, Codex, Cline, OpenCode, and Kilo. Its targeting of Model Context Protocol (MCP) configuration files means that stolen AI tokens may carry far more access than a stolen API key alone—granting attackers the same downstream permissions the developer's AI agent held, including access to source repositories, databases, cloud accounts, and internal APIs.
- Patches addressing CVE-2026-48558 were released by SimpleHelp on May 26, 2026 (versions 5.5.16 and 6.0 RC2). CISA added the vulnerability to its Known Exploited Vulnerabilities catalog on June 29, 2026 [12], with a federal patching deadline of July 7, 2026. Organizations that have not yet patched should treat remediation as an emergency response.
- Approximately 14,000 SimpleHelp servers are currently internet-facing; roughly 7.2% of sampled servers were found to be configured with the vulnerable OIDC authentication method [1][9].

Background

SimpleHelp and Remote Monitoring and Management Risk

SimpleHelp is a commercial remote monitoring and management (RMM) platform used by managed service providers (MSPs) and enterprise IT teams to administer endpoints across client organizations. Like other RMM tools, SimpleHelp provides legitimate technicians with the ability to remotely access systems, execute scripts, transfer files, and manage device configurations—capabilities that make it an attractive target for threat actors seeking not one compromised machine, but a vector into every machine a provider manages.

This dynamic is not new. Since at least early 2025, ransomware groups have treated SimpleHelp as a high-leverage initial access target. In January 2025, Horizon3.ai disclosed three vulnerabilities in SimpleHelp—CVE-2024-57726, CVE-2024-57727, and CVE-2024-57728—the most severe of which (CVE-2024-57727, a path traversal flaw) was exploited by the DragonForce ransomware group in a campaign observed by Arctic Wolf beginning January 22, 2025 [10][11]. CISA added CVE-2024-57727 to the Known Exploited Vulnerabilities catalog in early 2025 [11]. The 2026 campaign follows the same strategic logic—compromise the RMM platform, inherit technician access to all downstream endpoints—but introduces a payload specifically shaped to exploit AI-integrated development environments increasingly common in technology organizations.

CVE-2026-48558: An OIDC Signature Validation Failure

CVE-2026-48558 was assigned on May 21, 2026, reported to SimpleHelp on May 22, patched on May 26 (SimpleHelp 5.5.16 and 6.0 RC2), and publicly disclosed by Horizon3.ai on June 12, 2026 [2][8]. The vulnerability resides in SimpleHelp's handling of OpenID Connect (OIDC) authentication—a protocol commonly deployed in enterprise environments to delegate user management to an external Identity Provider (IdP) such as Azure Active Directory.

The root cause is a failure to verify the cryptographic signature of OIDC identity tokens submitted during login. In a correctly implemented OIDC flow, the relying party—in this case the SimpleHelp server—must validate the signature on the JSON Web Token (JWT) issued by the identity provider before accepting the claims within it. SimpleHelp did not perform this validation, meaning an attacker could craft a token bearing arbitrary identity claims and submit it without ever interacting with a legitimate IdP. The application would accept the forged token as authoritative, allowing the attacker to self-register as a new Technician account [1][8].

For this exploit to succeed, the targeted SimpleHelp server must have OIDC authentication enabled, have at least one TechnicianGroup associated with the OIDC provider, and have the "Allow group authenticated logins" setting enabled on that group [2][8]. These conditions are present in many enterprise deployments where OIDC with Azure AD is a common configuration for centralizing identity management [1][8]. The exploit also allows the attacker to bypass MFA enrollment requirements by registering their own authenticator device during account creation.

Security Analysis

The Attack Chain: From OIDC Bypass to Credential Harvest

Blackpoint Cyber researchers documented an incident in which the full attack progressed from authentication bypass through credential exfiltration within a single operational sequence [4][5]. The intrusion unfolded in four stages.

In the entry stage, attackers exploited CVE-2026-48558 against an internet-facing SimpleHelp server, forging an OIDC token to create and authenticate as a new Technician account with full platform privileges. SimpleHelp's legitimate RMM capabilities were then used as the delivery mechanism for the next stage—there was no need for a separate exploitation step to reach managed endpoints, because the technician role already authorized remote command execution across every connected machine.

The loader stage introduced TaskWeaver, a previously undocumented Node.js-based malware that Blackpoint describes as the operational backbone of the campaign. Attackers used SimpleHelp to download a file named `jquery.js`—a 1.08 MB obfuscated Node.js webpack bundle—from a randomly generated Cloudflare Tunnel URL (`*.trycloudflare.com`), then executed it via `node.exe`. TaskWeaver reconstructs access to Node.js's native `require()` function at runtime rather than referencing it statically, a technique intended to evade security tools that monitor literal `require()` calls. The loader then fingerprinted the host, generated process listings and environment inventories, and established an encrypted command-and-control (C2) channel to `a.dev-tunnels.com` using AES-256-GCM encryption with RSA-2048 key wrapping [4].

In the delivery stage, the C2 server responded with a "deliver" task containing Djinn Stealer as a 298,474-byte secondary payload. The RSA public key embedded in Djinn Stealer matches TaskWeaver's, confirming a common development origin for both tools [4]. In the final exfiltration stage, Djinn Stealer harvested credential material and transmitted it to an attacker-controlled IP address.

Djinn Stealer: Breadth of Collection with AI-Specific Precision

Djinn Stealer is cross-platform malware running on Windows, macOS, and Linux, engineered to strip a compromised system of credential material in a single pass. Its collection scope spans more than a dozen credential categories: cloud platform credentials for AWS, Azure, Google Cloud, Oracle Cloud, Okta, Cloudflare, DigitalOcean, Heroku, and Vercel; SSH keys; Docker authentication configurations; Terraform state files and HashiCorp Vault tokens; Git credentials; browser-stored credentials and history; and cryptocurrency wallet files for Bitcoin, Ethereum, Monero, Exodus, and Atomic Wallet [3][4].

Blackpoint Cyber researchers describe Djinn Stealer's explicit collection module targeting AI-assisted development toolchains as a distinguishing characteristic of the malware [4]. The malware searches for configuration, authentication, session, and project data associated with Claude, Gemini, Codex, Cline, OpenCode, and Kilo [3][4][7]. It also targets package registry credentials across npm, Yarn, NuGet, Composer, Maven, Gradle, Cargo, and PyPI—reflecting awareness that developer machines are trust anchors for both cloud infrastructure and software supply chains [6][7].

Many AI coding assistants rely on the Model Context Protocol to connect an AI assistant to external services on the developer's behalf—source repositories, databases, cloud accounts, and internal APIs are all accessible through MCP server connections. Djinn Stealer specifically searches for files such as `~/.claude/mcp.json` and analogous MCP configuration files for other assistants [4][6]. A stolen AI assistant API token is a narrow credential; a stolen MCP configuration grants an attacker the same downstream access the developer's AI agent held. In practice, this can mean access to repositories, CI/CD pipelines, database accounts, and internal APIs without the attacker needing to authenticate to each individually.

This targeting pattern reflects a deliberate threat actor decision to harvest not just today's credentials, but the compounded access that AI-integrated developer environments have assembled and trusted to a local agent. For organizations that have deployed AI coding assistants at scale, the blast radius of a single compromised developer machine may be substantially larger than it was before AI toolchain integration—a consequence of the compounded access that MCP configurations encode.

The RMM-to-AI-Credential Kill Chain

The full attack chain—RMM authentication bypass, Node.js loader, AI credential harvester—is consistent with a coherent adversarial strategy rather than opportunistic tooling: each component is purpose-fitted to the next, and Djinn Stealer's AI-credential targeting suggests advance planning to harvest developer machines specifically. SimpleHelp's legitimate access to managed endpoints removes the need for lateral movement or privilege escalation on individual hosts. TaskWeaver's minimal footprint and runtime

obfuscation minimize detection surface during the dwell period. Djinn Stealer's prioritization of AI development tool credentials indicates that the threat actor specifically intended to harvest developer machines, whose credential stores are disproportionately valuable.

This pattern should be understood alongside the 2025 SimpleHelp campaigns, in which the same RMM platform was used as initial access for ransomware deployment [10][11]. The transition from ransomware delivery to information stealing with AI-credential focus may reflect evolving threat actor economics: AI tokens can provide persistent, authenticated access to high-value cloud infrastructure without triggering the destructive signals that ransomware deployment generates. The stolen credentials may be used directly, sold to other actors, or retained for longer-term access to target environments.

Recommendations

Immediate Actions

Organizations running SimpleHelp should treat CVE-2026-48558 remediation as an emergency response, not a scheduled maintenance task. The patch—SimpleHelp versions 5.5.16 (stable) or 6.0 RC2 (pre-release)—should be applied immediately across all SimpleHelp server instances [2][8]. Administrators should not wait for a maintenance window.

Following patching, administrators must audit their technician account roster. SimpleHelp provides this through Administration → Technicians → Gear Icon → Show Group Authenticated Users [8]. Any accounts not corresponding to known, authorized personnel should be treated as indicators of compromise and investigated. Server logs should be reviewed for unexpected technician registrations and authenticated sessions from unfamiliar IP addresses, particularly during the exposure window between the June 12 public disclosure and the organization's patch date.

Organizations that cannot patch immediately should restrict technician authentication to approved IP addresses using SimpleHelp's access control settings, which limits the exploit surface without requiring a full version upgrade [8]. OIDC authentication should be temporarily disabled if it cannot be confirmed as required for current operations.

Any organization whose SimpleHelp server is used by an MSP or IT service provider should notify their service provider and request evidence of patching and account audit completion. An MSP's failure to patch a shared SimpleHelp deployment creates downstream exposure for all of its clients regardless of their own security posture.

Short-Term Mitigations

Security teams should deploy detection logic for the known IOCs associated with this campaign. Relevant indicators include the SHA-256 hashes for TaskWeaver (`00cc86d1144020c24c8fbb3a8dc6b908926497ebd23be3bf854360f93d1c8f4c`) and Djinn Stealer (`f4a72600a3735c2a4d843875ea61bbb6f935a1af51a81f2fbc992ce11ba94afc`), C2 communications to `a.dev-tunnels.com`, HTTP POST requests matching the pattern `/api/<base64url>.<base64url>.<base64url>`, staging connections to `*.trycloudflare.com`, outbound traffic to `96.126.130.126:58942`, and execution of `node.exe` with a `jquery.js` argument [4].

AI development credentials—API keys for cloud-hosted AI services, local configuration files for Claude, Gemini, Codex, and similar tools, and especially MCP configuration files—should be treated as potentially compromised on any endpoint that had network exposure during the vulnerability window. This means rotating API keys, revoking OAuth tokens, auditing MCP server permission grants, and reviewing access logs for AI services to detect unauthorized activity. Because MCP configurations can grant access to databases, repositories, and cloud accounts, the scope of credential rotation must extend to those downstream resources as well.

Endpoint detection rules should flag unusual use of `node.exe` for file execution not initiated by known development workflows, as TaskWeaver's execution pattern (`node.exe <path>\jquery.js`) represents an anomalous use of the Node.js runtime outside development contexts.

Strategic Considerations

This incident reflects a broader pattern that deserves organizational attention beyond emergency patching. RMM platforms represent a structural amplifier for initial access: a single compromised server grants access to all managed endpoints. Organizations should audit their RMM vendor inventory, confirm that all internet-facing RMM instances are covered by vulnerability monitoring and patch management processes, and ensure that RMM access is restricted to known-good IP ranges and subject to privileged access workstation policies.

The AI tool credential targeting in Djinn Stealer signals that developer machines hosting AI coding assistants have become high-value targets in their own right. Organizations deploying AI development toolchains should apply the same credential hygiene discipline to AI tool configurations that they apply to cloud IAM credentials: store API keys in secrets managers rather than local config files, scope MCP

server permissions to the minimum required for each workflow, and rotate credentials on a defined schedule. MCP configurations that grant access to production databases or critical cloud accounts should be reviewed and scoped down regardless of the current threat level; the principle of least privilege applies as much to AI agent access as to human user access.

Finally, organizations should develop clear policies governing the installation of AI development tools and MCP server connections on endpoints that also handle sensitive infrastructure access. This is a broader organizational control that extends beyond the current incident—the convergence of AI-integrated development environments with high-privilege infrastructure credentials is a structural tension that any credential harvesting campaign can exploit. Separating AI-assisted development workstations from production credential access reduces the blast radius of any future campaign targeting this class of tools.

CSA Resource Alignment

This incident maps directly to several areas of CSA's AI security framework and published guidance.

CSA's MAESTRO threat modeling framework for agentic AI addresses the risks posed by credential harvesting of AI agent configurations, recognizing that such compromises carry cascading implications across the agentic stack. Djinn Stealer's targeting of MCP configurations exemplifies the risk associated with improperly scoped agent permissions: when an AI agent's configuration encodes access to multiple downstream systems, any compromise of that configuration inherits all of the agent's granted access. Organizations building or deploying AI coding assistants should apply MAESTRO's threat modeling methodology to their MCP server permission structures.

CSA's AI Controls Matrix (AICM) provides controls applicable to this incident across multiple domains. Areas including AI lifecycle management address credential protection for AI system configurations, while supply chain risk management is directly relevant given SimpleHelp's role as an MSP tool providing access to downstream client organizations. CSA's work on shadow access is also pertinent here, addressing the category of implicit, compounded access that MCP configurations create—access that exists in configuration files rather than identity provider records and is therefore invisible to conventional IAM auditing tools.

The Zero Trust principles central to CSA's ongoing work are directly applicable to RMM platform access. The implicit trust model under which a SimpleHelp technician session inherits broad access to managed endpoints fundamentally contradicts zero trust architecture's requirement for continuous authorization

verification. Organizations should evaluate whether their RMM platforms support continuous authorization verification, privileged access workstation requirements, and session recording that would limit the impact of a compromised technician account.

CSA's work on non-human identity (NHI) management is also relevant here. AI tool API keys and MCP configurations are non-human identities with meaningful access permissions; the same lifecycle management, rotation, and auditing practices that apply to service account credentials should apply to AI tool configurations.

References

[1] Horizon3.ai. "CVE-2026-48558: SimpleHelp OIDC Auth Bypass." Horizon3.ai, June 12, 2026.

[2] Help Net Security. "SimpleHelp RMM flaw could give attackers full access to managed endpoints (CVE-2026-48558)." Help Net Security, June 16, 2026.

[3] Help Net Security. "SimpleHelp vulnerability exploited to deliver mighty Djinn Stealer (CVE-2026-48558)." Help Net Security, June 30, 2026.

[4] Blackpoint Cyber. "A Djinn in the Machine: TaskWeaver's Node.js Intrusion Chain." Blackpoint Cyber, June 29, 2026.

[5] The Hacker News. "Attackers Exploit SimpleHelp CVE-2026-48558 to Deploy TaskWeaver and Djinn Stealer." The Hacker News, June 30, 2026.

[6] BleepingComputer. "Critical SimpleHelp flaw exploited to deploy new stealer malware." BleepingComputer, June 30, 2026.

[7] Dark Reading. "Djinn Stealer Targets Cloud and AI Credentials." Dark Reading, June 30, 2026.

[8] Horizon3.ai. "CVE-2026-48558: SimpleHelp Auth Bypass IOCs." Horizon3.ai, June 2026.

[9] Cybersecurity News. "Nearly 14,000 SimpleHelp Servers Exposed Amid Critical Authentication Bypass Disclosure." Cybersecurity News, June 2026.

[10] Arctic Wolf. "Campaign Exploiting SimpleHelp RMM Software for Initial Access." Arctic Wolf, January 2025.

[11] CISA. "Ransomware Actors Exploit Unpatched SimpleHelp Remote Monitoring and Management to Compromise Utility Billing Software Provider." CISA Advisory AA25-163A, June 2025.

[12] CISA. "CISA Adds One Known Exploited Vulnerability to Catalog." CISA, June 29, 2026.