

Clop Returns with Custom Implant in Mass-Extortion Campaign

 reliaquest.com/blog/clop-returns-with-custom-implant-in-mass-extortion-campaign

August 18, 2026

- [Threat Intelligence](#)
- [Emerging Threats](#)
- [Agentic AI](#)



Editor's note: This report was authored by John Dilgen and Connor Short.

This is external threat intelligence from the ReliaQuest Threat Research team. The findings describe threats, vulnerabilities, and attacker activity affecting third parties and the broader threat landscape—not ReliaQuest's own environment. Nothing in this report should be interpreted as a vulnerability in ReliaQuest's systems or data.

Key Points

- “Clop's” exploitation of CVE-2026-12569 in PTC Windchill has returned the group to mass exploitation, delivering a custom web shell that provides full data-theft capability from the moment of deployment, with no additional tooling required.

- ReliaQuest identified the web shell as a fully equipped extortion platform: it maps sensitive vault data, decrypts every credential in the Windchill keystore, and includes a custom Java class loader that lets Clop execute any additional code inside the application process, extending the shell into an unlimited backdoor for follow-on activity such as lateral movement, ransomware, or persistence.
- Immediately apply the vendor fix for CVE-2026-12569, hunt for suspicious JavaServer Pages (JSP) files in Windchill codebase directories and rotate all credentials in the Windchill keystore on any server suspected of compromise.

ReliaQuest identified a custom web shell highly likely linked to "Clop" (aka ClOp), a financially motivated ransomware and extortion group known for mass-exploiting enterprise software vulnerabilities. The web shell is deployed after exploitation of CVE-2026-12569 (CVSS 9.3) in PTC Windchill, an industry-standard product lifecycle management (PLM) platform used by manufacturing enterprises worldwide to store engineering data and product designs. The web shell gives attackers a direct path to credential theft and large-scale data exfiltration, with no additional tooling required. Unlike generic command shells, this implant decrypts credentials, delivers malware, and maps stored files for exfiltration.

This appears to be an application-specific evolution of Clop's established mass-exploitation playbook. The web shell is purpose-built to target this tool: it embeds detailed knowledge of the application's APIs, database schema, keystore, and file-vault structure, enabling rapid movement from access to data theft, without external commands or additional tools. References to "Clop" throughout reflect this highly likely attribution.

Because this application centralizes high-value product designs and intellectual property, a single successful compromise could hand an attacker both a ready inventory of sensitive data and the credentials needed to reach systems well beyond the affected server. The web shell's built-in Java class loader can execute additional attacker-supplied code entirely in memory, leaving no files written to disk, enabling lateral movement, ransomware deployment, or long-term persistence at any point after initial compromise.

Read on to learn:

- How CVE-2026-12569, a remote code execution (RCE) flaw in PTC Windchill, likely gives Clop its initial foothold.
- Why a compromise can quickly escalate to an enterprise-wide credential compromise.
- What makes this web shell more dangerous than generic alternatives, and how to prioritize your response.

A Purpose-Built Data-Theft Platform, not a Generic Shell

Through investigation, we found that the web shell arrives as a fully equipped platform with built-in functions for credential harvesting, discovery of critical information locations, arbitrary file reads, and file transfers already present from the moment of deployment. It is more than the type of conventional web shell commonly used by other attackers. Beyond its built-in toolkit, a custom Java class loader lets the threat actor run attacker-supplied Java modules inside the application process. This allows the group to add capabilities on demand, including tools for sustained access, network traversal, or data encryption.

The combination of a feature-rich implant that requires no additional tooling to begin stealing data, paired with an extensible delivery mechanism for follow-on capability, gives the adversary a complete toolkit from the moment of access. The attacker can therefore move quickly from initial access through data theft to further post-exploitation activity entirely within the application's own trust boundary, using the web shell's purpose-built features without executing manual commands. The approach significantly limits defenders' ability to detect the activity, as it closely mimics the application's standard functions.

One Command Decrypts Credentials in Plaintext

A single "S" command to the web shell returns Windchill's directory-management and administrative credentials in plaintext. This is performed by a built-in function called **gs**, which works in three steps:

- It reads Windchill's `ieStructProperties.txt` configuration file.
- It decrypts the Lightweight Directory Access Protocol (LDAP) manager password from the application keystore.
- It iterates through all stored local properties, decrypting additional encrypted values including administrative account credentials, object storage credentials, and all site administrator keys.

During an active compromise, the S command returns the credentials Windchill holds, including those used to manage the organization's LDAP directory. A separate command is then used to exfiltrate the results. Because LDAP credentials typically govern access to Active Directory, email systems, VPN, and other enterprise services tied to directory authentication, their exposure could turn a single application compromise into an enterprise-wide credential compromise. The resulting privileged access fuels data theft from additional applications and storage locations, as well as persistence for follow-on attacks.

The Built-In Class Loader Serves as a Reusable Backdoor

One of the web shell's most dangerous components is a custom Java class loader (Cldr) that allows the attacker to deliver and run code on the server without modifying the web shell itself. It accepts a Base64-encoded ZIP file containing compiled Java bytecode, loads it directly into memory, and executes it. While Java class loading is not a novel technique, its use here means attacker-supplied code runs entirely in memory, leaving no files written to disk and reducing the artifacts available for detection.

This design makes the web shell a delivery mechanism for extensible follow-on capabilities. For example, the attacker could send new malware, including network propagation tools, targeted exploits, or file-encrypting payloads. This enables objectives beyond data theft and extortion, including pivoting into other sensitive data storage locations or network encryption for double extortion.

Built-In Vault Enumeration Targets Sensitive Engineering Files

The web shell's vault enumeration capability is purpose-built for Windchill, targeting the application's specific database structure to map sensitive files before any data leaves the server. By automating discovery within the application, the attacker can identify high-value engineering data without executing manual discovery commands, compressing the window between initial access and targeted data theft, leading to faster and more scalable attacks than manual operations.

The web shell includes a built-in function ("fl") and a class ("Flst1") that query the application's database for vault stream IDs, filenames, storage paths, and file sizes. The results are written to a file named flst.txt, giving the attacker a ready-made map of the repository for selecting files to steal. Flst1 uses a "gdb()" helper function to connect directly to Windchill's database using the application's own internal Java classes, making the enumeration specific to Windchill's architecture. The queries themselves are executed through Windchill's own database connection, a design that bridges directly into the detection challenge covered in the next section: because the activity runs through the application's existing access context, it is difficult to distinguish from normal application behavior.

Malicious Queries Blend into the Database's own Traffic

The web shell connects to Windchill's database through the application's own MethodContext and WTConnection classes, meaning its queries run under the application's existing database identity rather than through a separately configured attacker account. As a result, database telemetry may attribute this activity to the application's normal service identity, limiting the value of alerts that rely solely on detecting new accounts or unexpected source hosts.

The queries reference internal application field names, including specific identifiers such as "IDA3A4," reflecting detailed prior knowledge of the application's database schema. Because they execute through the application's own context, they are less likely to trigger database alerts, which are more likely to attribute the activity to standard operations. This also undermines containment plays that rely on flagging unusual database accounts or connections, removing an opportunity to stop the exfiltration that follows.

Custom Integration Makes this Implant Harder to Detect

The web shell imports and calls application-specific classes, including MethodContext, WTConnection, and WTKeyStoreUtil, and contains queries tailored to Windchill's vault structure, demonstrating that it was highly likely developed with detailed knowledge of the application rather than reused unchanged across unrelated technologies. Typically, threat actors reuse generic web shells across attacks for simplicity; however, Clop usually builds custom web shells specific to the software it is exploiting.

Generic web shells usually expose reusable operating-system or file management functions or simply offer a command prompt that allows threat actors to run their own instructions. "China Chopper," one of the most widely reused web shells in the threat landscape, illustrates the baseline: its two-component design has been deployed by a broad range of unrelated threat actors across different technologies because it's widely available and requires no knowledge of the underlying application to operate. These shells carry known patterns and signatures that detection tools commonly look for, making them easier to identify and block. This implant, by contrast, carries out Windchill-specific discovery and credential access from inside the application process, using the application's own classes and database connections. Because it mimics normal application behavior rather than following known attacker patterns, it is significantly harder to detect using signature-based controls.

Custom Headers and GZIP Blind Standard Network Monitoring

Because the web shell is highly likely built specifically for the application it targets, its traffic blends in with ordinary application requests at every layer, making it significantly harder for organizations to detect this activity using standard network monitoring. The shell routes commands through a custom HTTP header, "X-windchill-req," rather than a visible request body, and connects to the database through Windchill's own internal programming interfaces rather than a standard database connection method. The web shell also uses GZIP to compress its responses, making the returned data appear as standard compressed web content.

This limits detection opportunities for defenders: security controls that inspect only URL paths or request body parameters will not see the command traffic at all, and those that log headers without also decompressing responses will capture the instructions but miss the data being returned. Tools lacking Transport Layer Security (TLS) inspection will not see either.

Identifying this activity requires header logging that captures non-standard values, response decompression, and TLS inspection; without all three, coverage against this web shell's traffic is partial at best. Collectively, these design elements reduce detection opportunities and increase dwell-time potential for the attacker.

Extortion Emails and Reused Tradecraft Point to Clop

This activity was highly likely conducted by the Clop extortion group.

- Extortion emails match contact addresses currently displayed on Clop's active data-leak site (see Figure 1)
- The web shell uses the X-windchill-req header, previously linked to Clop in external reporting on CVE-2026-12569
- Tactics, techniques, and procedures (TTPs) align with Clop's established pattern: mass exploitation followed by custom web shell deployment

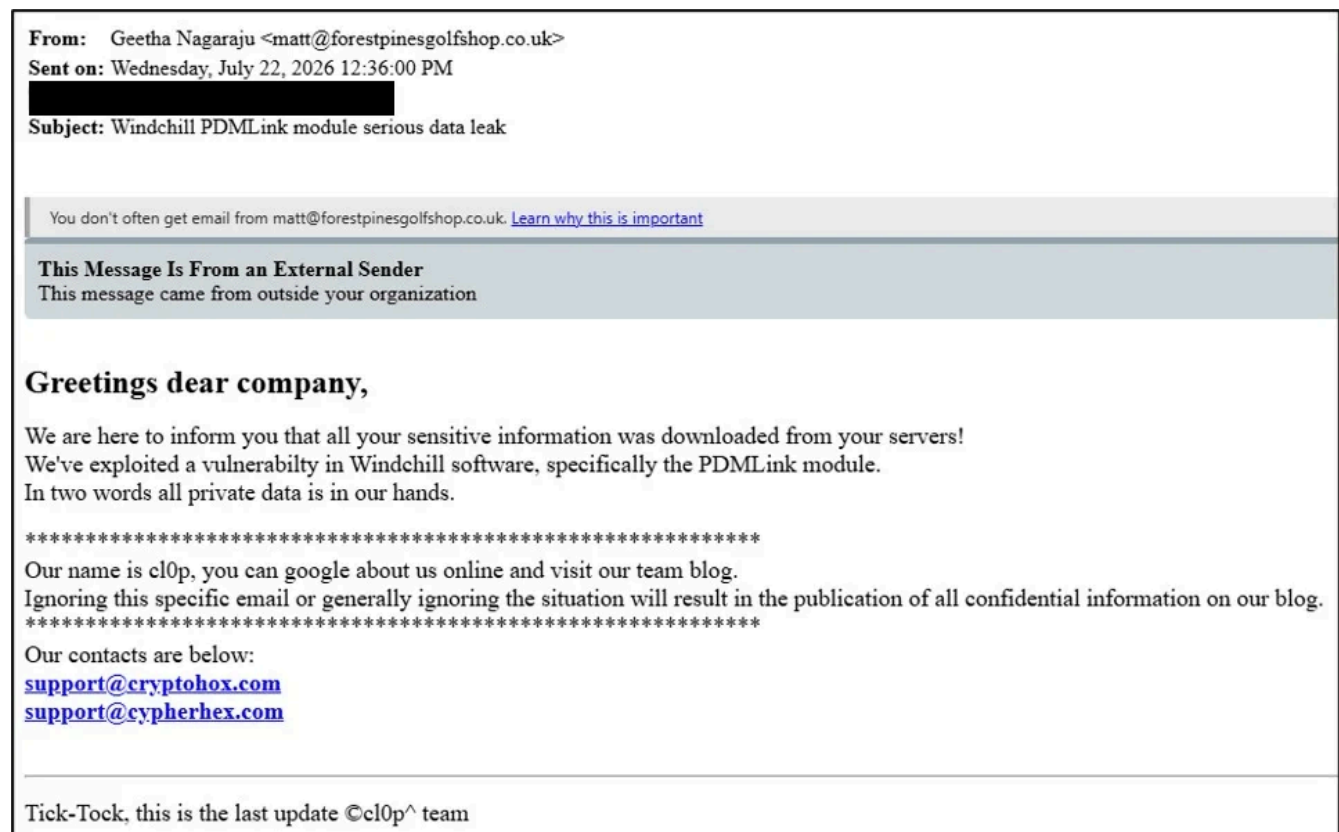


Figure 1: Clop's CVE-2026-12569 extortion email (Source: Ransom-ISAC)

[The group is known for mass exploitation of vulnerabilities](#) in software that holds sensitive data, followed by custom-built web shell deployment. For example, Clop previously deployed the custom web shell "DEWMODE" after exploiting the SQL injection vulnerability CVE-2021-27101, and it deployed "LEMURLOOT" after exploiting CVE-2023-34362.

This campaign is another reminder that Clop remains a sleeping dragon, always looking and preparing to mass exploit vulnerabilities in software that holds sensitive data. The group commonly goes inactive between campaigns but springs to life with custom-built web shells whenever there is another opportunity for mass extortion.

Step Up Your Defenses Against Clop

ReliaQuest's Approach

Clop's likely latest implant scatters its footprint across telemetry sources, making individual alerts appear routine. Detecting and containing it requires correlation across the full attack chain. ReliaQuest GreyMatter, an agentic AI security operations platform, provides capabilities directly relevant to the TTPs observed in this campaign:

GreyMatter Agentic AI: The implant's activity can appear across web, application, database, file, and identity telemetry, causing individual alerts to look unrelated or lower priority.

GreyMatter Agentic AI can correlate these signals across the attack chain—for example, suspicious Windchill requests, vault-table queries, creation of flst.txt, large outbound transfers, and subsequent use of recovered credentials—to prioritize the activity as a coordinated compromise and enable immediate containment actions.

GreyMatter Digital Risk Protection: If Clop successfully bypasses detections and steals credentials and sensitive data, the group might then post that data on its data-leak site and potentially sell the access on criminal forums to the highest bidder. GreyMatter's Digital Risk Protection can alert on these events that take place on the dark web and outside of your standard logging and detections to surface exposure early and reduce the window for brand damage.

ReliaQuest Detection Rules: ReliaQuest detection content is continuously updated using the latest relevant threat intelligence.

GreyMatter Automated Response Playbooks turn detection into immediate action. Once web shell activity or credential misuse is identified, prebuilt playbooks can block attacker infrastructure and disable affected accounts—all within minutes and across your existing security stack. Pairing detection rules with the following Automated Response Playbooks

helps your organization cut mean time to contain (MTTC) threats from hours to five minutes or less:

Block IP: Blocks the attacker infrastructure communicating with the web shell at the perimeter, severing the command-and-control (C2) path while host remediation begins. Because blocking the IP address does not remove the web shell itself or invalidate credentials that have already been harvested, this play should always be paired with host remediation and secret rotation.

Terminate User Session: Revokes active sessions tied to any account whose credentials were exposed through the web shell's credential harvesting function, including the LDAP manager and site administrator accounts recovered from the Windchill keystore. Session termination is critical because rotated passwords alone leave existing tokens valid.

Disable User: Suspends any account whose credentials were returned in plaintext by the S command, preventing credential replay across the environment while password rotation, LDAP directory review, and downstream authentication audits are completed.

Your Action Plan

The following recommendations address the main exposure chain: preventing CVE-2026-12569 exploitation, web shell deployment, and sensitive data theft.

- **Patch and Harden Internet-Facing Windchill Servers:** Apply the vendor-provided fix for CVE-2026-12569 immediately on all Windchill instances. Additionally, restrict internet exposure of Windchill management interfaces, place the servers behind a web application firewall (WAF), and monitor for exploitation attempts against the vulnerable endpoints.
- **Hunt for Suspicious JSP Files in Windchill Codebase Directories:** Review the windchill/codebase/login directory and other Windchill codebase paths on all Windchill servers for unexpected JavaServer Pages (JSP) files that could be web shells. Prioritize files with recent modification timestamps, unfamiliar filenames, or content that references the custom X-windchill-req header, MethodContext, WTConnection, or WTKeyStoreUtil. Any file matching these criteria should be treated as a potential web shell and escalated for full response and removal.

- **Rotate LDAP and Windchill Administrative Credentials on Any Suspected Compromised Server:** If a Windchill server is confirmed or suspected to be compromised, rotate the LDAP manager password and any other credentials stored in the application keystore. Because the web shell's S command returns every encrypted value in plaintext, assume the full set has been exfiltrated and rotate accordingly across every downstream system where those credentials are reused, starting with those that can access other databases and servers.

Key Takeaways and What's Next

ReliaQuest assesses with high confidence that exploitation of CVE-2026-12569 will expand to compromise more organizations in the coming weeks, consistent with Clop's established pattern of scaling access across vulnerable infrastructure following initial vulnerability disclosure. ReliaQuest also assesses with moderate confidence that copycat threat actors will adopt this vulnerability as public exploit code becomes more widely available. For already-compromised environments where extortion demands go unmet, the web shell's credential harvesting capability and extensible class loader provide the technical foundation for deeper network access. Based on Clop's history of escalating pressure against non-paying targets, follow-on attacks targeting broader network access represent a likely next stage in these environments.

Clop's previous mass exploitation campaigns followed a consistent pattern of rapid, wide-scale targeting immediately after vulnerability disclosure. CVE-2026-12569 is tracking the same trajectory: within days of PTC's patch release, CISA added the vulnerability to its Known Exploited Vulnerabilities (KEV) catalog and required federal agencies to patch within three days, a response level consistent with the confirmed active exploitation seen at the outset of previous Clop campaigns.

IOCs

IOCs	Description
321e1fb01eb3462b48ff6ccdef132acc1182e3f7456548439f0d4ead12fd98bf	Hash of Clop's custom webshell

5.180.41[.]35	IP address associated with CVE-2026-12569 exploitation
78.128.113[.]10	IP address associated with CVE-2026-12569 exploitation
104.194.9[.]14	IP address associated with CVE-2026-12569 exploitation
104.243.35[.]63	IP address associated with CVE-2026-12569 exploitation
185.227.83[.]236	IP address associated with CVE-2026-12569 exploitation
209.222.98[.]44	IP address associated with CVE-2026-12569 exploitation

216.152.151[.]204

IP address
associated
with CVE-
2026-12569
exploitation

Learn How GreyMatter Can Improve Your DIR Process

GreyMatter enables you to get visibility across your entire attack surface, reduce complexity of your security operations, and efficiently manage risk across the business.

[Request A Demo](#)

