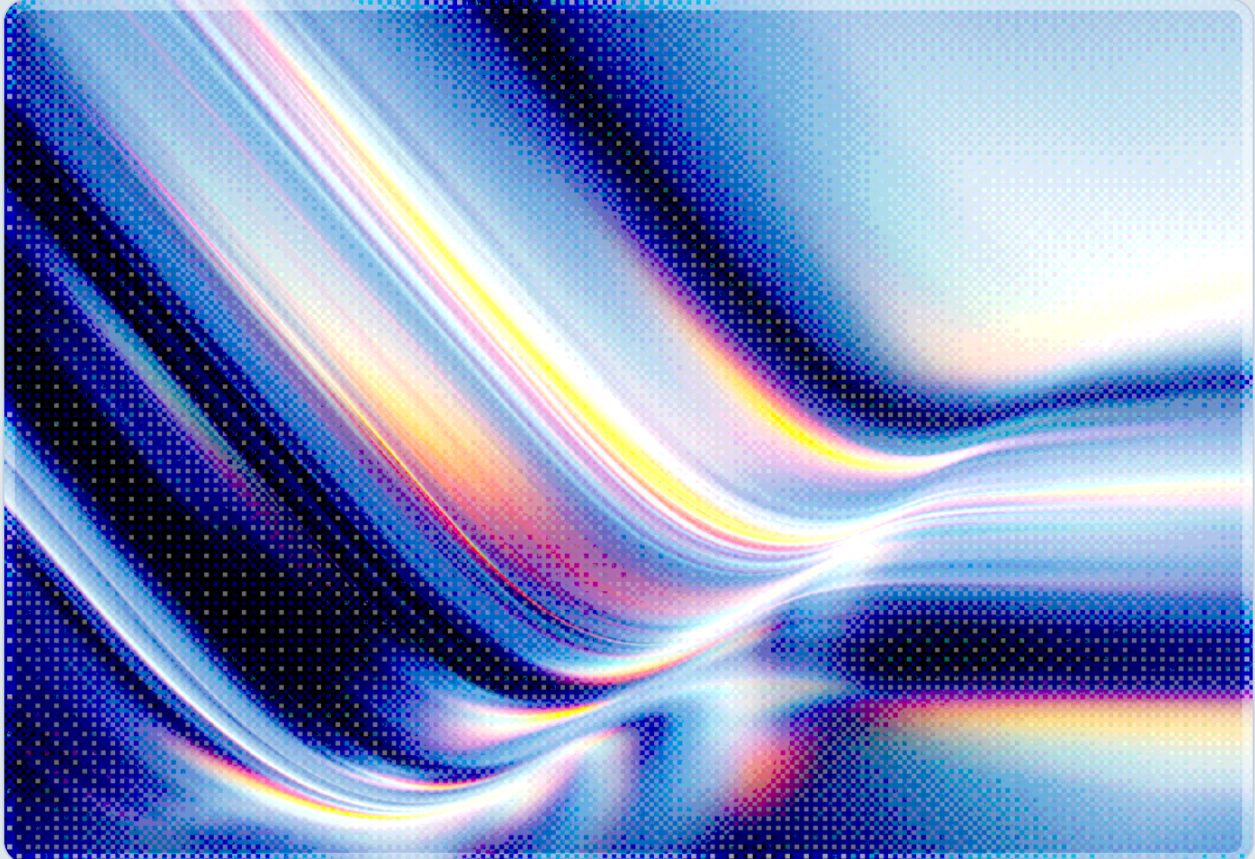


Mythos-Found Rejetto HFS Flaw Probed Within a Day of Public Write- Up

CVE-2026-61500 and the Shrinking Patch Window

2026-10-07

 AI-assisted Rapid Research



CSAI

cloud
CSA 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

CVE-2026-61500 is a critical authentication-bypass flaw in Rejetto HTTP File Server (HFS) 3.0.0 through 3.2.0 that allows an unauthenticated attacker to forge an administrator session and reach remote code execution [1][2]. According to press reports, the flaw was found by Zach Hanley of Horizon3.ai working with Anthropic's Mythos model [3][4]. The vendor fix, HFS 3.2.1, shipped on July 13, 2026, nearly three months before technical details became public [2][5].

Exploitation attempts were reported within roughly a day of Horizon3.ai's public write-up and proof-of-concept video [4][5][12]. The case illustrates a pattern CSA has described in earlier work on the shrinking window between disclosure and exploitation: a patch that already exists, combined with a public explanation, can be enough for attackers to move quickly [6][7]. Nothing in the reporting suggests that an AI-assisted discovery workflow was required on the attacker's side. The activity reported so far appears to be small-scale reconnaissance, and as of this writing the sources reviewed do not confirm successful compromises [2][3].

Three practical conclusions follow. Organizations running HFS should upgrade to 3.2.1 or later, preferably the current stable release, and should not expose the administrative interface to untrusted networks. Security teams should treat the publication of a detailed write-up as a trigger for re-prioritizing fixes that exist but have not been applied. Finally, defenders should not assume that a low-profile, niche product is outside the attention of opportunistic scanners once exploitation guidance is public.

Background

Rejetto HFS is a lightweight, commonly deployed HTTP file server. Older versions of the product have a record of exploitation, including CVE-2014-6287 and CVE-2024-23692, both of which appear in CISA's Known Exploited Vulnerabilities catalog [8][13]. The 3.x line is a rewrite on a JavaScript runtime, and CVE-2026-61500 affects versions 3.0.0 through 3.2.0 of that line. The record was published through the VulnCheck CNA on July 13, 2026, and carries a CVSS score of 9.3 [1][2].

The root cause is a combination of two weaknesses. HFS derives its session-cookie signing key from JavaScript's `Math.random()`, which is not a cryptographically secure generator, and it also discloses outputs of the same generator to unauthenticated clients during login [1]. The V8 engine implements

`Math.random()` with the xorshift128+ algorithm, whose internal state can be recovered from a small number of observed outputs [3][9]. A remote attacker can therefore collect several login responses, reconstruct the generator state, recover the signing key, and forge a valid administrator cookie. From administrator access, code execution on the host follows through the server's `server_code` configuration feature [1].

The discovery context is also notable. Horizon3.ai joined Project Glasswing, Anthropic's program giving vetted security partners access to Mythos, in July 2026 [4]. Reporting credits Hanley "in collaboration with" Claude and Anthropic Research, and describes the model as recognizing that the weak generator and the output leak formed a chain, and as proposing the Z3 SMT solver to recover the generator state [3][4]. These characterizations come from press coverage of the Horizon3.ai write-up; we have not reviewed the primary Horizon3.ai or VulnCheck posts or the underlying model transcripts. The Register reported, citing a public tracker maintained by Patrick Garrity, that 286 CVEs were credited to Mythos and Project Glasswing as of October 2 [4].

Reported Timeline

The sources agree on the sequence of events but differ by a day or two on exact dates. Secondary sources differ on the write-up date (September 30 or October 1) and on the first detected exploitation (October 1 or October 2), and we have not reviewed the primary Horizon3.ai and VulnCheck posts. The Register's weekday-based account (Wednesday disclosure, Thursday evening detection, Friday follow-on attempts) maps to September 30 through October 2, 2026, which is consistent with the dates reported by BleepingComputer and The Hacker News [4][5][12]. Readers quoting the timeline externally should consult the primary sources.

Event	Reported date	Source
CVE published; HFS 3.2.1 released	July 13, 2026	[2][5]
Horizon3.ai write-up and exploit video	September 30, 2026 (some sources report October 1)	[4][5][9][12]
First exploitation attempts detected by VulnCheck (reported from a China-based IP, targeting US and Japan canaries)	October 1, 2026 (some sources report October 2)	[3][4][5][12]
Additional attempts from two US-based addresses	The following day	[4]

Security Analysis

The reported facts invite several readings, and this section separates what the sources establish from what CSA infers. The first concerns the interval between fix and attack, the second concerns the role of AI-assisted discovery, and the third concerns software that sits outside normal inventory. A final subsection records what remains unknown.

The patch existed; the explanation appears to have been the trigger

The most notable feature of this case is that the fix was available for almost three months before exploitation attempts began [2][9]. Nothing about the vulnerability changed between July and October. What changed was that a clear technical explanation and a working demonstration became public, so an attacker no longer needed to find the flaw, only to read how it worked and reproduce it. The reporting does not say how the early scanners learned of the flaw, but the sequence suggests the write-up was the trigger. This matches the exploit-before-patch and collapsing-window dynamics CSA has previously analyzed, in which the interval between disclosure and weaponization shrinks toward hours and enterprise remediation timelines do not keep pace [6][7]. The difference here is the order of events: the patch came first, so the exposure appears attributable to unapplied updates rather than to a zero-day.

Organizations often triage patches using the severity score and exploitation status available at the time a fix is released. A flaw published quietly in July with no public proof of concept may reasonably have ranked below other work, and once the write-up appeared that ranking was out of date. Exposure management should therefore treat the release of a detailed technical analysis, a proof of concept, or a demonstration video as an event that reopens prioritization for any already-patched flaw in the asset inventory.

AI-assisted discovery changes defender economics, not attacker access

It would be easy to read this incident as "AI found a bug and attackers used it." The reported facts support a narrower reading. The discovery involved a human researcher at a security vendor working with a frontier model under a controlled program, and the disclosure followed a vendor fix [2][3]. The exploitation that followed was based on a human-authored public write-up. The reporting does not indicate that attackers used AI to find or weaponize the flaw, and we do not draw that inference.

The relevant effect is on the volume and quality of write-ups. The reported ability of the model to connect two individually modest weaknesses, a non-cryptographic generator and an output leak, and to reach for an SMT solver suggests that AI-assisted research can surface multi-step flaws that are easy to

overlook in manual review [3][4]. The Register's count of 286 credited CVEs shows that the program's output is already substantial [4]. CSA's inference is that the volume of well-explained disclosures will continue to rise. Each one is a potential trigger event of the kind described above, and the more of them that arrive, the more pressure falls on defenders' ability to apply patches quickly across less visible software.

Niche and self-hosted software is part of the attack surface

HFS is not an enterprise platform, and it is often installed by individuals, small teams, or departments outside central IT. Software of this kind is commonly absent from the asset inventories and vulnerability scanners that drive enterprise patching. The reported probing of VulnCheck canary systems indicates that internet-wide scanning for a newly explained flaw begins quickly, regardless of how prominent the product is [3][5]. We found no reliable public count of exposed HFS instances in the sources reviewed, and therefore make no estimate of the vulnerable population.

Limits of what is known

Several points remain unconfirmed in the sources reviewed. We did not find confirmation of successful compromise, of inclusion of CVE-2026-61500 in CISA's Known Exploited Vulnerabilities catalog, or of the identity or objectives of the actors behind the early probing. Some third-party aggregators may publish severity scores that differ from the 9.3 assigned by the VulnCheck CNA, so teams should rely on the CNA and NVD records rather than aggregator scores [1]. Because the flaw requires no credentials, the forged-session technique should at a minimum be considered applicable to any deployment where the vulnerable versions are reachable by an attacker, whether from the internet or from an internal network [1].

Recommendations

The recommendations below move from immediate response to longer-term program changes. Because the reported activity is reconnaissance-level and compromise is unconfirmed, the compromise-assessment guidance is precautionary rather than a statement that instances have been breached.

Immediate Actions

Teams should identify every HFS deployment, including those outside central IT, and upgrade to 3.2.1 or later; BleepingComputer reports that the latest stable release is 3.3.4, which is the preferable target [5]. As a precaution, any HFS instance running 3.0.0 through 3.2.0 that was reachable from the internet since the write-up should be treated as potentially compromised. That means rotating administrator credentials, reviewing configuration for unexpected `server_code` changes, checking for new files and processes on the host, and examining web logs for repeated login requests from a single source, which would be consistent with the state-recovery step [1][5]. Where an immediate upgrade is not possible, administrative access should be restricted to trusted networks or removed from public exposure.

Short-Term Mitigations

Vulnerability management teams should add a "new public technical detail" trigger to their prioritization process, so that the publication of a proof of concept, write-up, or demonstration for a flaw already patched upstream causes re-scoring and an accelerated remediation window. Teams should also expand discovery to software installed outside standard channels, using network scanning and endpoint inventory to locate file-sharing and utility servers. Detection engineering can add rules for the observed behavior of bursts of login requests followed by administrative actions from a new session. Hosts running such utilities should be segmented so that code execution on one does not provide a route to production systems.

Strategic Considerations

Over the longer term, organizations should plan for a higher cadence of credible, well-explained disclosures as AI-assisted research programs scale. This implies measuring remediation time against the interval between public explanation and exploitation, not only against the date a fix was released, and funding automation for patch deployment on low-risk assets. Procurement and third-party risk processes should ask whether suppliers and open-source dependencies have an update path that can be applied quickly. Software producers also have a part to play, and it is CSA's view that they carry a related responsibility. The root cause here belongs to a well-documented weakness class, the use of a non-cryptographic generator for security-sensitive values [1], and secure-by-design practice and code review that explicitly check for it could plausibly have caught the flaw before release.

CSA Resource Alignment

CSA's whitepaper *The Exploit-Before-Patch Gap* [6] examines how the assumption that public disclosure precedes exploitation is no longer reliable. CVE-2026-61500 is a less extreme instance of the same dynamic, since the patch preceded exploitation but the window between public explanation and attack was short. That paper's framing supports the recommendation to measure remediation against the time to exploitation rather than against the release of a fix.

The Collapsing Exploit Window [7] addresses how automated analysis tooling reduces the time attackers need to produce working exploits. It is relevant here because a public write-up with a demonstration video lowers that effort further, and it provides context for why teams should assume near-immediate scanning after publication.

For control mapping, the AI Controls Matrix (AICM) v1.1 [10] includes Threat and Vulnerability Management (TVM) and Application and Interface Security (AIS) domains that cover the patching and secure-development practices recommended above. Readers following CSA's Mythos-focused guidance for CISOs should also consult the Mythos CISO program page [11], which collects CSA material on preparing security programs for AI-accelerated vulnerability discovery.

References

- [1] CVE Program / VulnCheck CNA. "[CVE-2026-61500: Use of Cryptographically Weak Pseudo-Random Number Generator \(PRNG\) in rejetto hfs.](#)" NIST National Vulnerability Database, July 13, 2026.
- [2] Security Affairs. "[Anthropic Mythos Found A Bug in Rejetto HFS. Attackers Are Now Exploiting It.](#)" Security Affairs, October 5, 2026.
- [3] Cybersecurity News. "[Anthropic Mythos AI Finds Rejetto HFS Flaw That Lets Attackers Forge Admin Sessions and Execute Code.](#)" Cybersecurity News, October 7, 2026.
- [4] The Register. "[Anthropic's super bug-hunting model Mythos is hardcore good at math, as latest vuln under attack shows.](#)" The Register, October 3, 2026.
- [5] BleepingComputer. "[Rejetto HFS servers now actively scanned for critical RCE flaw.](#)" BleepingComputer, October 2026.
- [6] Cloud Security Alliance. "[The Exploit-Before-Patch Gap.](#)" CSA Labs, May 14, 2026.
- [7] Cloud Security Alliance. "[The Collapsing Exploit Window.](#)" CSA Labs, April 11, 2026.
- [8] CVEFeed. "[CVE-2024-23692: Rejetto HTTP File Server template injection, known exploited vulnerability.](#)" CVEFeed.
- [9] TechTimes. "[CVE-2026-61500: Anthropic Mythos Finds Rejetto HFS Flaw, Exploited Within One Day.](#)" TechTimes, October 5, 2026.
- [10] Cloud Security Alliance. "[AI Controls Matrix v1.1.](#)" CSA, June 22, 2026.
- [11] Cloud Security Alliance. "[Mythos CISO Program.](#)" CSA, 2026.
- [12] The Hacker News. "[Attackers Target Rejetto HFS Flaw That Anthropic's Mythos Found.](#)" The Hacker News, October 2026.
- [13] CVEFeed. "[CVE-2014-6287: Rejetto HTTP File Server remote command execution, known exploited vulnerability.](#)" CVEFeed, CISA KEV entry added March 25, 2022.