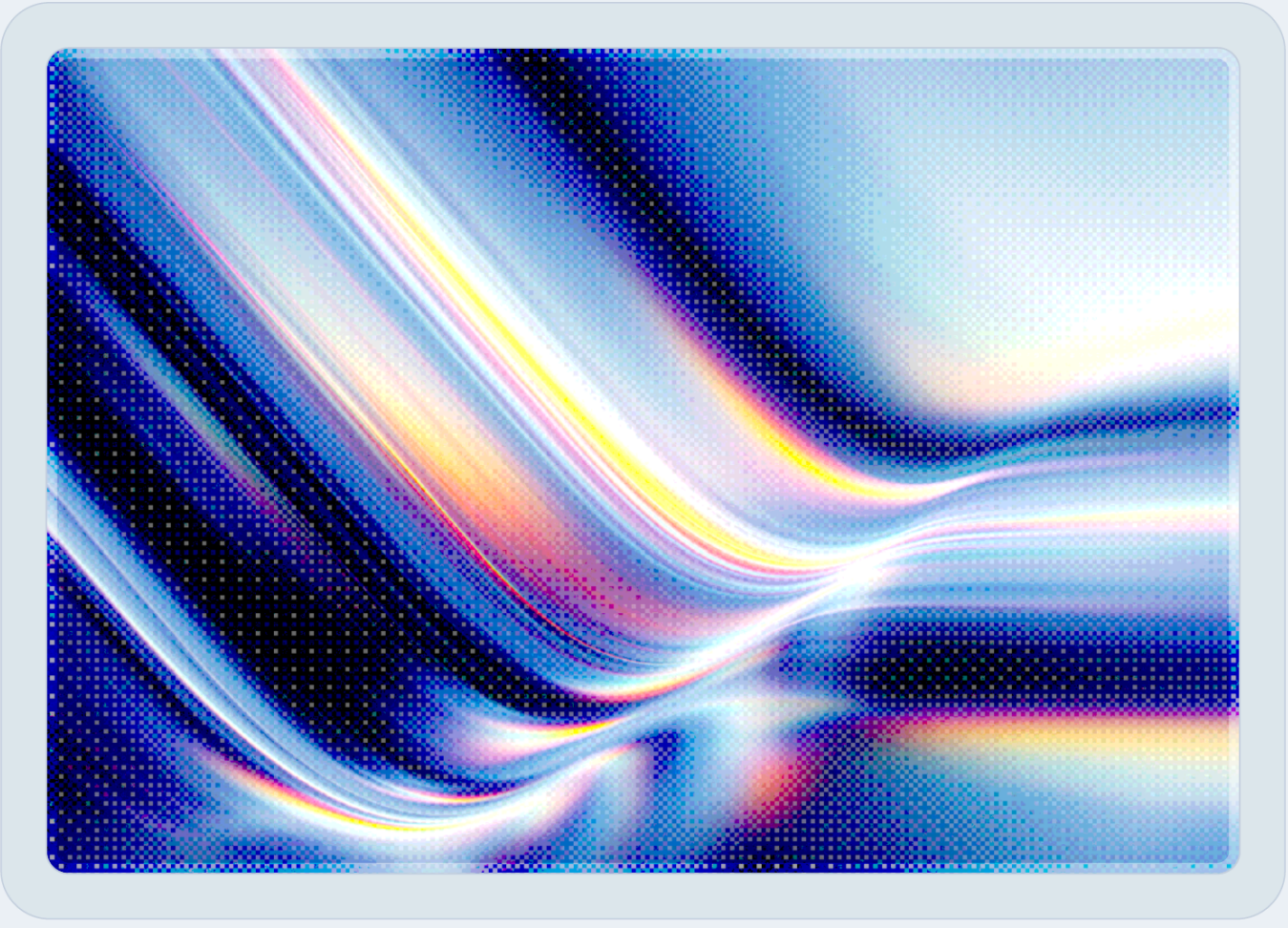
CSAI Foundation | Cloud Security Alliance

Open WebUI's Recurring Broken Access Control Problem

From Account Takeover to IDOR: A Cluster of Authorization Flaws in a Self-Hosted AI Platform

2026-07-10

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- Between November 2025 and June 2026, security researchers and the Open WebUI maintainers disclosed at least six distinct broken-access-control vulnerabilities in the platform, ranging from low-severity denial of service to high-severity account takeover and remote code execution — a pattern of recurring authorization failures rather than an isolated defect.
- CVE-2025-64496 (CVSS 7.3) chains a cross-site scripting flaw in the Direct Connections feature with unsandboxed Python tool execution, allowing a hostile model server to steal session tokens and, for users holding workspace.tools permission, achieve full remote code execution on the host [1][2][3].
- CVE-2026-44556 and CVE-2026-44557, both patched in version 0.9.0, show that newly added proxy and retrieval endpoints shipped without extending the per-model and per-collection permission checks that already protected equivalent legacy endpoints [4][5].
- CVE-2026-44564 and CVE-2026-54015 reveal a distinct but related failure mode: the application correctly verifies that a user belongs to a shared resource (a document room, a prompt they created) but never checks whether the specific action requested — write versus read, delete versus view — is one that user is entitled to perform [6][7].
- Open WebUI appears to be widely self-hosted directly by security-conscious teams and individual practitioners seeking to keep model interactions off third-party SaaS platforms, based on its GitHub community size and active development cadence. Because it is self-hosted rather than delivered as a managed service, operators — rather than a vendor — bear the compensating-control burden that a hosted product would otherwise absorb.
- One of the six vulnerabilities, CVE-2025-63681 (the task-cancellation IDOR, CVSS 2.1), has no released patch as of this writing; upgrading to the latest version does not close this cluster's exposure completely.

Background

Open WebUI is an open-source, self-hostable web interface for interacting with large language models, supporting both local runtimes such as Ollama and remote OpenAI-compatible APIs. Its appeal rests on a straightforward value proposition: organizations and individuals who want the conversational, tool-augmented experience of a commercial AI chat product without sending prompts, documents, or credentials to a third-party vendor can deploy Open WebUI on infrastructure they control. That proposition appears to have driven substantial adoption across enterprises experimenting with private LLM deployments, research labs, and individual developers running local models, based on the project's GitHub community size and active maintenance cadence, and it has pushed the project's feature set

to expand quickly: Direct Connections to arbitrary model servers, a workspace tools system that lets users author and execute custom Python functions, retrieval-augmented generation against uploaded knowledge bases, real-time collaborative document editing built on the Yjs CRDT framework, and versioned prompt libraries shared across teams.

Each of those features depends on the same underlying premise that any multi-tenant application depends on: that the system correctly determines, for every request, whether the requesting user is authorized to perform the specific action on the specific resource named in that request. Between the CVE-2025-64496 disclosure in November 2025 and the CVE-2026-54015 disclosure in June 2026, that premise broke down repeatedly across different subsystems of Open WebUI. The vulnerabilities were not concentrated in one module or introduced by one commit; they surfaced in the Direct Connections client, the OpenAI-compatible proxy router, the retrieval collection validator, the Socket.IO real-time collaboration layer, and the prompt version-history API. The common thread is not a single coding mistake but an architectural one: authorization logic implemented ad hoc, endpoint by endpoint, rather than enforced through a consistent, centrally reasoned policy layer. That pattern is the subject of this note.

Security Analysis

From Cross-Site Scripting to Remote Code Execution: CVE-2025-64496

The most severe vulnerability in this cluster centers on Open WebUI's Direct Connections feature, which lets a user point their client at any OpenAI-compatible model server, including ones outside the organization's control. Cato Networks researcher Vitaly Simonovich, who reported the flaw to Open WebUI's maintainers on October 8, 2025, found that the client trusted server-sent events of type "execute" and evaluated their contents using JavaScript's `new Function()` constructor — effectively treating output from an untrusted, attacker-controlled model server as executable code running inside the victim's authenticated browser session [1][2][3]. A hostile or compromised model server could therefore run arbitrary JavaScript in the user's session the moment that user sent it a message, and use that access to exfiltrate the authentication token Open WebUI stores in browser local storage.

Token theft alone would already constitute a serious account-takeover primitive, granting an attacker access to a victim's chat history, uploaded documents, and any API keys entered into the interface. But the second half of the vulnerability chain raised the stakes considerably. Open WebUI's workspace tools feature allows authorized users to define custom Python functions that the assistant can invoke, and the backend executes that Python code via `exec()` without sandboxing [1][3]. An attacker who hijacked the session of a user holding workspace.tools permission could create or modify a tool that ran arbitrary system commands, converting a client-side scripting flaw into full remote code execution on the server hosting the Open WebUI instance. The maintainers shipped an initial, incomplete fix on October 19, 2025, before a complete patch — middleware that blocks execution of server-sent events from Direct Connections servers — shipped in version 0.6.35 on November 5, 2025, with public disclosure and CVE assignment following on November 7, 2025 [1][2]. NIST's National Vulnerability Database later assessed the flaw at a CVSS score of 8, one full point above the CVSS 3.1 base score of 7.3 assigned in the original advisory [2][3].

A Pattern, Not an Incident: The 2026 Authorization Cluster

If CVE-2025-64496 stood alone, it might be read as a single serious but resolved defect. It did not stand alone. Over the following seven months, four additional broken-access-control vulnerabilities surfaced in unrelated parts of the Open WebUI codebase, each patched in the 0.9.x release line. The table below summarizes the cluster.

CVE	Affected Component	Core Flaw	CVSS	Affected Versions	Patched In
CVE-2026-44556	`/api/openai/responses` proxy endpoint	Forwards requests to any configured model without checking per-model access grants	7.1 (High)	≤ 0.8.12	0.9.0
CVE-2026-44557	Retrieval `_validate_collection_access` function	Incomplete allowlist lets any authenticated user query the system-wide knowledge-bases metadata collection	4.3 (Medium)	≤ 0.8.12	0.9.0
CVE-2026-44564	Socket.IO `ydoc:document:update` handler	Verifies room membership but not write permission, letting read-only users edit shared documents	5.4 (Medium)	≤ 0.8.12	0.9.0
CVE-2026-54015	Prompt version-history endpoints	Authorizes the prompt ID in the URL	6.4 (Medium)	≤ 0.9.5	0.9.6

CVE	Affected Component	Core Flaw	CVSS	Affected Versions	Patched In
		but not the caller-supplied history ID, enabling cross-prompt read and deletion			

CVE-2026-44556 illustrates the risk most directly: unlike Open WebUI's primary chat completion path, the newer `/responses` passthrough endpoint validated only that the caller held a valid session, not that the caller had been granted access to the specific model being requested, so any authenticated user could reach any model configured on the instance simply by naming its ID in a direct POST request [4]. CVE-2026-44557 followed the same pattern in the retrieval subsystem: a permission function written to check ownership of user-memory and file-scoped collections by matching name prefixes never accounted for the system's own internal metadata collection, so that collection passed through unchecked and exposed the names, identifiers, and descriptions of every knowledge base on the instance to any logged-in user [5]. Both flaws share a structural cause distinct from CVE-2025-64496's script-injection root: they are missing-authorization defects (CWE-862) introduced when a new endpoint or internal collection was added without updating the access-control logic that governed comparable, pre-existing resources.

CVE-2026-44564 and CVE-2026-54015 represent a related but analytically distinct failure mode, and one worth calling out separately because it recurs across collaborative and versioning features specifically. In both cases, the application performs a real authorization check — it confirms the user is a member of the document's collaboration room, or that the user has legitimate access to the parent prompt — and then treats that coarse-grained membership check as sufficient authorization for a much more specific action. In the Socket.IO case, a user with read-only access to a shared document can join its collaboration room (which only requires read access) and then emit `ydoc:document:update` events; because the handler checks room membership rather than write permission, the server applies the attacker's edits to the live in-memory document state, and if any collaborator with genuine write access subsequently saves, the tampering becomes persistent [6]. In the prompt-history case, the three affected endpoints correctly verify that the caller has access to the prompt named in the URL but then act on a separately supplied history-entry ID without confirming that entry actually belongs to that prompt, which lets an attacker who owns or has access to any single prompt read or delete another user's private prompt history simply by supplying the victim's history ID [7]. This is a textbook authorization-bypass-through-user-controlled-key defect (CWE-639) layered on top of an insecure direct object reference.

This is also not the first time Open WebUI's task-management logic has failed to check ownership. A lower-severity flaw disclosed alongside the account-takeover chain, CVE-2025-63681, found that the `/api/tasks/stop/` endpoint canceled in-flight LLM response generation tasks without verifying that the caller owned the task being stopped, letting any authenticated user disrupt another user's in-progress generations [8]. On its own this is a minor availability issue, rated CVSS 2.1, but it is evidence that the "authenticated equals authorized" shortcut has appeared

more than once in the codebase, independent of the more consequential 2026 cluster. As of this writing, CVE-2025-63681 remains unpatched: the GitHub Security Advisory lists no fixed version, so this particular gap cannot be closed by upgrading and instead requires either a compensating control or a maintainer fix [8].

Taken together, these six vulnerabilities do not point to a single defective module that can be patched once and forgotten. They point to an engineering practice in which authorization is decided locally, feature by feature, rather than enforced through one policy layer that every endpoint must pass through with a consistent question: not "is this a valid session" or "is this user in this room," but "is this specific user permitted to perform this specific action on this specific resource." Every feature Open WebUI has added since late 2025 — proxy routing, retrieval, real-time collaboration, prompt versioning — has needed its own bespoke answer to that question, and in five documented cases the answer implemented was incomplete.

Recommendations

Immediate Actions

Organizations running Open WebUI should confirm they are on version 0.9.6 or later, which incorporates fixes for CVE-2025-64496, CVE-2026-44556, CVE-2026-44557, CVE-2026-44564, and CVE-2026-54015. CVE-2025-63681, the unauthorized task-cancellation flaw, has no released patch as of this writing; administrators cannot remediate it through upgrading alone and should treat it as an accepted-risk item pending a maintainer fix. Instances on the 0.6.x branch remain exposed to the account-takeover and RCE chain in CVE-2025-64496 regardless of any later patches. Administrators should also audit which user accounts hold workspace.tools permission and revoke it from any account that does not require the ability to author executable functions, since that permission is the specific privilege that converts client-side token theft into server-side code execution. Teams that have enabled Direct Connections should review which external model servers are configured and disable connections to any server not fully controlled and trusted by the organization. Finally, administrators should review Socket.IO document-sharing and prompt-sharing settings for unintended cross-user exposure, since CVE-2026-44564 and CVE-2026-54015 both depend on legitimate, if limited, access having already been granted to the attacking account.

Short-Term Mitigations

Because the vulnerability class documented here is a pattern rather than a single defect, patching the currently known CVEs does not retire the risk. Instances should sit behind an authenticating reverse proxy or SSO gateway that enforces network-level access control independent of Open WebUI's own authentication, so that a future authorization bug in the application layer does not translate directly into external exposure. The administrative interface and API should never be exposed directly to the public internet; where remote access is required, it should be mediated through a VPN or zero-trust network access gateway. Security teams should enable logging on sensitive endpoints — the OpenAI proxy router, task management, and Socket.IO collaboration events — and watch for authenticated users reaching resources outside their expected scope. Because Open WebUI's maintainers have shipped several access-control patches over a short period, organizations running the platform should subscribe to

its GitHub security advisories directly and fold Open WebUI into the same vulnerability-scanning and patch-management cadence applied to any other network-facing application, rather than treating it as low-stakes internal tooling exempt from routine review.

Strategic Considerations

The broader lesson from this cluster is that self-hosted AI platforms deserve the same identity and access management rigor that organizations already apply to enterprise IAM systems, not the lighter governance often extended to internal developer tools. Open WebUI's repeated pattern — checking that a request is authenticated, or that a user belongs to a shared resource, without checking that the specific requested action is one that user is entitled to perform — is precisely the coarse-grained, binary trust model that Zero Trust architecture principles were designed to replace with fine-grained, context-aware, per-action authorization decisions. Organizations evaluating or continuing to operate self-hosted AI front-ends should weigh a vendor's or project's rate of access-control CVEs and time-to-patch as a maturity signal alongside its feature velocity, and should favor architectures — whether commercial or open source — that centralize authorization through a single policy-enforcement layer rather than distributing bespoke checks across every new endpoint a fast-moving feature roadmap introduces.

CSA Resource Alignment

The authorization failures documented in this note map directly onto guidance CSA has already published on identity-centric access control. CSA's Zero Trust Principles and Guidance for Identity and Access Management (IAM) describes exactly the shift Open WebUI's architecture has not yet made: away from binary, membership-based trust decisions and toward risk-based, contextual authorization enforced consistently through defined policy enforcement and policy decision points. CSA's Context-Based Access Control for Zero Trust develops that same shift in more operational detail, describing how contextual signals, not just identity or room membership, should determine whether a specific action on a specific resource is authorized; the CVE-2026-44564 Socket.IO handler, which checks only room membership before permitting a document edit, is a direct illustration of the coarse-grained model that guidance argues against. Every vulnerability in this cluster — from the /responses proxy that checked only session validity to the Socket.IO handler that checked only room membership — is a case study in what that guidance warns against: coarse trust signals substituting for fine-grained, per-action authorization.

CSA's survey report, Identity and Access Gaps in the Age of Autonomous AI, documents a parallel dynamic at the organizational level: 74% of surveyed security and IT professionals agree that AI-related systems often receive more access than necessary, and the report finds that confidence in access controls consistently outpaces their actual maturity. While that survey focuses on how enterprises govern AI agent identities rather than on application-level authorization bugs, the underlying failure is the same one this note documents inside Open WebUI's own codebase: access-control enforcement has not kept pace with the speed at which new AI-adjacent capabilities are shipped and adopted. CSA's Confronting Shadow Access Risks: Considerations for Zero Trust and Artificial Intelligence Deployments frames this same gap as a shadow access problem: unauthorized or undetected access paths that persist precisely because no single control point evaluates every request against what its specific action actually

requires. The IDOR-style flaws in this cluster, in which a user reaches a prompt-history entry or a knowledge-base collection never meant to be visible to them, are shadow access in exactly that sense, arising not from a compromised credential but from an authorization check that never verified the scope of what a valid credential is entitled to reach. Organizations assessing their exposure to the platform-level risks described here should use both that survey's findings and the shadow access framework to evaluate whether their broader AI governance program enforces identity-centric controls or relies on procedural stopgaps.

Finally, organizations building or auditing a formal security assessment program for self-hosted AI platforms should use the AI Controls Matrix (AICM) v1.1's identity and access management domain as the control baseline. AICM's IAM-domain controls address authentication and authorization requirements for AI system components and APIs, and applying them systematically, rather than trusting a vendor's own endpoint-by-endpoint judgment, gives organizations a structured way to test for gaps like CVE-2026-44556 before they reach production, regardless of whether the underlying application is Open WebUI or any comparable self-hosted AI tool [9].

References

[1] Infosecurity Magazine. "High-Severity Flaw in Open WebUI Affects AI Connections." Infosecurity Magazine, November 2025.

[2] SC World. "Open WebUI Account Takeover Flaw Could Lead to Remote Code Execution." SC World, November 2025.

[3] Cato Networks. "Cato CTRL Threat Research: Vulnerability Discovered in Open WebUI Enables Account Takeover and Remote Code Execution (CVE-2025-64496)." Cato Networks, November 2025.

[4] GitHub Advisory Database. "GHSA-hp5m-24vp-vq2q: Open WebUI's Responses Passthrough Endpoint Lacks Access Control Authorization (CVE-2026-44556)." GitHub Security Advisories, May 2026.

[5] GitHub Advisory Database. "GHSA-6c2x-gcp3-gp73: Open WebUI Vulnerable to Global Knowledge Base Enumeration via knowledge-bases Meta-Collection (CVE-2026-44557)." GitHub Security Advisories, May 2026.

[6] GitHub Security Advisories (open-webui/open-webui). "GHSA-vrfh-rj4q-rmhr: Read-Only Users Can Modify Collaborative Documents via Socket.IO (CVE-2026-44564)." GitHub, May 2026.

[7] GitHub Advisory Database. "GHSA-4r4w-2wgp-w7cj: Open WebUI Prompt History IDOR — Unbound history_id Allows Cross-Prompt Read and Deletion (CVE-2026-54015)." GitHub Security Advisories, June 2026.

[8] GitHub Advisory Database. "GHSA-frv8-gffc-37px: open-webui Is Vulnerable to Incorrect Access Control (CVE-2025-63681)." GitHub Security Advisories, November 2025.

[9] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2026.