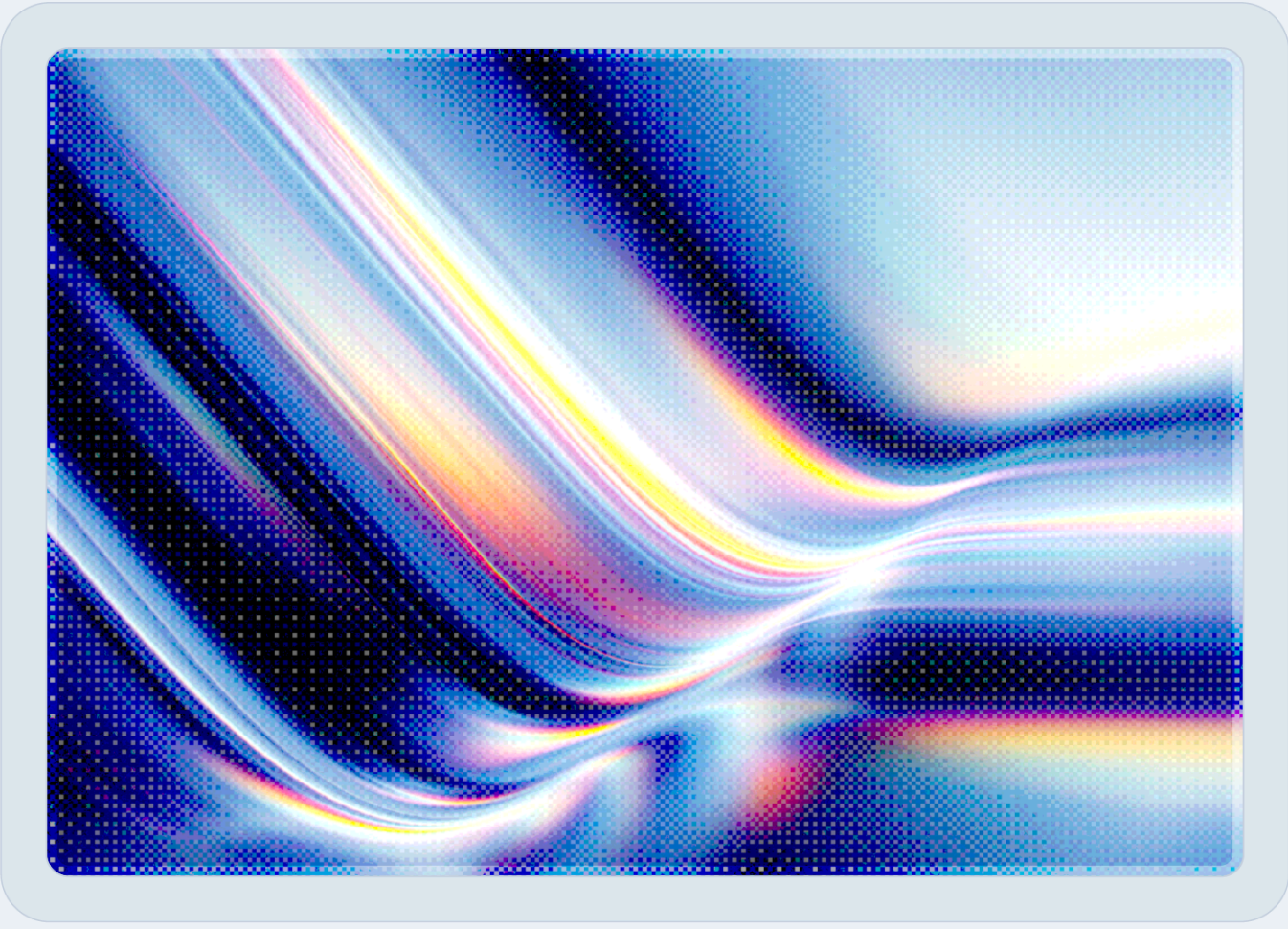
CSAI Foundation | Cloud Security Alliance

ChromaToast: Unauthenticated RCE in AI Vector Databases

CVE-2026-45829 and the Exposed AI Data Layer

2026-05-20

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

The following findings are drawn from technical analysis of CVE-2026-45829 and current deployment data for ChromaDB in production AI environments.

- CVE-2026-45829, nicknamed "ChromaToast" by the researchers at HiddenLayer who discovered it [6], is a pre-authentication remote code execution vulnerability in ChromaDB's Python FastAPI server. An unauthenticated attacker can supply a malicious HuggingFace model reference that the server executes before any authentication check occurs, yielding full server takeover without credentials [1][2].
- The vulnerability affects ChromaDB versions 1.0.0 through 1.5.8. The Rust-based ChromaDB server implementation is not affected. Organizations running the Python server should isolate it from untrusted networks immediately and verify patch availability in any version released after 1.5.8 [3].
- ChromaDB ships with authentication disabled by default. Security researchers have identified over 1,000 internet-accessible ChromaDB deployments, with many exposed instances appearing to contain real production data rather than test content [4][5].
- Vector databases represent a high-value target because they concentrate proprietary embeddings, RAG knowledge bases, and AI application context. Successful exploitation enables intellectual property theft, exfiltration of documents used to build enterprise AI systems, and covert manipulation of AI outputs through embedding poisoning.
- The combination of sensitive data holdings and historically weak access controls means vector database infrastructure warrants the same security standards applied to relational databases and object stores: network isolation, access controls, continuous monitoring, and structured vulnerability management.

Background

ChromaDB is an open-source, AI-native vector database designed to store and retrieve high-dimensional vector embeddings for use in retrieval-augmented generation (RAG) pipelines, semantic search systems, and other AI workloads requiring similarity-based data lookup. First released approximately 2022 by Chroma AI [17], the project has grown rapidly alongside enterprise adoption of

large language model (LLM) applications: as of mid-2026, HiddenLayer's security research team reported approximately 13 million monthly pip downloads, making ChromaDB one of the most widely deployed open-source vector databases in modern AI application stacks [6].

The database operates as a server exposing an HTTP API, typically implemented through a Python FastAPI layer. Developers connect to this API from LLM orchestration frameworks such as LangChain, LlamaIndex, and custom RAG pipelines to insert embeddings derived from documents, query embeddings to retrieve semantically relevant context, and update knowledge bases as source content changes. In production deployments, the knowledge base stored in ChromaDB commonly contains sensitive organizational content: internal policy documents, customer support histories, proprietary code, financial records, and other materials whose embeddings are used to ground AI responses in enterprise-specific context.

Despite this sensitivity, ChromaDB has historically offered authentication as an opt-in feature rather than a default. The product documentation describes token-based and basic authentication configurations, but neither is enabled in the default installation. This design choice has contributed to a pattern of organizations deploying ChromaDB in network positions where it is reachable by internal or external actors without any credential requirement. Security researchers at UpGuard documented this exposure surface, finding over 1,170 accessible instances containing what appeared to be real organizational data — suggesting that the default-unauthenticated posture has translated directly into production risk at scale [4].

CVE-2026-45829 adds a qualitatively different threat on top of this structural exposure. Even in deployments where authentication is enabled, the vulnerability allows an attacker to achieve remote code execution before the authentication mechanism is consulted, negating the protection that authentication would otherwise provide. The vulnerability was discovered by researchers at HiddenLayer and reported on November 28, 2025, with CVE-2026-45829 formally assigned in 2026. Technical details were disclosed publicly through the ChromaDB GitHub repository and coordinated security channels [1][3].

Security Analysis

Vulnerability Mechanics

The root cause of CVE-2026-45829, as documented in the ChromaDB security advisory [11], is an ordering failure in ChromaDB's Python FastAPI server: the server accepts and acts on client-supplied model identifiers during request processing before it evaluates whether the requesting client is

authenticated [1][2]. Specifically, an attacker can submit an HTTP request that references a malicious HuggingFace model identifier. The server instantiates the referenced model — pulling and executing attacker-controlled code from a remote model repository — prior to any authentication gate. By the time authentication would be checked, the payload has already executed.

This pattern — where a trusted internal component executes attacker-controlled input before authentication is evaluated — resembles what security researchers call the "confused deputy problem": a trusted system component (the model loader) is weaponized by untrusted input because the trust boundary was established at the wrong point in the processing chain. The impact is as severe as RCE can be: an attacker who triggers execution through this path gains the operating system privileges of the ChromaDB process, access to all files readable by that process (including mounted secrets, API keys stored in environment variables, and the full embedding database), and the ability to establish persistence or pivot to other services reachable from the host.

The Rust-based ChromaDB server implementation does not share this vulnerability. The Python FastAPI server is the affected component, and organizations that have migrated to the Rust server — or that have never deployed the Python server in a network-accessible configuration — are not exposed to CVE-2026-45829 via this path [2][3].

Exposure and Deployment Context

The scale of exposure matters because ChromaDB is not typically deployed as an isolated research tool. In production AI systems, it sits at the intersection of multiple high-trust data flows: documents from enterprise content repositories are embedded and written to ChromaDB, embeddings are queried at query time to assemble context passed to LLM APIs, and the quality and integrity of retrieved embeddings directly shapes what AI applications tell users. This architectural centrality means that a compromised ChromaDB instance can affect AI behavior across every application that draws from it.

HiddenLayer's own survey found that approximately 73% of internet-exposed ChromaDB instances ran version 1.0.0 or later — the version range that carries the vulnerability [6]. Earlier research from UpGuard identified over 1,170 publicly accessible ChromaDB deployments, with many appearing to contain real production data rather than test content [4]. These figures indicate that misconfiguration and default-open posture have created a material attack surface, and that CVE-2026-45829 is exploitable against a substantial fraction of that surface without any credential guessing or social engineering.

Data and AI System Integrity Risks

Exploitation of CVE-2026-45829 carries risks beyond the immediate server compromise. Because ChromaDB holds embeddings used by downstream AI systems, an attacker with write access — either through the pre-auth RCE or through simple unauthenticated API access to instances lacking authentication — can manipulate the knowledge base that AI systems rely on. Embedding poisoning, in which malicious or subtly altered vectors are inserted into the database, can cause RAG-enabled AI systems to retrieve attacker-controlled content as if it were authoritative organizational knowledge [14]. This technique is particularly difficult to detect because the AI system's outputs may appear plausible while subtly reflecting injected content, and because embeddings are not human-readable in their raw form.

The data extraction risk is equally significant. ChromaDB deployments in enterprise environments typically hold embeddings of documents that were never intended for external access. While embeddings are often assumed to be non-reversible representations, research has demonstrated that embedding inversion techniques can recover substantial amounts of the original text from stored vectors under certain conditions [7]. An attacker who exfiltrates a ChromaDB database via the RCE path may be able to partially reconstruct the source documents, turning an AI infrastructure compromise into a document exfiltration incident.

Connection to Systemic AI Infrastructure Risk

CVE-2026-45829 is not an isolated incident. It follows a pattern visible in several recent AI infrastructure vulnerabilities, including CVE-2026-42208 in LiteLLM — a pre-authentication SQL injection flaw exploited within 36 hours of public disclosure [13] — and prior vulnerabilities in AI orchestration frameworks. The pattern is consistent: components built to accelerate AI application development are deployed without the security hardening applied to comparable data infrastructure, and the cases reviewed suggest attackers are moving increasingly quickly to exploit these gaps after disclosure. The AI application stack — embedding pipelines, vector stores, orchestration frameworks, and model servers — is now sufficiently mature and widespread to attract the same adversarial attention historically directed at web servers and database management systems.

Recommendations

Immediate Actions

Organizations running ChromaDB in any environment where the Python FastAPI server is network-accessible should take the following steps without delay.

First, determine whether CVE-2026-45829 has been patched in the currently deployed version of ChromaDB. The vulnerability affects versions 1.0.0 through 1.5.8; any version released after 1.5.8 should be reviewed against the ChromaDB changelog and GitHub security advisories to confirm patch inclusion [3]. If no confirmed patch is available for the deployed version, upgrade should not be the only action taken — the mitigations below must be applied regardless.

Second, migrate to or verify use of the Rust-based ChromaDB server. The Rust server implementation (`chroma run`) does not carry CVE-2026-45829 and is the recommended deployment path for production workloads going forward. Organizations still on the Python FastAPI server should treat migration as an urgent operational priority, not a backlog item [2].

Third, enforce network isolation immediately. ChromaDB API ports should be reachable only from known, trusted application hosts. Firewall rules or security group policies should block all access to ChromaDB from untrusted network segments, including access from general internet egress. If ChromaDB must be accessible across a broader network segment, terminate all external access at a reverse proxy that enforces authentication before any request reaches the ChromaDB API.

Short-Term Mitigations

Beyond the immediate response, organizations should address the underlying authentication gap that makes unauthenticated access possible even in the absence of CVE-2026-45829. Enable ChromaDB's built-in token-based or basic authentication if the Python server remains in use during a migration window [12]. Authentication without transport encryption is insufficient: TLS must be enforced on all ChromaDB network connections, either at the service level or through a TLS-terminating proxy in front of the database [8].

Access to the ChromaDB API should follow least-privilege principles. Application service accounts used to read embeddings should not have write access; administrative access for embedding ingestion pipelines should be scoped to the specific collections they manage. Audit logging should capture all API

calls to the ChromaDB server, with log data shipped to a centralized SIEM where anomalous patterns — high-volume reads, unexpected collection modifications, access from unusual source IPs — can be detected and alerted on.

Organizations should also inventory the content of existing ChromaDB collections to understand what data is at risk. If a deployment was previously internet-accessible without authentication, it should be treated as potentially compromised: embeddings may have been exfiltrated, and the possibility of embedding poisoning should be investigated by reviewing recent collection modifications and, where feasible, rebuilding collections from verified source documents.

Strategic Considerations

At the strategic level, this vulnerability should prompt a broader review of how AI data infrastructure is governed within the organization. Vector databases, embedding pipelines, and model serving endpoints collectively handle data of the same sensitivity as production relational databases, but they have frequently been deployed without the access review, change management, and monitoring processes applied to traditional data infrastructure. Closing this gap requires organizational recognition that AI infrastructure is production data infrastructure, not experimental tooling.

Security teams should extend their asset inventory and vulnerability management programs to include vector database deployments, AI orchestration components, and model servers. Configuration audits should verify that authentication is enabled and correctly configured for all AI data layer components, not assumed on the basis of documentation or initial setup. Penetration testing of AI application stacks should explicitly include the vector database layer, testing for unauthenticated access, embedding manipulation, and privilege escalation through AI infrastructure paths.

Finally, organizations building AI systems should evaluate whether their architecture creates unnecessary exposure of the vector database layer. In many cases, applications can be designed so that ChromaDB is accessible only from a single backend service with a narrow, well-defined interface, rather than from multiple application tiers or developer workstations. Reducing the network surface area of the vector database through architectural decisions is more durable than relying on configuration controls that can be misconfigured or bypassed.

CSA Resource Alignment

CVE-2026-45829 maps to threat categories and control domains addressed across several CSA frameworks and publications.

Within CSA's **MAESTRO** agentic AI threat modeling framework [15], this vulnerability is most directly relevant to Layer 3 (Data and Storage Operations), which addresses the security of the data retrieval layer in AI systems, including vector databases and embedding stores. MAESTRO's treatment of adversarial data injection and retrieval manipulation is directly applicable to the embedding poisoning risk that unauthenticated ChromaDB access enables. Layer 5 (Deployment and Operations Infrastructure) addresses the broader concern of AI components deployed with insufficient network isolation and access controls — the systemic configuration posture that made this vulnerability exploitable at scale.

The **AI Controls Matrix (AICM)** [16], CSA's superset of the Cloud Controls Matrix designed for AI environments, includes control domains governing data security, access management, and vulnerability management that are directly implicated by this advisory. In particular, controls addressing data-at-rest and data-in-transit encryption, least-privilege access provisioning, and network segmentation for data services apply to ChromaDB deployments and should be verified against current configurations.

CSA's **"The AI Vulnerability Storm"** provides relevant broader context, documenting the acceleration of AI infrastructure vulnerability discovery and exploitation timelines [10]. The pattern observed in CVE-2026-45829 — a widely deployed AI infrastructure component with a critical unauthenticated access flaw — is consistent with the systemic exposure trends described in that research. Similarly, **"Data Security Within AI Environments"** addresses the RAG architecture security model, including the risks posed by an insecure retrieval layer and the importance of treating the embedding store as a high-sensitivity data asset subject to enterprise data governance [9].

Organizations implementing CSA's **Zero Trust guidance** should apply zero-trust network access principles to their AI data infrastructure. Implicit trust in any component of the AI application stack — including the vector database — is inconsistent with zero-trust architecture, and access to ChromaDB should be treated as a privileged data operation requiring explicit authentication and authorization at every request, regardless of network position.

References

[1] Hadrian. "CVE-2026-45829 — ChromaDB Python Server Hands You RCE Before It Asks Who You Are." Hadrian Security Research, 2026.

[2] BleepingComputer. "Max-Severity Flaw in ChromaDB for AI Apps Allows Server Hijacking." BleepingComputer, 2026.

[3] NIST National Vulnerability Database. "CVE-2026-45829 Detail." NVD, 2026.

[4] UpGuard. "Open Chroma Databases: The AI Attack Surface You Didn't Know You Had." UpGuard Research, 2025.

[5] SecurityWeek. "Unpatched ChromaDB Vulnerability Can Lead to Server Takeover." SecurityWeek, 2026.

[6] HiddenLayer. "ChromaToast: Served Pre-Auth." HiddenLayer AI Security Research, 2026.

[7] Huang, Tianhao et al. "Transferable Embedding Inversion Attack: Uncovering Privacy Risks in Text Embeddings without Model Queries." arXiv, 2024.

[8] Amikos Tech. "Secure Your ChromaDB Instance: Authentication." Amikos Tech Engineering Blog, 2024.

[9] Cloud Security Alliance. "Data Security Within AI Environments." CSA AI Safety Initiative, 2025.

[10] Cloud Security Alliance. "The AI Vulnerability Storm." CSA AI Safety Initiative, 2025.

[11] GitHub Advisory Database. "GHSA-f4j7-r4q5-qw2c: ChromaDB Python Server Pre-Auth RCE." GitHub Security Advisories, 2026.

[12] ChromaDB. "ChromaDB Security Cookbook." ChromaDB Documentation, 2025.

[13] The Hacker News. "LiteLLM CVE-2026-42208: SQL Injection Vulnerability Exploited Within 36 Hours." The Hacker News, 2026.

[14] Su, Yixuan et al. "Towards More Robust Retrieval-Augmented Generation: Evaluating RAG Under Adversarial Poisoning Attacks." arXiv, 2024.

[15] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA AI Safety Initiative, 2025.

[16] Cloud Security Alliance. "AI Controls Matrix (AICM)." CSA, 2025.

[17] Chroma AI. "chroma-core/chroma." GitHub, 2022.