INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-20 06:44 UTC

ChromaDB Authentication Bypass Enables Unauthenticated RCE via Hugging Face Model Loading (CVE-2026-45829)

CVE VULNERABILITY | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0196
Type	CVE Vulnerability
CVE ID	CVE-2026-45829
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.0014 (34th percentile)
Affected Products	ChromaDB Python FastAPI server, versions 1.0.0 through 1.5.8 (PyPI package); Rust frontend not affected
Published	2026-05-19T18:25:49
Discovery Source	Rss

Executive Summary

A critical unauthenticated remote code execution vulnerability (CVE-2026-45829) has been identified in ChromaDB's Python FastAPI server, affecting versions 1.0.0 through 1.5.8. Any attacker with network access to an exposed ChromaDB instance can execute arbitrary code on the server before authentication is ever checked, with no credentials required. Organizations using ChromaDB in AI pipelines or vector search infrastructure face complete server compromise; no confirmed vendor patch exists as of 2026-03-04.

Technical Analysis

CVE-2026-45829 affects ChromaDB Python FastAPI server versions 1.0.0 through 1.5.8 (PyPI). The Rust frontend is not affected. The vulnerability chain involves four weaknesses: improper authentication (CWE-287), incorrect authorization (CWE-863), inclusion of functionality from an untrusted control sphere (CWE-829), and improper control of code generation (CWE-94). The attack path allows a remote, unauthenticated attacker to force the ChromaDB server to load and execute a malicious model from Hugging Face before any authentication check is performed, resulting in arbitrary code execution on the host. CVSS base score is 9.5 pending NVD confirmation. EPSS score is 0.00139 (33rd percentile), indicating low observed exploitation probability at time of scoring, though this metric lags real-world activity. The CVE is not listed in CISA KEV as of

2026-03-04. No confirmed patch or vendor advisory has been issued. The 2026 CVE year and absence from NVD at time of writing warrant independent verification at https://nvd.nist.gov/vuln/detail/CVE-2026-45829 and https://www.cve.org/CVERecord?id=CVE-2026-45829 before acting on CVSS or patch status claims. Relevant MITRE ATT&CK techniques: T1190 (Exploit Public-Facing Application), T1203 (Exploitation for Client Execution), T1105 (Ingress Tool Transfer), T1059/T1059.006 (Command and Scripting Interpreter: Python), T1071 (Application Layer Protocol), T1195.001/T1195.002 (Supply Chain Compromise). Approximately 73% of internet-exposed instances are reported to remain on a vulnerable version [T3 source, indicative]; this figure should be treated as directional, not authoritative.

Action Checklist

1. Step 1: Containment, Immediately restrict network access to all ChromaDB Python FastAPI server instances (versions 1.0.0-1.5.8). Block inbound connections to ChromaDB's default port (8000/TCP) at the perimeter firewall or security group level. If the service is internet-facing with no WAF or IPS, take it offline until remediation is confirmed. Do not rely on application-layer controls given the pre-authentication nature of the flaw.
2. Step 2: Detection, Audit all ChromaDB deployments for version (pip show chromadb). Review FastAPI access logs for unexpected POST or GET requests to model-loading endpoints (e.g., /api/v1/collections or any endpoint invoking Hugging Face model references) originating from external IPs. Look for anomalous outbound connections to huggingface.co or hf.co from ChromaDB host processes. Check host process trees for Python child processes spawned by the ChromaDB service outside of normal operation. Correlate with T1190 and T1059.006 detection logic in your SIEM.
3. Step 3: Eradication, No confirmed vendor patch exists as of 2026-03-04. Monitor the ChromaDB GitHub repository (https://github.com/chroma-core/chroma) and PyPI package page for a patched release. If a patch is released, upgrade immediately via pip install --upgrade chromadb and verify the installed version. Until a patch is available, consider disabling Hugging Face model-loading functionality at the network layer (block outbound to huggingface.co/hf.co from ChromaDB hosts) and enforce strict network segmentation so ChromaDB is not reachable from untrusted networks.
4. Step 4: Recovery, After applying any released patch, verify the installed version with pip show chromadb and confirm it falls outside the 1.0.0-1.5.8 range. Re-audit FastAPI access logs for evidence of prior exploitation (unexpected model loads, unusual outbound connections, new files written to disk by the ChromaDB process). Rotate any credentials or API keys accessible from the ChromaDB host. Restore service only after confirming no indicators of prior compromise are present.
5. Step 5: Post-Incident, Review your AI/ML pipeline asset inventory to ensure all components with external model-loading capabilities are subject to the same network segmentation and authentication enforcement applied to other production services. Evaluate whether Hugging Face model ingestion should be restricted to a controlled, air-gapped process rather than live server-side fetching. Add ChromaDB and similar vector database components to your vulnerability management scope with automated version monitoring.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to CISO and legal/privacy counsel immediately if FastAPI access logs show any external-origin requests to model-loading endpoints during the exposure window, if the ChromaDB host had access to PII, PHI, or regulated data (triggering breach notification timelines under GDPR Art. 33, HIPAA 45 CFR §164.410, or applicable state laws), or if the responding team lacks the capability to perform memory forensics or Python runtime analysis on a potentially compromised host.
Recovery Notes	After patching, maintain elevated monitoring of the ChromaDB host for a minimum of 30 days: watch for Python child processes spawned by uvicorn (Sysmon EID 1 or auditd execve), outbound connections to huggingface.co or non-standard IPs from the ChromaDB process, and unexpected modifications to Python site-packages (SI-7). Verify that all credentials and API keys rotated during recovery have been invalidated in their respective platforms (Hugging Face token revocation at hf.co/settings/tokens, cloud IAM key deletion confirmed via provider console). Do not restore internet-facing exposure of the ChromaDB service until network-layer authentication enforcement (reverse proxy with mandatory auth, e.g., nginx + OAuth2 proxy) is in place, given the pre-authentication nature of CVE-2026-45829.
Forensic Artifacts	Uvicorn/FastAPI stdout logs (default: journalctl -u chromadb or /var/log/chromadb/access.log) — query for POST/GET requests to /api/v1/ endpoints with Hugging Face model references or base64-encoded payloads in request bodies, originating from non-RFC1918 source IPs, which represent the primary exploitation vector for CVE-2026-45829 | Hugging Face model cache directory (~/.cache/huggingface/hub/) — file creation timestamps for any model files (*.bin, *.safetensors, config.json, tokenizer.json) downloaded after initial deployment without authorized model update activity; attacker-controlled models loaded via the unauthenticated RCE path will appear here | Python process audit trail — Linux auditd EXECVE records or Windows Sysmon EventID 1 logs filtered on parent process matching uvicorn/chromadb with child processes python3, sh, bash, curl, or wget, which indicate successful RCE and post-exploitation command execution per MITRE T1059.006 | Host network connection logs — Sysmon EventID 3 (Network Connection) or Linux auditd CONNECT syscall records for the ChromaDB process PID, specifically outbound connections to huggingface.co (34.105.x.x), hf.co, or any non-organizational external IP initiated by the uvicorn worker process, evidencing the model-load exfiltration/execution chain | Python site-packages integrity — directory listing with mtimes for /usr/local/lib/python*/dist-packages/chromadb/ and all installed dependencies; any files modified after the ChromaDB installation timestamp and not corresponding to a pip upgrade event indicate attacker modification of the Python runtime for persistence (MITRE T1546.005 — Trap)

Per-Action IR Details

Step 1: Containment — Immediately restrict network access to all ChromaDB Python FastAPI server instances (versions 1.0.0–1.5.8). Block inbound connections to ChromaDB's default port (8000/TCP) at the perimeter firewall or security group level. If the service is internet-facing with no WAF or IPS, take it offline until remediation is confirmed. Do not rely on application-layer controls given the pre-authentication nature of the flaw.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 12.2 — Establish and Maintain a Secure Network Architecture (IG2/IG3)

Compensating: On Linux hosts: run `sudo iptables -I INPUT -p tcp --dport 8000 -j DROP` immediately, then `sudo iptables-save > /etc/iptables/rules.v4` to persist. On Windows: `netsh advfirewall firewall add rule name='Block ChromaDB 8000' protocol=TCP dir=in localport=8000 action=block`. For cloud-deployed instances (AWS/GCP/Azure), remove or restrict the inbound security group / VPC firewall rule for port 8000 to 0.0.0.0/0 immediately. A 2-person

team can execute all three in under 10 minutes without any SIEM or EDR tooling.

Evidence: Before blocking, capture a netstat snapshot to document all active inbound connections to port 8000/TCP: `ss -tnp sport = :8000 > /tmp/chromadb_connections_$(date +%Y%m%d%H%M%S).txt`. Also snapshot the running process tree of the ChromaDB FastAPI service: `ps auxf | grep -A 20 'uvicorn\|chromadb' > /tmp/chromadb_proctree.txt`. These preserve evidence of any active exploitation sessions or spawned child processes before the network cut.

Step 2: Detection — Audit all ChromaDB deployments for version (pip show chromadb). Review FastAPI access logs for unexpected POST or GET requests to model-loading endpoints (e.g., /api/v1/collections or any endpoint invoking Hugging Face model references) originating from external IPs. Look for anomalous outbound connections to huggingface.co or hf.co from ChromaDB host processes. Check host process trees for Python child processes spawned by the ChromaDB service outside of normal operation. Correlate with T1190 and T1059.006 detection logic in your SIEM.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-3 (Content of Audit Records), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Version audit: `pip show chromadb | grep Version` on each host; automate across fleet with `pdsh -g all_hosts 'pip show chromadb | grep Version'` or equivalent SSH loop. FastAPI log review — grep the uvicorn access log (default path: `~/.local/share/chroma/chroma.log` or stdout redirected to `/var/log/chromadb/access.log`): `grep -E '(POST|GET).*/api/v1.*(embed|model|load)' /var/log/chromadb/access.log | grep -v '10\.|172\.16\.|192\.168\.'` to isolate external-origin model-load requests. Outbound connection monitoring without EDR: deploy Sysmon with EventID 3 (Network Connection) filtering on the chromadb/uvicorn PID, then query: `Get-WinEvent -LogName 'Microsoft-Windows-Sysmon/Operational' | Where-Object {$_.Id -eq 3 -and $_.Message -match 'huggingface\.co|hf\.co'}`. On Linux: `auditctl -a always,exit -F arch=b64 -S connect -F ppid=$(pgrep -f uvicorn) -k chromadb_outbound` then review via `ausearch -k chromadb_outbound`. For child process detection (T1059.006), use Sysmon EventID 1 filtering parent process name matching 'uvicorn' or 'chromadb' with child process 'python' or 'sh' or 'bash'.

Evidence: Collect before analysis: (1) Full uvicorn/FastAPI stdout and stderr logs covering the exposure window — `journalctl -u chromadb --since '7 days ago' > /tmp/chromadb_journal.txt`. (2) Outbound DNS query logs from the host resolving huggingface.co or hf.co — check `/var/log/syslog` or `journalctl` for systemd-resolved entries, or run `tcpdump -nn -r /path/to/existing/pcap 'host huggingface.co'` if packet capture was active. (3) Python package cache and model download artifacts: `find ~/.cache/huggingface -newer /tmp/chromadb_install_timestamp -ls` — any model files downloaded after initial deployment without authorized model updates are exploitation artifacts specific to the Hugging Face model-loading vector of CVE-2026-45829.

Step 3: Eradication — No confirmed vendor patch exists as of 2026-03-04. Monitor the ChromaDB GitHub repository (https://github.com/chroma-core/chroma) and PyPI package page for a patched release. If a patch is released, upgrade immediately via pip install --upgrade chromadb and verify the installed version. Until a patch is available, consider disabling Hugging Face model-loading functionality at the network layer (block outbound to huggingface.co/hf.co from ChromaDB hosts) and enforce strict network segmentation so ChromaDB is not reachable from untrusted networks.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST SI-3 (Malicious Code Protection), NIST CM-6 (Configuration Settings), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Until a vendor patch exists, implement a two-layer network eradication control: (1) Outbound DNS/IP block for Hugging Face model-loading — on Linux: `iptables -I OUTPUT -d 34.105.0.0/16 -j DROP && iptables -I

OUTPUT -d 13.107.0.0/16 -j DROP` (verify current HF IP ranges via `dig huggingface.co` and `dig hf.co` first); also add `/etc/hosts` override: `echo '0.0.0.0 huggingface.co' >> /etc/hosts && echo '0.0.0.0 hf.co' >> /etc/hosts`. (2) Monitor PyPI for patch release without manual checking: use `pip index versions chromadb 2>/dev/null | head -1` in a cron job every 6 hours and alert on version string change above 1.5.8. If exploitation is confirmed and the host is compromised, do not patch-in-place — rebuild from a known-clean base image and restore ChromaDB configuration from pre-compromise backup, as a compromised Python runtime environment cannot be trusted after T1059.006 execution.

Evidence: Before eradication actions, preserve: (1) Full pip environment snapshot: `pip freeze > /tmp/pip_freeze_preremoval_$(date +%Y%m%d%H%M%S).txt` — establishes baseline of all installed packages, including any attacker-installed Python packages that may have been dropped via the RCE. (2) Hugging Face model cache contents and timestamps: `find ~/.cache/huggingface /tmp -name '*.bin' -o -name '*.safetensors' -o -name 'config.json' | xargs ls -la > /tmp/hf_cache_inventory.txt` — attacker-loaded models or payloads disguised as model files will appear here. (3) Filesystem changes since ChromaDB service start: `find / -newer /proc/$(pgrep -f uvicorn)/exe -not -path '/proc/*' -not -path '/sys/*' -ls 2>/dev/null > /tmp/fs_changes_since_service_start.txt` — new files written by the ChromaDB process are primary indicators of post-exploitation staging.

Step 4: Recovery — After applying any released patch, verify the installed version with pip show chromadb and confirm it falls outside the 1.0.0–1.5.8 range. Re-audit FastAPI access logs for evidence of prior exploitation (unexpected model loads, unusual outbound connections, new files written to disk by the ChromaDB process). Rotate any credentials or API keys accessible from the ChromaDB host. Restore service only after confirming no indicators of prior compromise are present.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST IA-5 (Authenticator Management), CIS 5.2 (Use Unique Passwords), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Version verification after patch: `pip show chromadb | awk '/Version/{print $2}' | python3 -c "import sys; v=sys.stdin.read().strip(); print('SAFE' if tuple(map(int,v.split('.'))) > (1,5,8) else 'STILL VULNERABLE')"`. Credential rotation scope for ChromaDB hosts: enumerate all secrets accessible to the process — check environment variables (`cat /proc/$(pgrep -f uvicorn)/environ | tr '\0' '\n' | grep -iE 'key|token|secret|password'`), `.env` files in the application directory, and mounted Kubernetes secrets if applicable. Rotate any OpenAI API keys, Hugging Face access tokens, AWS/GCP service account credentials, and database connection strings found. For integrity verification without a commercial tool, run `pip hash --algorithm sha256 chromadb` against the PyPI-published SHA256 to confirm the installed package was not tampered with post-exploitation.

Evidence: Before restoring service: (1) Capture final state of FastAPI access logs with exploitation timeline annotations — `grep -E '(POST|GET).*/api/v1' /var/log/chromadb/access.log | awk '{print $1,$2,$3,$4,$7,$8,$9}' > /tmp/chromadb_access_timeline.txt`. (2) Document all environment variables and secrets present at time of compromise (redact values, preserve key names for rotation tracking). (3) Run a YARA scan against the ChromaDB working directory and Python site-packages for common webshell and Python backdoor patterns: `yara /path/to/webshell.yar /usr/local/lib/python*/dist-packages/chromadb/ /tmp/ ~/.cache/huggingface/` — a successful CVE-2026-45829 RCE could drop a Python-based reverse shell or modify existing ChromaDB module files to maintain persistence (MITRE T1546).

Step 5: Post-Incident — Review your AI/ML pipeline asset inventory to ensure all components with external model-loading capabilities are subject to the same network segmentation and authentication enforcement applied to other production services. Evaluate whether Hugging Face model ingestion should be restricted to a controlled, air-gapped process rather than live server-side fetching. Add ChromaDB and similar vector database components to your vulnerability management scope with automated version monitoring.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), NIST SI-5 (Security Alerts, Advisories, and Directives), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain

a Software Inventory), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: AI/ML pipeline asset discovery without a CMDB: `find / -name 'chromadb' -o -name 'chroma.log' -o -name 'chroma_collections' 2>/dev/null` across hosts; supplement with `pip list --format=columns | grep -iE 'chroma|langchain|llama|faiss|pinecone|weaviate|qdrant'` to identify the broader vector DB and AI pipeline attack surface. For automated version monitoring of ChromaDB and similar packages, set up a free GitHub Actions workflow or cron job using `pip index versions ` and compare against your pinned version baseline weekly. For air-gapped model ingestion, establish a policy requiring all Hugging Face models to be downloaded by a dedicated internal model registry host (e.g., a hardened Python script on an isolated VM with outbound-only access to huggingface.co), cached locally, and served internally — eliminating live server-side fetching that CVE-2026-45829 exploits. Document this architectural control as a formal compensating control in your GRC system referencing NIST SC-7 (Boundary Protection) until a vendor patch is confirmed.

Evidence: Lessons-learned documentation should capture: (1) The full exposure window — date ChromaDB was first deployed at the vulnerable version versus date of containment — to support breach notification scope assessment if PII or regulated data was accessible from the ChromaDB host. (2) Inventory of all AI pipeline components (LangChain, LlamaIndex, embedding services) that interacted with the compromised ChromaDB instance, as these may have transmitted query data or vector embeddings containing sensitive organizational information. (3) Network flow records (NetFlow/sFlow or VPC flow logs) for the ChromaDB host covering the full exposure window, retained per NIST AU-11 (Audit Record Retention) requirements for potential regulatory or legal review.

Detection Guidance

Query FastAPI access logs for requests to any endpoint that triggers model loading or collection creation from external or unexpected source IPs, particularly referencing Hugging Face model paths. Monitor outbound network flows from ChromaDB host processes to huggingface.co, hf.co, or cdn-lfs.huggingface.co; legitimate use may exist, but unexpected or high-frequency connections warrant review. On the host, inspect process ancestry: ChromaDB spawning Python subprocesses executing unfamiliar scripts is a strong indicator. In SIEM, correlate with T1059.006 (Python interpreter abuse) and T1105 (tool/file transfer) alerts. If you have EDR, create a rule for file writes originating from the ChromaDB process outside its expected working directory. Log sources: FastAPI stdout/stderr, host process audit logs (auditd or equivalent), network flow logs (firewall, VPC flow logs), EDR telemetry.

Indicators of Compromise

Type	Value	Context	Confidence
DOMAIN	`huggingface.co`	Hugging Face model repository — ChromaDB exploitation involves forcing the server to load a malicious model from this domain; unexpected or anomalous outbound connections from ChromaDB hosts to this domain warrant investigation. Note: this domain has legitimate uses and is not inherently malicious.	LOW
DOMAIN	`hf.co`	Hugging Face short domain — same context as huggingface.co; monitor for unexpected outbound connections from ChromaDB host processes.	LOW

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1203** — Exploitation for Client Execution
- **T1105** — Ingress Tool Transfer
- **T1059** — Command and Scripting Interpreter
- **T1071** — Application Layer Protocol
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1195.002** — Compromise Software Supply Chain
- **T1059.006** — Python

NIST-800-53R5

- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-7** — Continuous Monitoring
- **CM-7** — Least Functionality
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **AC-3** — Access Enforcement
- **SI-10** — Information Input Validation
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-8** — Identification and Authentication (Non-Organizational Users)

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A03:2021** — Injection
- **A07:2021** — Identification and Authentication Failures

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.8** — Define and Maintain Role-Based Access Control
- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access

- **6.5** — Require MFA for Administrative Access
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets
- **CC9.2** — Manages risks associated with vendors and business partners

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1190	Exploit Public-Facing Application	Initial-Access
T1203	Exploitation for Client Execution	Execution
T1105	Ingress Tool Transfer	Command-And-Control
T1059	Command and Scripting Interpreter	Execution
T1071	Application Layer Protocol	Command-And-Control
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1195.002	Compromise Software Supply Chain	Initial-Access
T1059.006	Python	Execution

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/max-severity-flaw-in...	T3
CVE-2026-45829 Detail - NVD - NIST	https://nvd.nist.gov/vuln/detail/CVE-2026-45829	T1
CVE-2026-45829 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-45829	T3

Source	URL	Tier
CVE-2026-45829 — ChromaDB Python server hands you RCE ...	https://hadrian.io/blog/cve-2026-45829----chromadb-python-server-ha...	T3
CVE-2026-45829 - Exploits & Severity - Feedly	https://feedly.com/cve/CVE-2026-45829	T3

Generated 2026-05-20 06:44 UTC by TJS Security Command Center