contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

Spring AI: new CVE

CVE-2026-59415 · record published 24 November 2026

Analysis Report

Any team running Spring AI with TypesenseVectorStore should upgrade this week; the record gives this a score of 10 and the flaw is network-reachable. If you do not use Spring AI at all, you can stop …

PREPARED BY	NamiQ	DATE	24 November 2026
FACTS AS OF	24 November 2026	VERSION	1.0
SEVERITY	**Critical: CVSS 3.1 10.0**	STATUS	Not on CISA's exploited list; exploitation not recorded

NamiQ © 2026 NamiQ. All rights reserved. This document is the property of NamiQ. It may be shared unaltered, with attribution.
Companion article: namiq.io/insights/spring-ai-cve-2026-59415

The bottom line

> *Any team running Spring AI with TypesenseVectorStore should upgrade this week; the record gives this a score of 10 and the flaw is network-reachable. If you do not use Spring AI at all, you can stop reading. If you do, check the version and upgrade to 2.0.3 (OSS) first.*

At a glance

Fact	Detail
Product	Spring AI (CVE-2026-59415)
Severity	Critical: 10 (CVSS 3.1)
Status	A new CVE record, published by Vmware on 24 November 2026. Not on CISA's list of exploited vulnerabilities when we read it on 24 November 2026, which means nothing public says yet that anyone is using it.
Fixed in	2.0.3 (OSS); 2.0.2.1, 1.1.10, 1.0.11 (Enterprise Support only). The vendor's advisory lists all four.
Who looks first	Teams running a Java application built on Spring AI 2.0.0–2.0.2, 1.1.0–1.1.9, or 1.0.0–1.0.10, where the application uses TypesenseVectorStore and accepts document IDs from user input or an external caller.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Does your application use Spring AI?	Nothing to do for this CVE.	Check whether you use TypesenseVectorStore (next question).
Does your application use TypesenseVectorStore?	Nothing to do for this CVE.	Check the Spring AI version you run (next question).
Is your Spring AI version in the range 2.0.0–2.0.2, 1.1.0–1.1.9, or 1.0.0–1.0.10?	Nothing to do for this CVE.	Check whether user-supplied values can reach the delete call (next question).
Can an external caller or user control the document IDs passed to the delete operation?	Risk is lower; still plan to upgrade at next patch cycle.	Upgrade promptly; apply the ID-validation workaround immediately if upgrading is delayed.

What it is

An attacker who can supply document IDs to the vector store's delete call can inject filter-grammar characters, causing the application to delete documents it was not asked to delete, or to wipe every document in the collection in a single operation.

Why it matters

The impact is targeted or bulk deletion of a vector store collection — no data is read, but the application's knowledge base can be destroyed. The vendor's advisory describes the flaw as network-reachable with low complexity and low privileges required, and no user interaction needed. Our reading is that the practical risk depends on whether the application exposes a path where external input reaches the delete call; where it does, a single crafted ID can wipe the collection.

What the vendor's advisory says

We read one page on spring.io. In short:

- The vendor's advisory names no other CVEs fixed in the same release.
- The vendor's advisory says a workaround exists: validate every document ID before passing it to the delete call and reject any ID containing characters outside letters, digits, hyphens, underscores, dots, colons, forward slashes, and at-signs.
- 2.0.3 is available as an OSS release; 2.0.2.1, 1.1.10, and 1.0.11 are Enterprise Support only, per the advisory.
- The advisory was published 19 November 2026.

Things people will get wrong

- Any Spring AI user is affected. Only applications using TypesenseVectorStore and passing user-controlled IDs to the delete operation are at risk. The record and advisory do not name other VectorStore implementations.
- This flaw lets an attacker read stored data. The CVSS vector in the advisory shows no confidentiality impact. The flaw causes deletion, not disclosure.

How we would triage it

This is our judgement.

Step 1: Find the Spring AI version you run

- Check your project's dependency manifest (pom.xml or build.gradle) for the spring-ai or spring-ai-bom version. Affected versions are 2.0.0–2.0.2, 1.1.0–1.1.9, and 1.0.0–1.0.10.

- Confirm whether the application uses TypesenseVectorStore specifically, not another VectorStore implementation.

Step 2: Apply the workaround if you cannot patch immediately

- The vendor's advisory says to validate every document ID before calling delete and reject any ID containing characters outside letters, digits, hyphens, underscores, dots, colons, forward slashes, and at-signs.
- The advisory says document IDs from standard UUID generation or base64url encoding are safe and do not need additional validation.

Step 3: Upgrade to the fixed version

- The vendor's advisory lists 2.0.3 as the OSS-available fix for the 2.0.x line. Fixed versions 1.1.10 and 1.0.11 are Enterprise Support only for those lines.
- Update your dependency manifest to the fixed version and redeploy. Confirm the new version is in use after deployment.

Step 4: Check for signs of unintended deletion

- Review your Typesense collection document counts after patching. If unexpectedly low, check application logs for delete operations with unusual filter expressions.
- If document IDs are user-controlled, review recent delete requests for IDs containing commas, closing brackets, or operator tokens.

To close it, read the version from the product itself, not from an old inventory, and save the output with the date. If you cannot update today, write down who owns it, the new date, and what limits exposure until then.

Send this to your IT provider

Copy it, change the names and dates, and send it. It is a draft for you to check, not a template tested with any provider.

> *We are writing about CVE-2026-59415, a flaw in Spring AI's TypesenseVectorStore component published 24 November 2026. The flaw allows an attacker who can supply document IDs to the delete operation to delete unintended documents or wipe an entire Typesense collection. We would like to know whether we run Spring AI, whether we use TypesenseVectorStore, and which version we are on. The fixed open-source release is 2.0.3; Enterprise Support versions 2.0.2.1, 1.1.10, and 1.0.11 are also listed as fixed by the vendor. Could you confirm in writing which version we run and whether the fix is in place? We would appreciate a response by 1 December 2026.*

Likely ATT&CK mapping

MITRE ATT&CK is a public catalogue of the steps attackers take, from getting in to causing damage. No source has mapped this flaw to it yet, so the steps below are our prediction from what the record says the flaw allows. They are not MITRE's, the vendor's or CISA's, and an attacker may do other things.

Steps an attacker would likely take with this flaw

The step, in plain words	Stage of an attack	Why we think so
Data Destruction (T1485): Deleting or overwriting data so it cannot be recovered.	Impact: causing damage or disruption	Operator breakout would match every document in the collection, resulting in bulk deletion; data destruction is the primary impact the record describes.
Data Manipulation: Stored Data Manipulation (T1565.001): Changing stored data without permission.	Impact: causing damage or disruption	Comma injection selectively deletes documents other than the intended ones, manipulating stored data in the vector store beyond what the caller requested.
Exploit Public-Facing Application (T1190): Breaking in through a service that faces the internet, such as a web server, VPN or login page.	Initial Access: getting in	If the delete path is exposed through a public-facing API, an attacker with low-privilege network access would likely reach it via that application endpoint.

How to use this: ask your IT provider which control or log would show each step, for example whether sign-in attempts from the internet are recorded. If nobody can answer, that is the gap to close first.

MITRE ATT&CK® is a registered trademark of The MITRE Corporation. Technique names are MITRE's.

The sources, in their words

The CVE record:

> *TypesenseVectorStore is vulnerable to a document ID injection in its delete operation that can cause unintended documents to be deleted. TypesenseVectorStore.doDelete(List<String>) builds a Typesense filter_by expression by raw string concatenation of caller-supplied document IDs into the form id: [id1,id2,...] . The Typesense id field uses bare (unquoted) values in this list form with no escaping mechanism. An ID containing characters with syntactic meaning in the Typesense filter grammar — such as a comma (,), a closing bracket ( ]), or operator tokens (\|\| , &&) — can break out of the list literal and alter the structure of the filter expression. An attacker who controls a document ...*

Versions of Spring AI the record lists as affected:

- 2.0.0 up to, but not including, 2.0.3

- 1.1.0 up to, but not including, 1.1.10
- 1.0.0 up to, but not including, 1.0.11
- 2.0.0 up to, but not including, 2.0.2.1

The record gives no version ranges. Use the vendor's advisory for the affected versions.

Check your own project

If you build software, you can compare what a project declares with this CVE from a terminal. The tool reads requirements files, package.json, Maven, Gradle, Cargo, CMake and Dockerfile files, runs nothing, and only the CVE id leaves your machine. Naming the CVEs needs no key.

```
lattice-scan project . --cve CVE-2026-59415
```

Each row says in-range, outside-range, undetermined or lead, with the file and line. That is evidence about what a manifest declares, not about what runs, and a person decides. The verdicts, the CSV columns and a pipeline recipe are in the guide.

Check the servers you run

The same id can be checked against a host. Run this on the machine itself, not on the laptop that deploys to it: it reads installed packages (dpkg, rpm or apk), snaps, the hardware (maker, model, firmware), the running container images with their operating system and packages, and the kernel version, compares them on the spot, and only the CVE ids leave the machine; the inventory stays on it, in ~/.lattice-scan. Naming the CVEs needs no key. Do it on each server after a deployment, and again when a new entry lands.

```
lattice-scan --cve CVE-2026-59415
```

A match is a candidate for a person to confirm, not a finding. A distribution can fix a CVE under an unchanged version number, so a match is a candidate; on Debian-family hosts the report also shows whether the package's own changelog names the CVE. By default it opens the images your containers are running and reads their operating system and package list; a deep scan (--deep, a few minutes) opens every local image and the Python, Node, Java and Ruby packages inside them; for what an image declares, use the project check on its Dockerfile.

Guide: namiq.io/lattice/guides/choose-your-channel. MCP server (io.namiq/lattice in the MCP registry): namiq.io/lattice. Package: pypi.org/project/lattice-scan, or namiq.io/apt.

What we do not know yet

- The sources do not say whether any production application exposes a path where external input reaches the delete operation.

- The sources do not say whether other VectorStore implementations in Spring AI use a similar concatenation pattern and might be affected by the same class of flaw.
- The sources do not confirm whether the Enterprise-only fixed releases (2.0.2.1, 1.1.10, 1.0.11) carry source changes or are binary-only patches.
- Exploitation is not recorded. That is not the same as none.
- This is a new record: scores, versions and wording can change in the first days, and they are as at 24 November 2026. We read the vendor pages listed under Sources; nothing here has been tested.
- The sections headed "The bottom line", "Does this apply to you?", "What it is", "Why it matters", "Exposure by situation", "Things people will get wrong", "Likely ATT&CK mapping", "How we would triage it" and "Send this to your IT provider" are our interpretation of the sources, not a statement by CISA or the vendor.

Appendix 1. Severity decoded

CVE-2026-59415: score 10, `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H` As the CVE record gives it.

Metric	Value	What it means here
Attack vector	Network	Reachable over a network, with no need to be on the same segment.
Attack complexity	Low	No special conditions beyond the flaw itself.
Privileges required	None	No account is needed.
User interaction	None	No one has to do anything.
Scope	Changed	Damage reaches beyond the vulnerable component.
Confidentiality	High	Total loss of confidentiality.
Integrity	High	Total loss of integrity.
Availability	High	Total loss of availability.

Appendix 2. Timeline

Date	Event
24 November 2026	CVE record published by Vmware.

Appendix 3. Exposure by situation

These priorities are our judgement from the attacker prerequisites in the sources. They are not vendor ratings.

Your situation	Priority	Timeframe
Application uses TypesenseVectorStore and passes user-supplied IDs to the delete call, reachable over a network	Medium	This week
Application uses TypesenseVectorStore but document IDs are generated internally, not from user input	Low	Next patch cycle
Application uses Spring AI on an affected version but a different VectorStore implementation	None	None for this CVE
Application uses Spring AI 2.0.3 or later (OSS), or Enterprise releases 2.0.2.1, 1.1.10, or 1.0.11	None	None for this CVE

Appendix 4. Definition of done

- Every Spring AI you run is listed, with its version.
- Each is outside the affected list, or the vendor's fix or mitigation is in place.
- The version output or a screenshot is saved with the date.
- Anything not done has an owner and a date.

Sources

- CVE record for CVE-2026-59415: https://www.cve.org/CVERecord?id=CVE-2026-59415
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Advisory or release notes read: spring.io/security/cve-2026-59415/: https://spring.io/security/cve-2026-59415/
- Reference in the CVE record: spring.io: https://spring.io/security/cve-2026-59415