contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

# hexpm: new CVE

**CVE-2026-129514 · record published 24 November 2026**

## Analysis Report

Only teams running a self-hosted hexpm instance need act. The record states the flaw was actively exploited against the hex.pm service on 23 November 2026; if you host hexpm yourself, check your …

| | | | |
|---|---|---|---|
| PREPARED BY | NamiQ | DATE | 24 November 2026 |
| FACTS AS OF | 24 November 2026 | VERSION | 1.0 |
| SEVERITY | **Critical: CVSS 4.0 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/hexpm-cve-2026-129514

## The bottom line

> *Only teams running a self-hosted hexpm instance need act. The record states the flaw was actively exploited against the hex.pm service on 23 November 2026; if you host hexpm yourself, check your deployed commit against the fix immediately.*

## At a glance

| Fact | Detail |
|---|---|
| Product | hexpm (CVE-2026-129514) |
| Severity | Critical: 10 (CVSS 4.0) |
| Status | A new CVE record, published by EEF 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 | The source identifies the fix at commit 379b0a29bcc95164249f38ad7d772a37a8ff9533 in hexpm/hexpm. No release version number is named in any source. |
| Who looks first | Teams running a self-hosted instance of hexpm, the Elixir and Erlang package registry software. If you do not host hexpm yourself, there is nothing to do for this CVE. |

## Does this apply to you?

**Answer in order. Stop at the first no.**

| Question | If no | If yes |
|---|---|---|
| Do you run a self-hosted instance of hexpm? | Nothing to do for this CVE. | Check the deployed commit against the fix (next question). |
| Is your deployment at a commit before 379b0a29bcc95164249f38ad7d772a37a8ff9533? | You are on or past the fix commit; no further action needed. | Update to commit 379b0a29bcc95164249f38ad7d772a37a8ff9533 or later immediately. |
| Is your hexpm instance reachable from the internet or by untrusted users? | Exposure is reduced; still plan to update at your next maintenance window. | Treat this as urgent; restrict access while you prepare the update. |

## What it is

An unauthenticated attacker can send a crafted request body to any route on the application and cause the server to execute operating system commands as the user running the application process.

## Why it matters

Our reading is that the score of 10.0 reflects no authentication, no complexity, and full system impact in combination. The source says the parser runs before any authentication plug, so every route on the application is reachable without credentials. A successful exploit gives OS command execution as the application user, from which the source says the database URL, storage credentials, and registry signing key held in the process environment are all readable. The record states the flaw was actively exploited against the hex.pm service on 23 November 2026, confirming it is not theoretical.

## What the vendor's advisory says

We read three pages on github.com and cna.erlef.org and osv.dev. In short:

- The source says the fix makes the Erlang format decoder return an error when any decoded map has a __struct__ key, so a crafted request body now gets a 400 response rather than reaching the protocol dispatch chain. No other CVEs are named alongside this fix in the sources.

## Things people will get wrong

- Only the public hex.pm service was at risk, not self-hosted instances The record names hexpm/hexpm (the open-source codebase) as affected. Any deployment of that code at an affected commit carries the same flaw, regardless of who operates it.
- Authentication would stop an attacker from reaching the vulnerable parser The source says the parser runs in the endpoint before routing and authentication, so a POST to any route reaches it without any credentials.

## How we would triage it

This is our judgement.

## Step 1: Confirm whether you run hexpm

- Check your infrastructure inventory for any self-hosted hexpm deployment. If none is found, there is nothing further to do for this CVE.

## Step 2: Check the deployed commit

- Retrieve the commit hash of the hexpm build running in production. The source names 379b0a29bcc95164249f38ad7d772a37a8ff9533 as the fix; any commit before it in the

hexpm/hexpm history is affected.

## Step 3: Restrict access while updating

- If the instance is internet-facing and at an affected commit, restrict network access to the application while the update is prepared. The source says the parser runs before authentication, so a network-level block reduces exposure immediately.

## Step 4: Deploy the fix and review logs

- Deploy the code at commit 379b0a29bcc95164249f38ad7d772a37a8ff9533 or later. After deploying, review application and web server logs for POST requests carrying the content type application/vnd.hex+erlang sent before the fix was in place.

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 reviewing CVE-2026-129514, a critical unauthenticated remote code execution flaw (CVSS 4.0 score 10.0) in hexpm, the Elixir and Erlang package registry software. The record states this flaw was actively exploited against the hex.pm service on 23 November 2026. We need to establish whether we run a self-hosted hexpm instance anywhere, and if so, which commit is deployed. The fix is at commit 379b0a29bcc95164249f38ad7d772a37a8ff9533 in the hexpm/hexpm repository; no release version number is given in the sources. Please confirm in writing whether we are running hexpm, the deployed commit hash, and whether the fix has been or will be applied, by 27 November 2026.*

## The weakness behind it

The record classes this as CWE-502.

## 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 |
|---|---|---|

| | | |
|---|---|---|
| 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 | The flaw is in a public HTTP endpoint reachable before authentication; a crafted request body alone is sufficient to trigger OS command execution, with no login needed. |
| Command and Scripting Interpreter (T1059): Running commands or scripts on a machine, such as a shell or PowerShell. | Execution: running the attacker's code | Exploitation reaches an apply/3 call on attacker-controlled module, function and argument values, which would typically result in OS command execution via the BEAM process. |
| Data from Local System (T1005): Copying data stored on the machine. | Collection: gathering data | The source says the process environment, which holds the database URL, storage credentials and registry signing key, is readable once command execution is achieved. |

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:

> *Deserialization of Untrusted Data vulnerability in the application/vnd.hex+erlang request body parser in hexpm hexpm allows an unauthenticated remote attacker to execute operating system commands via a crafted Erlang external term format request body. This vulnerability was actively exploited against the hex.pm service on 2026-11-23. HexpmWeb.ErlangFormat.decode/1 in lib/hexpm_web/erlang_format.ex decodes the body with Plug.Crypto.non_executable_binary_to_term/2, which accepts maps that have a __struct__ key. HexpmWeb.PlugParser.parse/5 returns the decoded term as the request params. The parser runs in the endpoint before routing and authentication, so a POST to any route reaches it. Elixir ...*

Versions of hexpm the record lists as affected:

- 2016-02-23 up to, but not including, 2026-11-23
- ab1bbcef7cc8395d1056d3b1aa6a174353492d59 up to, but not including, 379b0a29bcc95164249f38ad7d772a37a8ff9533

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-2026-129514
```

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-2026-129514
```

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

- No release version number is given for the fix; sources identify it only by commit hash 379b0a29bcc95164249f38ad7d772a37a8ff9533.
- The scope and impact of the confirmed exploitation of hex.pm on 23 November 2026 are not described in the record.
- No workaround short of updating the code is described in any source.
- Whether any other Elixir or Erlang applications using similar Erlang term body parsers are affected by the same class of flaw is not addressed in the record.
- 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-2026-129514**: score 10, `CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H/E:A` 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. |
| Attack requirements | None | Works against a default deployment. |
| Privileges required | None | No account is needed. |
| User interaction | None | No one has to do anything. |
| Vulnerable system: confidentiality | High | Total loss of confidentiality. |
| Vulnerable system: integrity | High | Total loss of integrity. |
| Vulnerable system: availability | High | Total loss of availability. |
| Subsequent systems: confidentiality | High | Total loss of confidentiality. |
| Subsequent systems: integrity | High | Total loss of integrity. |
| Subsequent systems: availability | High | Total loss of availability. |

## Appendix 2. Timeline

| Date | Event |
|---|---|
| 24 November 2026 | CVE record published by EEF. |

## 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 |
|---|---|---|
| Self-hosted hexpm at an affected commit, reachable from the internet | High | Immediately — exploitation confirmed 23 November 2026 |
| Self-hosted hexpm at an affected commit, internal network only | Medium | This week, before your next release cycle |

| | | |
|---|---|---|
| Self-hosted hexpm at an affected commit, with network access controls restricting who can reach it | Medium | This week; controls reduce but do not remove the risk |
| Not running hexpm at all | None | None for this CVE |

## Appendix 4. Definition of done

- Every hexpm 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-2026-129514: https://www.cve.org/CVERecord?id=CVE-2026-129514
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Advisory or release notes read: github.com/hexpm/hexpm/security/advisories/GHSA-9pmm-hf4r-: https://github.com/hexpm/hexpm/security/advisories/GHSA-9pmm-hf4r-27pg
- Advisory or release notes read: cna.erlef.org/cves/CVE-2026-129514.html: https://cna.erlef.org/cves/CVE-2026-129514.html
- Advisory or release notes read: osv.dev/vulnerability/EEF-CVE-2026-129514: https://osv.dev/vulnerability/EEF-CVE-2026-129514
- Reference in the CVE record: hex.pm: https://hex.pm/blog/security-incident-on-hexpm
- Reference in the CVE record: github.com: https://github.com/hexpm/hexpm/commit/ab1bbcef7cc8395d1056d3b1aa6a174353492d59