contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

lxc incus: new CVE

CVE-2026-130778 · record published 18 November 2026

Analysis Report

If you run Incus before 7.6.0 and its HTTPS listener can be reached by anyone you do not trust, act now: restrict that access to trusted hosts, then upgrade to 7.6.0. If you do not run Incus, or the ...

PREPARED BY	NamiQ	DATE	19 November 2026
FACTS AS OF	19 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/lxc-incus-cve-2026-130778

The bottom line

> *If you run Incus before 7.6.0 and its HTTPS listener can be reached by anyone you do not trust, act now: restrict that access to trusted hosts, then upgrade to 7.6.0. If you do not run Incus, or the listener is not enabled, you can stop reading.*

At a glance

Fact	Detail
Product	lxc incus (CVE-2026-130778)
Severity	Critical: 10 (CVSS 4.0)
Status	A new CVE record, published by VulnCheck on 18 November 2026. Not on CISA's list of exploited vulnerabilities when we read it on 19 November 2026, which means nothing public says yet that anyone is using it.
Fixed in	Incus 7.6.0 (the vendor's advisory, 29 October 2026; the record lists 7.6.0 as unaffected).
Who looks first	Anyone running Incus below 7.6.0 with the HTTPS listener reachable from other machines, above all clusters and hosts open to untrusted networks.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run Incus, or does an IT provider run it for you?	Nothing to do for this CVE.	Check the version (next question).
Is your Incus version older than 7.6.0?	7.6.0 or later is not affected. Nothing more for this CVE.	You are in the affected range. Check who can reach the listener.
Is the HTTPS listener enabled on any Incus host?	The advisory describes the flaw in the HTTPS listener. Still plan the upgrade.	Check who can reach it (next question).
Can any machine you do not fully trust reach that listener over the network?	Lower urgency. Upgrade to 7.6.0 in your next patch cycle.	Restrict access now, then upgrade to 7.6.0 this week.

What it is

A client that can reach the HTTPS listener can log in as any trusted client, metrics or server certificate by adding that certificate's public part to its own chain. It does not need the private key. This bypasses authorization entirely.

Why it matters

Our reading is that this is a full bypass, not a partial one. The advisory says server certificates are handed out in every TLS handshake, so an attacker needs no prior access to get one. On a cluster, another member's certificate gives internal cluster access. The advisory says the flaw has existed since the initial release, so every older install is in range.

What the vendor's advisory says

We read one page on github.com. In short:

- The vendor's advisory rates this Critical, 10.0, and says it has been present since the initial release.
- It says a client can also renew or delete any trust store entry through the self-service paths.
- It says the agent connection between the host and virtual machines is also affected.
- It says the unverified chain also reaches the authorization scriptlet, so a client can try to influence scriptlet-based decisions.
- Workaround per the advisory: restrict network access to the HTTPS listener to trusted hosts. It says there is no configuration-based workaround.

Things people will get wrong

- "Clients need a valid certificate, so we are protected." The advisory says the handshake only proves the client holds the key for the first certificate. The rest of the chain is client controlled and was wrongly trusted.
- "The server's certificate is secret, so this is hard." The advisory says server certificates are returned by every TLS handshake, so no prior access is needed to obtain one.
- "We can switch off a setting to avoid it." The advisory says there is no configuration-based workaround. Restricting network access to the listener is the only mitigation it gives.

How we would triage it

This is our judgement.

The commands only read: they print a version or list what is installed. They come from general knowledge of this kind of product, not from the sources, so check them against the vendor's page before you run them. Words in capitals are for you to fill in.

Step 1: Find the Incus versions you run

- List every host running Incus, including clusters, and note each version. Anything below 7.6.0 is in the affected range per the record.

```
incus version
```

Step 2: Reduce exposure to the HTTPS listener

- Limit network access to the listener to trusted hosts only. This is the vendor's stated workaround, and it is the only one.
- Check whether the listener is enabled at all and which addresses it uses.

Step 3: Upgrade to 7.6.0

- Move every host to 7.6.0, the version the advisory lists as patched. Do clusters as a whole, not one member.

Step 4: Confirm the fix and review trust

- Re-run the version check on each host and keep the output.
- Review the list of trusted certificates for entries you do not recognise.

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.

> *Hello, we need to confirm our exposure to CVE-2026-130778 in Incus. Versions before 7.6.0 let a remote client log in as any trusted certificate. Please tell us whether we run Incus, on which hosts, and which version each is. Please also say whether the HTTPS listener is reachable from outside trusted hosts. The fixed version is 7.6.0. Please restrict listener access now and send us written confirmation of each host's version and the upgrade by a date you name.*

The weakness behind it

The record classes this as CWE-287.

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 network listener and needs no privileges, so it would likely be used against an exposed Incus service.
Valid Accounts (T1078): Logging in with a real account the attacker should not have, so the activity looks like normal use.	Initial Access: getting in	The record says attackers authenticate as a trusted certificate, so they would likely act as a legitimate client or cluster member.
Account Manipulation (T1098): Changing an existing account, for example adding permissions or keys, to keep access.	Persistence: staying in	The record says attackers can modify trust store entries, which would typically let them keep access by altering who is trusted.

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:

> *Incus before 7.6.0 contains an authentication bypass vulnerability that allows remote attackers to authenticate as any trusted certificate by appending its public certificate to their presented TLS client chain. Attackers can append the server's or a cluster member's certificate to gain cluster-level access, bypass authorization, modify trust store entries, and influence scriptlet-based authorization decisions.*

Versions of lxc incus the record lists as affected:

- all versions before 7.6.0

It lists these as unaffected:

- 7.6.0

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-2026-130778
```

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-2026-130778
```

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

- Whether this flaw has been used in attacks; the sources do not say.
- Whether any release before 7.6.0 on an older branch has a separate fix; the sources name only 7.6.0.

- Whether a compromise leaves visible traces; the sources describe none.
- 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 19 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-2026-130778: 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` 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
18 November 2026	CVE record published by VulnCheck.

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
Incus below 7.6.0 with the HTTPS listener reachable from untrusted networks	Medium	Restrict access today; upgrade this week

Incus cluster below 7.6.0, listener open to any host outside the cluster	Medium	Restrict access today; upgrade this week
Incus below 7.6.0, listener reachable only from a small set of trusted hosts	Medium	Upgrade this week
Incus below 7.6.0 with the HTTPS listener not enabled	Low	Next patch cycle
Incus 7.6.0 or later, or no Incus at all	None	None for this CVE

Appendix 4. Definition of done

- Every lxc incus 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-2026-130778: https://www.cve.org/CVERecord?id=CVE-2026-130778
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Advisory or release notes read: github.com/lxc/incus/security/advisories/GHSA-xg75-xf37-cp: https://github.com/lxc/incus/security/advisories/GHSA-xg75-xf37-cph9