contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

OpsTree Solutions BuildPiper: new CVE

CVE-2026-127073 · record published 13 November 2026

Analysis Report

If you run OpsTree BuildPiper to deploy to Kubernetes, check your version this week. Any version before 2.7.0 lets an ordinary project user seize cluster-admin, so upgrade to 2.7.0. If you do not run …

PREPARED BY	NamiQ	DATE	13 November 2026
FACTS AS OF	13 November 2026	VERSION	1.0
SEVERITY	**Critical: CVSS 3.1 9.9**	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/opstree-solutions-buildpiper-cve-2026-127073

The bottom line

> *If you run OpsTree BuildPiper to deploy to Kubernetes, check your version this week. Any version before 2.7.0 lets an ordinary project user seize cluster-admin, so upgrade to 2.7.0. If you do not run BuildPiper, you can stop reading.*

At a glance

Fact	Detail
Product	OpsTree Solutions BuildPiper (CVE-2026-127073)
Severity	Critical: 9.9 (CVSS 3.1)
Status	A new CVE record, published by VulnCheck on 13 November 2026. Not on CISA's list of exploited vulnerabilities when we read it on 13 November 2026, which means nothing public says yet that anyone is using it.
Fixed in	BuildPiper 2.7.0 — the first version not affected, per the VulnCheck advisory dated 12 November 2026. Every version before 2.7.0 is affected.
Who looks first	Anyone running BuildPiper to deploy into a Kubernetes cluster, above all where several teams or project users share one cluster. The person who manages your CI/CD pipeline and its cluster credentials should look first.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run OpsTree BuildPiper to deploy to Kubernetes?	Nothing to do for this CVE.	Check your version (next question).
Is your BuildPiper version earlier than 2.7.0?	You are on a fixed version; nothing to do.	You are affected; plan the upgrade.
Can more than one person or team use the Additional Manifest feature?	Lower risk, but still upgrade.	Any project user could take the cluster; upgrade now.

What it is

The record says the Additional Manifest feature does not limit which Kubernetes objects a project user may submit, and the pipeline applies them with its own cluster-wide credentials. So a user

allowed only to deploy their own app can bind a service account to cluster-admin and take over every workload and secret.

Why it matters

The weakness is improper privilege management: the pipeline trusts manifests a user supplies but runs them with its own cluster-wide rights. Our reading is that the real exposure is internal. Any holder of a low-level project account, or anyone who steals one, can cross from 'deploy my own app' to full control of the cluster, its other tenants and its secrets. On a shared cluster that turns a minor account into a master key. The advisory marks it critical and says the escalation needs no step beyond a normal deploy.

What the vendor's advisory says

We read two pages on www.vulncheck.com and gist.github.com. In short:

- The VulnCheck advisory rates it 9.4 on the newer CVSS version 4 scale and marks it critical, citing the same weakness, improper privilege management (CWE-269).
- The researcher's advisory says the flaw reaches the pipeline through three input methods: file upload, a Git source and the YAML editor. All three behaved the same way.
- Neither source names any other flaw fixed in the 2.7.0 release, nor a workaround short of upgrading.

Things people will get wrong

- It needs no account, so anyone on the internet can do it. No. The record says a project user account is required. The risk is the holder of a low-level account, or a stolen one, not an anonymous stranger.
- We limit who can deploy, so we are safe. The flaw is that an ordinary project user, allowed only to deploy their own app, can still escalate. Deploy rights are enough; upgrade to 2.7.0.

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 whether you run it and which version

- Confirm whether BuildPiper is in use and on which version; every version before 2.7.0 is affected.

- Note which clusters its pipeline holds credentials for, since those are what an escalation would reach.

Step 2: Reduce exposure until you upgrade

- Tighten who holds project user accounts on BuildPiper, since any such account can submit manifests.
- Where you can, narrow the cluster credentials the pipeline uses so they are not cluster-admin, reducing what a submitted manifest can do.

Step 3: Upgrade to the fixed version

- Plan the upgrade to BuildPiper 2.7.0, the first version the advisory says is not affected.
- Neither source names a workaround that fully fixes it, so treat upgrading as the fix.

Step 4: Check for misuse if it was exposed

- Look in the cluster for ClusterRoleBindings or ClusterRoles you did not create, especially any bound to cluster-admin, and for namespaces you did not expect.
- Review recent Additional Manifest deployments for RBAC or cluster-scoped objects.

```
kubectl get clusterrolebindings
kubectl get clusterroles
kubectl get namespaces
```

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 checking our exposure to CVE-2026-127073, a privilege escalation in OpsTree BuildPiper. Please confirm whether we run BuildPiper, where, and which version. Every version before 2.7.0 is affected, and the fix is version 2.7.0. The flaw lets any project user submit a Kubernetes manifest that grants them cluster-admin, so it matters most where the pipeline holds cluster-wide credentials or several people share a cluster. It is not on CISA's exploited list, but the severity is high. Please tell us in writing the version in use and your plan to upgrade to 2.7.0, by the end of this week.*

The weakness behind it

The record classes this as CWE-269.

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
Exploitation for Privilege Escalation (T1068): Using a flaw to gain higher rights than the account had, such as administrator or root.	Privilege Escalation: gaining higher rights	The record says a low-privileged project user can submit a manifest granting cluster-admin, so this software flaw would likely be used to raise rights from app-level deployment to full cluster control.
Account Manipulation (T1098): Changing an existing account, for example adding permissions or keys, to keep access.	Persistence: staying in	Deploying a ClusterRoleBinding that binds a service account to cluster-admin would typically add high rights that persist in the cluster after the deploy.
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 attack needs a working low-level project login; whoever holds or steals such credentials would likely be the one to exploit it.

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:

> *OpsTree BuildPiper before 2.7.0 contains a privilege escalation vulnerability that allows project users to gain Kubernetes cluster-admin rights by submitting unrestricted manifests through the Additional Manifest feature. Attackers can deploy a ClusterRoleBinding binding a service account to cluster-admin, which the pipeline applies with cluster-wide credentials, granting control over all workloads and secrets.*

Versions of OpsTree Solutions BuildPiper the record lists as affected:

- all versions before 2.7.0

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-2026-127073
```

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-2026-127073
```

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 upgrading to 2.7.0 changes the cluster credentials the pipeline uses, or only blocks unrestricted manifests.
- They do not say how the fix limits which Kubernetes resource kinds a manifest may contain.

- There is no workaround given for sites that cannot upgrade at once.
- They do not say whether the cluster records which project user submitted a given manifest.
- 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 13 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-2026-127073: score 9.9, `CVSS:3.1/AV:N/AC:L/PR:L/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	Low	An ordinary account is enough.
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
13 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
A shared Kubernetes cluster where several project users or teams can submit Additional Manifests	Medium	this week
A BuildPiper instance before 2.7.0 used by staff or contractors you do not fully trust	Medium	this week
A single-team cluster where only trusted admins use BuildPiper	Low	next patch cycle
You do not run BuildPiper	None	none for this CVE

Appendix 4. Definition of done

- Every OpsTree Solutions BuildPiper 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-2026-127073: https://www.cve.org/CVERecord?id=CVE-2026-127073
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Advisory or release notes read: www.vulncheck.com/advisories/opstree-buildpiper-before-2.7.0-priv: https://www.vulncheck.com/advisories/opstree-buildpiper-before-2.7.0-privilege-escalation-via-additional-manifest
- Advisory or release notes read: gist.github.com/neelkotnis/7bc715e190e5be2c2fc5be15c4035ce7: https://gist.github.com/neelkotnis/7bc715e190e5be2c2fc5be15c4035ce7
- Reference in the CVE record: www.buildpiper.io: https://www.buildpiper.io/