contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

kyverno: new CVE

CVE-2026-124440 · record published 8 November 2026

Analysis Report

Any team running Kyverno 1.19.1 on a multi-tenant cluster where namespace tenants can create policies must update to 1.19.2. The first step is to confirm which version is deployed — only 1.19.1 is …

PREPARED BY	NamiQ	DATE	8 November 2026
FACTS AS OF	8 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/kyverno-cve-2026-124440

The bottom line

> *Any team running Kyverno 1.19.1 on a multi-tenant cluster where namespace tenants can create policies must update to 1.19.2. The first step is to confirm which version is deployed — only 1.19.1 is listed as affected.*

At a glance

Fact	Detail
Product	kyverno (CVE-2026-124440)
Severity	Critical: 9.9 (CVSS 3.1)
Status	A new CVE record, published by VulnCheck on 8 November 2026. Not on CISA's list of exploited vulnerabilities when we read it on 8 November 2026, which means nothing public says yet that anyone is using it.
Fixed in	1.19.2 (GitHub advisory GHSA-39w8-hv5g-f73w, published 23 October 2026)
Who looks first	Teams running Kyverno 1.19.1 on Kubernetes clusters where namespace tenants — including developers or automated pipelines — have permission to create namespaced Policies.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run Kyverno on any Kubernetes cluster?	Nothing to do for this CVE.	Check the version (next question).
Is the Kyverno version exactly 1.19.1?	This specific flaw does not apply; check related advisories for your version.	Check who can create namespaced Policies (next question).
Can any untrusted user or pipeline create namespaced Policies?	Exploitation risk is lower; still update to 1.19.2 at the next cycle.	Update to 1.19.2 promptly — the full escalation path is open.

What it is

A user who can create policies in one namespace can append a malformed percent-escape to an API call path to bypass the namespace guard, run that call as the admission-controller service account against any namespace or at cluster scope, and from there reach cluster admin.

Why it matters

The record scores this 9.9. CWE-441 means the admission controller acts on a tenant's behalf but is routed to a scope the tenant was never granted. The bypass needs only the ability to create a namespaced Policy — a common developer right on shared clusters — and any malformed percent-escape triggers it; the vendor calls this an open class of inputs, not a single payload to block. Our reading is that any cluster with tenants holding Policy-create rights is at real risk until 1.19.2 is in place. The advisory says the same bypass reopens two escalation paths that 1.19.1 itself was meant to close.

What the vendor's advisory says

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

- The GitHub advisory (GHSA-39w8-hv5g-f73w, published 23 October 2026) says this flaw reopens GHSA-c5qq-7g2q-cpqp (cross-namespace read) and GHSA-5qq8-67g6-4h2w (cluster-scoped write to cluster admin), both of which were fixed in 1.19.1.
- The advisory says the bypass is not a single payload but an open class of inputs: any malformed percent-escape in a trailing query string forces the namespace guard to validate the wrong string.
- Patched version is 1.19.2, per the advisory.

Things people will get wrong

- Versions before 1.19.1 are safe because they predate this flaw. The advisory says 1.19.1 was itself a patch for related escalation flaws (GHSA-c5qq-7g2q-cpqp, GHSA-5qq8-67g6-4h2w). Earlier versions carried those predecessor paths.
- Because low privileges are required, only internal attackers matter. On a shared cluster, any developer or pipeline with Policy-create rights in one namespace qualifies as low-privileged. The record scores this 9.9 precisely because those rights are common.

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 Kyverno version running

- Check the image tag or Helm release for Kyverno. The record names 1.19.1 as the only affected version; 1.19.2 is the fix per the vendor's advisory.

```
helm list -n kyverno
```

Step 2: Identify who can create namespaced Policies

- List RBAC bindings that grant Policy-create access in any namespace. Any principal with that right on a 1.19.1 cluster can follow this escalation path.
- If only cluster administrators can create Policies, the exploitation risk is substantially lower while still requiring the update.

Step 3: Update to 1.19.2

- Update Kyverno to 1.19.2 using the same method used to install — Helm chart, plain manifests, or Operator.
- After rollout, confirm the new version is running and that admission-controller Pods have restarted.

Step 4: Confirm no unexpected webhooks were created

- List all MutatingWebhookConfiguration and ValidatingWebhookConfiguration resources and check for any that your team did not provision.
- The advisory says an attacker can create cluster-scoped webhooks using the admission-controller's service account; unexpected entries warrant investigation.

```
kubectl get mutatingwebhookconfigurations,validatingwebhookconfigurations -o wide
```

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-2026-124440, a privilege-escalation vulnerability in Kyverno 1.19.1. A namespace tenant with only the right to create policies in their own namespace can bypass the namespace guard and reach cluster-admin access. The fixed version is 1.19.2. Could you confirm whether we run Kyverno, which version is deployed, and whether any untrusted users or automated pipelines can create namespaced Policies? If we are on 1.19.1, please arrange an update to 1.19.2 and confirm the running version in writing. Please reply by 15 November 2026.*

The weakness behind it

The record classes this as CWE-441.

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 flaw would likely move a low-privileged namespace tenant to cluster-admin by exploiting the admission-controller's service account; the record describes cross-namespace reads and cluster-scoped webhook creation as outcomes.
Account Manipulation (T1098): Changing an existing account, for example adding permissions or keys, to keep access.	Persistence: staying in	The advisory says an attacker can create cluster-scoped admission webhooks they are not entitled to, which would likely be used to maintain elevated access beyond the initial exploit.
Exploitation of Remote Services (T1210): Exploiting a flaw in a service on another machine to move to it.	Lateral Movement: moving to other machines	The apiCall bypass routes requests across namespace boundaries via the admission controller, suggesting exploitation of that internal Kubernetes service for lateral movement to other namespaces.

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:

> *Kyverno from 1.19.1 before 1.19.2 contains an authorization bypass vulnerability in the apiCall namespace clamp that allows namespace tenants to issue cross-namespace and cluster-scoped API requests by appending malformed percent-escapes to urlPath. Attackers can create namespaced Policies whose apiCall runs as the admission-controller ServiceAccount to read other namespaces' resources and create mutating webhooks, escalating to cluster admin.*

Versions of kyverno the record lists as affected:

- 1.19.1 up to, but not including, 1.19.2

It lists these as unaffected:

- 1.19.2

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-2026-124440
```

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-2026-124440
```

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 state whether restricting Policy-create rights to trusted principals is an effective workaround short of patching to 1.19.2.
- The record does not say whether Kyverno versions before 1.19.1 share this exact percent-escape bypass, though the advisory says those versions carried related escalation flaws.
- 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 8 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-2026-124440: 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
8 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
Kyverno 1.19.1 on a multi-tenant cluster where namespace tenants can create Policies	Medium	This week — update to 1.19.2
Kyverno 1.19.1 on a single-operator cluster with no untrusted policy authors	Medium	Next scheduled patch cycle
Kyverno at a version other than 1.19.1	None	No action for this CVE; review related advisories for your version
Kyverno not deployed in any cluster	None	No action for this CVE

Appendix 4. Definition of done

- Every kyverno 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-2026-124440: https://www.cve.org/CVERecord?id=CVE-2026-124440
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Advisory or release notes read: github.com/kyverno/kyverno/security/advisories/GHSA-39w8-h: https://github.com/kyverno/kyverno/security/advisories/GHSA-39w8-hv5g-f73w
- Advisory or release notes read: www.vulncheck.com/advisories/kyverno-1.19.1-privilege-escalation-: https://www.vulncheck.com/advisories/kyverno-1.19.1-privilege-escalation-via-apicall-namespace-clamp-bypass