Reconciling the Past: Correcting Records for Unfixed Kubernetes CVEs

By **Pushkar Joglekar** **(Broadcom / SIG Security),** **Tabitha Sable** **(Datadog / K8s Security Response Committee / SIG Security)** | Tuesday, May 26, 2026

The Kubernetes project relies on transparency to empower cluster administrators and security researchers. One important way we do that is by publishing CVE records into the Common Vulnerabilities and Exposures database. As part of our ongoing effort to mature the official Kubernetes CVE Feed, we have identified some discrepancies. CVE records for a few older, unfixed issues incorrectly include a *fixed version* field.

The Kubernetes Security Response Committee (SRC) will correct the affected CVE records on June 1, 2026. This may result in vulnerability scanners identifying these vulnerabilities in places where they were previously not detected.

To help reduce confusion, this post provides a technical update on three vulnerabilities that were disclosed in previous years but remain unfixed: **CVE-2020-8561**, **CVE-2020-8562**, and **CVE-2021-25740**.

Why we are updating these records now

While these vulnerabilities have been public for several years, the recent work to generate official Open Source Vulnerabilities (OSV) files revealed that their corresponding CVE records did not accurately reflect their status. Specifically, some records suggested a *fixed* version existed, when in reality, these issues are architectural design trade-offs that cannot be fully remediated through code without breaking fundamental Kubernetes functionality.

Correcting these records is vital for the community for:

- **Automation Fidelity**: Modern vulnerability scanners depend on precise version ranges. Inaccurate *fixed* tags lead to false negatives, giving users a false sense of security.
- **Risk Documentation**: By formalizing these as *unfixed*, we ensure that platform providers and administrators are aware of the persistent need for administrative mitigations.

For completeness, we should also mention that CVE-2020-8554 is an unfixed CVE with a correct CVE record stating that it affects all versions. That record will also be updated to use a more-standardized version number format.

Technical analysis of unfixed architectural risks

The following vulnerabilities will not be fixed by the Kubernetes project. GitHub issues remain the best reference for the technical mechanics of these flaws.

CVE-2020-8561: Webhook redirect in kube-apiserver

- **Severity**: Medium (4.1).
- **The Issue**: The kube-apiserver follows HTTP redirects when communicating with admission webhooks. An actor capable of configuring an AdmissionWebhookConfiguration can redirect API server requests to internal, private networks.
- **Why it remains unfixed**: Restricting this behavior would require breaking the standard HTTP client behavior that many legitimate integrations rely on.
- **Mitigation**: Set the API server log level to less than 10 (to prevent logging response bodies) and disable dynamic profiling (`--profiling=false`) to prevent unauthorized log-level changes.

CVE-2020-8562: Proxy bypass via DNS TOCTOU

- **Severity**: Low (3.1).
- **The Issue**: A Time-of-Check to Time-of-Use (TOCTOU) race condition in the API server proxy allows users to bypass IP restrictions. The system performs a DNS check to validate an IP, but then performs a second resolution for the actual connection, which an attacker can manipulate.
- **Why it remains unfixed**: Fixing this requires pinning resolved IPs in a way that breaks complex split-horizon DNS or dynamic IP environments.
- **Mitigation**: Use a local DNS caching server like dnsmasq for the API server and configure `min-cache-ttl` to enforce consistent responses between the check and the connection.

CVE-2021-25740: Cross-namespace forwarding via Endpoints

- **Severity**: Low (3.1).

- **The Issue**: A design flaw in the Endpoints and EndpointSlice API objects allows users to manually specify IP addresses, which can be used to point a LoadBalancer or Ingress toward backends in other namespaces.
- **Why it remains unfixed**: This is a fundamental feature of the Endpoints API used by many networking tools and operators.
- **Mitigation**: Restrict write access to Endpoints (legacy) and EndpointSlices. Since Kubernetes 1.22, Kubernetes RBAC authorization mode no longer includes those permissions in the default *edit* and *admin* ClusterRoles. That removal applies to clusters created using Kubernetes v1.22; for clusters upgraded from older versions, administrators should manually audit and reconcile the `system:aggregate-to-edit` ClusterRole.

> **Note:**
> On June 1, 2026, these CVE records will be updated to correctly reflect the fact that all versions are affected. You may see them begin to appear in vulnerability scanner results.

Required actions for administrators

The Kubernetes project recommends a *secure by configuration* approach to manage these persistent risks:

Vulnerability	Action item	Severity score (Rating)	Command / configuration
CVE-2020-8561	Restrict Log Verbosity	4.1 (Medium)	Ensure `--v` is set to `< 10` and `--profiling=false`.
CVE-2020-8562	Enforce DNS Consistency	3.1 (Low)	Deploy dnsmasq or a similar caching resolver on control plane nodes.
CVE-2021-25740	Hardened RBAC	3.1 (Low)	`kubectl auth reconcile` to remove Endpoints write access from broad roles.

The RBAC action for CVE-2021-25740 applies when your cluster uses RBAC authorization mode, which is the default for clusters created with standard Kubernetes tooling.

Administrators should independently test and validate these configurations in a non-production environment, assessing the architectural risks against their specific threat model and risk tolerance.

Conclusion: maturity through transparency

The effort to reconcile these records is a sign of a maturing security ecosystem. By moving away from the "patch-only" mindset and accurately documenting architectural debt, the Kubernetes project provides the community with the high-fidelity data needed to secure modern cloud native infrastructure.

We would like to thank the security researchers—QiQi Xu, Javier Provecho, and others—who identified these risks, and the SIG Security Tooling contributors who continue to refine our official feeds. Special shoutout to Rory McCune for sharing information around these CVEs through his blog posts.

← Previous

Next →

Last modified May 23, 2026 at 3:50 PM PST: Schedule article publication (d4124ac1d1)