Upstream CVEs

Below is a list of Common Vulnerabilities and Exposures (CVE) in Kubernetes or Kubernetes related technologies VMware Customer Reliability Engineering (CRE) feels you should know about.

Warning

This list is not exhaustive for every technology you might use in your clusters. You should always check with an individual technology's security announcements and understand any risks before using that technology.

CVE-2019-16276: Go Server vulnerability

The Golang community issued CVE-2019-16276 for a vulnerability in Go's net/http library that causes invalid headers to be normalized and interpreted as valid by an HTTP server. If a reverse proxy in front of a Go HTTP server allows and forwards, but doesn't normalize invalid headers, the Go server could interpret those headers differently than the reverse proxy. See the Kubernetes upstream announcement for more information about this vulnerability.

We strongly recommend that you upgrade to a version of Go where this issue has been resolved. This issue affects all versions of Go prior to 1.13.1 and 1.12.10. It has been fixed in Go 1.13.1 and Go 1.12.10, and Kubernetes 1.16.1, 1.15.5, and 1.14.8 have all been updated to use a fixed Go version.

CVE-2019-16097: Critical Harbor vulnerability enables privilege escalation

The Harbor community issued CVE-2019-16097 for a vulnerability in `core/api/user.go` that allows non-admin users to create admin accounts via the POST `/api/users` API, allowing them full access to the registry and images. To exploit this vulnerability, bad actors must have API access to an instance of the system in a non-admin role to perform privilege escalation. A recent scan found that many registries are exposed to the internet, which widens their exposure to attack. Additionally, if any registry allows self-registration (the default setting), then any individual that can access the Harbor sign in page can exploit this to become an administrator of that Harbor installation.

This vulnerability affects Harbor versions 1.7.0 to 1.8.2. It has been fixed in 1.7.6, 1.8.3, and 1.9.0. If you are using Harbor, we strongly recommend that you upgrade to 1.7.6, 1.8.3, or 1.9.0 to make sure you are not impacted by this vulnerability. Use the Harbor upgrade guide for more information on how to get the new versions.

CVE-2019-11251: `kubectl cp` vulnerability

A vulnerability has been discovered in kubectl v1.13.10, v1.14.6, and v1.15.3 that allows a combination of two symlinks to copy a file outside of its destination directory when using `kubectl cp`. An attacker can use this vulnerability to place a malicious file using a symlink, outside of the destination tree. Read more information about this vulnerability in the community security announcement. This vulnerability is similar to previously reported CVE-2019-1002101 and CVE-2019-11246.

To address this vulnerability, support of symlinks in `kubectl cp` has been removed in versions 1.16.0 and later. It's recommended to use a combination of exec+tar instead in 1.16.0 and later. See more information about the fix in the upstream pull request.

A second fix has been made to 1.15.4 and backported to 1.14.7 and 1.13.11. This changes the `kubectl cp` un-tar symlink logic to unpack the symlinks after all the regular files have been unpacked. This method guarantees that a file can't be written through a symlink.

VMware CRE recommends upgrading to a version where this issue has been resolved.

Kubernetes patch for Go vulnerability

Go 1.11.13 and 1.12.8 have been released to fix CVE-2019-9512 and CVE-2019-9514. These CVEs disclose potential Denial of Service attack vectors in Go that affect server implementations of the HTTP/2 protocol. These vulnerabilities allow untrusted clients to allocate an unlimited amount of memory, until the server crashes. All previous versions of Go are affected.

Kubernetes 1.13.10, 1.14.6, and 1.15.3 have also been patched to use a version of Go where these issues have been fixed. All previous versions of Kubernetes and Kubernetes components are affected by these issues as they use unpatched versions of Go. Future releases of Kubernetes will use the patched versions of Go.

The new Kubernetes releases have been patched to use the fallowing versions of Go:

- Kubernetes v1.15.3 - 901.12.8
- Kubernetes v1.14.6 - 901.12.8
- Kubernetes v1.13.10 - 901.11.13

We strongly recommend that you upgrade to a version of Kubernetes where this issue has been resolved. See the downloads page for 1.13.10, 1.14.6, or 1.15.3 to get started.

Envoy 1.11.1 Security Release

The Envoy security team released 1.11.1 to fix vulnerabilities that cause excessive CPU or unbounded memory growth and resource starvation, as a result of unpatched Envoy proxies interacting with specially crafted packets. The packets in question use PING, PRIORITY, HEADERS, and SETTINGS Frame types in addition to an empty payload. This issue pertains to the following CVEs:

- CVE-2019-9512
- CVE-2019-9513
- CVE-2019-9514
- CVE-2019-9515
- CVE-2019-9518

This issue affects Envoy versions 1.5.0, 1.6.0, 1.7.0 - 1.17.1, 1.8.0, 1.9.0 - 1.9.1, 1.10.0, 1.11.0, and 1.12.0. For more details on this vulnerability, please read the community security announcement.

Customers should immediately upgrade Envoy to version 1.11.1 and update Contour to version 0.14.2. Envoy version 1.11.1 is available on the Envoy Releases page. Contour version 0.14.2 was released to correct this issue, and is available on the Contour Releases page. Also note that Envoy is the underlying technology for other open source projects in addition to Contour, such as Istio, Ambassador, and Cillium. These projects are also affected. Check the documentation associated with each affected product for appropriate methods of remediation.

CVE-2019-11247: API server vulnerability

The API server mistakenly allows access to a cluster-scoped resource if the request is made as if the resource were namespaced. Authorizations for the resource are enforced using roles and role bindings within the namespace, meaning that a user with access to a resource in one namespace could create, view, update, or delete the cluster-scoped resource according to their namespace role privileges. For more details on this vulnerability, please read the community security announcement.

This issue will be resolved in 1.13.9, 1.14.5, 1.15.2, and all future releases. This issue currently affects versions 1.7 to 1.12.10, 1.13.0 to 1.13.8, 1.14.0 to 1.14.4, and 1.15.0 to 1.15.1.

We recommend upgrading to a version of Kubernetes where this issue has been fixed. For users who cannot upgrade, we recommend removing authorization rules that grant access to cluster-scoped resources within namespaces. To check for exploitation of this vulnerability, search API server request logs for namespaced access of a cluster-scoped resource. Logging of API server requests must be enabled.

CVE-2019-11245: kubelet vulnerability

The container UID changes to root after the first restart, or if the image is already pulled to the node. When a container runs for the first time on a node, it respects the UID set by the container image. On the second run, the container will run as UID 0, or root, which can be an undesired escalated privilege. For more details on this vulnerability, please read the community security announcement.

This affects kubelet versions 1.13.6 and 1.14.2, but will be resolved in kubelet versions 1.13.7 and 1.14.3.

We recommend specifying `runAsUser` or `mustRunAsNonRoot:true` directives in pods to prevent starting as root, or downgrading kubelets to versions 1.13.5 or 1.14.1 as instructed by your Kubernetes distribution. Additionally, users should avoid upgrading until versions 1.13.7 and 1.14.3 are released.

CVE-2019-1002101: kubectl vulnerability

The `kubectl cp` command could allow a directory traversal, replacing or deleting files on the user's workstation. For more details on this vulnerability, please read the community security announcement.

This affects all previous Kubernetes versions, but has been fixed in 1.11.9, 1.12.7, 1.13.5, and 1.14.0 and will be fixed in all future Kubernetes versions.

We strongly recommend upgrading your kubectl version to a version where this issue has been fixed.