CVE-2018-1002105
Security Release Post Mortem

Authors: liggitt@google.com, tpepper@vmware.com, deads@redhat.com, wfender@google.com, cjcullen@google.com
Status: published
Last updated: 2018-12-17

Impact

- An API call to any aggregated API server endpoint can be escalated to perform any API request against that aggregated API server. In default configurations, all users (authenticated and unauthenticated) are allowed to perform discovery API calls that allow this escalation.
- A pod exec/attach/portforward API call can be escalated to perform any API request against the kubelet API on the node specified in the pod spec (e.g. listing all pods on the node, running arbitrary commands inside those pods, and obtaining the command output). Pod exec/attach/portforward permissions are included in the admin/edit/view RBAC roles intended for namespace-constrained users.

See #71411 for details of CVE-2018-1002105.

Timeline

(All times are in PT)

2018-08-03 Reported publicly to Rancher as bug report against proxy component
2018-11-01 Diagnosed as TCP reuse issue by Rancher dev
2018-11-02 12:15PM Reported privately to Kubernetes Product Security Team (PST) by Rancher dev as security-sensitive issue
2018-11-02 7:45PM Report acknowledged by PST (bphilips)
2018-11-03 10:30PM Proof of concept exploit for pod exec produced (liggitt)
2018-11-05 9:00PM Fix opened for review in kubernetes-security/kubernetes#23 (liggitt)
2018-11-06 12:20AM Fix review begun (sttts, deads2k, wfender)
2018-11-06 1:30PM Proof of concept exploit for anonymous upgrade attack produced (liggitt)
2018-11-06 9:00PM Fix review completed (sttts, deads2k, wfender, lavalamp)
2018-11-07 12:00AM Green CI
2018-11-09 12:00PM kubernetes-distributors-announce email sent
2018-11-26 3:00AM Patch PRs opened on public repo and merged:
    1.10 - https://github.com/kubernetes/kubernetes/pull/71415
    1.11 - https://github.com/kubernetes/kubernetes/pull/71414
    1.12 - https://github.com/kubernetes/kubernetes/pull/71413
    master - https://github.com/kubernetes/kubernetes/pull/71412
2018-11-26 10:45AM Releases published (v1.10.11, v1.11.5, v1.12.3)

2018-12-03 9:00AM Security announcement published, details added to CVE issue

Root Causes and Trigger

- Existing proxy components were reused for new use-cases as the Kubernetes API server evolved, without verifying the security-related aspects functioned properly
- Error-handling paths for those proxy components were not fully tested
- Kubernetes API server proxy components still use http/1.1 upgrade-based connection tunneling, which does not distinguish between request data sent by the apiserver while establishing the backend connection, and data sent by the requesting user
- High and low-privilege API requests to aggregated API servers are proxied via the same component with the same high-permission transport credentials

Lessons Learned

What worked

- Private review cycle went smoothly
- Feedback on timeline for distributors to update was positive
- Canonical release plan doc worked well for release coordination
- Embargo was maintained

What can go better

- Some confusion around the Availability metric of the CVSS (availability severity was based on the ability of an attacker to disrupt workloads, affecting availability)
- No ability to cut patch releases without making source public via PRs
- CI in private repo had atrophied, made getting CI signal difficult
- Patch managers' time zones made cutting releases during North America business hours cut into their evenings
- Confusion about impact from users
    - Questions about vulnerable versions from distributors (non-OSS builds)
    - Two known escalation paths (one typically exploitable by anonymous requests, one typically not) confused users trying to determine if their clusters were vulnerable
- Distributor handling of embargoed information was unclear
    - "Non-embargoed patch, embargoed security implications" was confusing
    - Include security team, fix team, release team in review of the distributor pre-disclosure prior to sending it
        - Help catch any technical issues
        - Help point out questions about what is allowed with the embargoed data
    - Pre-disclosure announcement needed more explicit "you may do X" and "you may not do X" statements
        - Distributors that ship binaries weren't sure if they could ship prior to disclosure (because the patch wasn't under embargo)

        - Distributors that host solutions weren't sure if they could upgrade hosted environments prior to disclosure
        - Is there a channel for distributors to ask for clarification of embargo handling for a specific issue?
    - Coordination between distributors was difficult
        - How do distributors know who has access to the embargoed information?
        - Membership of the distributors list is not visible
        - Is there a channel for discussion between members of the distributors list
        - Multiple distributors re-implemented proof of concept exploit code (or at least a programmatic test that shows the patch was effective). Should we target including a test like that in the pre-disclosure?
        - The old (non-kubernetes.io) mailing lists are confusing
- Announcement friction
    - Security process refers to kubernetes-users@googlegroups.com which was deprecated in favor of https://discuss.kubernetes.io, found out when the announcement email bounced
    - Posting announcement to https://discuss.kubernetes.io was held for moderation (need to make sure all PST members responsible for posting are approved for all channels)
    - Last-minute coordination was required to post to slack #announcements channel

Where we got lucky

- Security issue was reported privately
- Timeline for disclosure made it in prior to 1.13.0 so it could be included in release announcements
- Realized the potential for exploit before handling publicly

Where we got unlucky

- Bug was discussed (without security implications) months earlier in https://github.com/rancher/rancher/issues/14931
- Since there was no other content since 1.10.10 and its release announcement was exactly in line with the 1.11 and 1.12 release Nov. 26, but its changelog was solely "Fixes an issue with stuck connections handling error responses (#71415, @liggitt)" this shines a light such that trivial source analysis of the only fix in 1.10.11 could have led to folks realizing there was a severe CVE in flight a whole week ahead of the embargo.

Followup

Action Item	Issue, PR, or owner
Investigate switching from http/1.1 upgrades to http/2 tunneling	#7452

Lock down allowed upgrade protocols in apiserver proxy to spdy and websockets	#72112
Add test coverage of error conditions for proxy components	#71768
Aggregator: serve downstream apiserver discovery documents from cache rather than proxying	#71754
Aggregator: prevent upgrade on known paths that do not serve upgrade requests (like discovery, healthz, etc)	#72113
Identify key components for security audit (either fold into existing planned audit or schedule dedicated ones)	liggitt
Investigate adding fuzz testing of request handling	#59829
Investigate defaulting anonymous discovery requests to disallowed. Specifically, the system:discovery and system:basic-user rolebindings	#72115
Investigate adding per-request binding or assertions that would let backends verify a particular request was intentionally forwarded	#72116
Investigate alternatives to proxying through the apiserver for things like pod/exec	#72117
Consider multiple patch managers per release, spread across timezones	#368
freeze the old distributor/security lists, move all members to new lists	#326
Figure out how to publish membership of the distributors-announce list	#416
Update security process to clarify how distributors on the pre-disclosure list should communicate with each other about embargoed data	#417
Add distributor pre-announcement template that has placeholders for explicit “you may do X with the embargoed patch or information” statements, add instructions for how to ask for clarification (who to contact), and inclusion of tests for cluster vulnerability	#418
Add announcement template that includes placeholders for concrete steps to determine if a cluster has a vulnerable version/configuration	#419
Update security release process and disclosure checklist to include fix team, release team, and Kubernetes Product Security Team in review of pre-disclosure announcement draft	#420