~~Bug 1177361 - (CVE-2020-8030) VUL-0: CVE-2020-8030: skuba: Insecure /tmp usage when joining node to cluster~~

Status:	RESOLVED FIXED
Classification:	Novell Products
Product:	SUSE Security Incidents
Component:	Audits
Version:	unspecified
Hardware:	Other Other
Priority:	P3 - Medium **Severity**: Normal
Target Milestone:	---
Assigned To:	Containers Team
QA Contact:	Security Team bot
URL:	
Whiteboard:	CVSSv3.1:SUSE:CVE-2020-8030:5.3:(AV:L...
Keywords:	
Depends on:	
Blocks:	
	Show dependency tree / graph

- Create test case
- Clone This Bug

Reported:	2020-10-06 11:35 UTC by Johannes Segitz
Modified:	2020-12-11 17:16 UTC (History)
CC List:	6 users (show)
See Also:	
Found By:	---
Services Priority:	
Business Priority:	
Blocker:	---

Attachments

Add an attachment (proposed patch, testcase, etc.)

Note

You need to log in before you can comment on or make changes to this bug.

Johannes Segitz 2020-10-06 11:35:08 UTC Description

```
While looking into our registry I noticed that skuba uses constant filenames in
/tmp and also sets insecure permissions:

1, Static name: /tmp/crio.conf.d
Didn't manage to exploit this, but I didn't try hard. With the right timing this
might be possible.

2, Static name + insecure permissions: /tmp/kubeadm-init.conf
This is more problematic. With simple tools (inotifywait) I couldn't exploit it to
set the permissions via symlinks, but that might be possible. More problematic is
that the file is created with 644 and is therefore readable by all users. This
leaks the bootstrapToken. Together with unsafeSkipCAVerification this is a bad
combination because "If an attacker is able to steal a bootstrap token via some
vulnerability, they can use that token (along with network-level access) to
impersonate the control-plane node to other bootstrapping nodes."

Unfortunately it not only leaks this token, but if the file exists then it will be
used without any precautions. E.g. if a user creates
-rw-r--r-- 1 sles users 495 Oct  6 08:20 /tmp/kubeadm-init.conf
this file will be overwritten. By carefully timing another write kubeadm can be
used with a changed configuration to e.g. join the control plane instead of
becoming a worker.

I'm not clear why the information leak happens as the permission seem to be
correctly set in pkg/skuba/actions/cluster/init/init.go
303         if err := ioutil.WriteFile(skuba.KubeadmInitConfFile(),
initCfgContents, 0600); err != nil {

As I was actually reviewing the registry and not skuba I didn't dive into this
more.

Please have a look. I think we need a CVE for issue 2.
```

Johannes Segitz 2020-10-06 11:37:38 UTC Comment 1

```
This is an embargoed bug. This means that this information is not public.

Please do NOT:
- talk to other people about this unless they're involved in fixing the issue
- make this bug public
- submit this into OBS (e.g. fix Leap/Tumbleweed) until this bug becomes public
(e.g. no EMBARGOED tag on the header)

Consult with security team if you think that the issue is public and the bug is
still private (e.g. subject still contains "EMBARGOED"). Please do NOT make the bug
public yourself.

Please be aware that the SUSE:SLE-15-SP3:GA codestream is available via OBS, so do
NOT submit there before this is public.

These are the steps that are asked from you:
1, Your primary responsibility is to submit a fix for this issue. Here's a how-to
for submitting packages for maintenance releases in IBS:

https://confluence.suse.com/display/maintenance/How+to+Submit+Packages+or+Containers+
   Apart from the GA codestreams mentioned above, you can submit to IBS anytime.
This is private and allows us to start testing as soon as possible.
2, We also want to fix openSUSE if it's affected.
   $ is_maintained $PACKAGE
   will tell you if the package is inherited from SLES or if it is branched for
openSUSE. There are two cases:
  - It's coming from SLES: The update will automatically be released for openSUSE.
Nothing to do for you.
  - It's branched for openSUSE: You need to submit AFTER the bug became public, to
```

the current openSUSE codestreams.
   For openSUSE Factory please submit to the devel project of you package AFTER the bug became public.

Security will then take the following steps:

- We wait for your submission and package them into an incident for QA testing. The QA tester might reach out to you if they find issues with the update.
- Once the coordinated release date (CRD), the date this issue should become public, is reached (or for internal findings: once we're done testing), we remove the EMBARGOED tag from this bug and publish the updates.
- Only if the bug here is public you may submit to public repositories (OBS).

You can contact us at:

* IRC: irc.suse.de #security
* RocketChat: https://chat.suse.de/channel/security
* Email: security-team@suse.de

Internal CRD: 2021-01-04 or earlier

Comment 2

cc Rafa (for skuba), Sascha (for cri-o)

Sascha Grunert 2020-10-07 07:01:35 UTC

Comment 3

(In reply to Klaus Kämpf from comment #2)

> cc Rafa (for skuba), Sascha (for cri-o)

Looks like the issue is only scoped to skuba and how it handles the config setup. Feel free to reach out to me if I miss anything.

Rafael Fernández López 2020-10-07 08:37:52 UTC

Comment 4

(In reply to Johannes Segitz from comment #0)

> While looking into our registry I noticed that skuba uses constant filenames
> in /tmp and also sets insecure permissions:
>
> 1, Static name: /tmp/crio.conf.d
> Didn't manage to exploit this, but I didn't try hard. With the right timing
> this might be possible.

ACK. This is an issue and we should use a tempdir here.

> 2, Static name + insecure permissions: /tmp/kubeadm-init.conf
> This is more problematic. With simple tools (inotifywait) I couldn't exploit
> it to set the permissions via symlinks, but that might be possible. More
> problematic is that the file is created with 644 and is therefore readable
> by all users. This leaks the bootstrapToken. Together with
> unsafeSkipCAVerification this is a bad combination because "If an attacker
> is able to steal a bootstrap token via some vulnerability, they can use that
> token (along with network-level access) to impersonate the control-plane
> node to other bootstrapping nodes."
>
> Unfortunately it not only leaks this token, but if the file exists then it
> will be used without any precautions. E.g. if a user creates
> -rw-r--r-- 1 sles users 495 Oct 6 08:20 /tmp/kubeadm-init.conf
> this file will be overwritten. By carefully timing another write kubeadm can
> be used with a changed configuration to e.g. join the control plane instead
> of becoming a worker.
>
> I'm not clear why the information leak happens as the permission seem to be
> correctly set in pkg/skuba/actions/cluster/init/init.go
> 303   if err := ioutil.WriteFile(skuba.KubeadmInitConfFile(),
> initCfgContents, 0600); err != nil {

This code only executes locally on the bootstrapper machine. The flow looks like this:

- User runs `skuba cluster init` in a local workstation -- this operation writes a set of files on the local machine, at the current folder.
- User tweaks configuration locally if desired.
- User runs `skuba node bootstrap`.
  - This command will copy `kubeadm-init.conf` (desired to be hardcoded) from the current folder to the remote machine. (*)
  - Once copied, this command will execute `kubeadm init` on the remote machine with the copied `kubeadm-init.conf`. (**)

So, looking at the local operation (cluster/init/init.go) it's fine (and desired) that this file is hardcoded to `kubeadm-init.conf`. What needs fixing is the logic that performs (*) and (**) -- because on the remote machine [the machine that will become part of the Kubernetes cluster] we want to prevent timing attacks on these files.

So, the course of action that I suggest (please confirm if that looks correct) would be:

- Keep generating `kubeadm-init.conf` on the local machine during `skuba cluster init`. This is an offline command, expected to write some files locally to the current machine, so the user can customize some of them if desired.

- When executing `skuba node boostrap` or `skuba node join`, copy the different files to the remote machine being bootstrapped, making sure that:
  - If the target file exists on the remote machine, make sure that its mode is at least as restrictive as the one we would have created before writing any real contents.
  - A temporary directory is used on the remote machine if any kind of processing of the file is needed prior to copy to final destination.

@Johannes, does this sound good?

Johannes Segitz 2020-10-08 11:23:17 UTC

Comment 5

> - Keep generating `kubeadm-init.conf` on the local machine during `skuba cluster

That is fine since the user will likely do this in a directory they control. Also they seem to be created 600 anyway

> - When executing `skuba node boostrap` or `skuba node join`, copy the different f
> - If the target file exists on the remote machine, make sure that its mode is at
> - A temporary directory is used on the remote machine if any kind of processing

I would suggest that a temporary, randomly named directory is created ala mktemp -d. The config files should be copied into this directory. Then the case where the files already exist don't have to be expected and even if the file permissions of the config files are suboptimal (which the of course shouldn't be) they would be protected.

Plese use CVE-2020-8030 to track the second issue. I don't think we need one for the first.

Rafael Fernández López 2020-10-26 11:01:41 UTC Comment 6

Adding David Ko and Jenting.

jenting hsiao 2020-11-05 03:23:16 UTC Comment 7

v4.2 fix: https://github.com/SUSE/skuba/pull/1424
v4.5 fix: https://github.com/SUSE/skuba/pull/1415

code merged, waiting to be released.

Swamp Workflow Management 2020-12-11 17:16:33 UTC Comment 10

SUSE-SU-2020:3761-1: An update that solves four vulnerabilities and has 11 fixes is now available.

Category: security (important)
Bug References:
1172270,1173055,1173165,1174219,1174951,1175352,1176225,1176578,1176903,1176904,11773
CVE References: CVE-2020-15106,CVE-2020-8029,CVE-2020-8564,CVE-2020-8565
JIRA References:
Sources used:
SUSE CaaS Platform 4.5 (src): caasp-release-4.5.2-1.8.2, cri-o-1.18-1.18.4-4.3.2, etcd-3.4.13-3.3.1, helm2-2.16.12-3.3.1, helm3-3.3.3-3.8.1, kubernetes-1.18-1.18.10-4.3.1, patterns-caasp-Management-4.5-3.3.1, skuba-2.1.11-3.10.1, velero-1.4.2-3.3.1

NOTE: This line indicates an update has been released for the listed product(s). At times this might be only a partial fix. If you have questions please reach out to maintenance coordination.