New issue

Jump to bottom

podman top not work with userns=keep-id container #10941

Closed **pendulm** opened this issue on Jul 15, 2021 · 37 comments

Assignees

Labels kind/bug

pendulm commented on Jul 15, 2021 Contributor

Is this a BUG REPORT or FEATURE REQUEST? (leave only one on its own line)

/kind bug

Description

Steps to reproduce the issue:

1.

top with --userns=keep-id container

```
[mike@x240 podman]$ podman  run --name redis1 -d --userns=keep-id redis
d2f77ab7b4ebbd22a36b9853d28f6bdcdc452a7544f5432fea79d6f9006c151e
[mike@x240 podman]$ podman top redis1  user,huser
Error: error executing "nsenter -U -t 1 cat /proc/1/status": exit status 127
```

2.

top with normal container

```
[mike@x240 podman]$ podman  run --name redis2 -d  redis
e87792f5a1c5f8bf850cbee5ce91fef259819d9f790b061f864fcec820a5c53f
[mike@x240 podman]$ podman top redis2  user,huser
USER        HUSER
redis       100998
```

3.

Describe the results you received:

Describe the results you expected:

Additional information you deem important (e.g. issue happens only occasionally):

Output of `podman version`:

```
Version:      3.2.2
```

openshift-ci bot added the kind/bug label on Jul 15, 2021

vrothberg commented on Jul 15, 2021 Member

Thanks for reporting, **@pendulm**.

That's something we need to fix in `containers/psgo`. Not yet sure if we need to pass some information from Podman down to psgo or if it can auto-detect.

vrothberg commented on Jul 15, 2021 Member

I am a bit lost. It looks like we shouldn't join the user NS but do. **@giuseppe**, do you know what's going on?

giuseppe commented on Jul 15, 2021 Member

`nsenter -U -t 1 cat /proc/1/status` looks like the wrong command to run? Why is it trying to join PID 1?

Does it work differently if you run `podman unshare podman top redis1 user,huser` ?

giuseppe commented on Jul 15, 2021 Member

From what I can see, the error seems to be in the target pid used to join the user namespace.

The command should look like `nsenter -U -t $CONTAINER_PID cat /proc/1/status`

vrothberg commented on Jul 28, 2021 Member

> `nsenter -U -t 1 cat /proc/1/status` looks like the wrong command to run? Why is it trying to join PID 1?
>
> Does it work differently if you run `podman unshare podman top redis1 user,huser` ?

Same issue.

giuseppe commented on Jul 28, 2021 — Member

I think it is trying to join the wrong pid, it should not be 1

p0da commented on Aug 15, 2021 • edited

Any update? I encountered this as well, however I seem to be getting another error:

```
$ podman run --detach --userns keep-id --name a busybox sleep 20
$ podman top a
Error: error executing "nsenter -U -t 1 cat /proc/1/status": exec: "nsenter": executable file not found in $PATH
```

Without keep-id everything works perfectly.

rhatdan commented on Aug 16, 2021 — Member

@vrothberg PTAL

rhatdan assigned **vrothberg** on Aug 16, 2021

github-actions bot commented on Sep 15, 2021

A friendly reminder that this issue had no activity for 30 days.

github-actions bot added the stale-issue label on Sep 15, 2021

rhatdan commented on Sep 16, 2021 — Member

Seems to work on main branch.

```
$ podman -v
podman version 4.0.0-dev
$ podman run --detach --userns keep-id --name a busybox sleep 20
Resolved "busybox" as an alias (/etc/containers/registries.conf.d/000-shortnames.conf)
Trying to pull docker.io/library/busybox:latest...
Getting image source signatures
Copying blob 24fb2886d6f6 done
Copying config 16ea53ea7c done
Writing manifest to image destination
Storing signatures
622e891d74910139caa659567e326218a88b141462b080d5cd71f13dbb409d51
$ podman top a
USER        PID         PPID        %CPU        ELAPSED       TTY         TIME        COMMAND
dwalsh      1           0           0.000       5.234386111s  ?           0s          sleep 20
```

 **rhatdan** closed this as completed on Sep 16, 2021

---

dcermak commented on Jan 3 — Contributor

I can reproduce this issue with the latest podman version from main as well as with 3.4.4, but only with certain images.

For example:

```
$ podman run --userns=keep-id --rm -d registry.opensuse.org/opensuse/tumbleweed-dnf sleep infinity
$ podman top $CONTAINER_ID
Error: error executing "nsenter -U -t 1 cat /proc/1/status": exec: "nsenter": executable file not fou
$ podman run --userns=keep-id --pull always --rm -d registry.opensuse.org/opensuse/busybox sleep infi
$ podman top $CONTAINER_ID
Error: error executing "nsenter -U -t 1 cat /proc/1/status": exit status 1
```

whereas the above commands work for `docker.io/alpine` or `registry.opensuse.org/opensuse/leap`.

I have tried using strace to figure what exactly is going wrong here, but I fail to find the place where the actual issue appears.

 **rhatdan** reopened this on Jan 3

---

rhatdan commented on Jan 3 — Member

@vrothberg PTAL

github-actions bot removed the stale-issue label on Jan 3

cyphar commented on Jan 5 • edited

I took a look at this (it's happening with Tumbleweed images). It turns out it just boils down to `util-linux` not being present in Tumbleweed containers -- installing `util-linux` fixes the issue for such containers. The reason why busybox works is because it has an `nsenter` binary (or rather a busybox entrypoint).

However I'm slightly more confused why `nsenter` is necessary at all (and not in the regular case), and why the `nsenter` inside the container is being used. I get this is happening inside `github.com/containers/psgo` (in particular `readStatusUserNS`) but given that we have the PID 1 details for the container, why join the container namespace at all (and why do it from inside the container itself -- surely if you've joined the container you don't need to join the user namespace of PID 1 -- surely you're already in the right user namespace?).

EDIT: Ah, the join namespace code gets triggered automatically if the namespaces don't match. Why isn't `podman top` joining the user namespace in the `keep-ids` case?

The `registry.opensuse.org/opensuse/busybox` container is failing for a separate reason, though I haven't figured out why. In both busybox images `nsenter -U -t 1 cat /proc/1/status` fails due to an `-EINVAL` from `setns(3, CLONE_NEWUSER)` but I haven't figured out whether this is due to the seccomp profile of the container and why this doesn't affect `podman top` on the Docker Hub busybox container but it does affect the openSUSE one...

rhatdan commented on Jan 5 Member

@vrothberg is on PTO and will be back on Monday, so he should be able to answer. I think the issue is psgo attempts to show IDs from the User Namespace perspective as well as from the hosts. So entering the User Namespace is attempting to show what the HUSER versus USER, HGROUP versus GROUP of processes within the container.

vrothberg commented on Jan 10 Member

I need some time to look into it since I haven't looked at this code in a long while.

cyphar commented on Jan 10 • edited

A quick update from looking at this again:

This happens no matter what userns mode you use, so long as the container uses user namepsaces you hit this issue.

The `nsenter` is coming from github.com/containers/psgo, seemingly podman top hasn't joined the user namespace of the container. It seems that either `podman top` should join all of the namespaces (including the user namespace) or `psgo` should be called from the host so you can use the host `nsenter` binary. Given that `psgo` appears to be able to handle namespaces, it might not be necessary to join the container at all outside of calling `psgo` ? I suspect that not joining the user namespace was a security decision, though as long as `podman top` doesn't do an `exec` inside the user namespace and you have the non-dumpable bit set you should be relatively safe.

However if the only reason the namespace is being joined is in order to convert uids and gids, this can be done in Go entirely by just reading `/proc/1/[ug]id_map` and then applying the mappings to all of the users. That way podman doesn't need to cross any security boundaries and it would probably make things faster as well.

cyphar commented on Jan 11 • edited

It should also be noted that this behaviour means that `podman top` is relying on code inside the container as part of its operations, but it would be trivial for a malicious container to create a bad `nsenter` binary that returns different results. I don't know how hardened psgo is against bad input, but at the very least this could result in `podman top` returning bad information to the caller.

I'm also fairly sure that `podman top` is not being run with all of the container security profiles applied (right now it appears it runs as root without joining the container user namespaces which is a bit concerning), which means a container could (by placing a malicious binary in `/bin/nsenter`) potentially make syscalls and do other operations that the container itself couldn't make. If we had a bug like this in runc I would probably consider assigning a CVE for it, but I don't know how much of an issue this is for podman.

If `nsenter` is only being used to translate UIDs and GIDs, then it seems a lot safer to me to be doing the translation without running code in the container. At the very least, it seems inadvisable to run code inside the container like this (in fact it smells a bit like a security bug IMHO).

  **cyphar** mentioned this issue on Jan 11

internal: proc: do not join the process user namespace containers/psgo#92

Merged

cyphar commented on Jan 11

containers/psgo#92 should resolve the issue, though I'm not quite sure how to test whether the mapping functionality works. I couldn't figure out how to get the mapping functionality to work with the `bin/psgo` sample binary even on HEAD.

vrothberg commented on Jan 12 — Member

With containers/psgo#92 merged, we can close the issue. It'll get vendored before we cut a new release.

 **vrothberg** closed this as completed on Jan 12

---

cyphar commented on Jan 12

@vrothberg Did you plan to make a security advisory for the issue? Unless I'm mistaken this is technically a privilege escalation vector (though not a complete escalation since most of the namespaces are still joined).

vrothberg commented on Jan 12 — Member

> **@vrothberg** Did you plan to make a security advisory for the issue? Unless I'm mistaken this is technically a privilege escalation vector (though not a complete escalation since most of the namespaces are still joined).

Thanks for checking! I am not (yet) convinced there's a security issue since we joined the user namespace and mount namespace of the container. But I may be off. Can you elaborate how we could escalate? We should probably move over to email (rothberg@redhat.com) to not paste an attack here.

dcermak commented on Mar 28 — Contributor

This issue has now been fixed in the `main` branch, but released podman versions like 3.4 still suffer from this bug. Could this be backported to the 3.4 branch as well please?

rhatdan commented on Mar 28 — Member

We don't backport unless it is to supported Branches like in RHEL, and even there we frown on it.

vrothberg commented on Mar 28 — Member

> We don't backport unless it is to supported Branches like in RHEL, and even there we frown on it.

That doesn't apply to the current state since we cannot bump Fedora 35 to v4.0. We actually did backport quite some patches to Podman v3.4 because of that.

dcermak commented on Mar 29 — Contributor

Daniel J Walsh ***@***.***> writes:

> We don't backport unless it is to supported Branches like in RHEL, and even there we frown on it.

This issue most certainly impacts RHEL as well. It impacts everything
but the main branch (and possibly the 4.0 release)

nickthetait commented on Mar 29

I'm trying to determine the security impact for this bug. It might deserve assigning a CVE (which I can take care of). Forgive my basic questions as I'm not very familiar with Podman yet. Below are my guesses/questions to the CVSS score, used in determining severity. Please correct me if any assumptions are wrong.

Attack Vector
Local - An attacker would need direct access to the host machine?

Attack Complexity
High
Some restrictions are beyond the attackers control: it must be a container using a type of `--userns`. This is not the default correct?
Another hurdle would be creating a bad `nsenter` binary.

Privileges Required
Low - Any regular user could execute `podman top`

User Interaction
None - No phishing required, attacker can just run the command at their leisure. Or would an attacker implant their binary and wait for an admin to run `podman top` which could trigger something evil?

Scope
Unchanged

Confidentiality Impact
Low? - Help me understand the implication of this statement: "since most of the namespaces are still joined". What boundaries remain around the container?

Integrity Impact
None?

Availability Impact
None?

vrothberg commented on Mar 30 — Member

> Local - An attacker would need direct access to the host machine?

No. As an attacker I would try push a malicious image on a public registry and wait for users to run it and run `podman top`.

> Some restrictions are beyond the attackers control: it must be a container using a type of --userns. This is not the default correct?

It is largely independent from how the container is run.

> Another hurdle would be creating a bad nsenter binary.

I don't think that's a hurdle. A simple Dockerfile is sufficient.

> None - No phishing required, attacker can just run the command at their leisure. Or would an attacker implant their binary and wait for an admin to run podman top which could trigger something evil?

The latter. As mentioned above, I would try to push an evil image on a public registry.

> Low? - Help me understand the implication of this statement: "since most of the namespaces are still joined". What boundaries remain around the container?

It is possible to send signals to the host's systemd which allows, in principle, to reboot the machine. AFAICS, we're not in the user NS of the container but the main one of Podman (if run rootless). We're only in the container's mount NS.

nickthetait commented on Mar 30

That helps a lot! I will be assigning a CVE.

Can we come up with a better title for the security issue? The current title "podman top not work with userns=keep-id container" was how it was discovered but the userns switch is actually irrelevant to the security impact. How about "Privilege escalation in 'podman top'"?

Podman will need to be updated to use a fixed version of psgo right?

vrothberg commented on Mar 31 Member

> How about "Privilege escalation in 'podman top'"?

Sounds good to me, thanks.

> Podman will need to be updated to use a fixed version of psgo right?

v4.0 has the fix. Older ones need to be updated.

cyphar commented on Apr 4 • edited

> Low? - Help me understand the implication of this statement: "since most of the namespaces are still joined". What boundaries remain around the container?
>
> It is possible to send signals to the host's systemd which allows, in principle, to reboot the machine. AFAICS, we're not in the user NS of the container but the main one of Podman (if run rootless). We're only in the container's mount NS.

Yeah we're only in the mount ns. There are a few other attacks you should be able to do carry out in addition to the one mentioned (you might be able to get access to the host filesystem using `open_by_handle_at` using a similar exploit to CVE-2014-3519 because the seccomp profile is not applied -- and from there you have full system filesystem access).

You can also probably access any databases on the host through SYSV memory sharing and so on, and you could intercept (and send out) any network traffic you like. Basically, you can probably gain full access to the host with a clever enough exploit AFAICS.

The impact should be high.

nickthetait commented on Apr 5

Thanks for helping figure out the severity. CVE-2022-1227 has been assigned.

dcermak commented on Apr 6 Contributor

Nick Tait ***@***.***> writes:

> Thanks for helping figure out the severity. CVE-2022-1227 has been assigned.

Does that mean we'll get a backport to 3.4?

vrothberg commented on Apr 6 Member

Yes.

cyphar commented on Jun 14

@vrothberg Now that there is a CVE assigned, can you add me to the credits section of the vulnerability entry? Thanks.

vrothberg commented on Jun 14 Member

> **@vrothberg** Now that there is a CVE assigned, can you add me to the credits section of the vulnerability entry? Thanks.

That is outside my competences but **@nickthetait** may be able to help.

nickthetait commented on Jun 15

Yep, done.

vrothberg commented on Jun 16 Member

Thank you, **@nickthetait**!

Assignees

vrothberg

Labels

kind/bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

8 participants