New issue

Jump to bottom

Source IP always 127.0.0.1 in rootless Podman 1.8.0 #5138

Closed **basvdlei** opened this issue on Feb 9, 2020 · 25 comments · Fixed by #9052

basvdlei commented on Feb 9, 2020

/kind bug

Description

For a rootless container the source IP of incoming packets on a publish port is always `127.0.0.1`. Even if the request is made from an external host.

Steps to reproduce the issue:

1. Start a NGINX container:

```
machine-1$ podman run -p 8888:80 docker.io/library/nginx:latest
```

2. Make a request from another node.

```
machine-2$ curl http://machine-1:8888
```

3. Look at the source ip of the request in NGINX stdout log:

```
127.0.0.1 - - [09/Feb/2020:21:54:17 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.66.0" "-"
```

Describe the results you received:

The logged source address is always 127.0.0.1

Describe the results you expected:

The logged source ip address to match the ip of the host the request was coming from.

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

In Podman 1.7 this worked as expected. And it's probably related to:

> Rootless Podman now uses Rootlesskit for port forwarding, which should greatly improve performance and capabilities

Output of `podman version`:

```
Version:            1.8.0
RemoteAPI Version:  1
Go Version:         go1.13.6
OS/Arch:            linux/amd64
```

Output of `podman info --debug`:

```
debug:
  compiler: gc
  git commit: ""
  go version: go1.13.6
  podman version: 1.8.0
host:
  BuildahVersion: 1.13.1
  CgroupVersion: v2
  Conmon:
    package: conmon-2.0.10-2.fc31.x86_64
    path: /usr/libexec/crio/conmon
    version: 'conmon version 2.0.10, commit: 6b526d9888abb86b9e7de7dfdeec0da98ad32ee0'
  Distribution:
    distribution: fedora
    version: "31"
  IDMappings:
    gidmap:
    - container_id: 0
      host_id: 1000
      size: 1
    - container_id: 1
      host_id: 100000
      size: 65536
    uidmap:
    - container_id: 0
      host_id: 1000
      size: 1
    - container_id: 1
      host_id: 100000
      size: 65536
  MemFree: 239222784
  MemTotal: 16487555072
  OCIRuntime:
    name: crun
    package: crun-0.12.1-1.fc31.x86_64
    path: /usr/bin/crun
    version: |-
      crun version 0.12.1
      commit: df5f2b2369b3d9f36d175e1183b26e5cee55dd0a
      spec: 1.0.0
      +SYSTEMD +SELINUX +APPARMOR +CAP +SECCOMP +EBPF +YAJL
    SwapFree: 8187277312
    SwapTotal: 8329883648
    arch: amd64
    cpus: 8
    eventlogger: journald
    hostname: prefect
    kernel: 5.4.17-200.fc31.x86_64
    os: linux
    rootless: true
    slirp4netns:
      Executable: /usr/bin/slirp4netns
      Package: slirp4netns-0.4.0-20.1.dev.gitbbd6f25.fc31.x86_64
      Version: |-
        slirp4netns version 0.4.0-beta.3+dev
        commit: bbd6f25c70d5db2a1cd3bfb0416a8db99a75ed7e
    uptime: 1h 17m 29.71s (Approximately 0.04 days)
  registries:
    search:
    - docker.io
    - registry.fedoraproject.org
    - registry.access.redhat.com
    - registry.centos.org
    - quay.io
  store:
    ConfigFile: /home/bas/.config/containers/storage.conf
    ContainerStore:
      number: 22
    GraphDriverName: overlay
    GraphOptions:
      overlay.mount_program:
        Executable: /usr/bin/fuse-overlayfs
        Package: fuse-overlayfs-0.7.5-2.fc31.x86_64
        Version: |-
          fusermount3 version: 3.6.2
          fuse-overlayfs: version 0.7.5
          FUSE library version 3.6.2
          using FUSE kernel interface version 7.29
    GraphRoot: /var/home/bas/.local/share/containers/storage
    GraphStatus:
      Backing Filesystem: extfs
      Native Overlay Diff: "false"
      Supports d_type: "true"
      Using metacopy: "false"
    ImageStore:
      number: 232
    RunRoot: /run/user/1000
    VolumePath: /var/home/bas/.local/share/containers/storage/volumes
```

Package info (e.g. output of `rpm -q podman` or `apt list podman`):

```
podman-1.8.0-2.fc31.x86_64
```

Additional environment details (AWS, VirtualBox, physical, etc.):

Silverblue 31.20200209.0 (Workstation Edition)

openshift-ci-robot added the kind/bug label on Feb 9, 2020

mheon commented on Feb 9, 2020 Member

This may be a side-effect of the swap to RootlessKit for port forwarding - **@AkihiroSuda** PTAL

AkihiroSuda commented on Feb 10, 2020 Collaborator

Intentional #4586 (comment)

AkihiroSuda removed the kind/bug label on Feb 10, 2020

basvdlei commented on Feb 10, 2020 Author

Ah, I missed this was an intentional change. While I do see the benefits of the new solution, I'm wondering if we are losing something valuable.

Just for context, this change broke two setups I run with rootless Podman for local development/testing:

- Custom server running in a Podman container where nodes from a Vagrant project connect to. This server relies on the source IP of the VM for identification.
- Tests setup for an authentication server with ACL's that in include source IP range.

Since AFAIK rootless Podman has no network support, even container connections need to go through exposed ports, effectively making all traffic local. Any use-cases that cover either security or audibility relying on source IP will no longer be possible in rootless Podman.

It can also change the behavior of containers running as root or rootless. Since the incoming connections now arrive in on the loopback interface instead of the container's virtual interface. The process is required to listen on the loopback interface and may have different (security) policies there.

AkihiroSuda commented on Feb 10, 2020 Collaborator

Thanks for the context and sorry.
Maybe we should add a new flag like `podman run --port-driver=(builtin|slirp4netns)` . Not sure `podman` flag or `podman run` flag.

@giuseppe

👍 2

giuseppe commented on Feb 10, 2020 Member

@AkihiroSuda we could probably add another network type:

`podman run --network slirp4netns ....` <- works as on Podman 1.7
`podman run --network rootlesskit ....` <- works as on Podman 1.8

What do you think?

AkihiroSuda commented on Feb 10, 2020 Collaborator

That may give people false sense that slirp4netns was not used at all when `--network=rootlesskit`

AkihiroSuda commented on Feb 10, 2020 Collaborator

Also, in future, rootlesskit MAY implement built-in slirp functionality using slirpnetstack as a Go library if there is a demand.

I think `--network=rootlesskit` should be reserved for that case.

giuseppe commented on Feb 10, 2020 Member

> I think `--network=rootlesskit` should be reserved for that case.

wouldn't that be an implementation detail for users?

So if that is the case, should we have `slirp4netns-rootlesskit` ? I'd just like to avoid another option tailored for slirp and re-use what we have now

👍 1

rhatdan commented on Feb 10, 2020 Member

Yes, I would go along with **@giuseppe** Just use slirp4netns or rootlesskit. We can document the difference, most users will have no idea the difference,and will just go with whatever the distro or podman chooses as the default.

rhatdan closed this as completed on Feb 18, 2020

aleks-mariusz commented on Apr 22, 2020 • edited Contributor

i unfortunately ran into this today (using centos 7)..

looked at the `--network` parameter for `podman pod create` :

```
$ podman pod create --help
Create a new empty pod
[...]
      --network string         Connect a container to a network (default "slirp4netns")
[...]
```

but alas doesn't like it:

```
$ podman pod create --name=foo --share net --network slirp4netns
ERRO[0000] Error freeing pod lock after failed creation: no such file or directory
Error: unable to create pod: error adding Infra Container: cannot join CNI networks if running rootless: invalid argument
```

@AkihiroSuda @giuseppe @rhatdan so what actually is the right way/options to switch and be able to revert away from the default of using rootlesskit optimization (for example, i need the source address more than i need 27gbps and the 8gbps bandwidth using slirp4netns is more than sufficient for my use-case :-)

i also can't work-around this by using `--net=host` because the container is configured to bind to port 80, which is privileged, and centos 7 seems to lack the sysctl net.ipv4.ip_unprivileged_port_start :-/

giuseppe commented on Apr 23, 2020 Member

I am just worried on the configuration side and have to maintain two different backends for minimal differences.

Would you like to open a PR to address it? We need to take pieces from `da7595a` that were removed and make it configurable.

We accept `--net=slirp4netns` now. We could extend it to have extra options like `--net=slirp4netns:rootlessport` and `--net=slirp4netns:slirplisten` .

What do you think?

👍 1

aleks-mariusz commented on Apr 23, 2020 • edited Contributor

i tried `--net=slirp4netns` and still see source ip always 127.0.0.1.. :-(

it's my understanding that slirp4netns is the "traditional" way of doing rootless networking, and as of `da7595a` it defaults to rootlesskitport for rootless port forwarding.. so why does `--net=slirp4netns` not change anything?

also i'm unclear what the two values on either side of the colon are in `slirp4netns:rootlessport` vs `slirp4netns:slirplisten` will end up being when that param handling is updated?

giuseppe commented on Apr 23, 2020 Member

> it's my understanding that slirp4netns is the "traditional" way of doing rootless networking, and as of da7595a it defaults to rootlesskitport for rootless port forwarding.. so why does `--net=slirp4netns` not change anything?

it won't change anything in the current version since `--net=slirp4netns` is already the default mode for rootless, and that mode uses rootlesskitport for listening.

We'd need to add back the support for listening through slirp4netns, and I was thinking of a way to expose it to users.

aleks-mariusz commented on Apr 23, 2020 Contributor

Ahh thanks for the explanation, so there's two components in play here.. setting up the networking capabilities for the rootless container AND doing port-forwarding.. slirp4netns can do both, but as of `da7595a` it only does the former, and the latter has been defaulted since 1.8.0 to be done by rootlessport, where the source always is 127.0.0.1 (any reason why?).. is that summary correct/accurate?

As for the PR to re-enable the ability to choose, the colon notation lets you pick the two different parts.. Were you asking for me to create this PR or for someone else (such as **@AkihiroSuda** ?).. me i can give it the old college-try, but *go* is not my language of preference (i'm more of a pythonista)

giuseppe commented on Apr 23, 2020 Member

> As for the PR to re-enable the ability to choose, the colon notation lets you pick the two different parts.. Were you asking for me to create this PR or for someone else (such as **@AkihiroSuda** ?).. me i can give it the old college-try, but *go* is not my language of preference (i'm more of a pythonista)

at the moment I've no time to look at it, but if you could give it a try and get to a point where we have both methods living in the code base (i.e. recover the existing method from `da7595a`) I could help with the remaining plumbing and have something faster.

Also please use the v1.9 stable branch as the base for development.

👍 1

aleks-mariusz added a commit to aleks-mariusz/libpod that referenced this issue on Apr 28, 2020

`allow switching of port-forward approaches when rootless and using sl…` … `876314a`

aleks-mariusz mentioned this issue on Apr 28, 2020

allow switching of port-forward approaches when rootless using slirp4netns #6025

Closed

aleks-mariusz commented on Apr 28, 2020 Contributor

I've put together PR #6025 to restore this according to your guidance **@giuseppe** please review and advise if/how it should be changed in order to be merged

(also is it worth re-opening this issue until the PR is accepted?)

AkihiroSuda reopened this on Apr 28, 2020

aleks-mariusz added a commit to aleks-mariusz/libpod that referenced this issue on Apr 28, 2020

`allow switching of port-forward approaches when rootless and using sl…` … `8674925`

aleks-mariusz added a commit to aleks-mariusz/libpod that referenced this issue on Apr 29, 2020

`allow switching of port-forward approaches when rootless/using slirp4…` … `1b75c78`

aleks-mariusz added a commit to aleks-mariusz/libpod that referenced this issue on May 10, 2020

`allow switching of port-forward approaches when rootless/using slirp4…` … `dd93a7b`

aleks-mariusz added a commit to aleks-mariusz/libpod that referenced this issue on May 12, 2020

`allow switching of port-forward approaches when rootless/using slirp4…` … `41044b2`

disaster123 commented on May 19, 2020

+1 this broke two of my setups as well also I find it very strange to see 127.0.0.1 as source IP for exposed ports

disaster123 commented on May 19, 2020

i think all users expect that the source IP is the real source ip even in rootless mode

mheon commented on May 19, 2020 Member

I think that's a little hyperbolic - while we've seen complaints about the switch to rootlesskit port forwarder, for the most part it doesn't seem to have been a problem for our users.

We have no plans to revert the change of port forwarder, but **@aleks-mariusz** is working on a PR to allow optionally selecting the old port forwarding method for containers where the new approach does prove problematic.

disaster123 commented on May 19, 2020 may be ;-) at least i'm thinking of log files where remote ip is logged. I already tried the PR but src IP is still 127.0.0.1 - commented in the PR

3 hidden items
Load more...

rhatdan commented on Jun 9, 2020 Member

The PR is still being worked.

aleks-mariusz added a commit to aleks-mariusz/libpod that referenced this issue on Jun 16, 2020

`allow switching of port-forward approaches when rootless/using slirp4…` ··· `802433b`

aleks-mariusz added a commit to aleks-mariusz/libpod that referenced this issue on Jun 16, 2020

`allow switching of port-forward approaches in rootless/using slirp4netns` ··· `c6cdbcd`

aleks-mariusz added a commit to aleks-mariusz/libpod that referenced this issue on Jun 22, 2020

`allow switching of port-forward approaches in rootless/using slirp4netns` ··· `3c2cbcb`

giuseppe pushed a commit to giuseppe/libpod that referenced this issue on Jul 14, 2020

`allow switching of port-forward approaches in rootless/using slirp4netns` ··· `5dcb6ba`

giuseppe mentioned this issue on Jul 14, 2020

allow switching of port-forward approaches in rootless/using slirp4netns #6965

Merged

giuseppe pushed a commit to giuseppe/libpod that referenced this issue on Jul 14, 2020

`allow switching of port-forward approaches in rootless/using slirp4netns` ··· `f683586`

giuseppe pushed a commit to giuseppe/libpod that referenced this issue on Jul 15, 2020

`allow switching of port-forward approaches in rootless/using slirp4netns` ··· `8d12f19`

aleks-mariusz commented on Jul 17, 2020 Contributor

update: this will be fixed thanks to #6965 being merged, either by building from master or waiting until the next release where this PR is picked up - you'll want to adding parameter `--network slirp4netns:port_handler=slirp4netns` to your command line.. config-file handling to be handled in a future PR

👍 1

AkihiroSuda closed this as completed on Jul 17, 2020

andrewgdunn mentioned this issue on Jul 25, 2020

port-forwarded service always appears to originate from 127.0.0.1 rootless-containers/rootlesskit#155

Closed

barosl commented on Jan 1, 2021 • edited

I want to note that this also has a security implication. PostgreSQL (via the library/postgres image) by default regards all local connections as trusted. This means that anyone can connect to the database when Podman is operating under the rootless mode with default options.

👀 2

AkihiroSuda commented on Jan 1, 2021 Collaborator

That sounds like an issue on Postgres side. Connections from localhost should never be trusted. https://cathyjf.com/articles/local-servers-can-get-you-compromised

giuseppe mentioned this issue on Jan 20, 2021

port: add ChildIP rootless-containers/rootlesskit#206

Merged

giuseppe reopened this on Jan 21, 2021

giuseppe added a commit to giuseppe/libpod that referenced this issue on Jan 21, 2021

`rootlessport: set source IP to slirp4netns device` ··· `6ad04c4`

giuseppe mentioned this issue on Jan 21, 2021

rootlessport: set source IP to slirp4netns device #9052

Merged

giuseppe commented on Jan 21, 2021 Member

opened a PR to address this issue: #9052

giuseppe added a commit to giuseppe/libpod that referenced this issue on Jan 21, 2021

`rootlessport: set source IP to slirp4netns device` ··· 6ba2fc1

giuseppe added a commit to giuseppe/libpod that referenced this issue on Jan 21, 2021

`rootlessport: set source IP to slirp4netns device` ··· d3a9cd7

giuseppe added a commit to giuseppe/libpod that referenced this issue on Jan 21, 2021

`rootlessport: set source IP to slirp4netns device` ··· 4c1a459

giuseppe added a commit to giuseppe/libpod that referenced this issue on Jan 21, 2021

`rootlessport: set source IP to slirp4netns device` ··· d0521ce

giuseppe added a commit to giuseppe/libpod that referenced this issue on Jan 22, 2021

`rootlessport: set source IP to slirp4netns device` ··· 5e65f0b

openshift-merge-robot closed this as completed in #9052 on Jan 22, 2021

iwita pushed a commit to iwita/podman that referenced this issue on Jan 26, 2021

`rootlessport: set source IP to slirp4netns device` ··· be778de

mheon pushed a commit to mheon/libpod that referenced this issue on Feb 3, 2021

`rootlessport: set source IP to slirp4netns device` ··· a8dac71

mheon pushed a commit to mheon/libpod that referenced this issue on Feb 3, 2021

`rootlessport: set source IP to slirp4netns device` ··· ca3caa3

mheon pushed a commit to mheon/libpod that referenced this issue on Feb 3, 2021

`rootlessport: set source IP to slirp4netns device` ··· 5172cfe

ahwayakchih added a commit to ahwayakchih/nobbic that referenced this issue on Feb 20, 2021

`fix: try to make podman keep source IP, instead of changing it to local` ··· 6992d9c

xatier added a commit to xatier/dockerfiles that referenced this issue on Feb 23, 2021

`Make toy-socks5 listen to 0.0.0.0 inside the container` ··· 8bf277e

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

rootlessport: set source IP to slirp4netns device
giuseppe/libpod

9 participants