Redpill Linpro

Redpill Linpro

Securing your container environment

Replacing Docker containers with root-less Podman.

About me

- Name: Erik Kaareng-Sunde
- Senior Consultant @ Redpill Linpro Devops Oslo
- Twitter: @doktor_erik
- Github: https://github.com/drerik

In the beginning....

- 1979: Unix V7
- 2000: FreeBSD Jails
- 2004: Solaris Containers
- 2005: Open VZ
- 2006: Process Containes
- 2008: LXC
- 2011: Warden
- 2013: LMCTFY (Google)

2013: Docker

... made containers easy!

How did Docker make containers easy?

Simplified:

- Deployment
- Network
- Volumes

... It just works...

Why Docker just works

- Client <=> Server/service architecture
- The service is running as root
- To manage a docker container, users either have to use sudo or gets added to the `docker` group.

```
erik@erik13:~$ ps aux | grep dockerd
root      625302  0.0  0.1 1752820 46812 ?       Ssl  mai09   1:28 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
erik      972300  0.0  0.0  20740  2708 pts/2    S+   10:36   0:00 grep --color=auto dockerd
erik@erik13:~$
```

Is running Docker as root a problem?

As a regular user i can gain access to protected files.

```
docker run -it --rm \
    -v /etc/passwd:/files_to_edit/passwd \
    -v /etc/shadow:/files_to_edit/shadow \
    -v /etc/group:/files_to_edit/group \
    alpine bash
```

But is this realy a problem?

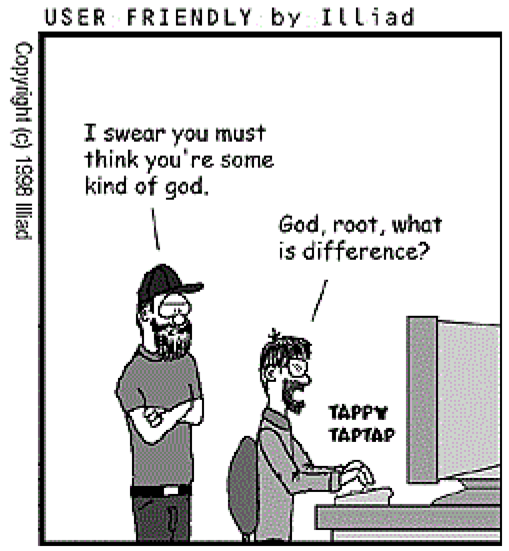
"Nobody can access my docker service or server"

"My servers security is hackerproof! nobody can gain access to it!"

Yes, it can be/is a problem!

- First rule of security: Never have just one layer of security.
- If someone gains access to the docker service, they own your server.
- If "they" can manipulate the creation of a container, they own the server.
- It's easy to expose the docker service as a network service.
- People make mistakes
- Someone is smarter than you!

Docker access == Root


USER FRIENDLY by Illiad
Copyright (c) 1998 Illiad
I swear you must think you're some kind of god.
God, root, what is difference?
TAPPY TAPTAP

(Please scan you network for port `2375/tcp` or `2376/tcp`)
(Only root should have access to `/run/containerd/containerd.sock`)

Can we run docker rootless?

Yes, but needs modification/configuration.

Limits:

- storage drivers
- overlay network
- and more...

...it's hard!

Podman to the rescue

https://podman.io

What is podman?

"Podman is a daemonless container engine for developing, managing, and running OCI Containers on your Linux System. Containers can either be run as root or in rootless mode. Simply put: alias docker=podman."

What is a rootless container?

- A container process that is running as a regular user.
- Has it's own uid/gid mapping
- `root` (uid 0) inside the container is not mapped to the "real" uid 0.
- Not the same as running a process inside a container as another user

Whats the catch with rootless containers?

Yes there are som disadvantages..

Network limitations

- Can only publish to ports above 1024 (unprivileged ports)

File access

- Container does not have access to "all" files (because your are not running it as root).
- You might get errors around using "special" filesystems life nfs/smb etc.

Limitation workarounds

Challenge accepted!

Networking workarounds

System modifications

- Set kernel parameter `net.ipv4.ip_unprivileged_port_start` to a lower port.

Iptables portforwarding

- "Forward port 80 to localhost:8080"

Reverse proxy/load balancer service

- Install a loadballancer (haproxy) on server and route traffic to "correct" port.
- Seperate certificate or sensitive files away from the code in your container makes it unavailable for intruders.

File/storage/dev access

- Make sure the user in the container has access to the files (uid mapping)

Run the one container that need access as root

- "its ok to run a container as root, if there is no other alternative."
- But maybe set up uidmap.

uidmap and gidmap

Without uidmapping:

```
$ ls -lah /tmp/test_root
-rw------- 1 root root 0 Jun  2 13:18 /tmp/test_root
$ podman run -it --rm -v /tmp:/hostfs/tmp alpine ls -lah /hostfs/tmp/test_root
-rw-------    1 root     root           0 Jun  2 13:18 /hostfs/tmp/test_root
$
```

With uidmapping:

```
$ ls -lah /tmp/test_root
-rw------- 1 root root 0 Jun  2 13:18 /tmp/test_root
$ podman run -it --rm --uidmap 0:100000:5000 -v /tmp:/hostfs/tmp alpine ls -lah /hostfs/tmp/test_root
-rw-------    1 nobody   nobody         0 Jun  2 13:18 /hostfs/tmp/test_root
$
```

What about other docker tools/commands i use?

API access

- Docker compatible api
- libpod api for podman's unique features

```
podman system service unix://$PWD/podman.sock  --time 0
```

docker-compose

`podman-compose` to the rescue!

- "An implementation of Compose Spec with Podman backend."

```
$ pip3 install podman-compose
$ podman-compose up -d
```

Accessing rootless podman trough automation (ansible)

- Issues with running podman tasks as a spesific user in ansible

Set `executable` with `sudo -i -u <user> podman`?

ansible example

```yaml
- name: Create a podman user
  ansible.builtin.user:
    name: "{{ podman_user }}"
    comment: podman service user
- name: Create "special" podman executable
  ansible.builtin.copy:
    dest: /usr/local/bin/podman_service_user.sh
    mode: 0700
    content: |
      #!/bin/bash
      sudo -i -u {{ podman_user }} /usr/bin/podman $@
- name: Create a nginx container
  containers.podman.podman_container:
    name: nginx1
    image: nginx:latest
    executable: /usr/local/bin/podman_service_user.sh
    ports:
    - "8082:80"
    volume:
      - /srv/nginx/www/:/usr/share/nginx/html:ro
```

Alternative ways to start up containers?

What about systemd?

```
$ loginctl enable-linger $(whoami)
$ podman run -d --volume /home/erik/nginx/www:/usr/share/nginx/html:ro --name nginx -p 8080:80 nginx:latest
$ podman generate systemd --new --files --name nginx
$ mkdir -p $HOME/.config/systemd/user
$ cp container-nginx.service $HOME/.config/systemd/user/.
$ systemctl --user enable container-nginx.service
$ systemctl --user start container-nginx.service
```

Are you amazed yet?

If not, give it a try (it's open source)

Redpill Linpro

Thank you!

References

- https://podman.io
- https://github.com/containers/podman-compose
- https://docs.ansible.com/ansible/latest/collections/containers/podman/index.html
- https://developers.redhat.com/blog/2020/09/25/rootless-containers-with-podman-the-basics
- https://docs.docker.com/engine/security/rootless/
- https://github.com/rootless-containers/rootlesskit
- https://www.redhat.com/sysadmin/podman-run-pods-systemd-services
- https://github.com/containers/podman/blob/main/rootless.md
- https://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016
- `man podman-generate-systemd`