Container Security

So Many Options, Use Them All!

Sally O’Malley @somalley108
Urvashi Mohnani @umohnani8

Open Container Initiative!

- Container Image (packaging)
- Container Runtime (launching)
- OCI: Any image can run on any runtime

What are Containers?

Normal Linux processes with...

- Constrained Resources - **cgroups**
- Isolation - **namespaces**
- Extra Security - **SELinux, Seccomp, Capabilities**

What are Container Images?

- **Base layer**: rootfs + json file description
- **Additional layers**: packages + updated json file
- Tarball of above

What do Container Engines do?

- Reassembles rootfs from the layers in the image onto local disk (COW)
- Creates a container runtime config
- Launches a container runtime (runc)

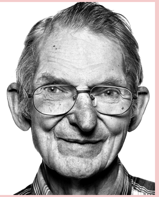
When working with containers...

- Build
- Run & Develop locally
- Store/Share
- Run in a Production Cluster

Unix Philosophy

Design programs to do a single thing, but to do it well, and to work well with other programs.

~ http://www.linfo.org/unix_philosophy.html



Douglas McIlroy

Ken Thompson and Dennis Ritchie

Unix Founders

When working with containers...

- Build - **buildah**
- Run & Develop locally - **podman**
- Store/Share - **skopeo**
- Run in a Production Cluster - **CRI-O**

buildah

Build Securely

- Shrink attack surface with minimal images
- Run builds isolated in a container
- Run without root

Give Me Demos!

(offering of cheetos and Mtn. Dew to Demo Gods)

podman

Run & Develop Securely

- Run without root
- Isolate with user namespaces
- Audit who runs what
- #nobigfatdaemons

skopeo

Share Securely

- Inspect Remote Images
- Move images between environments
- Run without root

cri-o

Run Securely in a Production Cluster

- Read-only container filesystem
- Enable fewer capabilities
- User namespaces (coming soon in Kubernetes)
- FIPS mode support

Seriously, use all these Security Features in your Linux Containers.

Every time you don’t do so, you make Dan Walsh Weep.

Dan is a nice guy and he certainly doesn’t deserve that.

CVE-2019-5736

With a compromised image or environment, processes can “escape” and execute programs on the host, by overwriting runc.

Oh No!!

This affects all container engines (CRI-O, Docker, Containerd, Buildah, Podman) that use runc container runtime.

Good News!! You can easily avoid this, and other future vulnerabilities with the following:

- Don’t run random images off the internet
- Run containers as non-root (default in OpenShift)
  Or, when possible, run containers with user namespaces
- Do run SELinux in enforcing mode **setenforce 1**
  runc file label: **container_runtime_exec_t**
  container processes SELinux type: **container_t**
  container_t types can only write to files labeled **container_file_t**.
  container_file_t != **container_runtime_exec_t**

Questions?

Resources

CRI-O: cri-o.io

Buildah: buildah.io

Podman: podman.io

Skopeo: https://github.com/containers/skopeo

Coloring Book:
https://github.com/mairin/coloringbook-container-commandos/blob/master/Web.pdf

Demo Script:
https://github.com/containers/Demos/tree/master/security