Docker Security Considerations & Incident Analysis

Percona

Unbiased Open Source Database Experts

PERCONA

Whoami

John Lionis

Junior Security Engineer @ Percona (<1 year)

(its the year of transforming my hobby(infosec) into a profession, formally known as a career change)

Several years of experience in electronics engineering
&
Freelancing as an IT field engineer & consultant

Why this talk?

Docker:

- Use is spreading
- Easy to use , fast to deploy
- Isolation, portability etc.
- From a security POV two aspects arise

  1) Is it secure?

  2) How to prevent & handle security related incidents?

Some key facts about it

- Kernel Namespaces and cgroups
- Docker daemon runs as root
- Containers run by default without cgroups limitations
- Default profile for syscall whitelisting
- Linux capabilities

Some key facts about it

Default Container Capabilities

- chown
- dac_override ←
- fowner
- kill
- setgid
- setuid
- set_pcap
- net_bind_service
- net_raw
- sys_chroot
- mknod
- audit_write
- Setfcap

discretionary access control_override. Reading the man pages, it is stated that this capability allows root to bypass file permissions (rwx) .It is as odd as it sounds.
As noted by a RedHat security standards expert
"no application should need this.If your container needs this, its probably doing something horrible"

For a brief overview of these you can have a look at:

https://www.percona.com/blog/2019/07/11/docker-security-considerations-part-i/

Motivation

Very very easy to use and deploy (so we like it)

- No need to setup a VM – plenty of images to use and spawn your containers
- Trap on the above point (not all of them are secure !)
- But the default configuration is NOT suitable for production purposes from a security pov.
- Docker leaves to the user A LOT of decisions to be made. While this offers flexibility it also increases Risk.
- Well known Vulnerabilities difficult to be redesigned
- Try it your self . Install community edition. You are now one line away from pivilege escalation.

```
$ docker run -v /home/${user}:/h_docs ubuntu bash -c "cp /bin/bash
/h_docs/rootshell && chmod 4777 /h_docs/rootshell;" && ~/rootshell -p
```

Motivation

Having said about pulling images...

- A study was made that was published in 2017 (Rui Shu, Xiaohui Gu, William Enck)
- In scope: 356.218 images
- Both official and community images contained more that 180 vulnerabilities on average
- Many have not been updated for hunders of days
- Vulns commonly propagated from parent to child images
- The study concluded that these findings demonstrate a strong need for more automated and systematic methods of applying security updates to Docker images

Secure your engine

- Secure the host
- Drop **all** capabilities , then add

  in a “need to go basis”
- Trusted Images / Certificate based authentication
- Protect the Docker daemon socket
- Apparmor support
- Selinux support
- Map root user in container to a non-root user
- Rootless daemon (not there yet)
- Use Docker CIS Benchmark

Incident Analysis

…or what to do when things go bad

In principle , a forensics analysis on a Docker starts like on a regular system

HD and memory dumps

However, analysis of the dumps may provide incomplete results

Container specifics must be taken into account

Mapping to containers – info about whether certain files are relevant for the reconstruction of the file system

File **recovery** and **accountability**

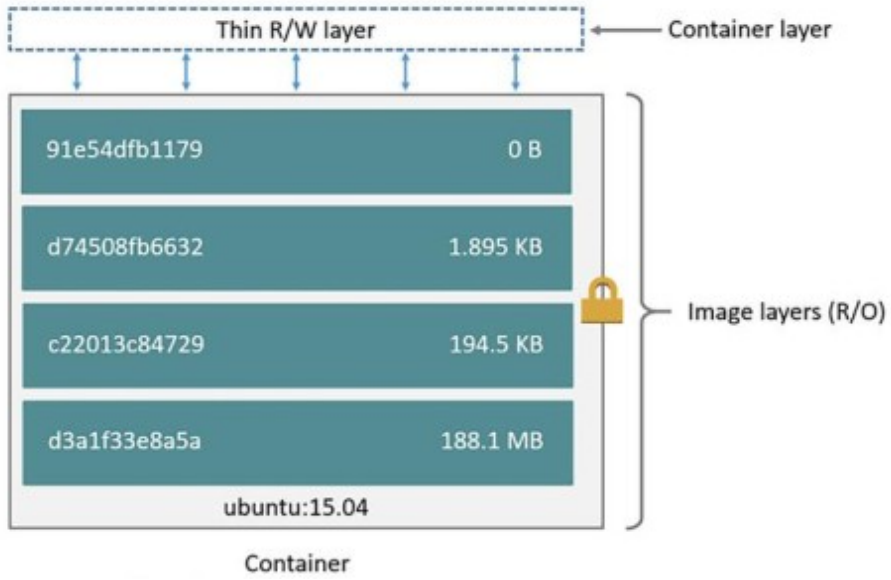
File Carving & Filesystem analysis

Filesystem

...the filesystem itself

Docker uses a layered filesystem.

A layered file system is based on a file system driver, which offers the possibility to build a single file system from different layers to present it in a uniform and abstract manner to a process.



Docker Illustration of the Layered Filesystem Model (Docker Inc, 2018).

File accountability

...the filesystem itself

Docker uses a layered filesystem.

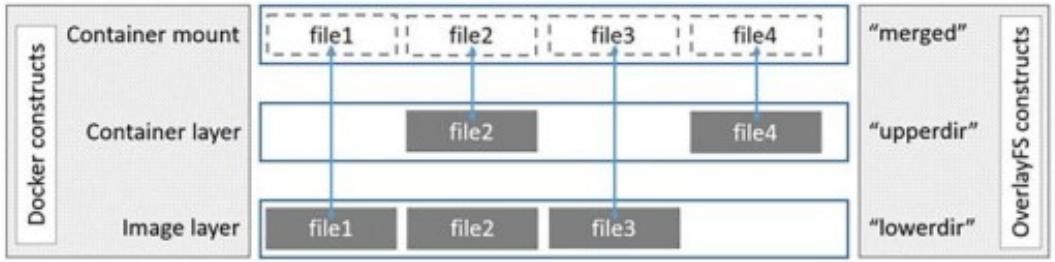
fs is not mapped to block devices as in traditional Vms

So when a disk dump of the host is made we must answer:

1)Which image provided the given file?

2)Which container used the given file?

3)Was the file deleted at container level?



Layered File Systems (Docker Inc, 2018).

PERCONA

File Carving

Method of linear searching a volume,disk image or file

Characteristic patterns (magic bytes)

File system is not considered , can recover files not yet overwritten

No metadata

Filesystem analysis

Filesystem analysis uses management structures stored in the file system

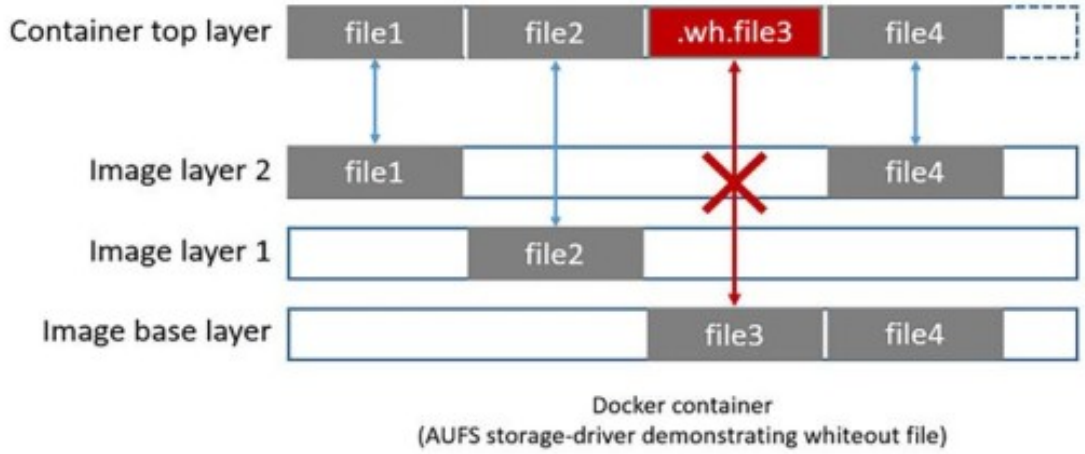
Master File Table (NTFS)

Inode tables (extfs)

Metadata availiable !

Image layer recovery

Filesystem analysis uses management structures stored in the file system



Deletion in LayerFS (Docker Inc, 2018)

In this case, a deletion reference is stored in the r/w layer, but the file remains in the image layer.

find /var/lib/Docker/overlay2/$ContainerID/diff -type c .

PERCONA

Namespaces

Linux Namespaces have various effects

PID mapping (host ↔ conainer)

UID mapping -//-

Runtime info needed!

PERCONA LIVE 2020

MAY 18 20
AUSTIN, TEXAS

Percona Live is the one and only event where all of the open source database solution companies come together with the community

MySQL, Mongo, Postgres, Elastic, Redis and more
Percona Live brings them to you.

- 3 Days
- Hands-on tutorials,
- Breakout sessions,
- Keynote addresses,
- Expo Hall
- Networking
- Lots of Fun!

Use **PRESENTER** for 20% off! Register now at perconalive.com