Enforcing a Docker container security policy

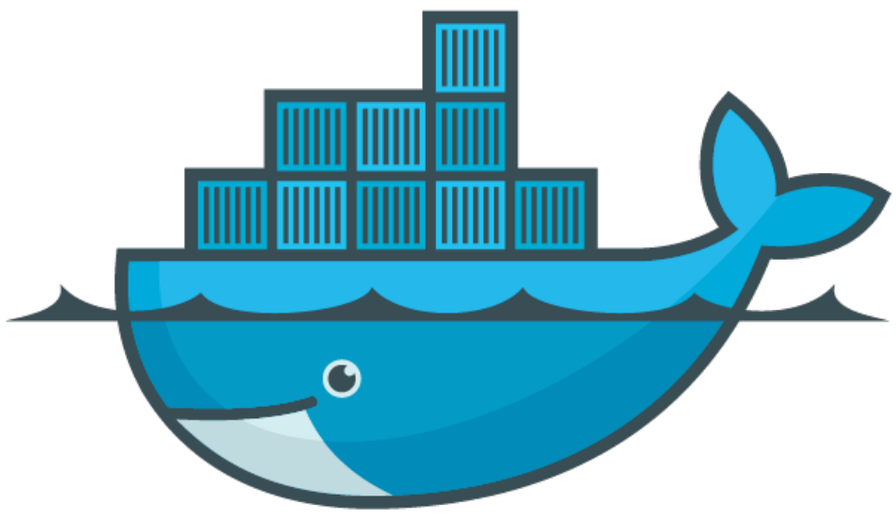
About me / Thomas Sjögren

- System tech at AB Svenska Spel
- Maintainer docker/docker-bench-security
- Author CIS Docker Benchmark

/konstruktoid

Agenda

- Daemon
- Images
- Runtime
- Events
- Enforcement



Daemon

- 1.12.0 (2016-07-28)
  `--live-restore`

- 1.10.0 (2016-02-04)
  `--userns-remap`
  `--security-opt`
  `--authorization-plugin`

- 1.8.3 (2015-10-12)
  `--disable-legacy-registry`

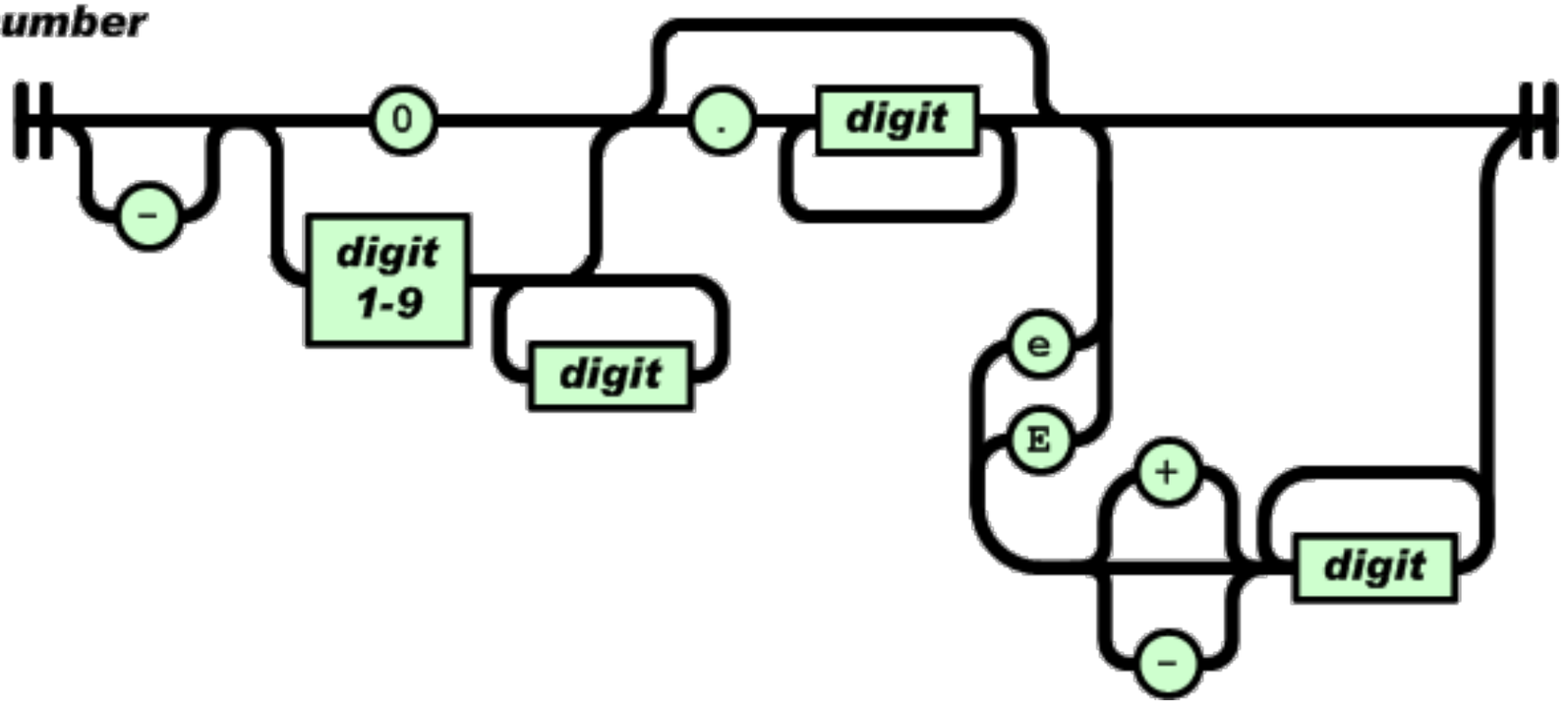
- 1.6.0 (2015-04-07)
  `--default-ulimit`

- 0.6.5 (2013-10-29)
  `--icc=false`

- ?
  `--tlsverify`

```
/usr/bin/dockerd –userns-remap=default \
–disable-legacy-registry \
–live-restore \
–icc=false \
–tlsverify \
–tlscacert=/etc/ssl/docker/ca.pem \
–tlscert=/etc/ssl/docker/server-cert.pem \
–tlskey=/etc/ssl/docker/server-key.pem \
–default-ulimit nproc=512:1024 \
–default-ulimit nofile=50:100 \
-H=0.0.0.0:2376
```

or use `daemon.json`



```
[Service]
Type=notify
ExecStart=/usr/bin/dockerd
ExecReload=/bin/kill -s HUP $MAINPID
LimitNOFILE=1048576
LimitNPROC=infinity
LimitCORE=infinity
TimeoutStartSec=0
Delegate=yes
KillMode=process
```

Base image

```
FROM scratch
ADD ./wheezy-1603172157.txz /
ENV SHA 00c3cc1b8968d3b5acf2ac9fc1e36f2aa…

ONBUILD RUN apt-get update && apt-get -y upgrade
```

```
git -s -S -m '…'
```

Container image

```
FROM alpine:3.3

ENV VERSION 1.12.1
ENV SHA256 05ceec7fd937e1416e5dce12b0b6e1c655907d349d52574319a1e875077ccb79

HEALTHCHECK CMD /health.sh

WORKDIR /usr/bin

RUN apk update && \
    apk upgrade && \
    apk --update add coreutils wget ca-certificates && \
    wget https://get.docker.com/builds/Linux/x86_64/docker-$VERSION.tgz && \
    wget https://get.docker.com/builds/Linux/x86_64/docker-$VERSION.tgz.sha256 && \
    wget https://get.docker.com/builds/Linux/x86_64/docker-$VERSION.tgz.asc && \
    sha256sum -c docker-$VERSION.tgz.sha256 && \
    echo "$SHA256 docker-$VERSION.tgz" | sha256sum -c - && \
    gpg --batch --verify docker-$VERSION.tgz.asc docker-$VERSION.tgz && \
    tar -xzvf docker-$VERSION.tgz -C /tmp && \
    mv /tmp/docker/docker . && \
    chmod u+x docker* && \
    apk del wget ca-certificates && \
    rm -rf /tmp/docker* /var/cache/apk/* docker-$VERSION.tgz*

COPY ./script.sh /script.sh

USER unpriv

ENTRYPOINT ["/bin/sh", "/script.sh"]
```

```
git -s -S -m '…'
```

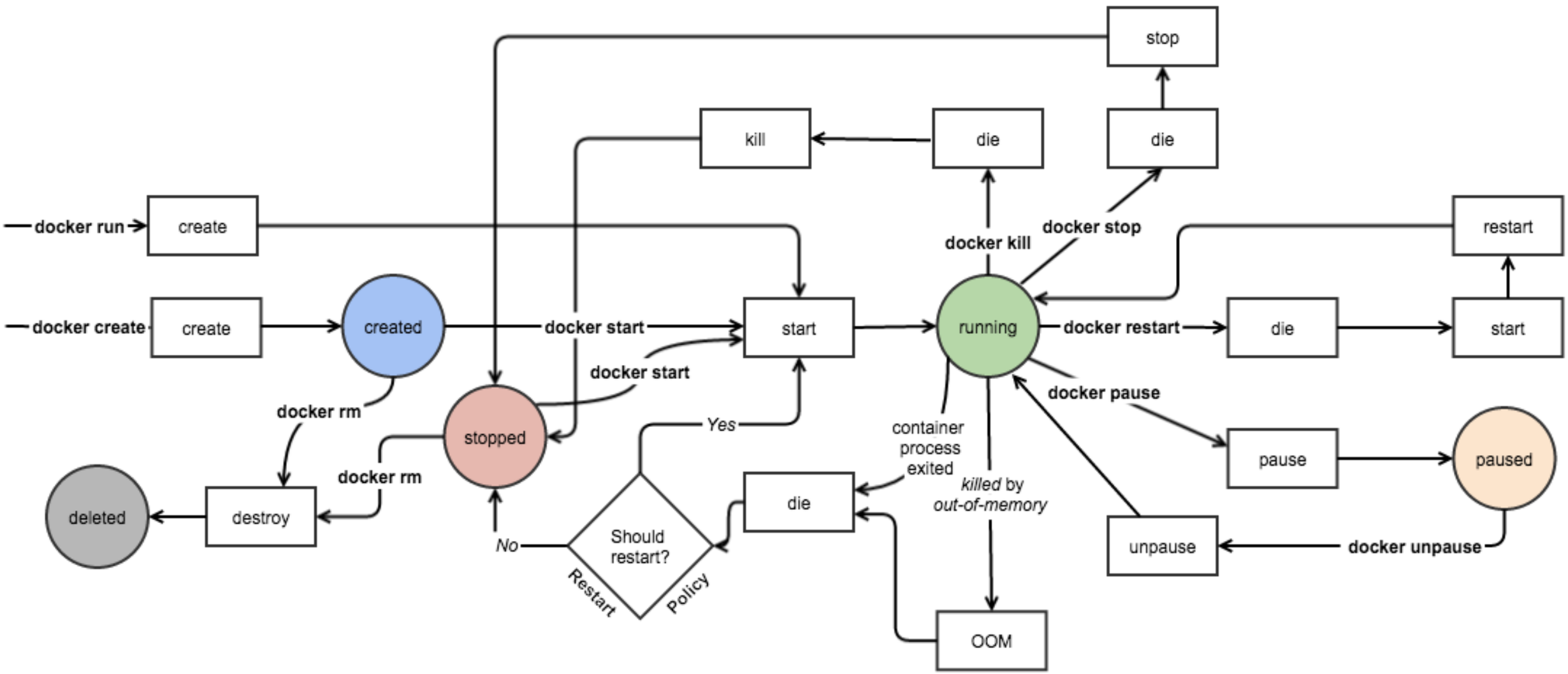
```
$ docker run -d \
--restart=”on-failure:3” \
--security-opt="apparmor:publicserver" \
-p 0.0.0.0:8123:8123 \
--cap-drop=all \
--read-only \
--tmpfs /tmp:rw,nosuid,nodev,noexec,size=100m \
--tmpfs /var/log:rw,nosuid,nodev,noexec \
--tmpfs /var/cache:rw,nosuid,noexec,nodev \
--tmpfs /run:rw,noexec,nodev,nosuid \
--ulimit nofile=10:20 \
--ulimit nproc=5 \
--name server \
public/server:latest
```

```
$ docker run -d \
-p 0.0.0.0:8123:8123 \
--name server \
public/server:latest
```

```
$ docker run -ti \
--entrypoint=/bin/bash \
-v /:/host/ \
--privileged \
--user root \
--userns=host \
public/server:latest
```

```
/usr/bin/dockerd –userns-remap=default
```

Events


stop
kill
die
die
docker run
create
docker kill
docker stop
restart
docker create
create
created
docker start
start
running
docker restart
die
start
docker start
docker rm
docker pause
stopped
Yes
container
process
exited
pause
paused
docker rm
killed by
out-of-memory
deleted
destroy
die
Should
restart?
No
Restart
Policy
unpause
docker unpause
OOM

docker run:
  container create
  network connect
  container start

docker restart:
  container kill
  container die
  network disconnect
  container stop
  network connect
  container start
  container restart

docker stop:
  container kill
  container die
  network disconnect
  container stop

docker rm:
  container destroy

Event enforcement

Create a security baseline
Watch for the `start` event
Verify
Enforce if necessary

```
---
syslog_ident: docker-covenant
debug: yes

docker-covenant:
  privileged: no
  cap_drop_required: yes

docker-bench-security:
  privileged: yes
  cap_drop_required: no

privoxy:
  cap_drop_required: yes
  security_opt_required: no
...
```

```
if "start" in container['status']:
        try:
            containerEventID = container['Actor']['ID']
            containerInspect = client.inspect_container(containerEventID)
            containerId = containerInspect["Id"]
            containerName = containerInspect["Name"].replace('/', '')

            containerCapDrop = containerInspect["HostConfig"]["CapDrop"]
            containerCapAdd = containerInspect["HostConfig"]["CapAdd"]
            containerPrivileged = containerInspect["HostConfig"]["Privileged"]
            containerSecurityOpt = containerInspect["HostConfig"]["SecurityOpt"]

            noSecurityOpt = "%s: no security options has been set" % (containerName)
            privNotAllowedLog = "%s: privileged set but not allowed" % (containerName)
            privNoPolicyLog = "%s: privileged set but no policy" % (containerName)
            capDropLog = "%s: all capabilities not dropped" % (containerName)
            capAddAllLog = "%s: capability ALL has been set" % (containerName)
            stopContainerLog = "%s: stopping container" % (containerName)
```

DEMO

- https://www.svenskaspel.se/
- https://github.com/konstruktoid
- https://github.com/docker/docker-bench-security
- https://benchmarks.cisecurity.org/downloads/multiform/index.cfm
- https://github.com/docker/docker/blob/master/contrib/init/systemd/docker.service
- https://docs.docker.com/engine/reference/commandline/dockerd/#/linux-configuration-file
- http://json.org/
- https://github.com/konstruktoid/Debian_Build
- https://git-scm.com/book/en/v2/Git-Tools-Signing-Your-Work
- https://docs.docker.com/engine/reference/api/docker_remote_api/#/docker-events
- https://github.com/konstruktoid/docker-covenant
- https://github.com/konstruktoid/Docker/blob/master/Security/CheatSheet.adoc
- http://yarchive.net/comp/linux/security_bugs.html