Docker Security

Architektur, Design, Penetration Testing

$ whoami

- Martin Pizala

- 2014 System Engineer bei Micromata
- 2015 Fokus auf IT-Security & InfoSec
  - Schwachstellenmanagement
  - Security Best Practices & Hardening
  - Infrastruktur & Web Application Penetration Testing, OSCP
  - Sichere Architekturen / CI/CD / SecDevOps
- 2017 IT-Security ~~Engineer / Analyst / Consultant~~ Enthusiast

Agenda

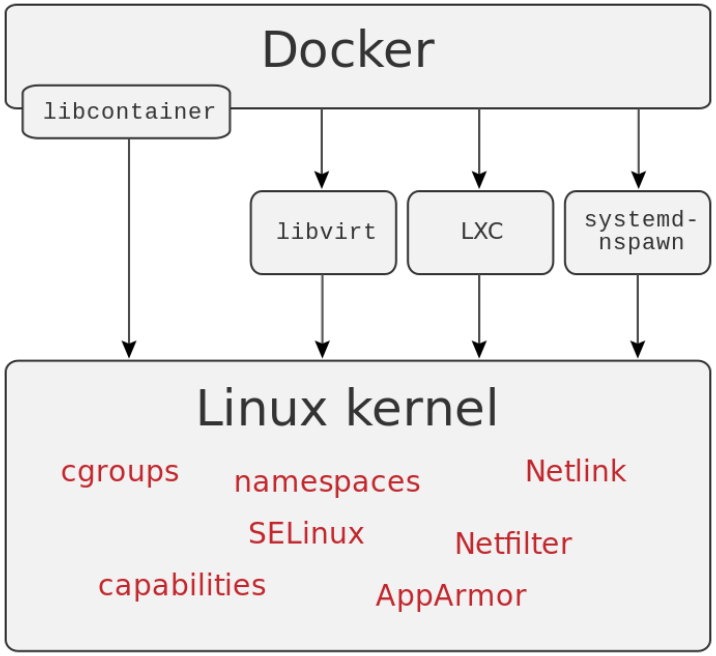
- Keine Einführung für Docker

- Architektur und Design (-Flaws)
- Docker Security
- Docker Penetration Testing

Architektur Docker – Linux Kernel

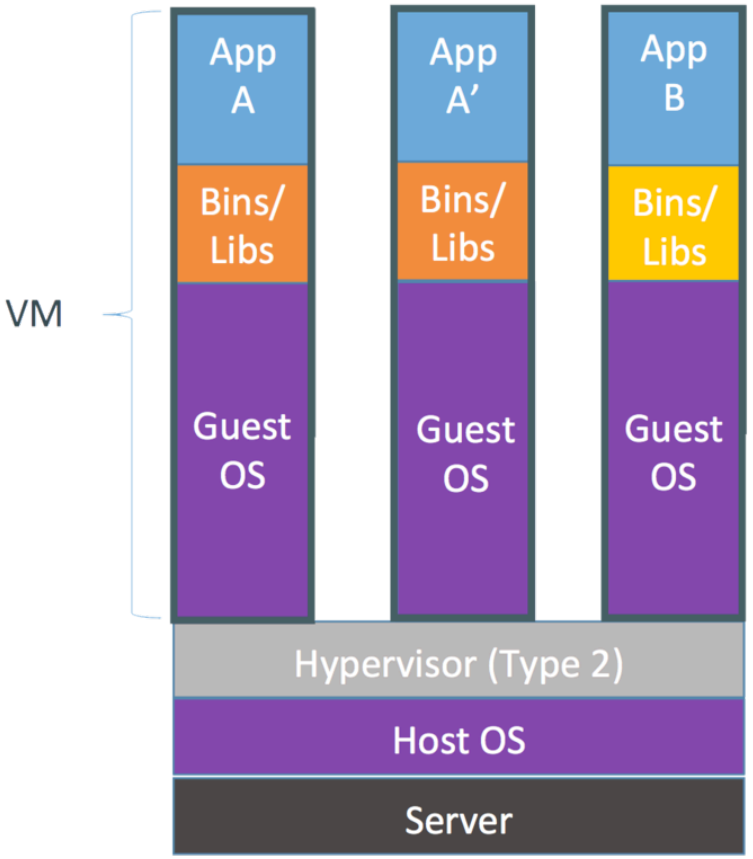
- Linux basierte Container Lösung

- Verbindet Funktionen des Linux-Kernels zur Isolierung
  - Namespaces
  - Control groups
  - Capabilities
  - Union file systems
  - Netlink und Netfilter
  - Linux Kernel security modules (SELinux or AppAmour)
  - seccomp (Begrenzung von Syscalls)



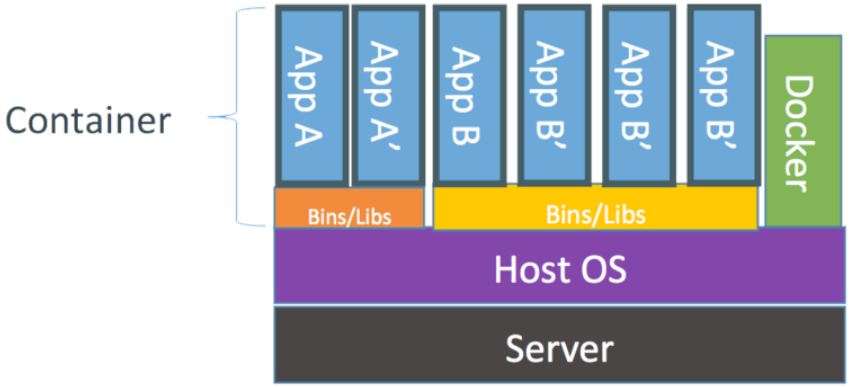
https://en.wikipedia.org/wiki/Docker_(software)

Architektur Docker – Container vs VM



Containers are isolated, but share OS and, where appropriate, bins/libraries

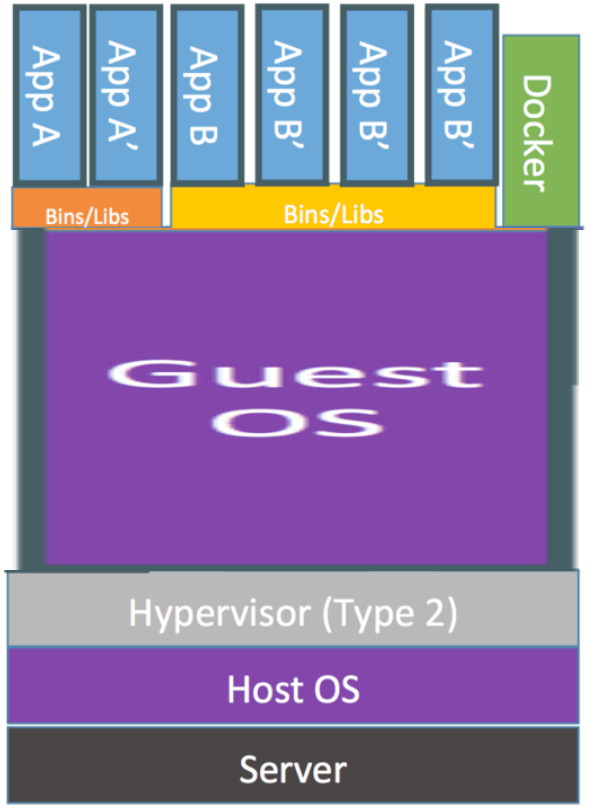
...result is significantly faster deployment, much less overhead, easier migration, faster restart



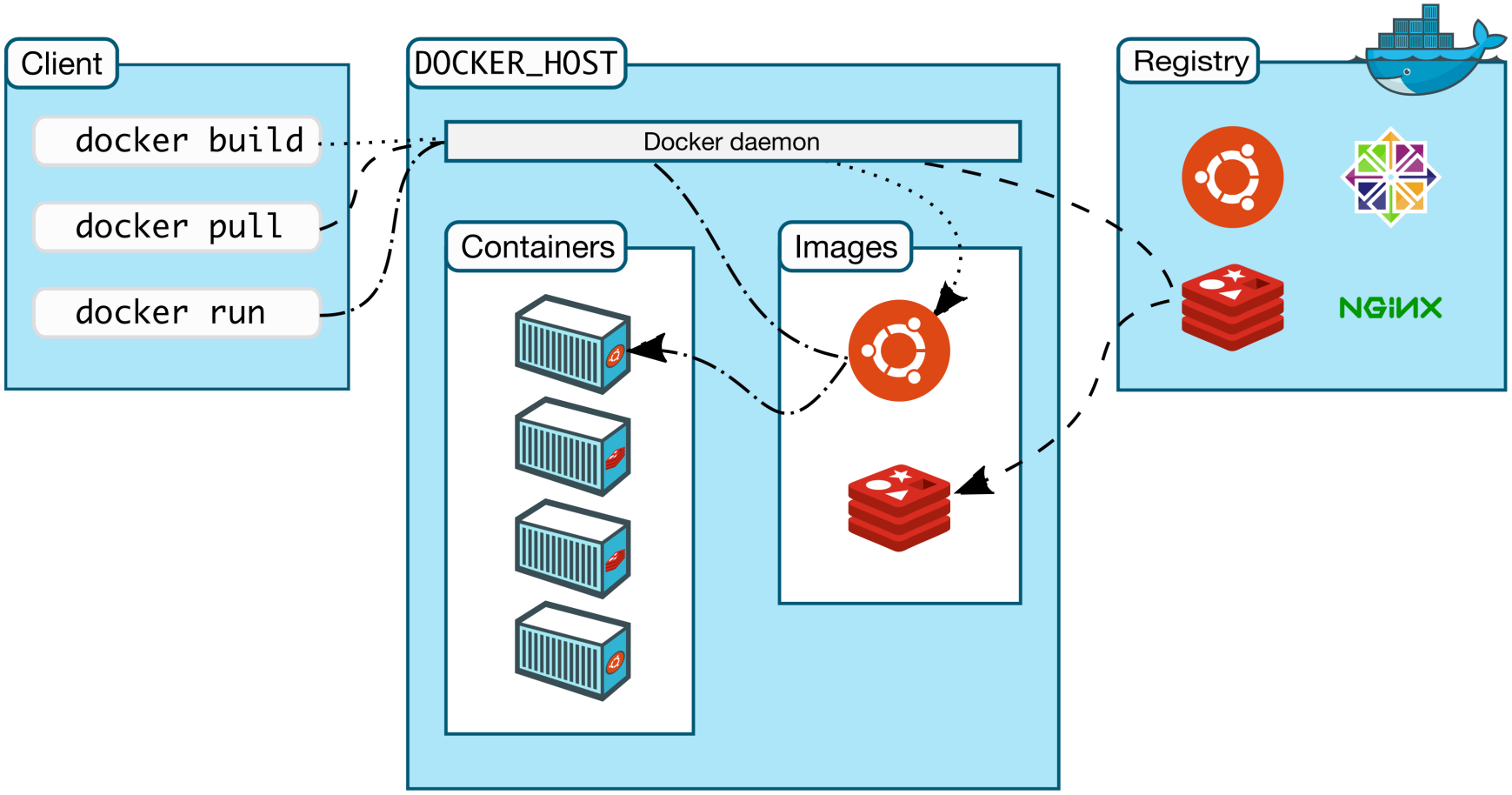
https://www.sdxcentral.com/cloud/containers/definitions/containers-vs-vms/

Architektur Docker – VM + Container

- Docker for Mac = HyperKit
- Docker for Windows = Microsoft Hyper-V



Architektur Docker – Workflow



https://docs.docker.com/engine/docker-overview/#docker-architecture

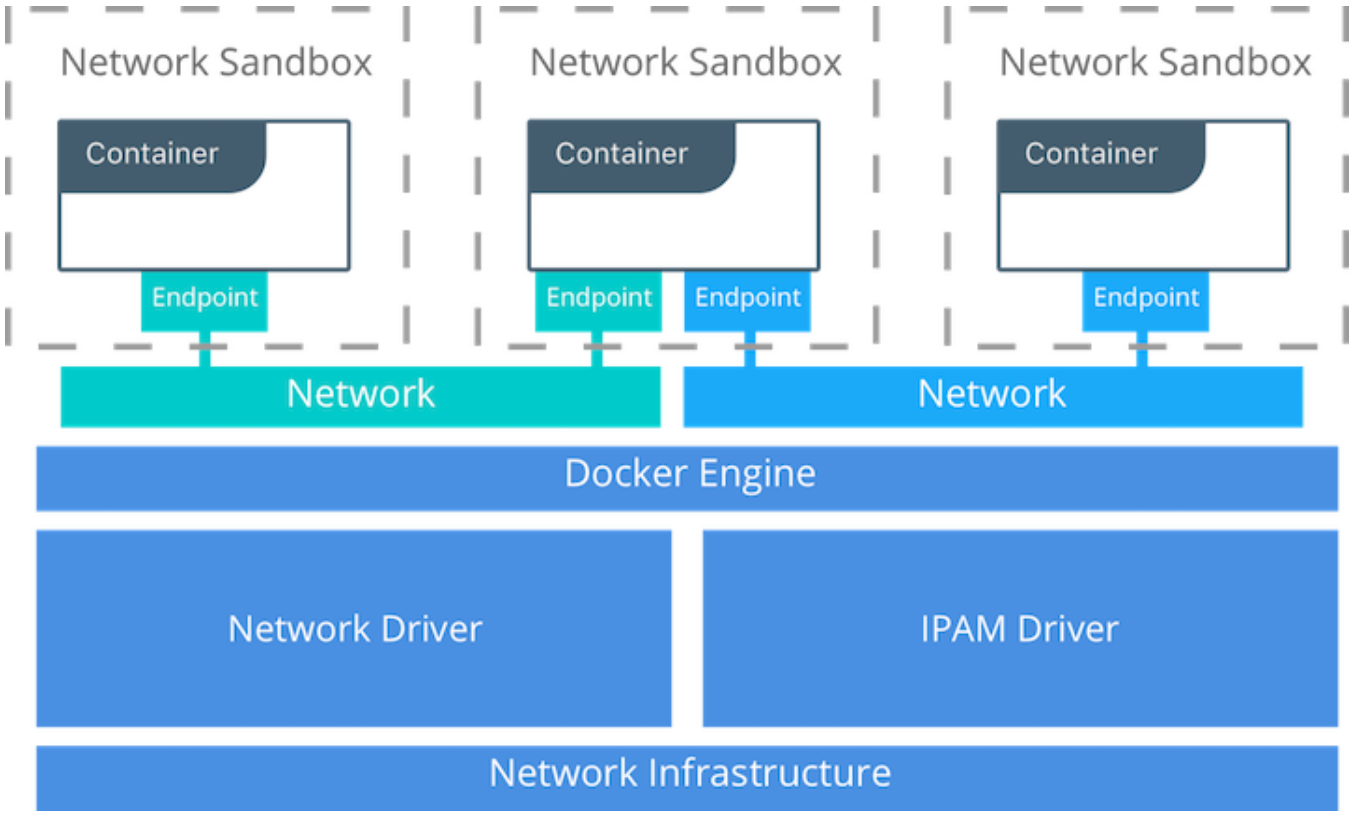
Architektur Docker – Basics

```
root@debian:~/demo/1-docker-build# docker images -a
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE

root@debian:~/demo/1-docker-build# cat Dockerfile
FROM centos:7
RUN yum install -y nmap >/dev/null 2>&1

root@debian:~/demo/1-docker-build# docker build -t nmap .
Sending build context to Docker daemon  2.048kB
Step 1/2 : FROM centos:7
7: Pulling from library/centos
74f0853ba93b: Pull complete
Digest: sha256:26f74cefad82967f97f3eeeef88c1b6262f9b42bc96f2ad61d6f3fdf544759b8
Status: Downloaded newer image for centos:7
 ---> 328edcd84f1b
Step 2/2 : RUN yum install -y nmap >/dev/null 2>&1
 ---> Running in 20b163aa9f7b
 ---> 46d5ae6e6ebe
Removing intermediate container 20b163aa9f7b
Successfully built 46d5ae6e6ebe
Successfully tagged nmap:latest

root@debian:~/demo/1-docker-build# docker run --rm -ti nmap bash
[root@5ad4587ea5dd /]# nmap -version

Nmap version 6.40 ( http://nmap.org )
Platform: x86_64-redhat-linux-gnu
Compiled with: nmap-liblua-5.2.2 openssl-1.0.1e libpcre-8.32 libpcap-1.5.3 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
[root@5ad4587ea5dd /]#
```

Architektur Docker – PID Namespaces

```
dockeruser@debian:~$ ps aux | wc -l
106
dockeruser@debian:~$ docker run --rm -ti alpine sh
/ # ps aux
PID   USER     TIME   COMMAND
    1 root       0:00 sh
    7 root       0:00 ps aux
/ #
```

Architektur Docker – Container Networking



https://success.docker.com/Architecture/Docker_Reference_Architecture%3A_Designing_Scalable%2C_Portable_Docker_Container_Networks

Architektur Docker – Container Networking

```
root@debian:~# ip a l
   ens32:
    inet 172.16.76.100/24
   docker0:
    inet 172.17.0.1/16

root@debian:~# docker run --rm -ti debian:7 bash
root@d6eb55aa2fb7:/# ip a l
    eth0@if19:
    inet 172.17.0.2/16

root@d6eb55aa2fb7:/# ip route
default via 172.17.0.1 dev eth0
172.17.0.0/16 dev eth0  proto kernel  scope link  src 172.17.0.2


root@debian:~# iptables-save | grep MASQUERADE
-A POSTROUTING -s 172.17.0.0/16 ! -o docker0 -j MASQUERADE
```

Architektur Docker – Container Networking

```
root@debian:~# docker run --rm -ti -d -p 8080:80 centos:7 bash
469c2334fe5c18661f015bd5365bed33059927fc7336f16ad1c4abe20fb9ab01

root@debian:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
469c2334fe5c        centos:7            "bash"              6 seconds ago       Up 5 seconds        0.0.0.0:8080->80/tcp

root@debian:~# netstat -tlnp | grep 8080
tcp6       0      0 :::8080                 :::*                    LISTEN      4703/docker-proxy

root@debian:~# iptables-save | grep 8080
-A DOCKER ! -i docker0 -p tcp -m tcp --dport 8080 -j DNAT --to-destination 172.17.0.2:80
```

Architektur Docker

SECURE BY DEFAULT

Docker is secure out-of-the-box with no additional setup required. Mutual TLS, certificate rotation, image signing and container isolation makes for bullet-proof and easy-to-use container app runtimes.

https://www.docker.com/get-docker

Secure by Default - Daemon

- Docker Daemon
  - /usr/bin/dockerd == root-Prozess
  - root@container == root@host (default = kein Username Namespace remapping)
  - user:docker@host == root@host

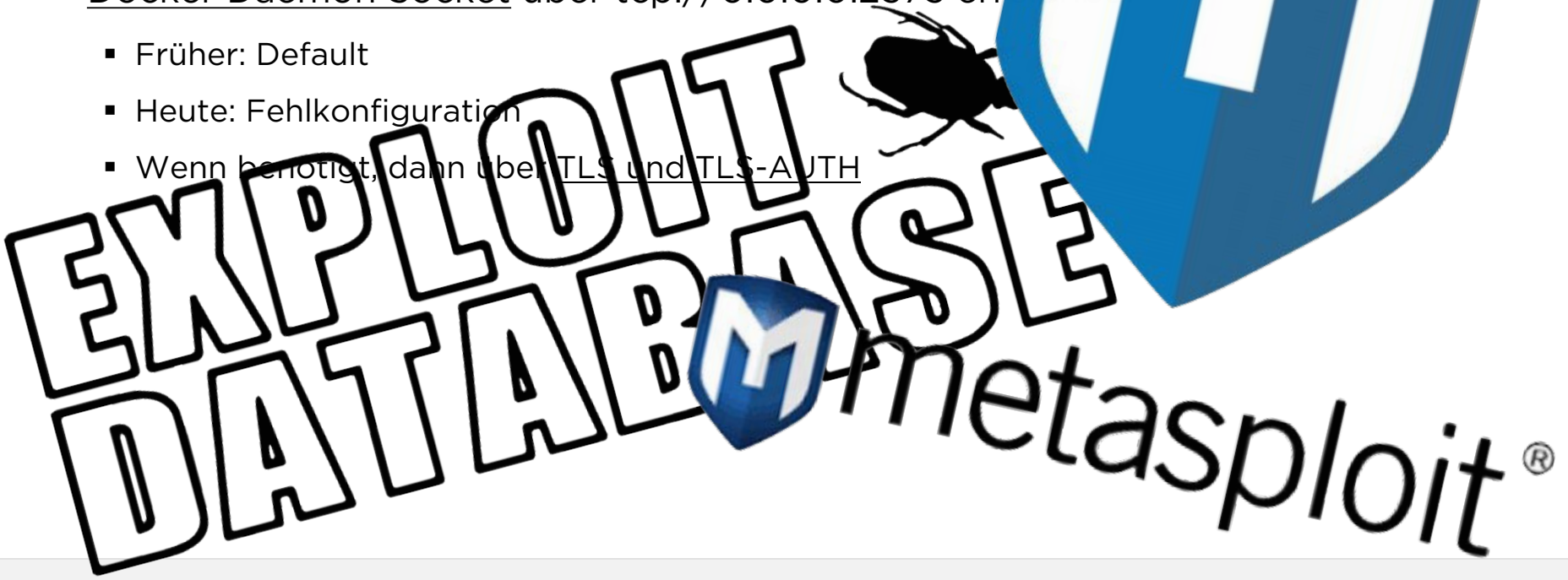
- Docker Daemon Socket über tcp://0.0.0.0:2375 erreichbar
  - Früher: Default
  - Heute: Fehlkonfiguration
  - Wenn benötigt, dann über TLS und TLS-AUTH

DEMO

Secure by Default - Daemon

- Docker Daemon
  - /usr/bin/dockerd == root-Prozess
  - root@container == root@host (default = kein Username Na
  - user:docker@host == root@host

- Docker Daemon Socket über tcp://0.0.0.0:2375 err
  - Früher: Default
  - Heute: Fehlkonfiguration
  - Wenn benötigt, dann über TLS und TLS-AUTH



EXPLOIT DATABASE

Home Exploits Shellcode Papers Google Hacking Database Submit Search

Search the Exploit Database

Search the Database for Exploits, Papers, and Shellcode. You can even search by **CVE** and **OSVDB** identifiers.

docker

Ich bin kein Roboter. reCAPTCHA Datenschutzerklärung - Nutzungsbedingungen

Search

More Options

5 total entries

Date ▾	D	A	V	Title	Platform	Author
2017-09-11	⬇	-	✔	**Docker Daemon - Unprotected TCP Socket (Metasploit)**	**Python**	**Metasploit**
2017-07-20	⬇	-	◷	Docker Daemon - Unprotected TCP Socket	Linux	Martin Pizala
2017-06-07	⬇	-	✔	DC/OS Marathon UI - Docker Exploit (Metasploit)	Python	Metasploit
2016-09-19	⬇	-	✔	Docker Daemon - Privilege Escalation (Metasploit)	Linux	Metasploit
2014-06-18	⬇	-	✔	Docker 0.11 - VMM-Container Breakout	Linux	Sebastian K...

Docker Daemon - Unprotected TCP Socket

Fefes Blog

Wer schöne Verschwörungslinks für mich hat: ab an felix-bloginput (at) fefe.de!

Fragen? Antworten! Siehe auch: Alternativlos

Mon Aug 7 2017

- [1] Kurzer Hinweis für Docker-Benutzer. Mir mailt gerade jemand folgende Warnung:

  > mir hat heute ein Kollege erzählt er hätte am Wochenende Shipyard installiert, via Deppeninstaller:
  >
  > `curl -s https://shipyard-project.com/deploy | bash -s`
  >
  > und das hätte bei ihm ein Image "memtest" von dockerhub automatisch installiert.
  > Bis er der ganzen Sache auf die Schliche gekommen war, hatte der miner schon 400€ "monero"s erzeugt und abgeführt.
  >
  > Ich habe daraufhin ein wenig recherchiert und kann zu der Geschichte rein gar nichts finden.

  Ich schätze die Glaubwürdigkeit erstmal für eher gering ein, denn übers Wochenende mined man nicht mal eben Crypto-Coins im Wert von 400€, außer man hat da eine Mining-Farm mit spezilisierter Hardware stehen. Aber als Hinweis, auf was man ein Auge haben sollte, ist das sicher richtig und wichtig. Wer sich per curl|bash Software installiert, und unbesehen irgendwelcher Dockerhub-Container ausführt, der fährt erhöhtes Risiko.

  Update: Nachtrag des Einsenders:

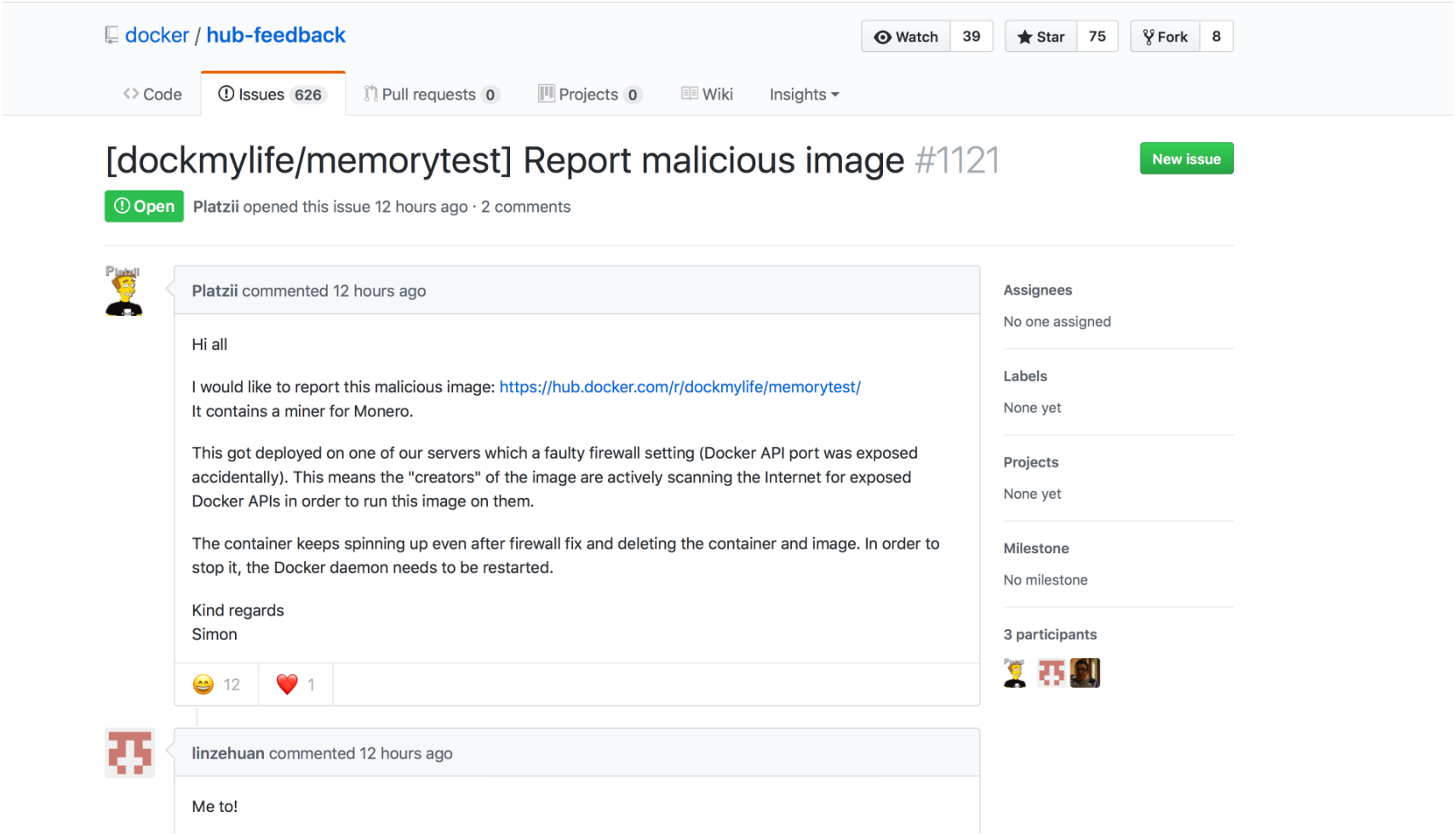
  > mittlerweile konnte ich einen Report finden, offenbar war das Problem des Kollegen nicht der Dienst, sondern eine offene API:
  >
  > https://github.com/docker/hub-feedback/issues/1121

  Same difference :)

ganzer Monat

Proudly made without PHP, Java, Perl, MySQL and Postgres
Impressum

Docker Daemon - Unprotected TCP Socket



docker / hub-feedback

Watch 39 Star 75 Fork 8

Code Issues 626 Pull requests 0 Projects 0 Wiki Insights

[dockmylife/memorytest] Report malicious image #1121

New issue

Open Platzii opened this issue 12 hours ago · 2 comments

Platzii commented 12 hours ago

Hi all

I would like to report this malicious image: https://hub.docker.com/r/dockmylife/memorytest/
It contains a miner for Monero.

This got deployed on one of our servers which a faulty firewall setting (Docker API port was exposed accidentally). This means the "creators" of the image are actively scanning the Internet for exposed Docker APIs in order to run this image on them.

The container keeps spinning up even after firewall fix and deleting the container and image. In order to stop it, the Docker daemon needs to be restarted.

Kind regards
Simon

12 1

linzehuan commented 12 hours ago

Me to!

Assignees

No one assigned

Labels

None yet

Projects

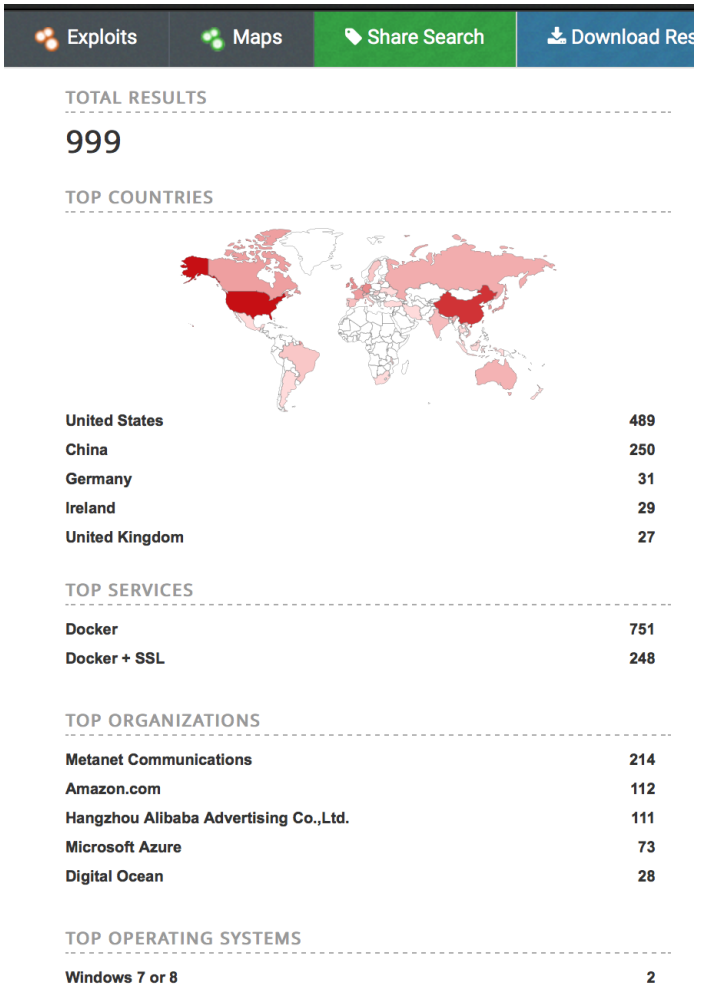
None yet

Milestone

No milestone

3 participants

Docker Daemon - Unprotected TCP Socket



https://www.shodan.io

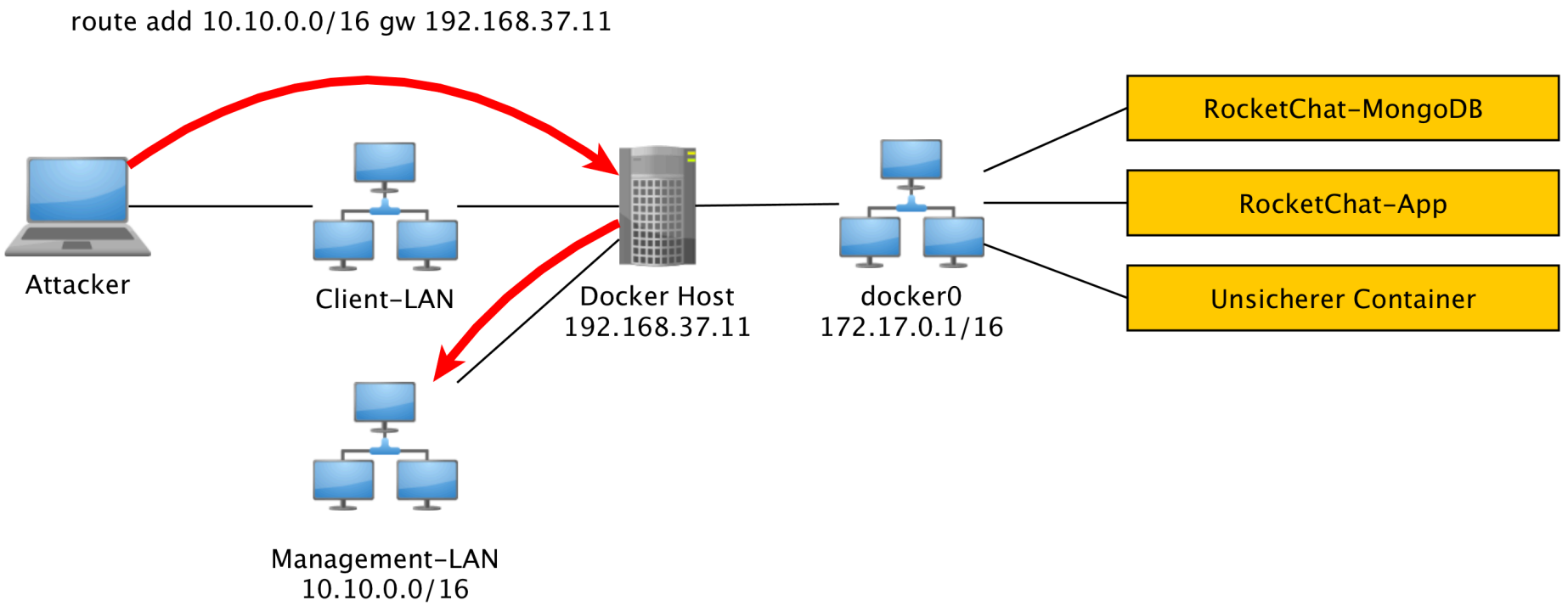
Secure by Default - Network

- Communicating to the outside world
  - Docker Daemon aktiviert ip_forwarding (kommentarlos)
- Communication between containers
  - Inter-Container-Communication (default=true)
  - Problem: Bequeme/Unsichere Images
- Container communication between hosts

https://docs.docker.com/engine/userguide/networking/default_network/container-communication/

Secure by Default - Network

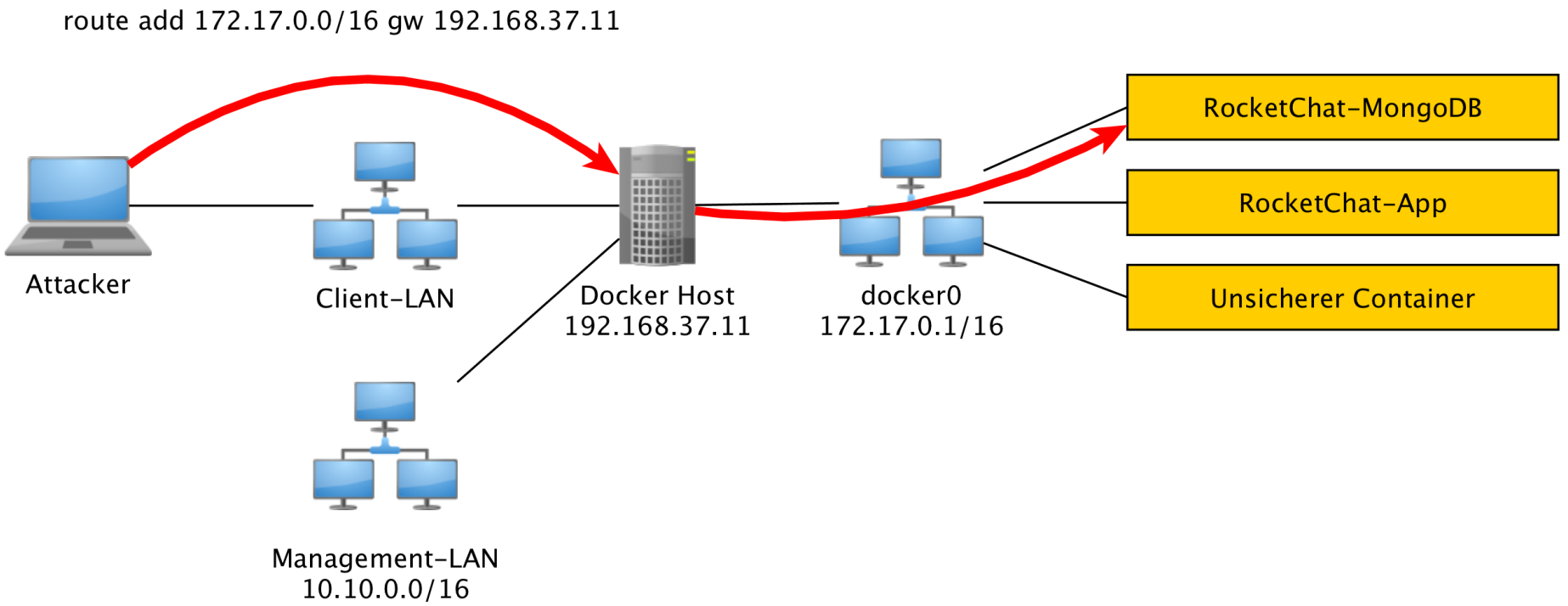
- Docker Daemon aktiviert ip_forwarding



- Fixed on 1.13.0 (2017-01-18) Change the default FORWARD policy to DROP #28257

Secure by Default - Network

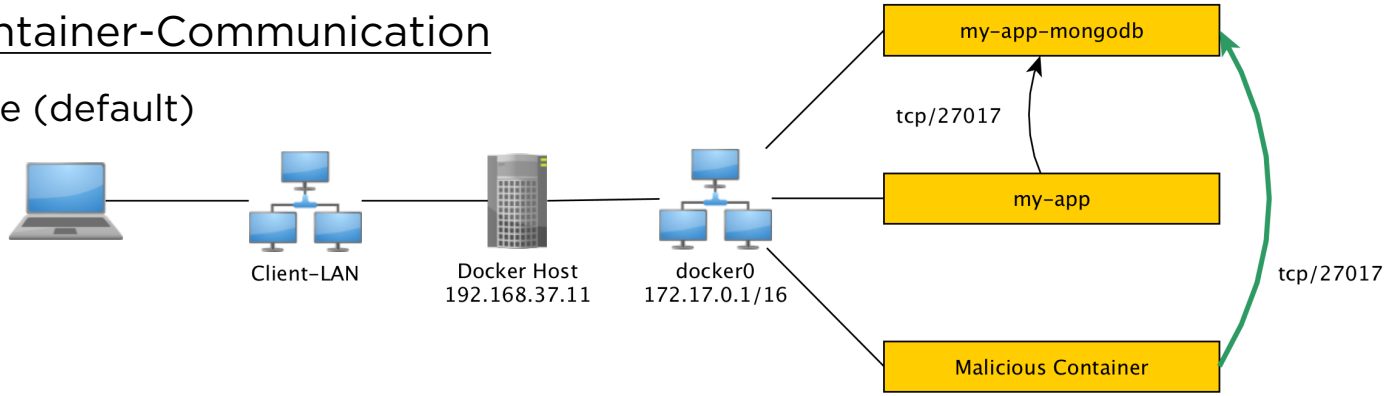
- Docker Daemon aktiviert ip_forwarding



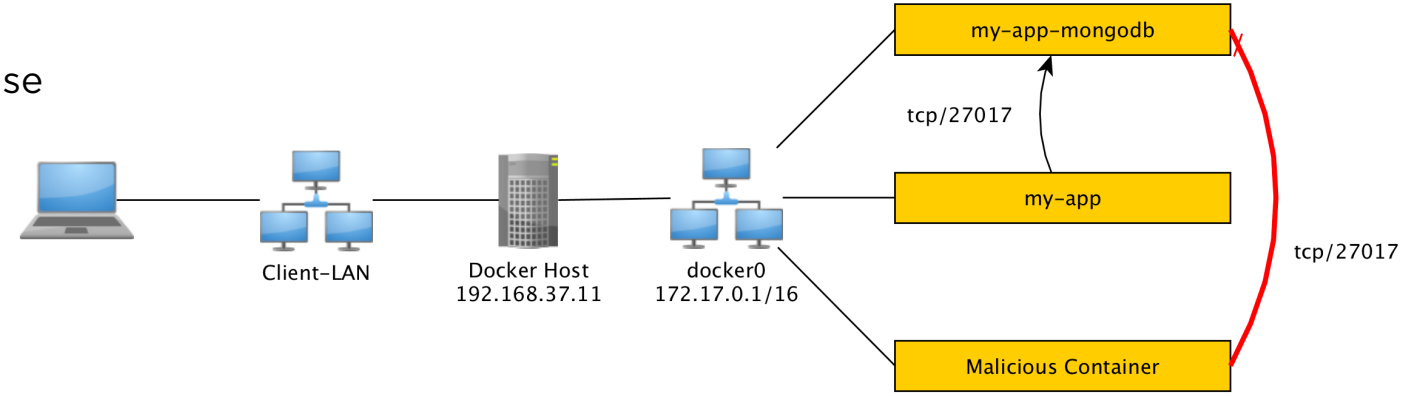
- Fixed on 1.13.0 (2017-01-18) Change the default FORWARD policy to DROP #28257

Secure by Default - Network

- Inter-Container-Communication
  - icc=true (default)



  - icc=false



DEMO

Secure by Default

- Docker Network
  - Kann sicher werden
    - In- und Outbound Firewall Regeln
    - icc=false
- Docker Images
  - Offizielle Images
    - Bequem (Dev 😍), aber Sicherheit sollte geprüft werden (Prod 😱)
  - Community Images
    - Bequem und mit Vorsicht zu genießen 😱
    - Prinzipiell wie Code (docker hub ~ github)
- Docker Container
  - Können sicher sein

Docker Best Practices

- Docker Security
  https://docs.docker.com/engine/security/security/

- Understanding Docker Security and Best Practices
  https://blog.docker.com/2015/05/understanding-docker-security-and-best-practices/

- Dockerfile Best Practices
  https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/

- Docker Security Best Practices 2017
  https://blog.aquasec.com/docker-security-best-practices

- 10 things to avoid in docker containers
  https://developers.redhat.com/blog/2016/02/24/10-things-to-avoid-in-docker-containers/

10 things to avoid in docker containers

First: Containers are immutable

Second: Containers are lightweight

Third: Containers are fast

1) Don’t store data in containers

2) Don’t ship your application in two pieces

3) Don’t create large images

4) Don’t use a single layer image

5) Don’t create images from running containers

6) Don’t use only the “latest” tag

7) Don’t run more than one process in a single container

8) Don’t store credentials in the image. Use environment variables

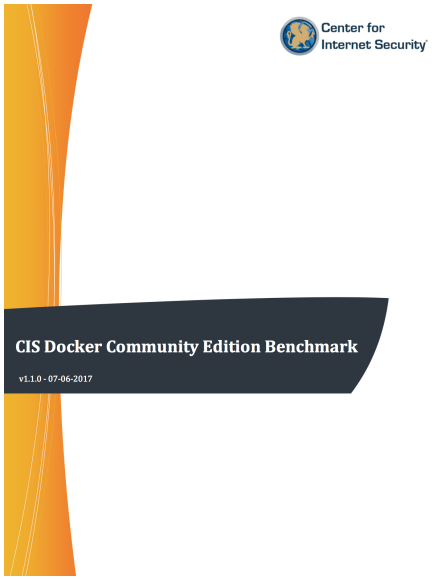
9) Don’t run processes as a root user

10) Don’t rely on IP addresses

https://developers.redhat.com/blog/2016/02/24/10-things-to-avoid-in-docker-containers/

Docker Best Practices

- CIS Benchmark for Docker
  1. Host Configuration
  2. Docker Daemon Configuration
  3. Docker Daemon Configuration Files
  4. Container Images and Build Files
  5. Container Runtime
  6. Docker Security Operations



- Center for Internet Security / NGO
- Kostenloser Download nach Registrierung

- Audit Tool Docker Bench for Security

Docker Bench for Security

- https://github.com/docker/docker-bench-security

```
# ------------------------------------------------------------------------------
# Docker Bench for Security v1.3.3
#
# Docker, Inc. (c) 2015-
#
# Checks for dozens of common best-practices around deploying Docker containers in production.
# Inspired by the CIS Docker Community Edition Benchmark v1.1.0.
# ------------------------------------------------------------------------------

Initializing Fri Jul 14 09:18:42 UTC 2017


[INFO] 1 - Host Configuration
[WARN] 1.1  - Ensure a separate partition for containers has been created
[NOTE] 1.2  - Ensure the container host has been Hardened
[PASS] 1.3  - Ensure Docker is up to date
[INFO]      * Using 17.06.0 which is current
[INFO]      * Check with your operating system vendor for support and security maintenance for Docker
[INFO] 1.4  - Ensure only trusted users are allowed to control Docker daemon
[INFO]      * docker:x:992:vagrant
[WARN] 1.5  - Ensure auditing is configured for the Docker daemon
[WARN] 1.6  - Ensure auditing is configured for Docker files and directories - /var/lib/docker
[WARN] 1.7  - Ensure auditing is configured for Docker files and directories - /etc/docker
[WARN] 1.8  - Ensure auditing is configured for Docker files and directories - docker.service
[INFO] 1.9  - Ensure auditing is configured for Docker files and directories - docker.socket
[INFO]      * File not found
[INFO] 1.10 - Ensure auditing is configured for Docker files and directories - /etc/default/docker
[INFO]      * File not found
[INFO] 1.11 - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json
[INFO]      * File not found
[WARN] 1.12 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-containerd
[WARN] 1.13 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-runc


[INFO] 2 - Docker daemon configuration
[WARN] 2.1  - Ensure network traffic is restricted between containers on the default bridge
[PASS] 2.2  - Ensure the logging level is set to 'info'
[PASS] 2.3  - Ensure Docker is allowed to make changes to iptables
[PASS] 2.4  - Ensure insecure registries are not used
[PASS] 2.5  - Ensure aufs storage driver is not used
```

Docker Penetration Testing

Docker für Penetration Tester

- Tools/OS installieren
  - https://github.com/enaqx/awesome-pentest#docker-for-penetration-testing
  - https://hub.docker.com/r/kalilinux/kali-linux-docker/
  - https://hub.docker.com/u/wpscanteam/

- Testumgebungen einrichten
  - Beliebige Linux-Varianten und Applikationen unter Linux / Windows / Mac
  - Verwundbare Container
    - https://hub.docker.com/u/vulnerables/
      - PHPMailer < 5.2.18 Remote Code Execution vulnerable image
      - SambaCry remote vulnerable environment with Samba 4.5.9

- Forschungsdrang

Enumeration

- Docker Host Identifizieren
  - gateway-finder (PentestMonkey) (nur im LAN)
- Docker Daemon
  - nmap -p 2375-2376 [ip]
- Docker nahe Dienste
  - Docker Registry
  - Container Verwaltungstools
  - ...

Exploitation

- Docker Daemon Unprotected TCP Socket
  - Ohne Username Namespace = ROOT
  - Mit Username Namespace = NOBODY

- Shell/RCE in einem Container
  - checkcontainer (Metasploit Module)
    - /proc/1/cgroup | grep docker
    - ls -la /.dockerenv
  - ps aux (Prozess 1 != /sbin/init)

Privilege Escalation

- Kernel Exploits
- Docker Exploits

Docker 0.11 - VMM-Container Breakout

EDB-ID: 33808	Author: Sebastian Krahmer	Published: 2014-06-18
CVE: N/A	Type: Local	Platform: Linux
E-DB Verified: ✔	Exploit: Download / View Raw	Vulnerable App: N/A

« Previous Exploit Next Exploit »

```
/* shocker: docker PoC VMM-container breakout (C) 2014 Sebastian Krahmer
 *
 * Demonstrates that any given docker image someone is asking
 * you to run in your docker setup can access ANY file on your host,
 * e.g. dumping hosts /etc/shadow or other sensitive info, compromising
 * security of the host and any other docker VM's on it.
 *
 * docker using container based VMM: Sebarate pid and net namespace,
 * stripped caps and RO bind mounts into container's /. However
 * as its only a bind-mount the fs struct from the task is shared
 * with the host which allows to open files by file handles
 * (open_by_handle_at()). As we thankfully have dac_override and
 * dac_read_search we can do this. The handle is usually a 64bit
 * string with 32bit inodenumber inside (tested with ext4).
 * Inode of / is always 2, so we have a starting point to walk
 * the FS path and brute force the remaining 32bit until we find the
 * desired file (It's probably easier, depending on the fhandle export
 * function used for the FS in question: it could be a parent inode# or
 * the inode generation which can be obtained via an ioctl).
 * [In practise the remaining 32bit are all 0 :]
 *
 * tested with docker 0.11 busybox demo image on a 3.11 kernel:
 *
```

- Linux Privilege Escalation
  - https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

- Ziel ist nicht der Container, sondern der Host!

Post-Exploitation

- Enumeration (lokal)
  - Abhänge Container (Bsp. App+DB)

- Umgebungsvariablen checken
- /etc/hosts

- Tools nachinstallieren/hochladen
  - nmap
    - Docker Network
    - Nochmal den Host (evtl Firewall von vorne, aber nicht von hinten)
  - tcpdump
  - arp-spoof / mitm

- Pivoting

Post-Exploitation – Umgebungsvariablen

- Secrets
  - in Umgebungsvariablen
  - Dateien werden „gesource“ als Umgebungsvariablen
    - /proc/$(pidof app)/environ
  - docker run -d –e USER=dbuser -e PASSWORD=geheim –v secret:/app/secret **--link my-php-app-db:db** --name my-php-app my-php-app-image

- Beispiel
  - PHP Anwendung mit MYSQL Datenbank

Post-Exploitation – Linking Containers

https://docs.docker.com/engine/userguide/networking/default_network/dockerlinks/ - connect-using-network-port-mapping

Vulnerable Docker Machines

- DonkeyDocker (vulnhub)
  - Beginner / Intermediate
- Vulnerable Docker VM (NotSoSecure)
  - Easy and Hard Modes (via grub)

Ausblick

- Fortsetzung Docker Security Vortrag
  - Best Practices / Betrieb
  - Images / Dockerfile
    - Container
    - Schwachstellenscanner
  - Auditing

Mehr

- https://www.ernw.de/download/ERNW_Stocard_Docker-Devops-Security_fbarth-mluft.pdf

- https://www.blackhat.com/docs/eu-15/materials/eu-15-Bettini-Vulnerability-Exploitation-In-Docker-Container-Environments.pdf

- https://www.blackhat.com/docs/us-17/thursday/us-17-Cherny-Well-That-Escalated-Quickly-How-Abusing-The-Docker-API-Led-To-Remote-Code-Execution-Same-Origin-Bypass-And-Persistence_wp.pdf