How to Lose a Container in 10 Minutes

Sarah Young
whoami
whoami
whoami

Azure
Cloud
Security
and
Compliance
Global
Black
Belt

@_sarahyo
whoami

[@_sarahyo](https://twitter.com/_sarahyo)
Yes, everything in Australia is trying to kill you
So, what am I going to talk about today?
How to Lose a Container in 10 Mins

@_sarahyo
What am I talking about today?

- Good security practices for containers, Kubernetes and related tools.
What am I talking about today?

- Protecting your data.
- Caring for your OS and orchestrators.
- Checking your privileges.
- Shifting left with containers.
- Getting my containers pwned.
This sums up container and k8s security for me
Good data protection practices
Well begun is half done

- Sounds basic right?
- If you’re making the move to the cloud and containerizing your application, use this as a opportunity to tidy it up.
- Use TLS 1.2 or 1.3.
- Don’t send things in the clear.
- Everything should be encrypted at rest and in transit.
- Remove deprecated protocols.
- Tidy up your code.
- i.e. simplify.
Shared responsibility model

SaaS
Hotel room

PaaS
Furnished apartment

IaaS
Rental apartment

Private Cloud
Private House
Traditional security model
“Isolation doesn’t bother me at all. It gives me a sense of security.”

-Jimmy Page
Caring for your OS and orchestrator
Do you care about your image?

- Make sure you know where your container images come from.
- Try and minimize your use of images from the Internet, keep your own base images.
- Don’t pull images from sources you can’t trust i.e. the whole of the Internet.
- Use a private image repository, there are many to choose from:
  - Clair
  - Notary
  - All major cloud platforms offer them
The “fault” in default

- Kubernetes default configs aren’t too secure.
- You need to work through the orchestrator configs to secure them.
- Notable baddies in Kubernetes are:
  - The API server listening on 8080 where no checks take place.
  - Secrets management in Kubernetes using etcd.
- Use the CIS Kubernetes benchmark.

https://www.cisecurity.org/benchmark/kubernetes/

@_sarahyo
It’s a secret to everybody

- Don’t bake creds and secrets into containers.
- Pass them into your container as environment variables.
- Kubernetes stores secrets in etcd, encoded in base64.
- All major cloud providers have inbuilt secrets management that can be used.
- Utilize a third party secrets management system.
- Rotate your keys regularly.
Horror story #1

- Dev needed a slight change to the app.
- Pulled from a public repo.
- We know what happened next.
Check your privilege
Check your privilege (containers)

- Don’t run as root.
- Don’t run as root.
- Don’t run as root.

- If you must run as root (e.g. if your container needs to modify the host system) use runtime security tools to limit what is accessible.
- Honorable mentions in this area go to Aqua Enforcer, SELinux, AppArmor and seccomp.
Check your privilege (orchestrators)

- Kubernetes had (and still has) some terrible defaults here:
  - Anonymous user access isn’t disabled.
  - The dashboard had full admin privileges by default (prior to v1.7).
  - No RBAC before v1.8.
- If it sounds too onerous to go through and do yourself:
  - Use a managed Kubernetes cluster e.g. EKS, AKS, etc.
  - Have a play with interesting open source tools.
Horror story #2

- Kubernetes cluster left exposed
- Left exposed to the world
- We know what happened
Contain your enthusiasm
(for shifting left)
Deploying a container/k8s aware security toolset

- Don’t assume your old toolset will be adequate for your needs when you move to the cloud/when containerizing applications.
- Most security tools need to be specifically container/k8s aware, or may need additional plugins:
  - IDS/heuristics
  - Vulnerability scanning
  - SIEM
  - Runtime security
  - Auditing
Get your plumbing in order

- Same goes for your CI/CD pipeline.
- Tools may need to be altered to work in your pipeline.
- Some may need to be replaced entirely.
- Do your research.
Actually go benchmark your tools, seriously

- Benchmark your tools.
- Get both developers and security involved in this process.
Horror story #3

- I have seen an organization try to use old-school vulnerability scanners on their containers.
- Unsurprisingly, the results weren’t very useful.
Getting myself pwned
Have I been pwned?

- I’ve been spinning up containers, Kubernetes clusters and leaving them open to the Internet for a few months now.
- Spun up some standalone containers on a cloud hosting provider.
- Yes, I’m not brave enough to have them on any infrastructure I own.
- Also I could pay for it through PayPal so to not run myself up a huge bill.
- Basically I tried to do the opposite of everything that I’ve just encouraged you to do.

@_sarahyo
Guessing game
So what happened...?
So what happened...?

Heh, it took a month before the WordPress site I set up experienced a sustained hacking attempt. I'm only surprised it didn't happen sooner.

Source: Twitter, @attacus_au
So what happened...?

Inbound Rules

Set the Firewall rules for incoming traffic. Only the specified ports will accept inbound connections. All other traffic will be blocked.

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>All TCP</td>
<td>TCP</td>
<td>All ports</td>
<td>All IPv4, All IPv6</td>
</tr>
<tr>
<td>All UDP</td>
<td>UDP</td>
<td>All ports</td>
<td>All IPv4, All IPv6</td>
</tr>
</tbody>
</table>
So what happened...?

**Remarks:**
Please note that CNNIC is not an ISP and is not empowered to investigate complaints of network abuse. Please contact the tech-c or admin-c of the network.

---

**TCP scan signatures:**

- "MISC Microsoft SQL Server communication attempt"
- dst port: 1433 (no server bound to local port)
- flags: SYN
- psad_id: 1000205
- chain: INPUT
- packets: 1
- classtype: attempted-admin

---

**inetnum:** 185.0.0.0 - 185.142.236.255
**netname:** BlackHOST-CLOUD
**descr:** BlackHOST CLOUD Network
**descr:** Specially crafted and optimized for bandwidth hungry applications
**descr:** Direct all copyright, legal, spam and abuse complaints to:

---

**mnt-by:** MAINT-CNNIC-AP
**last-modified:** 2017-12-15T05:57:39Z
**source:** APNIC

---

**organisation:** ORG-BLACKRIPE
**org-name:** BLACKHOST Ltd.
**descr:** Take advantage of the best deal of bandwidth on the planet.
**descr:** UNMETERED Dedicated & VPS Servers, Premium web & email hosting
**descr:** Check out our offer on: https://black.host

---

**security trouble**

- e-mail: cloud-co-sqcloud@list.alibaba-inc.com
- address: 5th floor, Building D, the West Lake International Plaza of S&T, 391# Wen’er Road
- phone: +86-0571-85022600
- country: CN
- mnt-by: MAINT-CNNIC-AP
In conclusion...
Tidy up your application before your cloud migration and/or containerization.
Orchestrator defaults are terrible: change them.
Please, please, please make sure you know where your container images come from.
Don’t run as root.*

* Unless you have very good reasons for doing so.
Keep your secrets secret.
Shift left, but make sure you have the right tools to support this.
Purposely trying to get containers hacked is harder than one would expect. *

* Or maybe I was just unlucky. Don’t quote me on this.
Useful links

- CIS Kubernetes security benchmark - https://www.cisecurity.org/benchmark/kubernetes/
- CIS Docker security benchmark - https://www.cisecurity.org/benchmark/docker/
Thank you!

Any questions?

@_sarahyo