CONTAINERS AT-RISK

A Review of 21,000 Cloud Environments
Overview

I. Executive Summary
II. Introduction
III. The Eroding Perimeter
IV. Open Management Interfaces and APIs
V. Kubernetes Specifics
VI. Recommendations for Container Security Best Practices
VII. FAQ
I. Executive Summary

Securing your workloads in public clouds requires a different approach than that used for traditional data centers. The need to operate security at cloud speed, respond to continuous change, adapt at scale, and operate with a new operating model all require a dramatic shift in the type of security solution required by today’s operation. In a world where APIs drive the infrastructure and create ephemeral workloads, organizations can develop control over their cloud security posture through real-time visibility, anomaly detection, and deep understanding of the behaviors of users, resources, and connections.

The reality of the risks of operating workloads in the cloud is highlighted in this research conducted by Lacework. In early June 2018, Lacework discovered more than 21,000 container orchestration and API management systems on the Internet, and these results highlight the potential for attack points caused by poorly configured resources, lack of credentials, and the use of non-secure protocols.

This report describes the risks and threats that can be created by deploying workloads in public cloud without the proper security guardrails, security services, and the systematic use of security best practices.

Note: there is an FAQ at the bottom of the report.

Summary of findings (downloadable infographic)
Over the last few years we have seen a dramatic rise in the use of containers and container orchestration systems for the coordination and management of cloud services. Among other things, containers allow for rapid deployment, ephemeral workloads, and autoscaling of applications at scale. For organizations that work in an agile way and deploy services continuously, it's an enormously popular piece of their infrastructure. Popular types of containers include: Kubernetes, Docker Swarm, OpenShift, and Mesosphere.

There are typically two critical pieces to managing these systems. First is a web UI and associated APIs. Secondly, an administrator dashboard and API are popular because they allow users to essentially run all aspects of a container cluster from a single interface. Access to the dashboard gives you top level access to all aspects of administration for the cluster it is assigned to manage. That includes managing applications, containers, starting workloads, adding and modifying applications, and setting key security controls.

Here are some examples of these systems dashboards:

**Kubernetes Management UI**
Marathon / Mesos

Red Hat OpenShift
Prior to public clouds, enterprises used to have something called a perimeter, which operated much like something you would see on a Game of Thrones set. At the risk of oversimplifying things, enterprises had their own castle to protect enterprise assets and all things that wanted to come inside the castle had to cross the drawbridge. Furthermore, IT and security owned the moat, in case evildoers attempted to gain access without passing through the bridge. Basically, winter was always imminent, but the moat did the trick.

Now imagine if someone had the keys to your datacenter: access to all servers, privileged accounts, and administrator passwords on all servers. Then, consider what would happen if they had all this but could operate their attack all from the Internet, hiding behind proxy servers, VPN concentrators, and compromised routers, essentially masking who they are and where they are coming from. Basically, your data, your customer’s data, and the foundation on which you’ve built your organization would be in major trouble.

Let’s be clear. We are BIG BELIEVERS in all things public cloud, but we need to raise the bar, and raise it quick.
Research Overview

In the past there have been reports that revealed that some companies accidentally left their computing resources open to the world with no username and password and, in turn, were taken over by hackers with a motive of deploying machines and code to perform cryptomining from the abused infrastructure. This can certainly be costly, but a greater risk is that an outsider gains the highest level of privileges to your cluster.

Research conducted by Lacework discovered more than 22,000 publicly accessible management nodes connected to the Internet. These nodes are essentially openings to these organization’s cloud environments to anyone with basic skills at searching the web. Although the vast majority of these management interfaces have credentials set up, there is little reason why they should be world-accessible and are far more vulnerable than they should be. Additionally, just by being open, you are potentially disclosing information that can give attackers sensitive information on their targets. Within most discovered systems, the company name could be derived from certificates and hostnames even without access. These organizations, and the others who will replicate their mistakes, are opening themselves up to brute force password and dictionary attacks.

In order to identify these nodes, a combination of web crawling, Shodan, SSL data mining, and some internal tools were used - all this data being available from publicly-accessible sources.

Note: Lacework will not release any company information or details on specifics around discovered hosts. Additionally, no access was attempted to any of the nodes that were open.
High Level Findings

- 22,672 OPEN ADMIN DASHBOARDS DISCOVERED ON INTERNET
- 95% HOSTED INSIDE OF AMAZON WEB SERVICES (AWS)
- 55% HOSTED IN AN AWS REGION WITH THE US (US-EAST MOST POPULAR)
- > 300 OPEN ADMIN DASHBOARDS OPEN WITH NO CREDENTIALS

Platforms Discovered

We discovered the following applications during our research:

- Kubernetes
- Mesos Marathon
- Swagger API UI
- Red Hat Openshift
- Docker Swarm:
  - Portainer
  - Swarmpit
IV. Open Management Interfaces and APIs

During the research we noticed an alarming number of systems with no authentication whatsoever. Some were clearly in the midst of being setup, but some were in full production. In cases where full access was available, one can perform operations like add and deploy their own applications, delete infrastructure, change credentials, and potentially exfiltrate data.

Some example screenshots of management dashboards:

Open Mesos Marathon Screenshot
V. Kubernetes Specifics

Kubernetes, or “K8s” as it’s often referred, is by far the most popular and fastest growing orchestration and container management system. It's incredibly powerful and provides a great deal of value to developers because it is optimized to support deployment of large scale stable infrastructure.

Although there are several new security features that are helping to secure Kubernetes such as default SSL and default authentication, we focused on Kubernetes due to the popularity of the platform. The general issues found were:

- Open dashboards that were in the midst of being setup,
- Open dashboards with no authentication,
- Open dashboards that possibly could be brute forced, and
- Information disclosure of the organizations that have deployed Kubernetes.

In cases where having the management UI open to the world is intentional - and it's unclear what the use case would be - administrators and security operators for these companies should be aware that their exposure is transparent and that it poses a huge potential for risk of their data and cloud infrastructure.
Open Kubernetes Admin Dashboard

Kubernetes Dashboard

- Kubeconfig
  Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

- Token
  Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

Choose kubeconfig file

SIGN IN  SKIP

Kubernetes Admin Dashboard Authentication

Sign in

https://[removed]

Username

Password

Cancel  Sign In
## Screenshot Showing Non-Trusted Certificate

![Warning]

**This Connection Is Not Private**

This website may be impersonating "****" to steal your personal or financial information. You should go back to the previous page.

Go Back

Safari warns you when a website has a certificate that is not valid. This may happen if the website is misconfigured or an attacker has compromised your connection.

To learn more, you can [view the certificate](#). If you understand the risks involved, you can [visit this website](#).

## Screenshot Showing Information Disclosure

<table>
<thead>
<tr>
<th>Extension</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extension</strong></td>
<td>Subject Alternative Name (2.5.29.17)</td>
</tr>
<tr>
<td><strong>Critical</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>DNS Name</strong></td>
<td>api.internal.k8s-staging.****.com</td>
</tr>
<tr>
<td><strong>DNS Name</strong></td>
<td>api.k8s-staging.****.com</td>
</tr>
<tr>
<td><strong>DNS Name</strong></td>
<td>kubernetes</td>
</tr>
<tr>
<td><strong>DNS Name</strong></td>
<td>kubernetes.default</td>
</tr>
<tr>
<td><strong>DNS Name</strong></td>
<td>kubernetes.default.svc</td>
</tr>
<tr>
<td><strong>DNS Name</strong></td>
<td>kubernetes.default.svc.cluster.local</td>
</tr>
<tr>
<td><strong>IP Address</strong></td>
<td>100.64.0.1</td>
</tr>
<tr>
<td><strong>IP Address</strong></td>
<td>127.0.0.1</td>
</tr>
</tbody>
</table>
Our researchers also discovered what appeared to be a popular container health check service which is part of the Kubernetes branch named `healthz`.

`Healthz` is described as follows:

"The exec healthz server is a sidecar container meant to serve as a liveness-exec-over-http bridge. It isolates pods from the idiosyncrasies of container runtime exec implementations."
During our research, 38 servers running healthz live on the Internet with no authentication whatsoever were discovered. AWS and Alibaba were the most popular cloud platforms supporting this activity.

While it's unclear whether you can perform full remote code execution (it looks like it could be set up), by default you can monitor workloads and even stop them from running via their UI.

Web screenshot of open container running Healthz

**Kubernetes healthz sidecar container**

`/healthz: healthz probe. Returns "ok" if the command given through -cmd exits with 0. /quit: Cause this container to exit.`
VI. Recommendations for Container Security Best Practices

During our research we learned that there are a lot of different ways to manage your containers, and that they are all incredibly flexible and powerful. With each one you essentially have the keys to the castle from deployment, discovery, deletion, and manageability.

We suggest that if you are a security professional and you don’t know you are running a container orchestration system, you should definitely find out ASAP. From there you need to determine the acceptable level of outside visibility and the policy determined for access.

Additional recommendations:

- Regardless of network policy, use MFA for all access;
- Apply strict controls to network access, especially for UI and API ports;
- Use SSL for all servers and use valid certificates with proper expiration and enforcement policies;
- Investigate VPN (bastion), reverse proxy or direct connect connections to sensitive servers;
- Look into product and services such as Lacework in order to discover, detect, prevent, and secure your container services.

Kubernetes specific recommendations:

- Configure your Kubernetes pods to run read-only file systems;
- Restrict privilege escalation in Kubernetes;
- Build a pod security policy.
VII. Frequently Asked Questions (FAQ)

Q: What is the threat announced?

A: Containers that are not secured with proper configurations and settings can pose major risks that can turn into threats. We believe there is little reason to leave your administration interface open to the world without a bastion jump, VPN, or proxy ACL. More importantly, you may be running a vulnerable version of Kubernetes which could lead not just a brute force attack but potentially an exploit-based and often there are more services than the management applications running. Lastly, we discovered hundreds of UI’s open to the world with no credentials needed and also sites not running SSL.

Q: Am I safe if my organization requires secure passwords on our servers?

A: If you use MFA then yes, you are certainly more safe than having a weak password. However you are still leaving yourself potentially open to exploitation and information disclosure. We did not verify or validate if companies were using MFA on their sites. Also, we discovered hundreds of sites still using HTTP vs HTTPS and sending credentials in insecure methods.

Q: Why are you reporting this research?

A: Because we believe that organizations should actively evaluate the configuration of their container orchestration systems for risks that could potentially lead to a breach. In the case where admin access is compromised, there is significant damage that could be done. This includes remote code execution, abuse of services, and data destruction.

Q: How do I know if my company is at risk?

A: We are not releasing a list of IP addresses; doing so would be unethical and could put organizations at risk. You can check however whether you are using an orchestration system by looking into your AWS Logs. In particular you should focus on open ports and services running. You can also do a free risk assessment with our service https://www.lacework.com/free-trial. We will share information to trusted security researchers through typical secure channels.
Q: Do you know what companies are using these services and have this risk?

A: In many cases the certificates of the server name and the names of domains and URLs have information that could lead to the companies. That said, we are not tracking nor releasing any company names.

Q: Did you brute force any accounts or passwords, execute code, or configure anything during this research?

A: Absolutely not. Such activities would be contrary to our mission and not pertinent to the type of research we conduct.

Q: What cloud / datacenters did you discover where the workloads were hosted?

A: In alphabetical order:

A100 ROW GmbH
Amazon.com
Digital Ocean
Gtd Internet S.A.
Hangzhou Alibaba Advertising Co.,Ltd.
Hetzner Online GmbH
Iliad-Entreprises
Microsoft Azure
Nine Internet Solutions AG
ONLINE SAS
OVH Hosting
OVH SAS
Tencent cloud computing
University of California at Berkeley
WorldStream B.V.
Interested in more? Try Lacework for free and validate your security configuration:

www.lacework.com/free

Get an immediate audit of your AWS configuration for security best practices, an interactive report with detailed information on how to fix violations, and more.

© 2018 Lacework, Inc. Lacework and Polygraph are registered trademarks of Lacework. All other marks mentioned herein may be trademarks of their respective companies. Lacework reserves the right to change, modify, transfer, or otherwise revise this publication without notice.