Practical Lessons from Cloud Secret Exploitation

Neil Carpenter / Orca Security

Research Methodology

The purpose of this research

was to achieve a better understanding of how quickly attackers find assets and use secrets in each scenario. Armed with this information, security teams can establish the right protections to keep assets from being exposed, and perform the most effective remediations when an exposed asset has been found.

In our research we measured the following:

- Probability of asset access
- Tactics applied in asset access
- Probability of secret usage
- Tactics applied in secret usage

Data collection

Our research was conducted between January and May 2023.

1 KEY FINDING

Vulnerable assets are discovered rapidly

GITHUB	2 mins
SSH	4 min
HTTP	3 mins
S3 BUCKETS	1 hours

source: go.orca.security/honeypot-report

2 KEY FINDING

Time to key exploitation varies per asset type

GITHUB	2 mins
S3 BUCKETS	8 hours
AMAZON ECR	4 months

source: go.orca.security/honeypot-report

3 KEY FINDING

Different assets, different tactics

GITHUB	24/7 scanning
SSH	Malware, Cryptominers
S3 BUCKET	Follow breadcrumbs

source: go.orca.security/honeypot-report

4 KEY FINDING

Don’t rely on automated key protection

Keys blocked on GitHub but nowhere else

Even if blocked, there is still risk

source: go.orca.security/honeypot-report

Summary

A comparison of access times, key usage, top actions, and attack vector popularity for each of our honeypots.

Resource	Type	Time to Access	Top Access Action	Time to Key Use	Top Key Action	Attack Vector Popularity
GitHub	Service	2 minutes	N/A	2 mins	GetCallerIdentity	High
HTTP	TCP Port	3 minutes	Reconnaissance	No use	N/A	High
SSH	TCP Port	4 minutes	shell	No use	N/A	High
AWS S3 Bucket	Service	1 hour	HeadBucket	8 hours	GetCallerIdentity	High/Medium
Elasticsearch	TCP Port	2 hours	Reconnaissance	No use	N/A	High/Medium
Redis	TCP Port	2.5 hours	NewConnect	No use	N/A	High/Medium
Elastic Container Registry	Service	4 months	Reconnaissance	4 months	GetCallerIdentity	Medium
Amazon EBS (AMI)	Service	No access	N/A	N/A	N/A	Low
DockerHub	Service	No access	N/A	N/A	N/A	Low

POPULARITY

Orca Security, Highly Confidential

Summary: Why are some resources targeted more than others?

Attackers run their operations like a business. It basically comes down to:

“Where can I get the best bang for my buck?”

Cost/benefit ratio:

- Text here

How much the resource is used:

- Text here

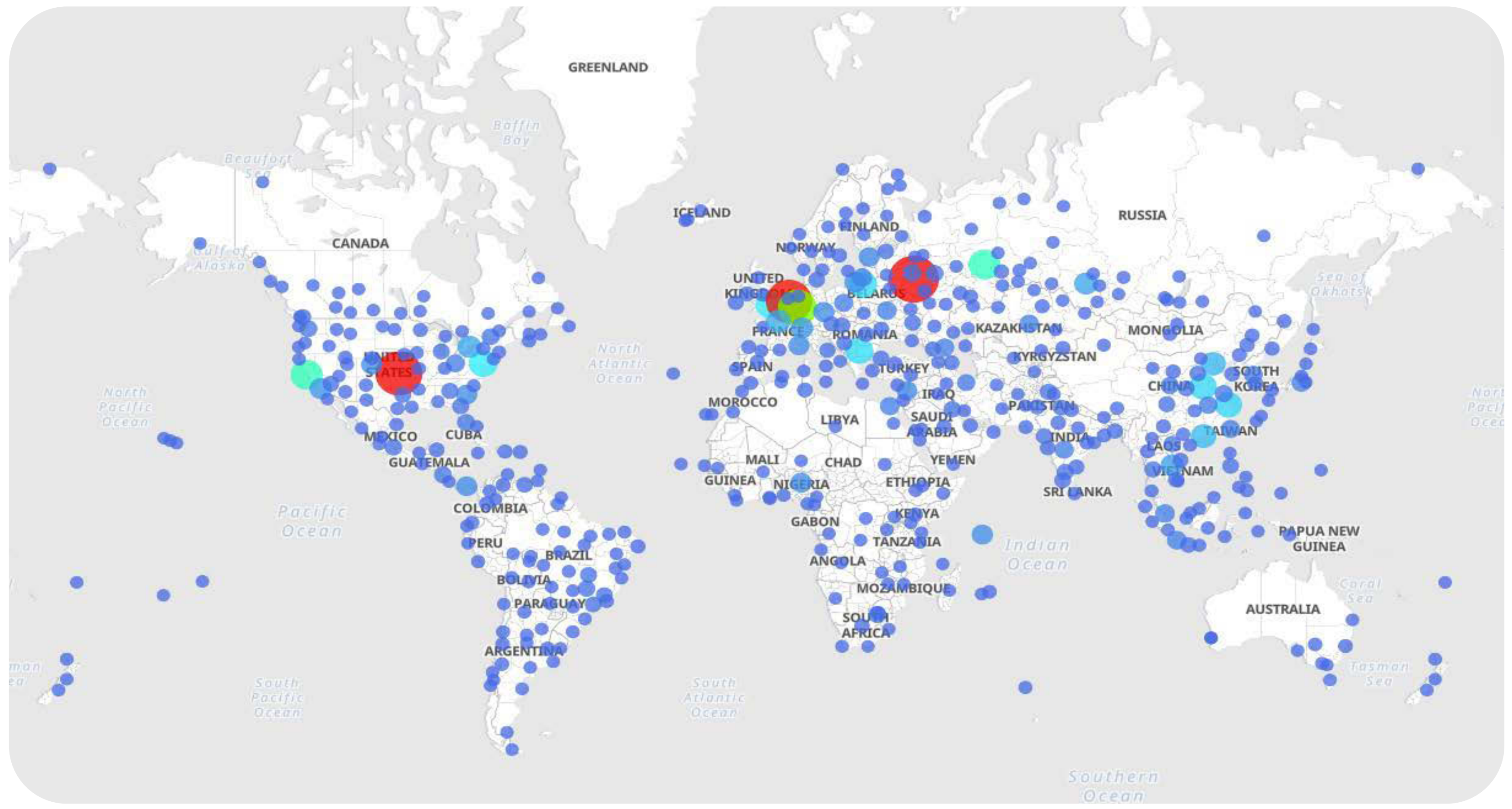
How prone it is to contain secrets:

- Text here

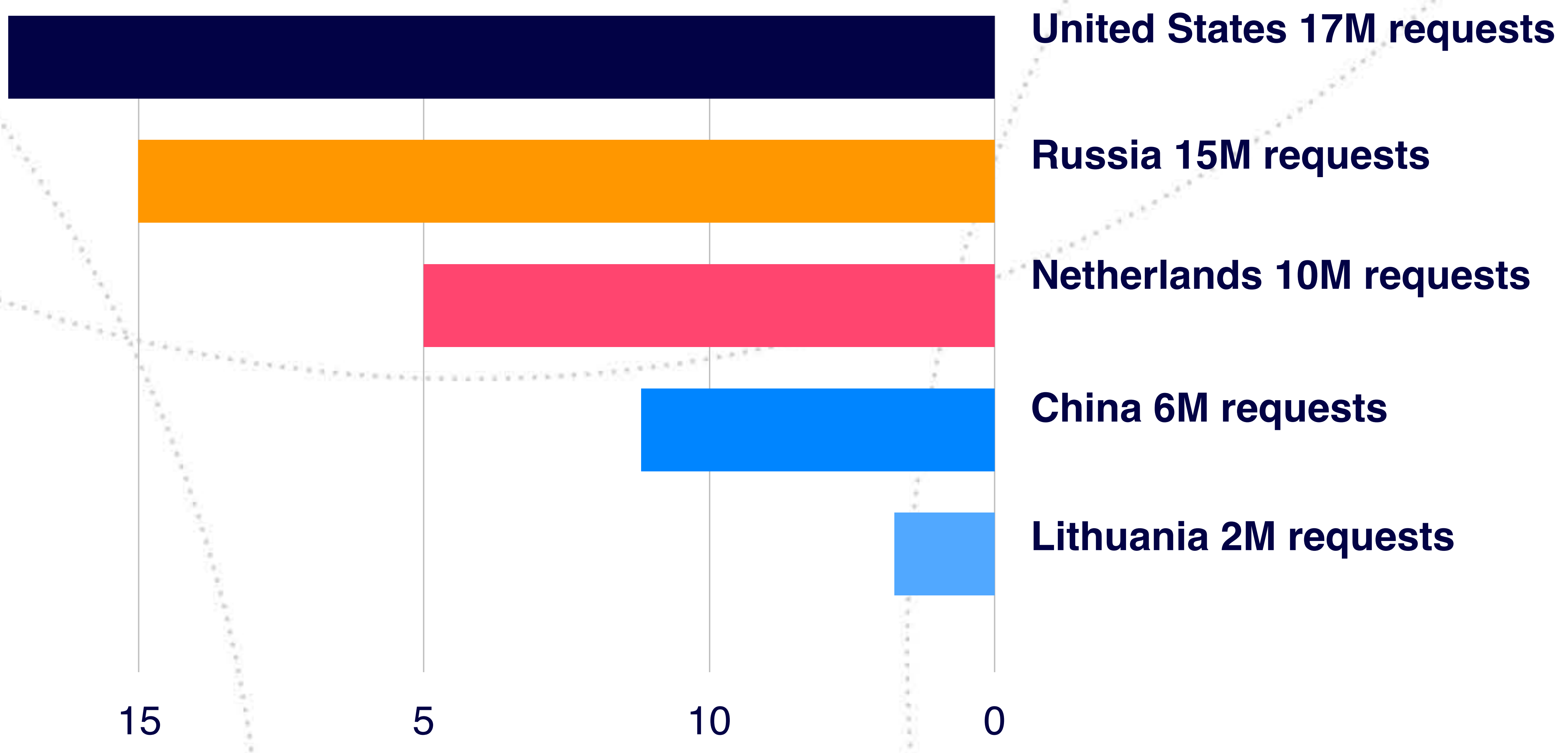
Summary: Attacker Heatmap

Most attacks originated from IP addresses in the US, which makes sense because most IPs are from the US. However, in second place is Russia, indicating that a fair number of attacks are originating from there. Surprisingly, third in the list is the Netherlands.

IP origins of our honeypot attackers



Top 5 countries



Key Recommendations

So how can defenders up their game and stay one step ahead of the attackers? We have listed our key recommendations below:

01

Identify & Manage Your Secrets

If you don't properly store your secrets, it's just a matter of time before attackers will find them. Combining a strong secret management approach using vaults with automatic identification of where secrets are stored, enables you to reduce the chance a secret is compromised and identify where sensitive data & secrets is inadvertently exposed. With Orca's data classification and attack path analysis, you can identify that exposure even if, unlike our honeypots, it would require an attacker to exploit multiple weaknesses in the environment to get to the data.

01

Check for Secrets Before Deploying Code

It is essential to scan for secrets prior to committing code. For developers working in a platform like GitHub, in particular, it may make sense to combine Orca's Shift-Left secret scanning with pre-commit hooks, allowing you to scan code on a developer's workstation before it's pushed to a platform like GitHub. When it only takes attackers minutes to find your mistake, it makes sense to go to great lengths to prevent this sort of exposure. Even if GitHub does include push protection, it's better to be safe than sorry and not rely on GitHub to find your secrets, since the stakes can be high. In addition, even though the permissions of our honeypot key were greatly reduced, this does not mean that an attacker could not still do damage using the remaining permissions.

03

Check your Git History

Don't forget that attackers are not only scanning new GitHub commits, but are also looking for secrets in your Git History. So make sure that when you do remove a secret from a commit, you also remove it from your history. The Orca platform detects when keys and other secrets are in your Git History and need to be removed.

Copyright Orca Security 2023

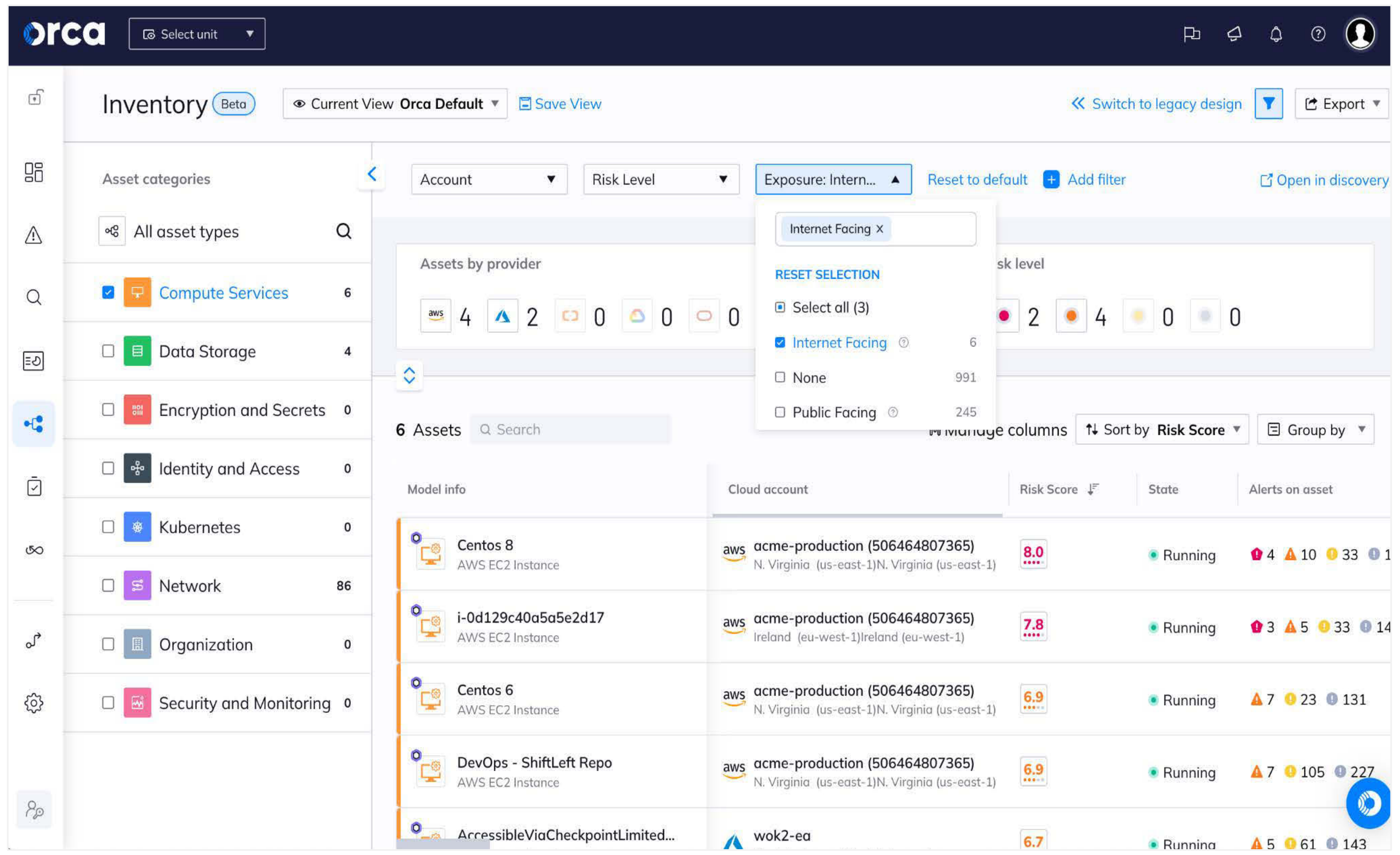
Key Recommendations

04

Set Public Access Only When Strictly Needed

Although it may seem obvious that you should limit public access to cloud assets whenever possible, unfortunately, in real-world deployments, accidental public exposure is common enough that attackers are scanning for many of these exposed resources *all the time*. Organizations should continually assess which resources in their cloud estate are publicly exposed and ensure that there is an important business reason for doing so.

To further reduce the chances of human error opening something to the outside world, it also makes sense to lock down cloud environments so that changes are made as code ("Infrastructure as Code", or IaC), This can be combined with [security policy scanning of IaC artifacts](#) to ensure that mistakes are caught & addressed before they're deployed in the cloud.



Asset categories

All asset types	
✓ Compute Services	6
Data Storage	4
Encryption and Secrets	0
Identity and Access	0
Kubernetes	0
Network	86
Organization	0
Security and Monitoring	0

Assets by provider: aws 4, Azure 2, 0, 0, 0

Exposure filter:

Select all (3)	
✓ Internet Facing	6
None	991
Public Facing	245

Risk level: ● 2, ● 4, ● 0, ● 0

6 Assets

Model info	Cloud account	Risk Score	State	Alerts on asset
Centos 8 AWS EC2 Instance	acme-production (506464807365) N. Virginia (us-east-1)N. Virginia (us-east-1)	8.0	Running	4 10 33 1
i-0d129c40a5a5e2d17 AWS EC2 Instance	acme-production (506464807365) Ireland (eu-west-1)Ireland (eu-west-1)	7.8	Running	3 5 33 14
Centos 6 AWS EC2 Instance	acme-production (506464807365) N. Virginia (us-east-1)N. Virginia (us-east-1)	6.9	Running	7 23 131
DevOps - ShiftLeft Repo AWS EC2 Instance	acme-production (506464807365) N. Virginia (us-east-1)N. Virginia (us-east-1)	6.9	Running	7 105 227
AccessibleViaCheckpointLimited...	wok2-ea	6.7	Running	5 61 143

Having a cloud security solution in place that can alert when assets are publicly exposed is essential.

Copyright Orca Security 2023

Key Recommendations

05

Follow “Security by Obscurity”

We saw that attackers often take an opportunistic approach and try to guess usernames, passwords and asset names by simply going down a list of the most commonly used names. For instance with our SSH honeypots, attackers were simply trying the ones that were the most commonly used.

As defenders, it is therefore recommended to use obscure usernames and asset names that cannot easily be guessed (admin as a username should be off the table, and customer-database is not a great name for a bucket either!). Although this should of course *not be your only security recourse*, the more difficult you can make it for the attacker, the sooner they will get tired and move on to the next ‘low hanging fruit’.

06

Bolster Authentication and Limit Authorization

One of the commonalities across our honeypots was that they had weak authentication – while this is what we wanted in a honeypot, it is a common contributing factor in security incidents. For those assets that must be exposed to the Internet, ensure that strong authentication is enabled. These options differ from service to service but may include certificate-based authentication and, if the service is accessed directly by humans, multi-factor authentication (MFA).

Additionally, where possible, ensure that the accounts that connect to the service have the minimum possible access. Limiting the scope of authorization can help to reduce the impact of an attacker managing to authenticate to the service.

Copyright Orca Security 2023

Key Recommendations

07 Monitor for Malicious Process Execution and Malware

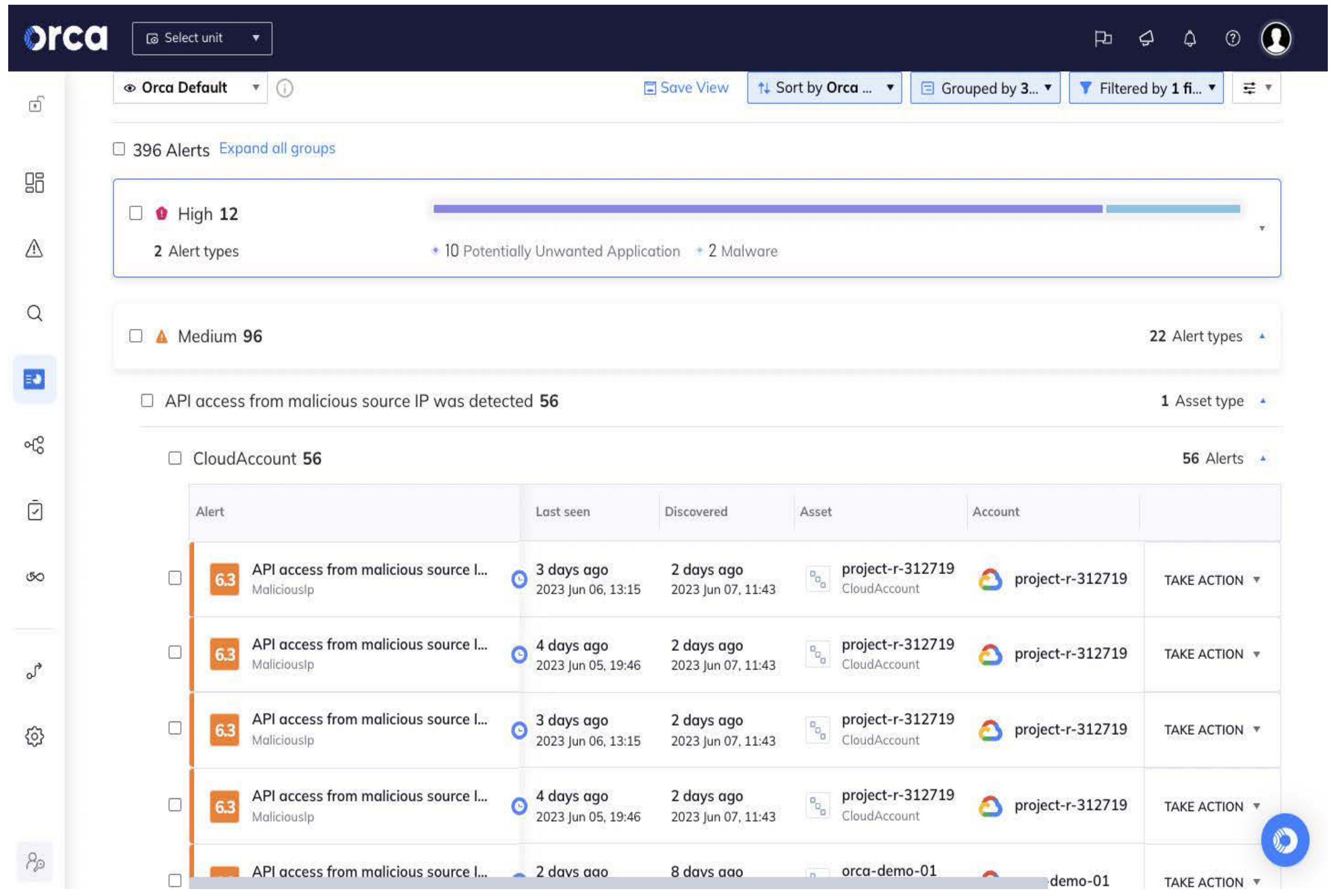
For those services where process execution is a possibility, monitor the systems for abnormal execution and potentially malicious files & processes. SSH is a particular risk here as it is, explicitly, a shell for executing things; however, other services may also explicitly or, via a vulnerability, allow an attacker to execute code. In our RDS honeypot, we saw evidence of an attempt to execute a cryptominer.

08 Assess and Patch Vulnerabilities

Vulnerability assessment and patching should be a priority for assets directly exposed to the Internet. CVEs that allow for remote code execution or privilege escalation may be used by attackers to bypass many of the other recommendations in this list.

09 Implement Port Hygiene

Determine which ports are necessary for your specific applications and services to function properly and limit access to only those applications. If you are using cloud providers like AWS, Azure, or Google Cloud, configure security groups or firewall rules to limit access to only required ports and restrict access from any unauthorized sources.



396 Alerts Expand all groups

High 12
2 Alert types
10 Potentially Unwanted Application · 2 Malware

Medium 96 — 22 Alert types

API access from malicious source IP was detected 56 — 1 Asset type

CloudAccount 56 — 56 Alerts

Alert	Last seen	Discovered	Asset	Account	
6.3 API access from malicious source I... MaliciousIp	3 days ago 2023 Jun 06, 13:15	2 days ago 2023 Jun 07, 11:43	project-r-312719 CloudAccount	project-r-312719	TAKE ACTION
6.3 API access from malicious source I... MaliciousIp	4 days ago 2023 Jun 05, 19:46	2 days ago 2023 Jun 07, 11:43	project-r-312719 CloudAccount	project-r-312719	TAKE ACTION
6.3 API access from malicious source I... MaliciousIp	3 days ago 2023 Jun 06, 13:15	2 days ago 2023 Jun 07, 11:43	project-r-312719 CloudAccount	project-r-312719	TAKE ACTION
6.3 API access from malicious source I... MaliciousIp	4 days ago 2023 Jun 05, 19:46	2 days ago 2023 Jun 07, 11:43	project-r-312719 CloudAccount	project-r-312719	TAKE ACTION
API access from malicious source I...	2 days ago	8 days ago	orca-demo-01	demo-01	TAKE ACTION

[Cloud Detection & Response](), which automatically detects anomalies and suspicious events in cloud environments, should be deployed to alert security teams to possible attacks in progress.

Copyright Orca Security 2023

Key Recommendations

10

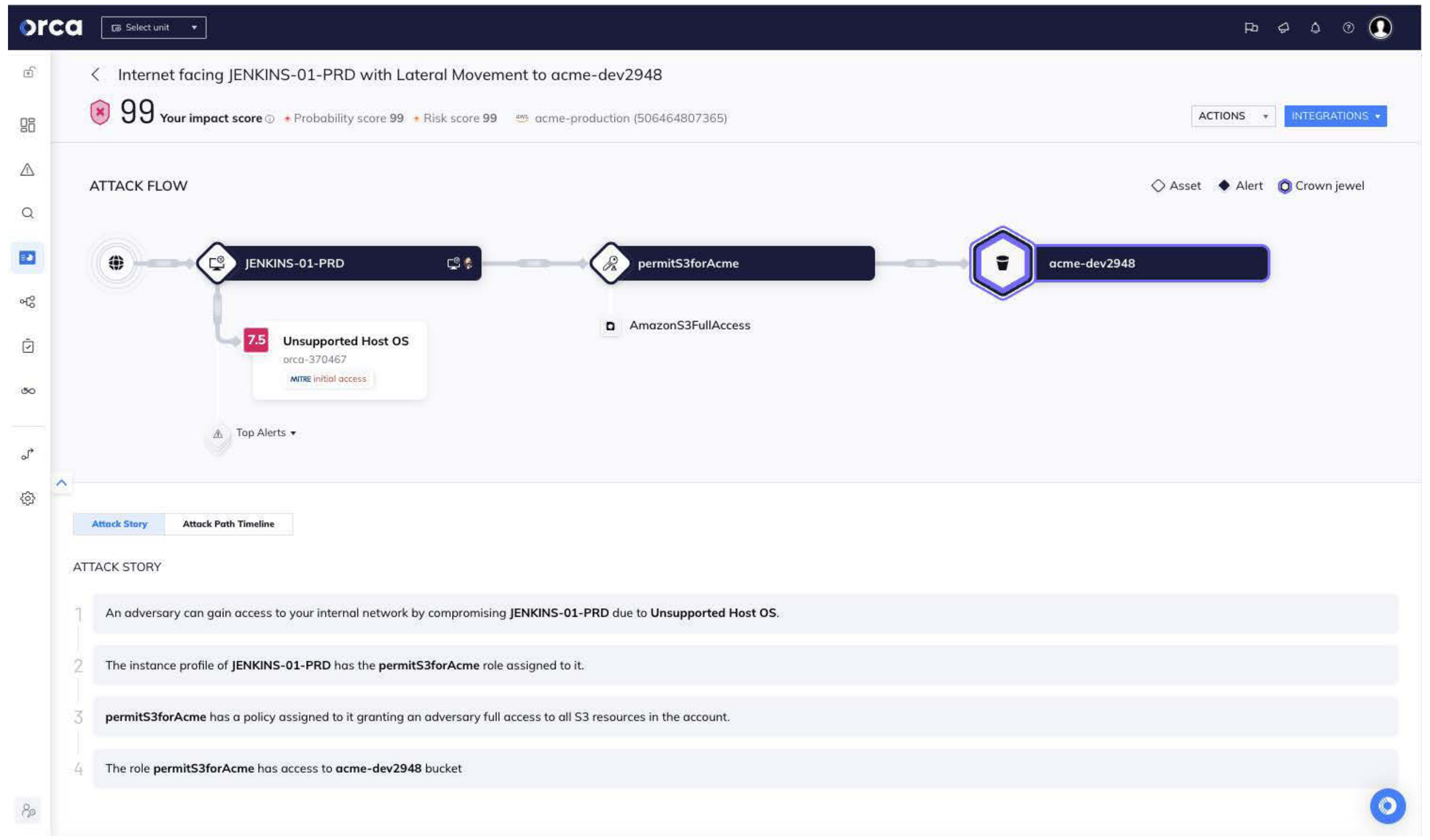
Prioritize Protection of Your Crown Jewels

We know that attackers are continually scanning cloud environments looking for vulnerable resources. Unfortunately, it's just a matter of time before attackers are going to find a vulnerability in your defense.

Therefore, instead of trying to fix *all* vulnerabilities and remediate *all* risks, which is frankly a Sisyphean task, a better strategy is to ensure that your crown jewels, such as PII, intellectual property, financial information, and other sensitive data, are protected like Fort Knox. Any risks that endanger your crown jewels should always be prioritized and fixed *first*.

However, this does not mean that you only focus on risks that are directly connected to your crown jewels. Attackers will take advantage of different weaknesses in your environment to move laterally and ultimately reach their target.

Therefore, it's essential that security teams have insight into the different attack paths (i.e. combinations of risks) that endanger the organization's crown jewels and then ensure that these attack paths are deactivated in the fastest and most effective way.



Internet facing JENKINS-01-PRD with Lateral Movement to acme-dev2948

99 **Your impact score** • Probability score **99** • Risk score **99** acme-production (506464807365)

ATTACK STORY

1 An adversary can gain access to your internal network by compromising **JENKINS-01-PRD** due to **Unsupported Host OS**.

2 The instance profile of **JENKINS-01-PRD** has the **permitS3forAcme** role assigned to it.

3 **permitS3forAcme** has a policy assigned to it granting an adversary full access to all S3 resources in the account.

4 The role **permitS3forAcme** has access to **acme-dev2948** bucket

Attack paths show the combinations of risks that are a direct path to your critical assets

Copyright Orca Security 2023