Lessons Learned on Attack Detection in the Cloud

Nick Jones

Cybersikkerhed i den finansielle sektor 2023

Who Am I?

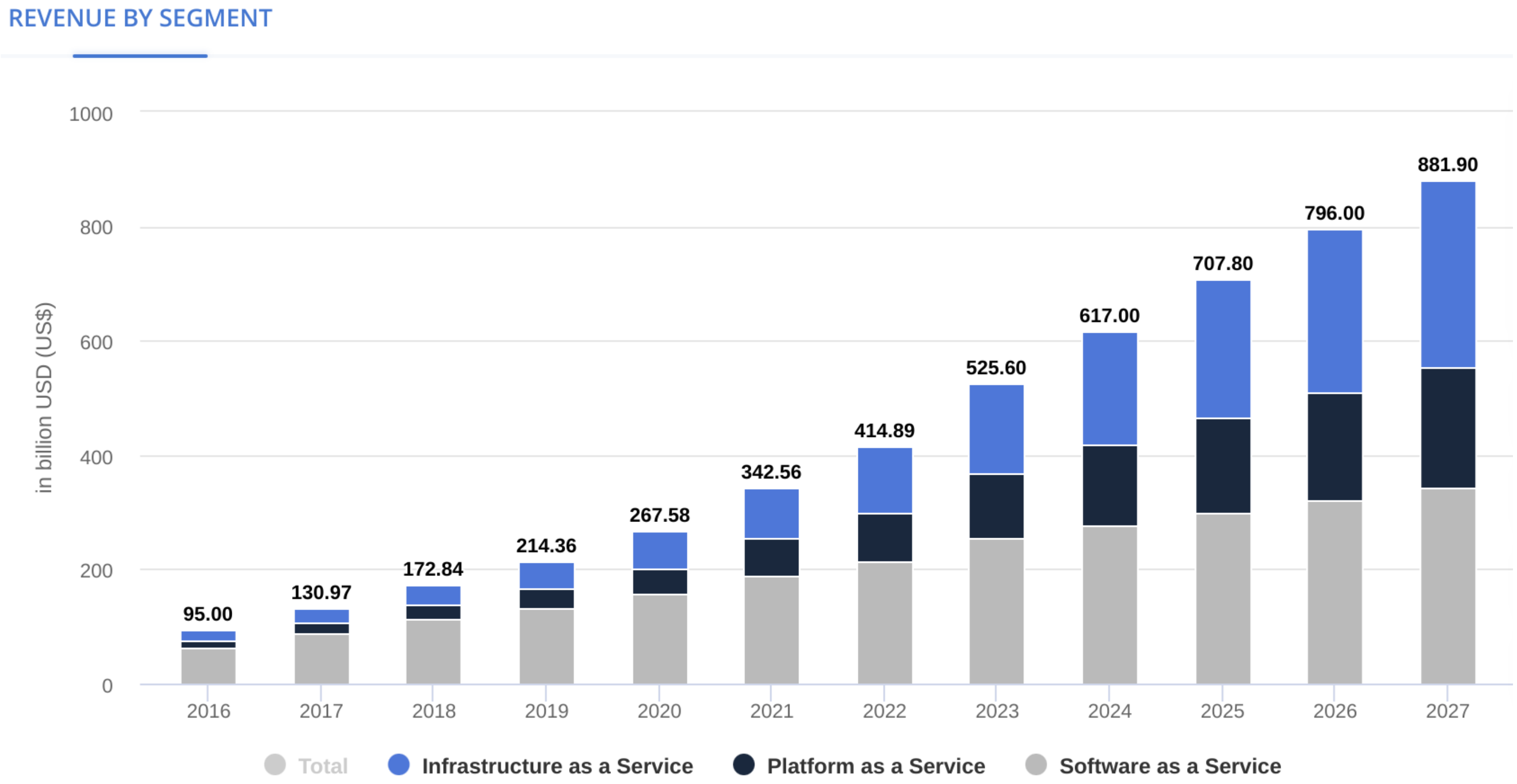
Nick Jones – @nojonesuk

- Principal Consultant @ WithSecure
- Cloud Security Consulting Lead
- AWS Community Builder
- Focus on:
  - Security automation & DevSecOps
  - Attack detection
  - Offensive Security

Agenda

1. Why Does This Stuff Matter?
2. Common Attacks
3. Cloud Attack Detection
4. Designing Your Detection Stack
5. Building Effective Detections

Everyone’s Using Cloud



https://www.statista.com/outlook/tmo/public-cloud/worldwide#revenue

Security Teams Slow To Adapt

OUR EDR WILL PROTECT YOUR AWS ACCOUNTS TOO
WE HAVE NO VIRTUAL MACHINES

Common Attacks

Inherently Flawed Data

Not all breaches get spotted

Providers hate talking about it

Focus on low hanging fruit

Open S3 Buckets

The perennial problem

- Biggest source of breaches for years now
- Trivial to find and exploit

Situation is Improving

- AWS providing good options now to prevent
- Enable block public buckets everywhere!

Photo from https://www.flickr.com/photos/electronicfrontierfoundation/50617066023

WITH secure

What Else are Attackers Doing?

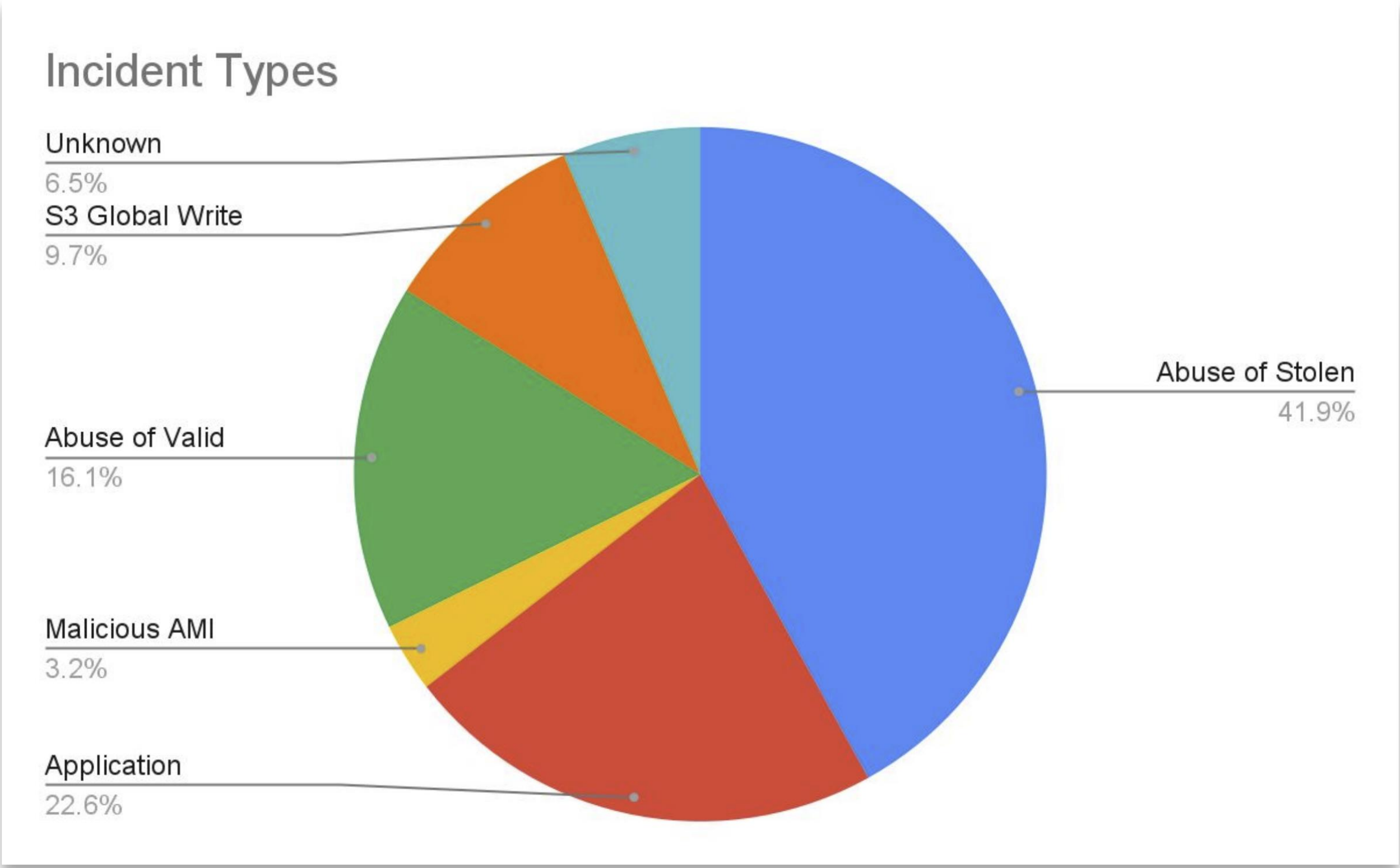


Image from https://speakerdeck.com/ramimac/learning-from-aws-customer-security-incidents-2022?slide=20

@ramimacisabird

W/TH secure

A Note on Cloud Zero Days

Cool but mostly irrelevant

- CloudVulnDB tracking 138 vulns
- One exploited in the wild, no breaches reported
- https://www.cloudvulndb.org

Expect this to change

- **Very** active research area
- Expect APTs to catch up to security researchers here

Real-World Attacks in Summary

Attackers look for the easiest path

Most attacks are opportunistic

The basics helps stop APTs too

Most get hit by the basics:

Public S3 buckets

Forgotten accounts

Leaked credentials

Bad leaver handling

You **probably** won't get breached by:

Encryption at rest

Not using fancy CSP security features

Zero days

CSP Insider threat

Other Attack Paths

Cloud Native Phishing

Identity Platforms / SSO

- Okta, Ping, OneLogin, Auth0…
- Single point of access
- Supply chain risk too

Interesting security properties

- MFA, CAPs etc etc
- Often poor session management
- Get the session token, get access to ***everything***

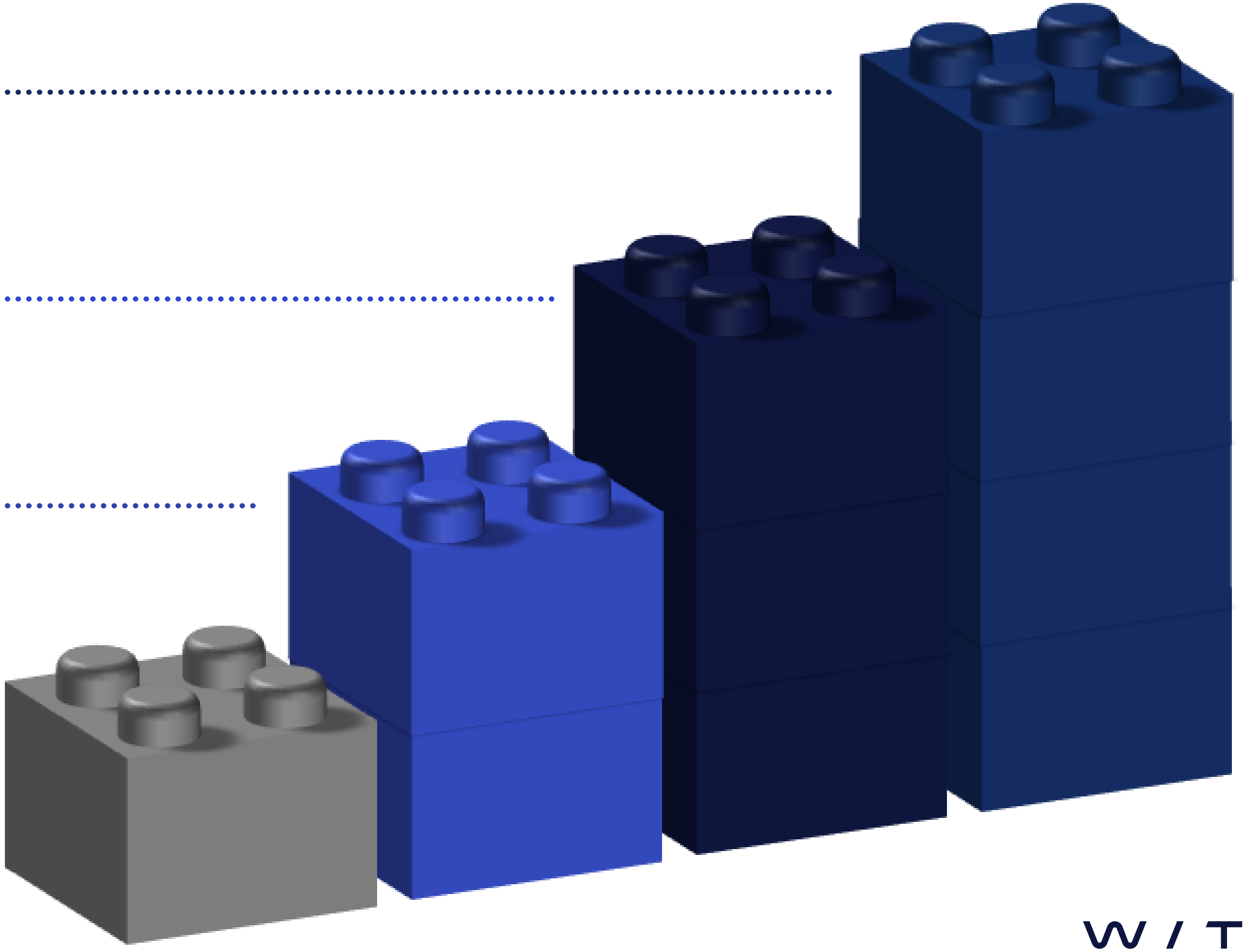
Exploiting Development Workflows

Source Code Management
Everyone uses GitHub or similar to develop and collaborate on their code

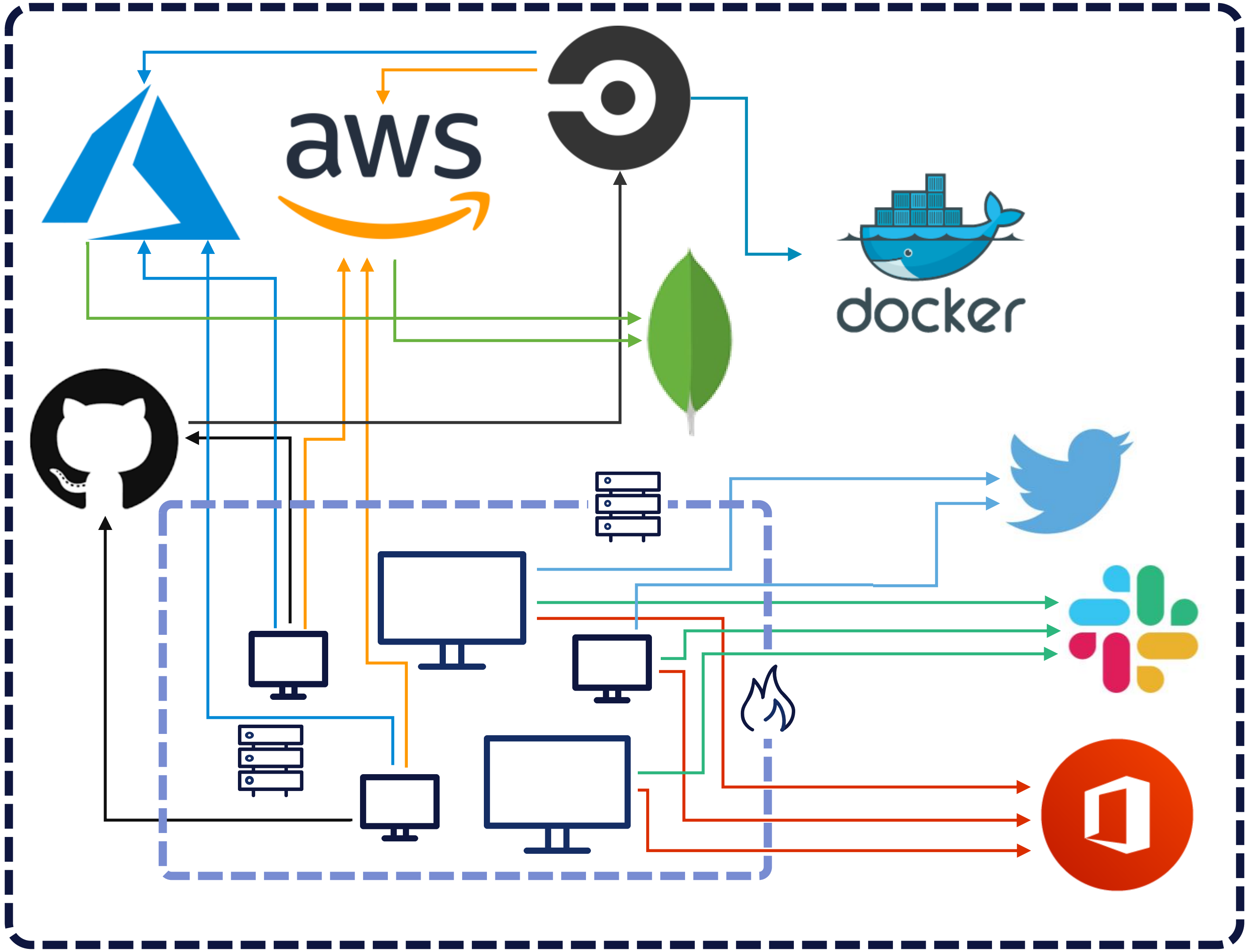
CI/CD
Continuous integration and continuous delivery to automate testing and deployment of cloud workloads

Dev Usability > Security
Enabling devs to move at speed often means system architectures and controls are not well hardened

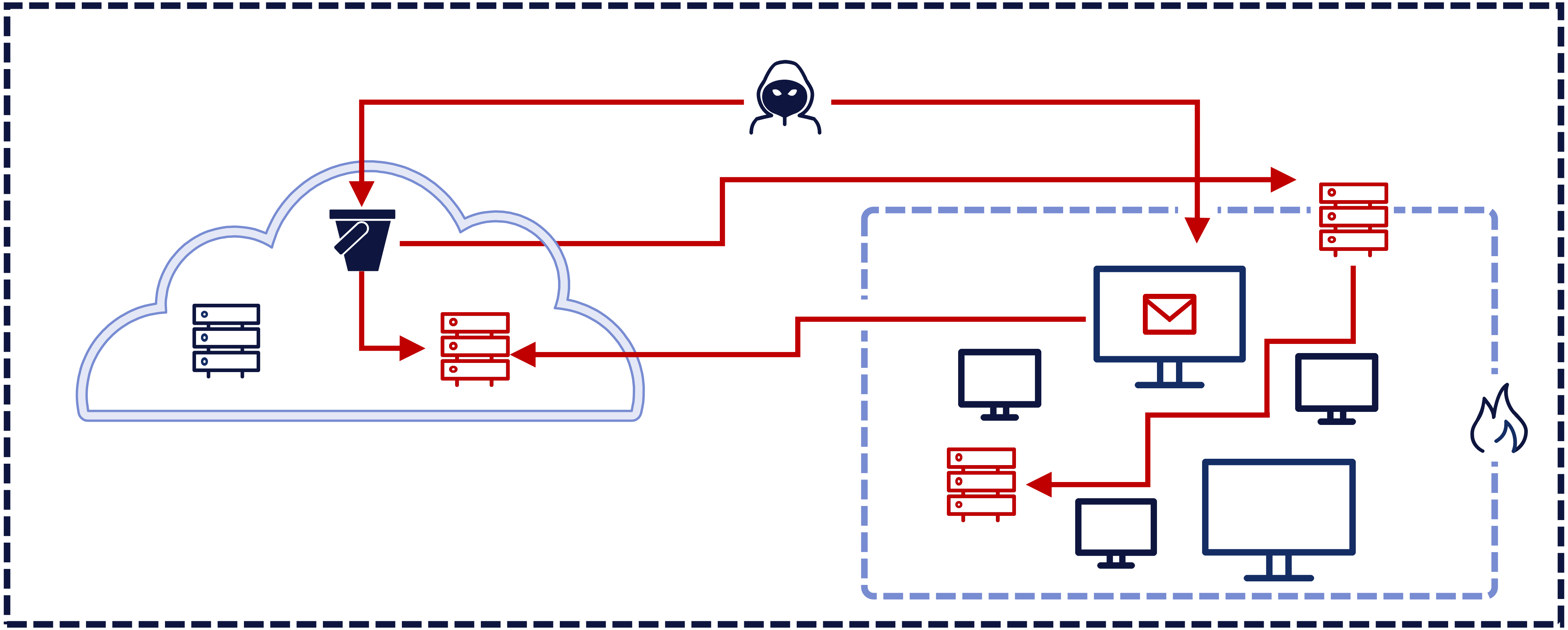
Automatic IaC Deployments
IaC changes often automatically deployed after merging – can we bypass approvals process?



Enterprise Cloud Adoption



Attackers Pivot



“

An ounce of prevention is worth a pound of cure

-- Benjamin Franklin

”

Exposed Resources

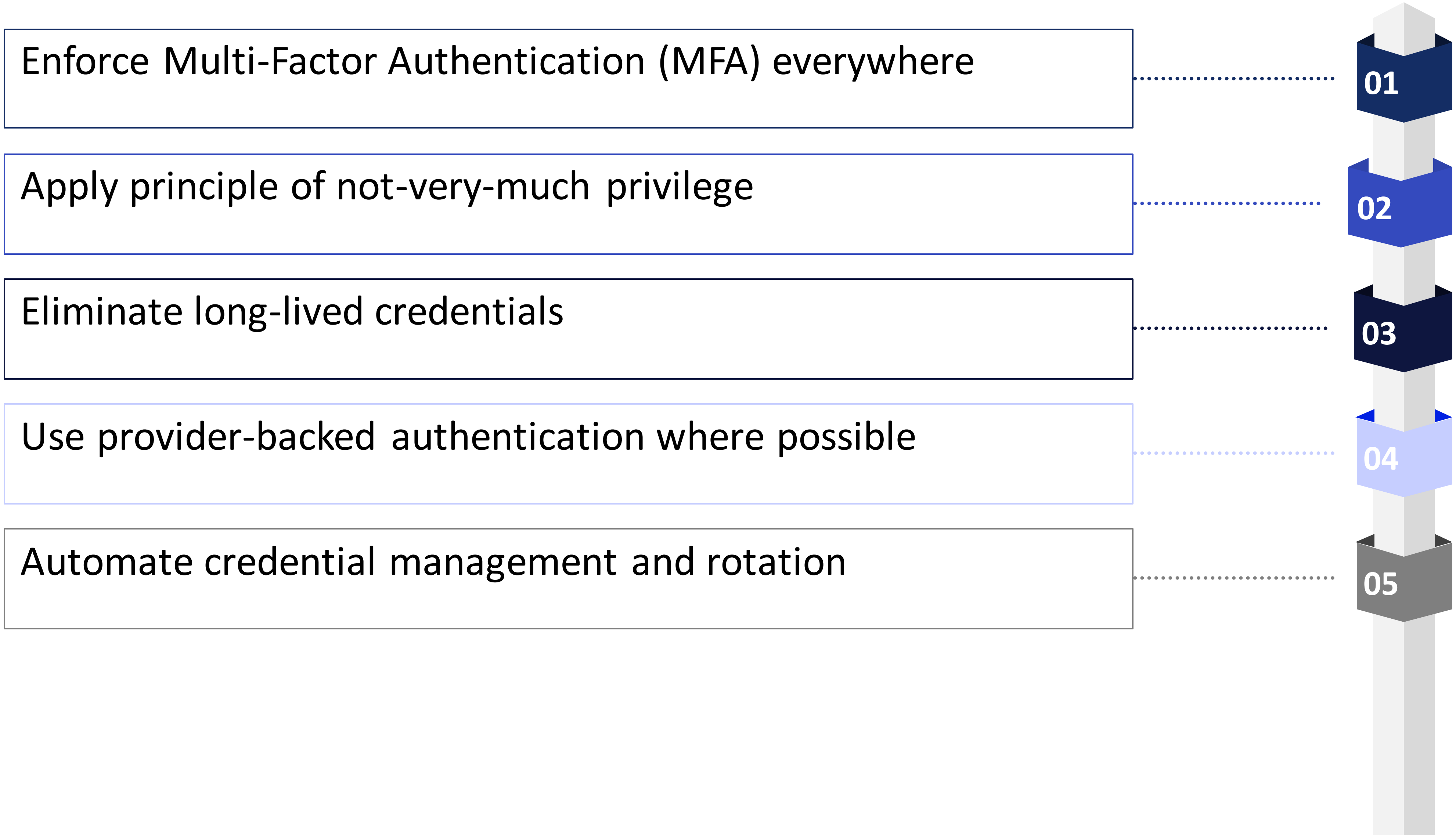
The biggest data leak risk

- Trivial to find
- Trivial to exploit

Relatively easy to find/fix

- AWS Security Hub, Azure Security Center
- Free/Open Source Scanners – prowler, scoutsuite etc

Strong Identity Controls



01 Enforce Multi-Factor Authentication (MFA) everywhere

02 Apply principle of not-very-much privilege

03 Eliminate long-lived credentials

04 Use provider-backed authentication where possible

05 Automate credential management and rotation

Secrets Management

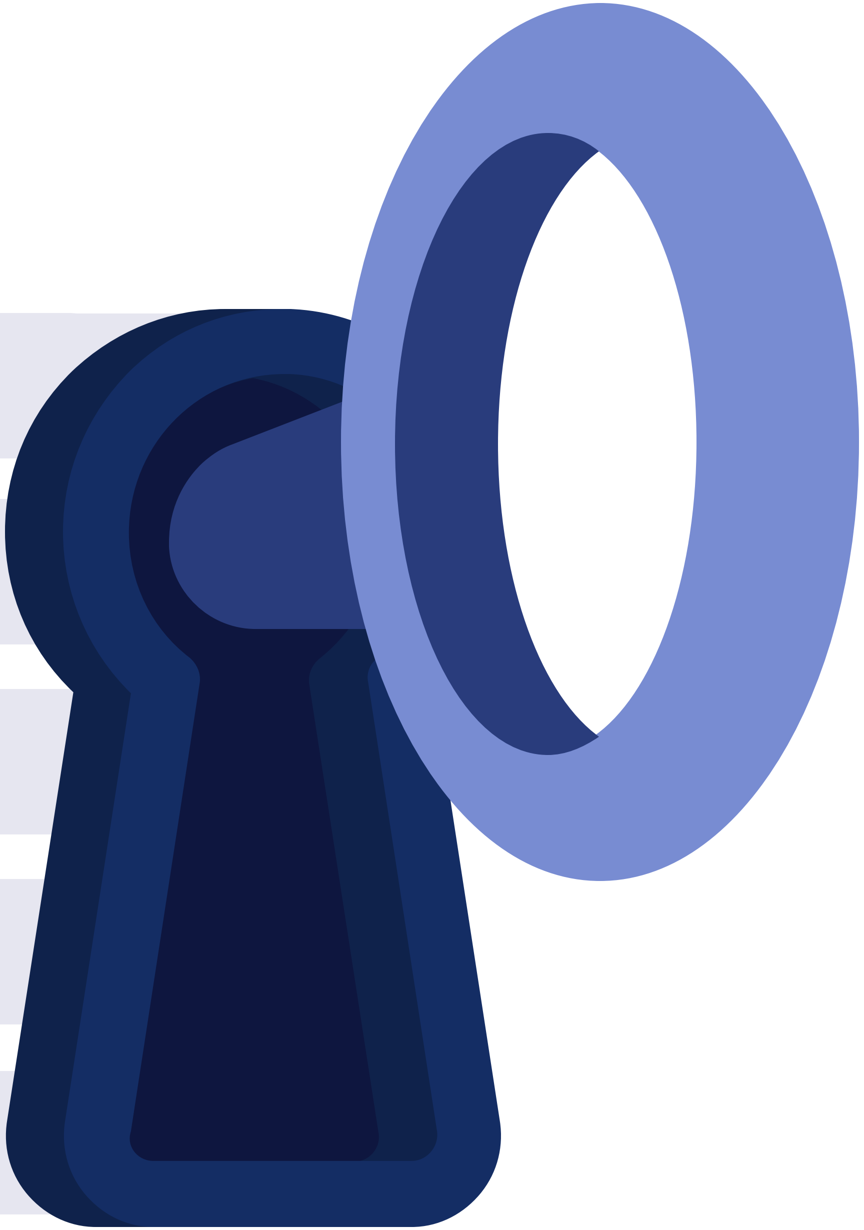
Often the key point of failure

Where do applications store their secrets?

How are credentials shared and rotated?

How do you know when secrets are leaked?

Use provider-offered secret storage services!



Cloud Attack Detection

How Cloud Detection Differs

UNCERTAINTY OF MALICIOUS INTENT

Fewer actions in the cloud are obviously bad compared to on-premise, making generic detection rules harder

CONTEXT IS KEY

Anomalies will vary by environment. Behavioral analytics are important here, so is developing environment-specific alerting.

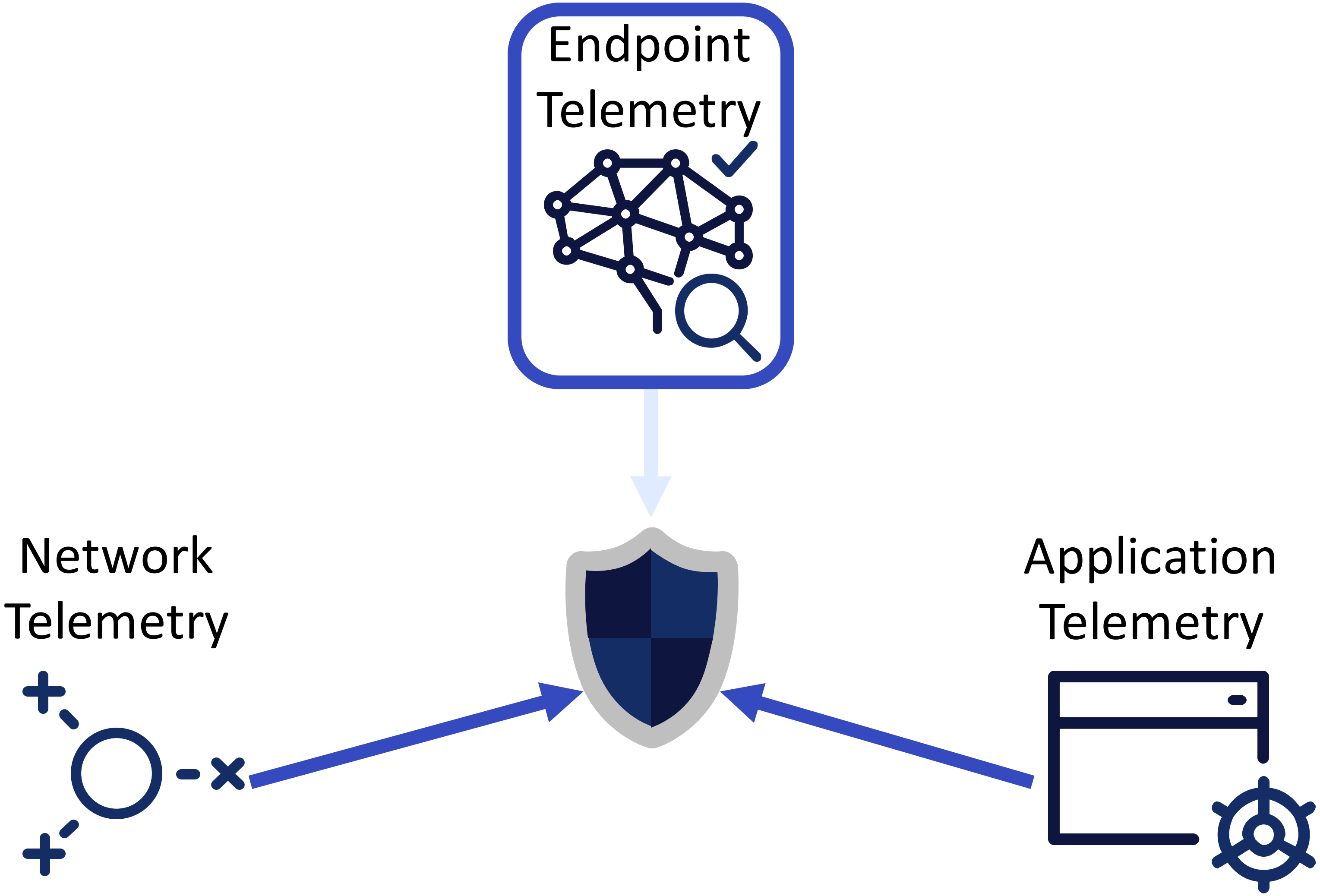
GAINING VISIBILITY IS EASIER

Org-wide CloudTrail, etc. makes it easier to gain visibility into much of your estate. Shadow IT now the primary issue, rather than coverage of known assets.

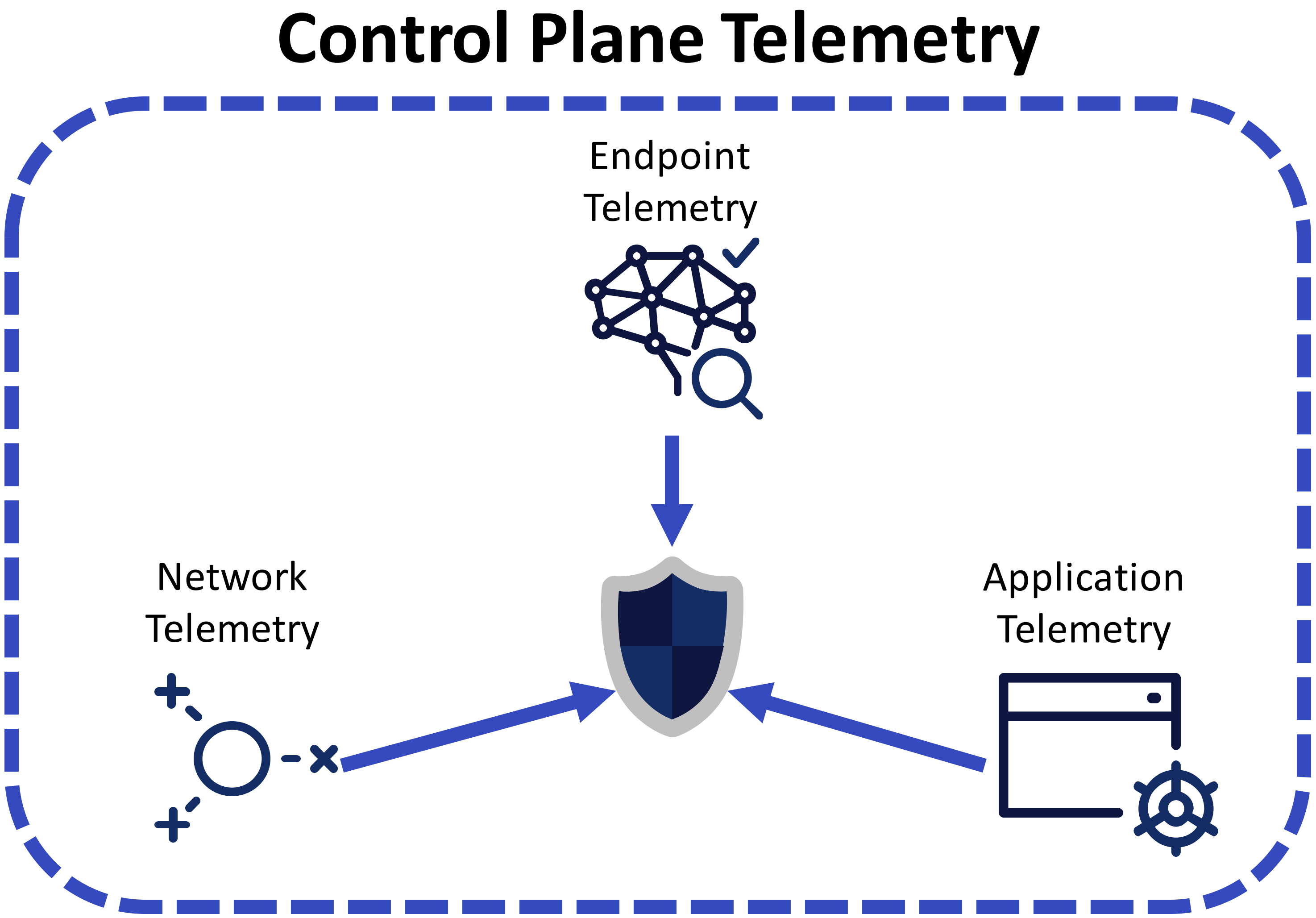
ATTACKERS AUTOMATE

Attackers leveraging scripted attacks to abuse stolen credentials for cryptocurrency mining. With an API-driven attack surface by-design, it’s easier to automate targeted attacks too.

On-Premises Telemetry



Cloud Telemetry

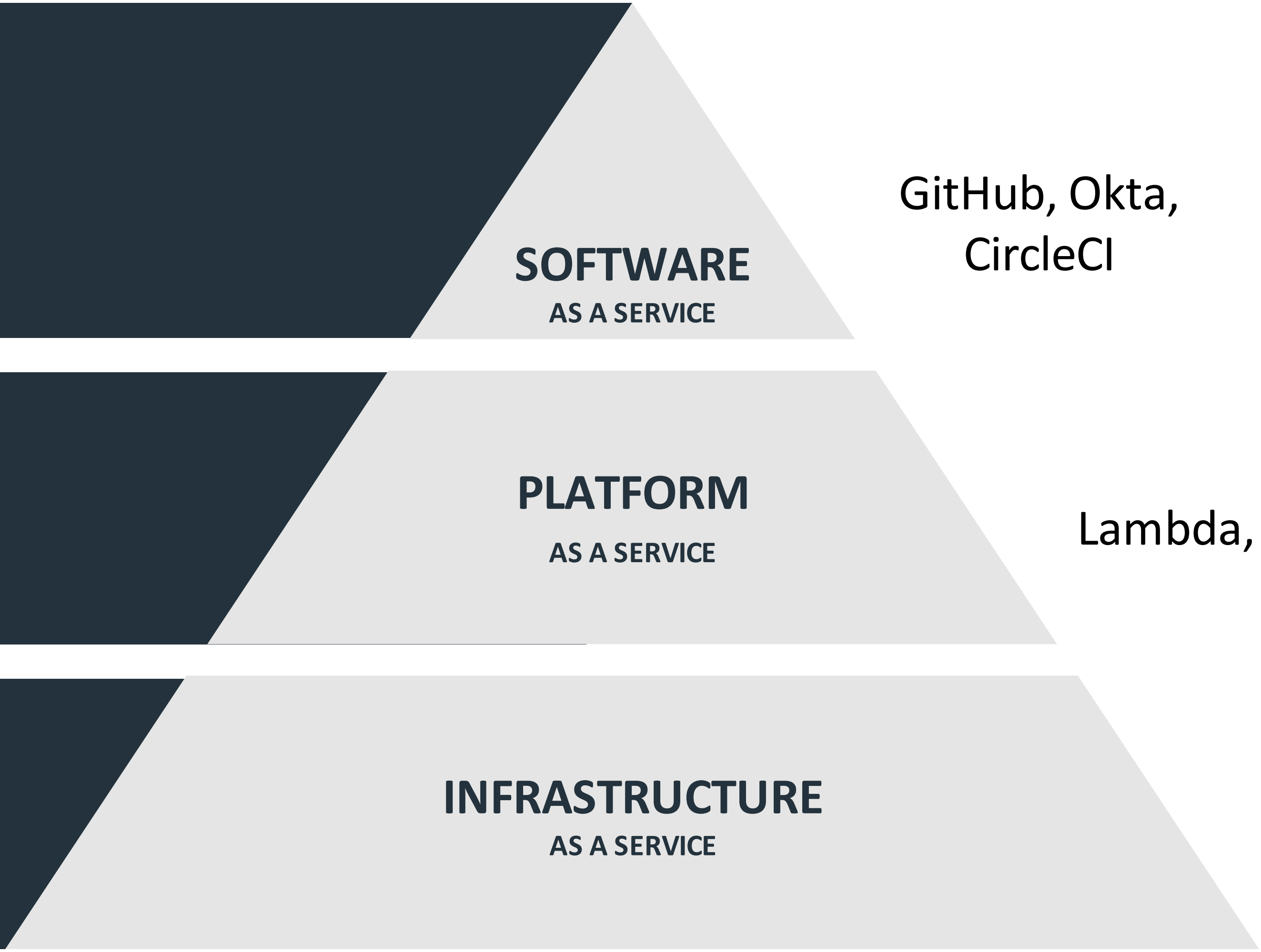


Cloud Services

🤷

- CloudTrail + Object-level Data Events
- Azure Audit Logs etc

- EDR / VPC Flow Logs / CloudTrail
- App Logs



SOFTWARE
AS A SERVICE

PLATFORM
AS A SERVICE

INFRASTRUCTURE
AS A SERVICE

GitHub, Okta, CircleCI

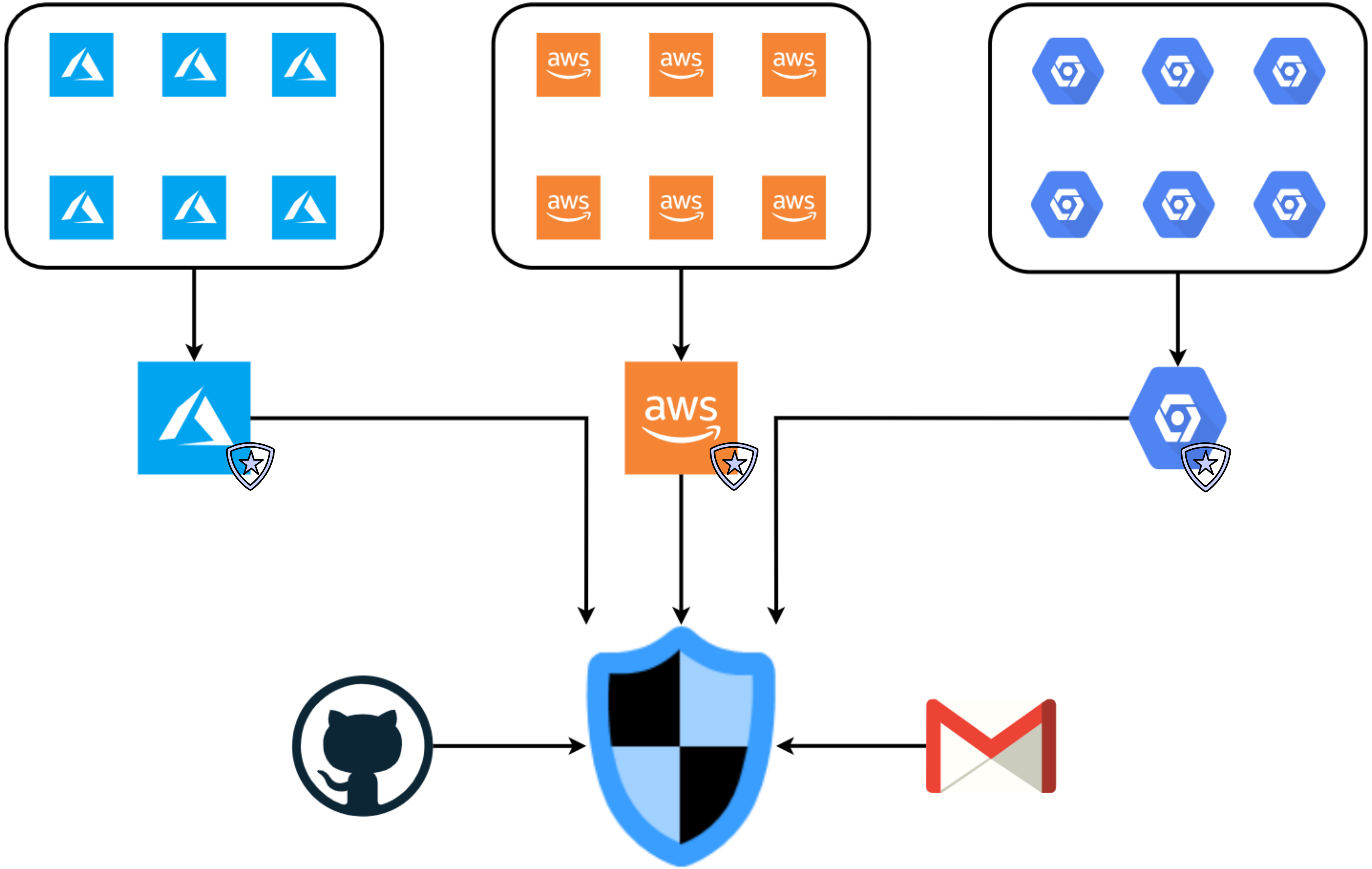
Lambda, S3

EC2

← Administrative Requirements of the Customer →

Designing Your Cloud Detection Stack

Centralise Everything



W/TH secure

Data Sources

SOURCE	BENEFIT
Control Plane audit logs (CloudTrail, Audit Log etc)	**Visibility of all administrative actions**
Service Specific Logs (storage access logs, function executions, KMS key access etc.)	**Shows access and usage of specific resources and services, which may help to track lateral movement or actions on objective**
Cloud-native detection services	**Detection of known bad activity**
API Gateway/WAF Logs	Identify malicious requests to applications
Network flow logs	Identify anomalous traffic by source/destination, volumes
System logs from any VMs	Grants OS-level visibility of potential attacker activity
Endpoint Detection and Response agents in VMs	Detects malicious activity within VMs as with on premises
Application logs	Provides app-specific contextual information

Control Plane Audit Logs

Provider specifics

- AWS – CloudTrail
- Azure – Audit Log
- GCP – Audit Log
- Kubernetes – Audit Log

Why bother?

- The key data source for all cloud native exploitation
- Logs (almost) every control plane level event

Considerations

- "Data events" not always enabled
- For AWS, enable global events and multi-region logging

Service-Specific Telemetry

Provider Specifics

- AWS – S3 access/object logs, Lambda executions, KMS key access
- Azure – Storage account access logs, function executions
- GCP – Storage Logs, Cloud Function Executions etc

Why bother?

- Can generate high fidelity telemetry on critical actions

Considerations

- Requires that use cases and hunt queries are developed per environment
- Enable on a case-by-case basis

Cloud-Native Detection Services

Provider Specifics

- AWS – GuardDuty
- Azure – Advanced Threat Protection
- GCP – Security Command Center

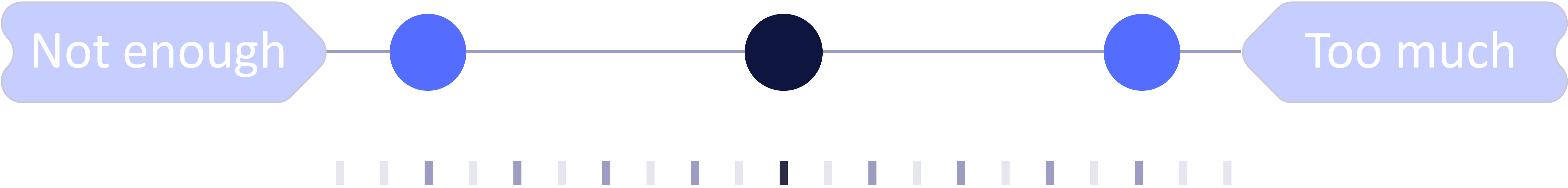
Why bother?

- Minimal integration effort compared to other sources
- Cost-effective way to detect low sophistication attacks

Considerations

- Typically signatures on known bad, though some ML/AI now too
- Optimised for low false positive across all cloud users

Visibility vs Cost & Usability



Common Mistakes and Pitfalls

Telemetry aggregated with no provided (or available) context

Bad in one account, **Good** in another

Overlooking authentication logs

Interfaces between On-premise/Cloud, management interfaces, etc

Never too early to threat model and test offensive scenarios

Common Mistakes and Pitfalls

Build the context from the architectural stage

What should the environment do?

What shouldn't it do?

Share with analysts, give them the insight into what is normal

BONUS: Exercising this with analysts gets them used to investigation in cloud

Building Effective Detections

Where To Start

01 Threat model your environment, identify attack paths and likely attacker actions

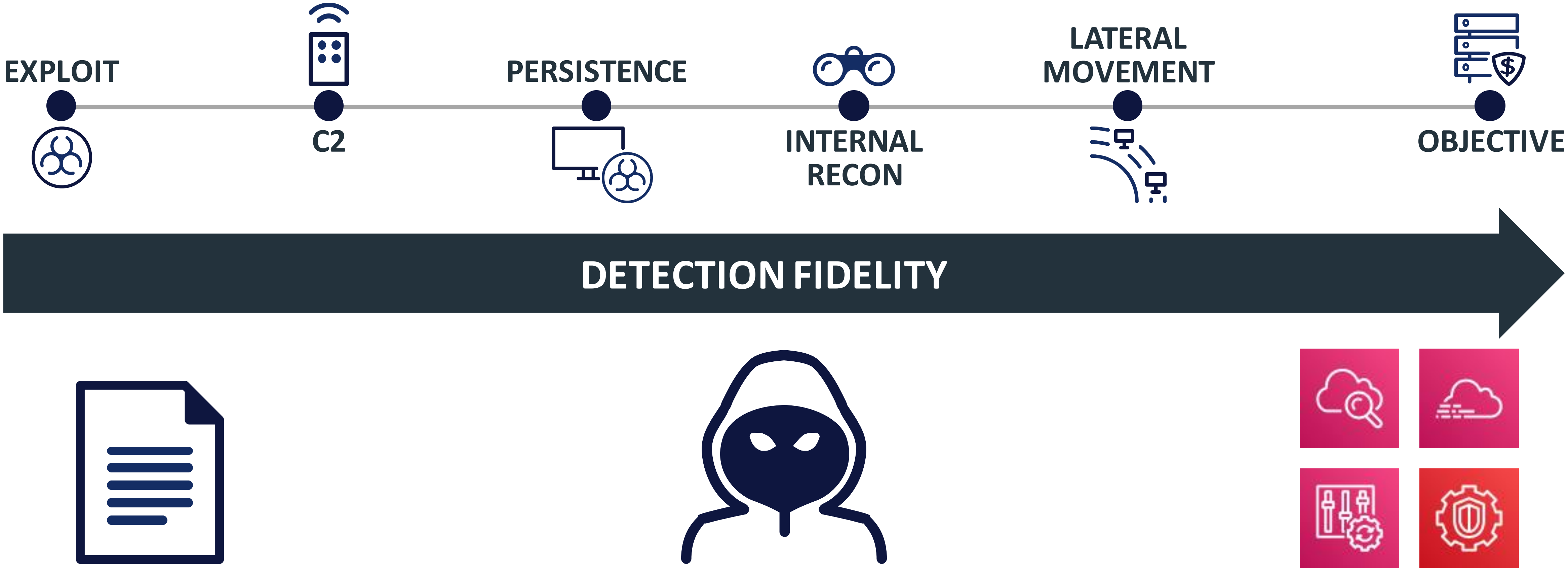
02 Prioritise attack paths and actions

03 Pick the most important attack paths, codify them

04 Verify telemetry is available to defenders

05 Execute attacker actions as attack paths, verify detection cases work as expected.

Where To Start



EXPLOIT

C2

PERSISTENCE

INTERNAL RECON

LATERAL MOVEMENT

OBJECTIVE

DETECTION FIDELITY

Detection Development Process

A IMPLEMENT DETECTIONS

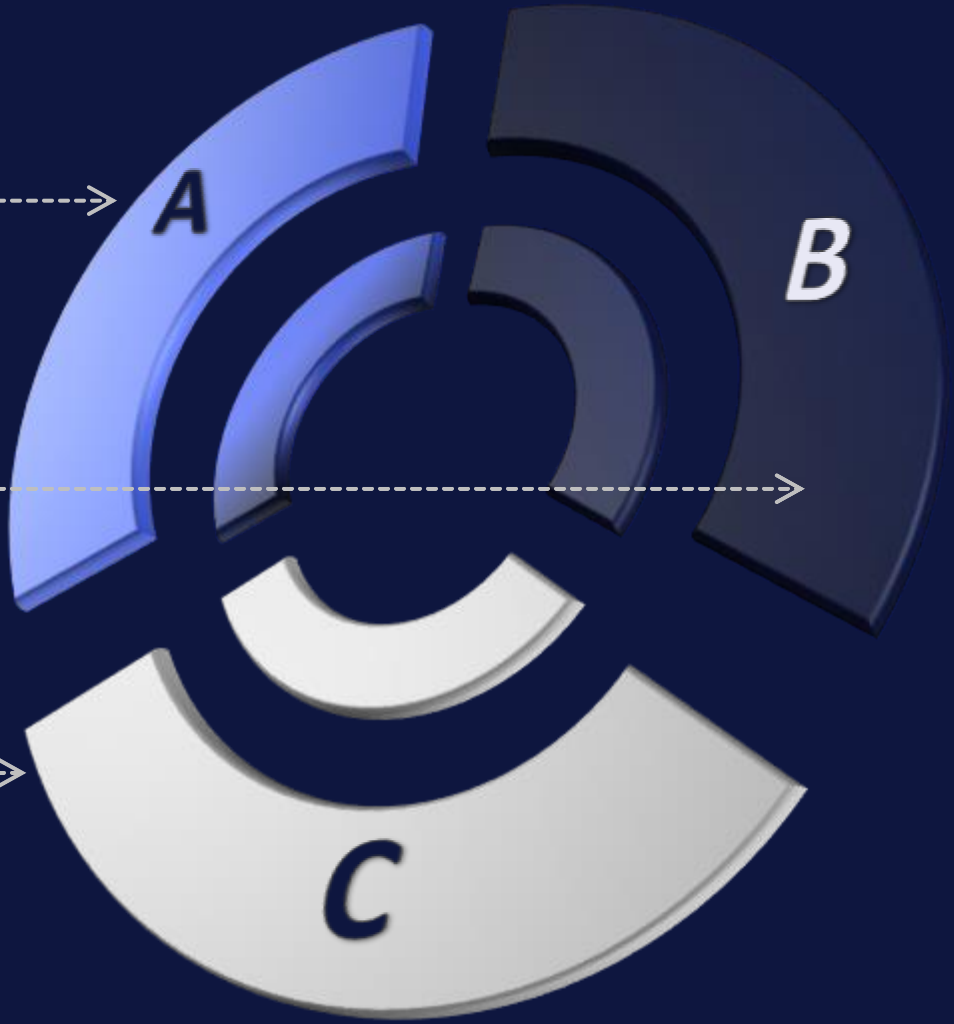
Develop a set of use cases for the app, given the threat model

B SIMULATE ATTACKS

Execute TTPs from the threat model against the application

C EVALUATE RESULTS

Confirm detections behaved as expected, confirm necessary improvements or next detections to implement



Detection is a Journey

Track your core assets, review and evolve detections against them over time

Cloud environments change, your detection will too

Codify use cases (and attacks) to aid knowledge sharing

Conclusions

Conclusions

Two biggest causes of cloud breaches:

- Exposed resources
- Mismanaged credentials

Get the basics right, then connecting systems – SSO, CI/CD, Dev Tools

Cloud Attack Detection is a mindset shift, requires new approaches

WITH®

secure