Lessons Learned in API Protection

Les Correia

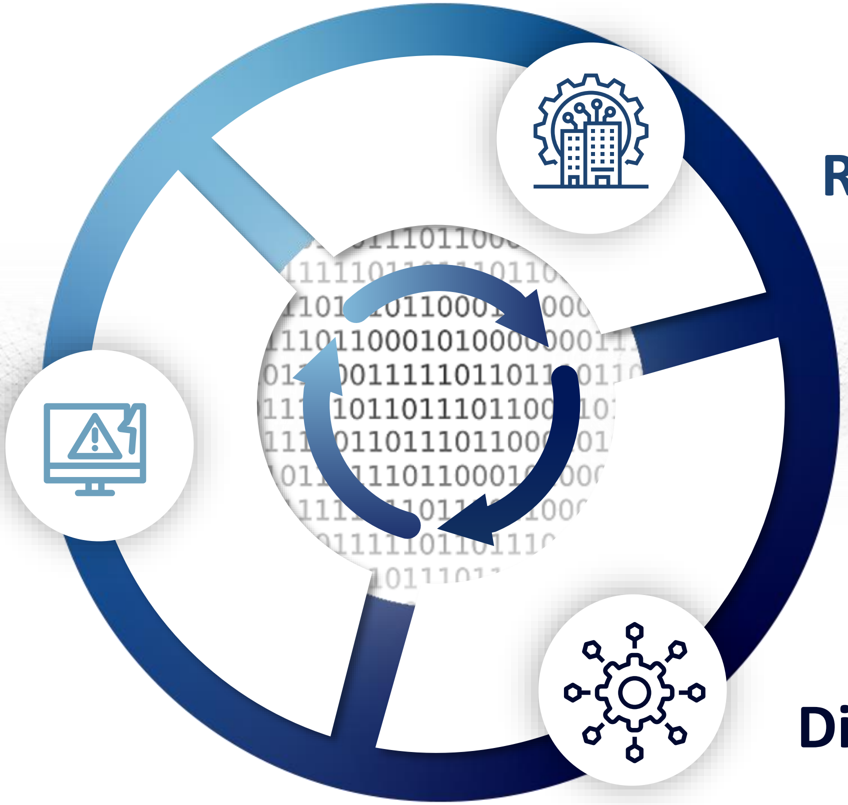
Founder, Secure Karma

Agenda

- Why API Security Is On Every CISO’s Mind
- Typical Web/API Security Challenges
- Original Goals
- An Approach to Building a Robust API Security Program
- Some Findings
- Q&A

Why API Security Is On Every CISO's Mind

- Everything is Code -

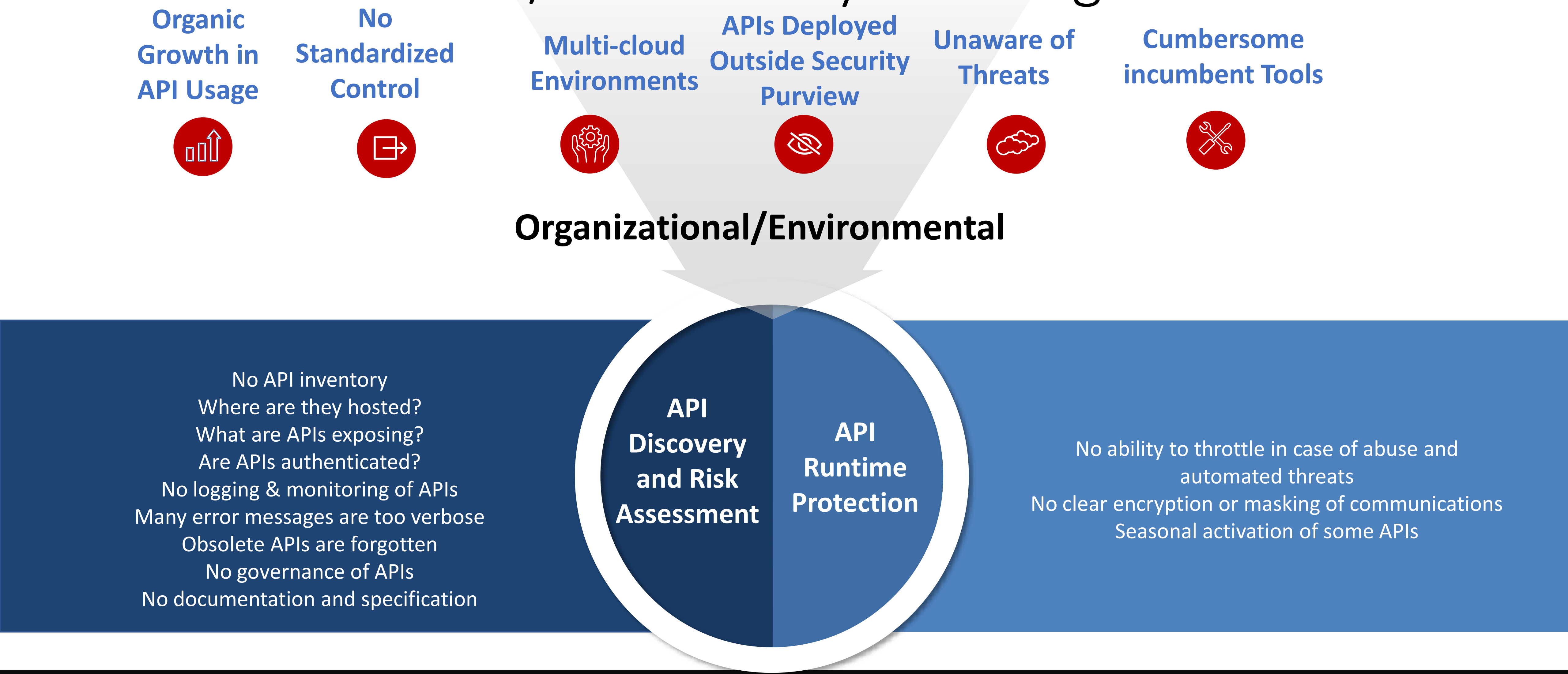


API Proliferation Resulting from Digital Transformation

Increasing Breaches Involving APIs

Applications Shift to Distributed Microservices Architecture

Web/API Security Challenges



Organic Growth in API Usage

No Standardized Control

Multi-cloud Environments

APIs Deployed Outside Security Purview

Unaware of Threats

Cumbersome incumbent Tools

Organizational/Environmental

API Discovery and Risk Assessment

No API inventory
Where are they hosted?
What are APIs exposing?
Are APIs authenticated?
No logging & monitoring of APIs
Many error messages are too verbose
Obsolete APIs are forgotten
No governance of APIs
No documentation and specification

API Runtime Protection

No ability to throttle in case of abuse and automated threats
No clear encryption or masking of communications
Seasonal activation of some APIs

Original Goals

1

Identify and document API bill-of-materials

Create an accurate, living inventory.

2

Identify risks and vulnerabilities

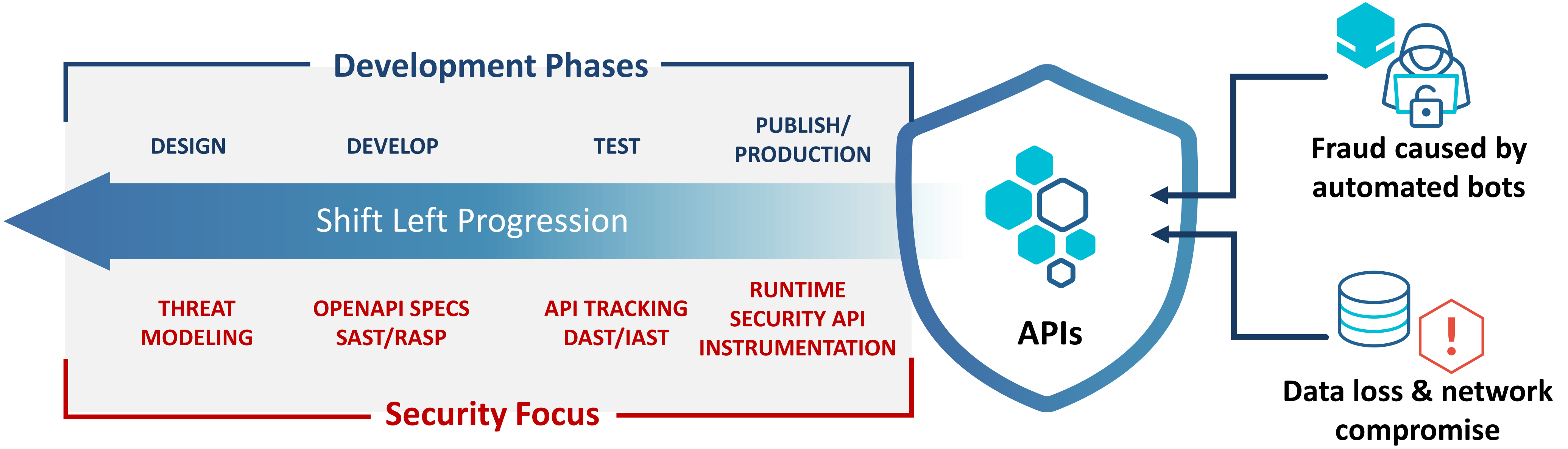
Assess our API risks using OWASP API top 10 as a benchmark.

3

Method to monitor and remediate seamlessly

Attack detection and mitigation responses.

Approach - Shield Right and Then Shift Left: Building a Robust API Security Program



Shift Left

- Uncover vulnerabilities before they go-live
- Elevated security focus throughout the development cycle
- Improves security overall

Shield Right

- Catalog APIs – external and internal
- Risk Assessment
- Protect APIs from cyber attacks

Findings:

API Discovery:

Focused heavily on finding the unknown

Challenges

- How many locations do we have?
- How many shadows and approved APIs?
- How many inactive/deprecated APIs do we have?

What We Discovered

- Shadow cloud usage and APIs
- Internal APIs accidentally exposed publicly
- No formal, automated process
- Many possible locations, widely distributed development teams
- A high number of inactive APIs
- Inconsistent coding
- Poor use of authentication
- Sensitive data exposure

Findings:

Risk & Threat Detection and Prevention

Challenges

- Low efficacy detection
- APIs simplify scraping, account takeover, and enumeration attacks
- Attacks appear legitimate, fall outside of OWASP top 10 lists
- Inconsistent Prevention:
  - Unable to stop what was not identified
  - Blocking based on known signatures

How We Addressed It

- Extend beyond OWASP lists
- Baseline normal behavior, use for detection AND prevention
- Understand attack origins – country and infrastructure
- Automate policy creation and response

Findings:

Other Critical Considerations

Challenges

- Develop guidance, policies, standards
- Improve secure design process
- Improve development awareness
- Select tooling that can assist and complement our incumbent set

How We Addressed It

- Developed guidelines, policies, and standards
- Awareness training is work-in-progress due to conflicting priorities and maturity
- We needed to get tools that meet our:
  - Business drivers – costs, references, replacement/consolidation, etc.
  - Operational drivers - are flexible, non-intrusive, create API specs for development feedback, API detection at scale (CI/CD and in production), bot and fraud detection, integrate seamlessly in our environment, and intuitive contextual reporting, ease of use, centralized dashboard, etc.
  - Security drivers – protection at scale for APIs, bot and fraud, compliance/audit support, contextual risk categorization, threat intelligence support, etc.

Questions?

linkedin.com/in/les-correia
https://securekarma.io/