
cisco Live!
TURN
IT
UP
#CiscoLive

The bridge to possible

API service reputation scoring for cloud applications

Cisco ET&I

Peter Bosch, Distinguished Engineer

February 2021

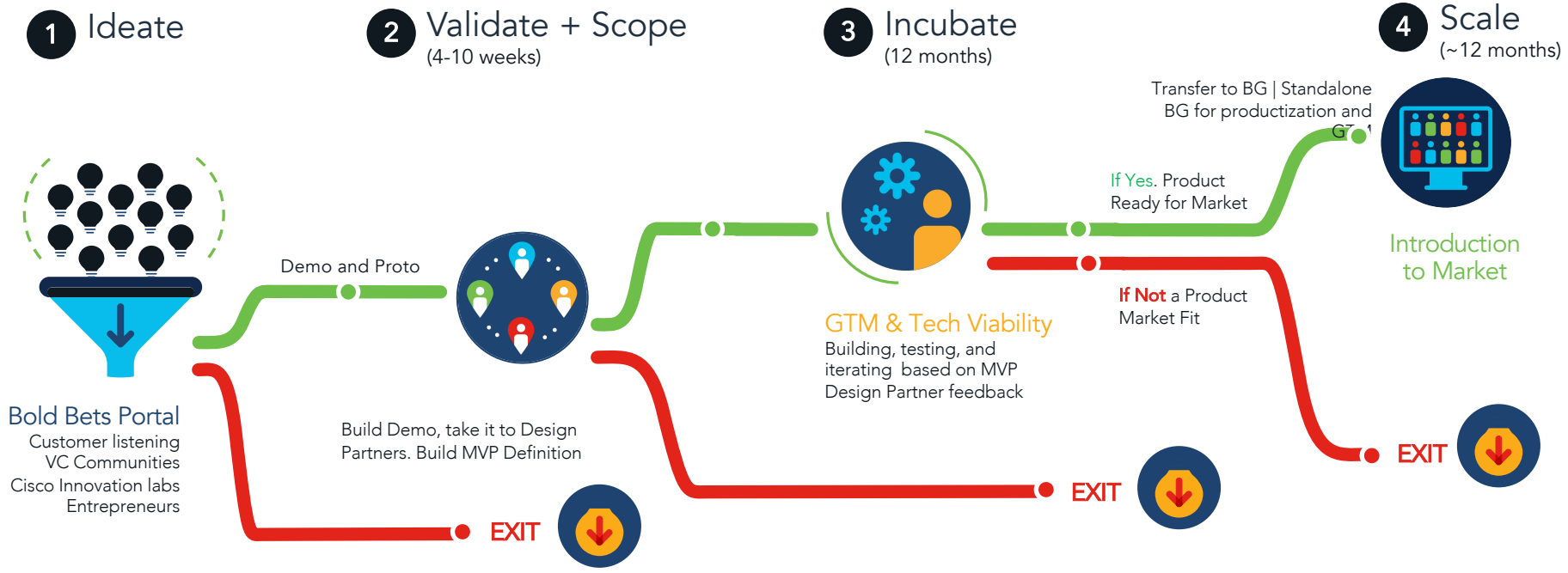
BRKNWT-2003

Welcome

Emerging Technologies & Incubation team

Corporate Strategy & Strategic Alliances team

Ideation Process

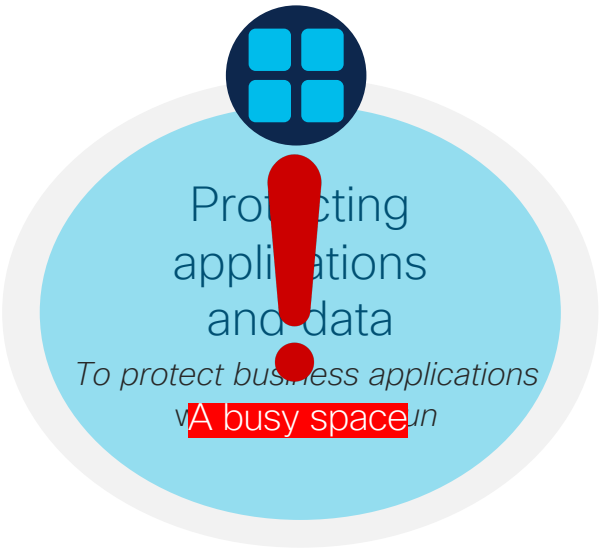


Application Security

Why application security?

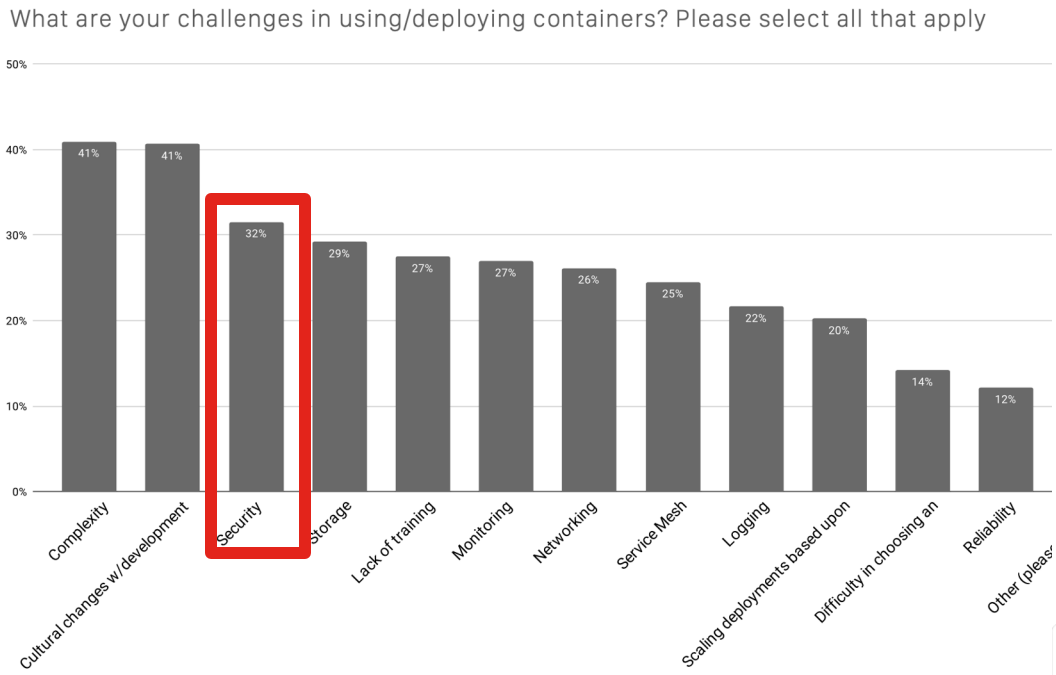
Security solutions are split protecting people and protecting applications and their data

"ACME, Inc."
Cisco customer's **enterprise** focus



"acme.com"
Focus on protecting customer's enterprise **APP**

Challenges with deploying containers



CLOUD NATIVE COMPUTING FOUNDATION

CNCF survey 11/20

Cloud-native application security
Introduction and goals

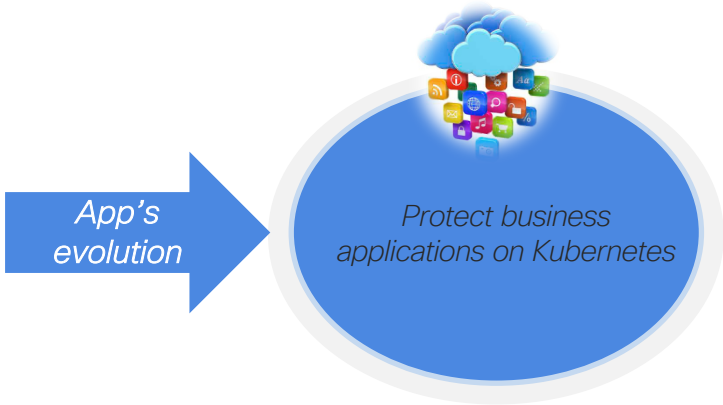
Kubernetes is a technical and a business revolution

A large open-source community enabling startups to enter and grow FAST

Enterprises are re-inventing themselves via cloud-native apps to accelerate TTM for applications

Key personas in such companies include the cloud architect, cloud SecOps/AppSec engineer and DevOps – built around enabling the developer

These are not the traditional stakeholders



"born-in-the-cloud.com"

K8s is a business game changer

What is the problem?

Applications move to the cloud and have different compute and security requirements

DevOps/DevSecOps is taking the lead and the security responsibility has shifted left

Developers lack the security tools to address the knowledge gap. Security tools need to be integrated in developer's tools

First generation cloud application security focusses on deployment/infra, vulnerability analysis, simple attacks

Tools need to be developed that address more sophisticated attacks, and focus on architectural soundness and observability of application operating in the cloud

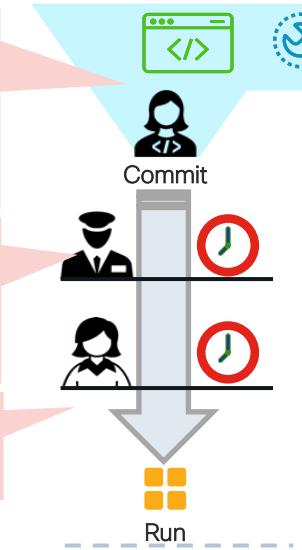
What dilemmas exist in organizations?

From rigid …

I use whatever library, function and API that is most convenient; I'll quickly put together a solution and roll it out; I need help with the security aspects, but really this is all too complicated for me

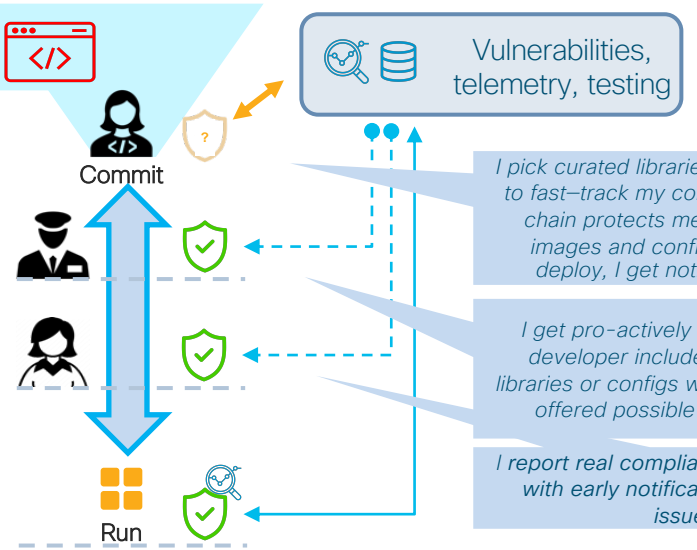
No idea if this app is safe; The app may have compromised libraries, configs, API services. I really should chase this down, but am I willing to delay roll outs?

My team has built a bunch of applications, but I really cannot oversee risk or easily show compliance



- Process takes months ❑
- Manual review gates ❑
- No/poor runtime visibility ❑

… to integrated development



I pick curated libraries, functions, APIs to fast–track my commits; my CI/CD chain protects me against rogue images and configs and when I deploy, I get notified of issues

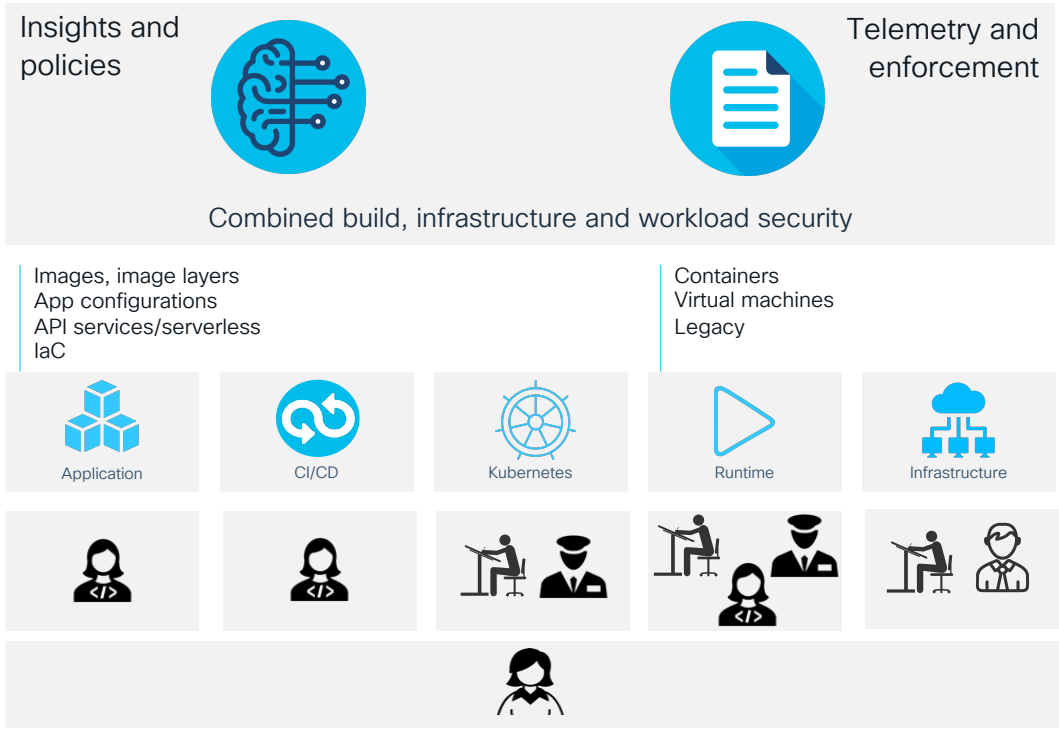
I get pro-actively notified when a developer included APIs, image libraries or configs with issues and am offered possible remediations.

I report real compliance to customers with early notification of security issues

- ✓ Instant – library, function and API feedback
- ✓ Minutes – automatic reviews
- ✓ Continuous – runtime enforcement

Combining image, metadata, function and API scoring into the development and run-time cycle – an insurance policy for DevSecOps

Personas and stack



CISO
SecOps
IT
Cloud architect
Developer, DevOps

Cloud applications security challenges
Cloud native computing

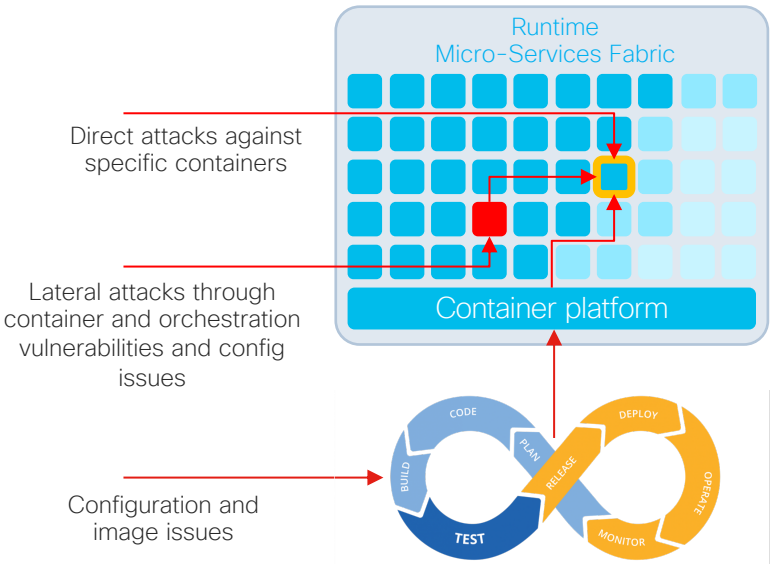
Cloud-native deployments more than ever rely on container deployments. Image vulnerabilities and security misconfigurations are the main security challenges for containers

Orchestration configuration, permission handling and cluster activities are critical building blocks for secure deployment of containers

Early detection of misconfigurations and vulnerabilities in CI/CD automated pipelines is key for early detection and security mitigation for cloud-native applications

Container networking requires dedicated policies to govern internal networking and external networking; properly secured communication requires an identity and key infrastructure

Container security
Containers native security approach



Container deployments create unique security challenges not addressed by existing security tools

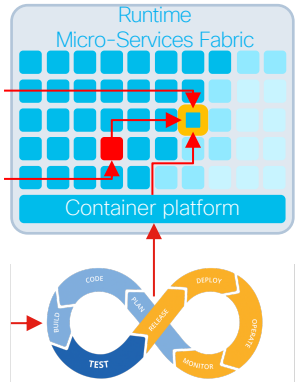
Security vulnerabilities can originate from cloud service providers, orchestration stack and the networking topology

Decomposed applications interact frequently with each other and rely extensively on internal and external API services

Cloud deployments are typically run on the Internet – exposed to all

Relying on 3rd party APIs may expose the applications to further attacks through their vulnerabilities (CVE violations, poor implementations, …)

Containers Security
Containers native security approach



(Docker) image, image layer, packages and their dependencies vulnerabilities are the main security challenges for container deployments

(Kubernetes) configurations and permissions (RBAC) and cluster activities (API calls) are critical for secure deployments of containers

Multi-pronged security enforcement approach:

- Early detection and quick mitigations of vulnerabilities in the CI/CD chain
- Uniform container-networking for internal- and external API services and common security policy enforcement
- Automatic traffic encryption for all interactions, even for defunct software
- Common public key infrastructure for all computing assets in the application

Security policy enforcement in CI/CD, deployment and during connection establishment

Cloud applications security challenges
APIs

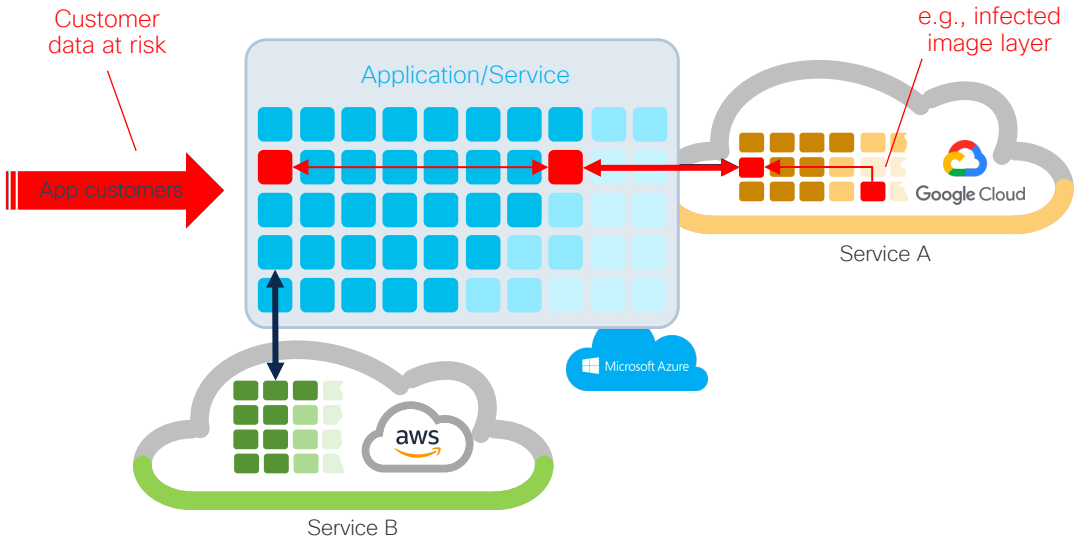
Contemporary applications rely on a loosely-coupled architecture, with well-defined APIs between internal *and* external API services

Use of internal and external APIs requires API security tools to inspect and validate the security of these API preventing malicious manipulations

External APIs require special attention due to the lack of control and limited visibility into CVE violations, poorly implemented interfaces, missing parameters and insecure tokens

Observe and analyze internal and external APIs, test those where possible, and enforce connection policies on such observations

What is the problem API scoring addresses?

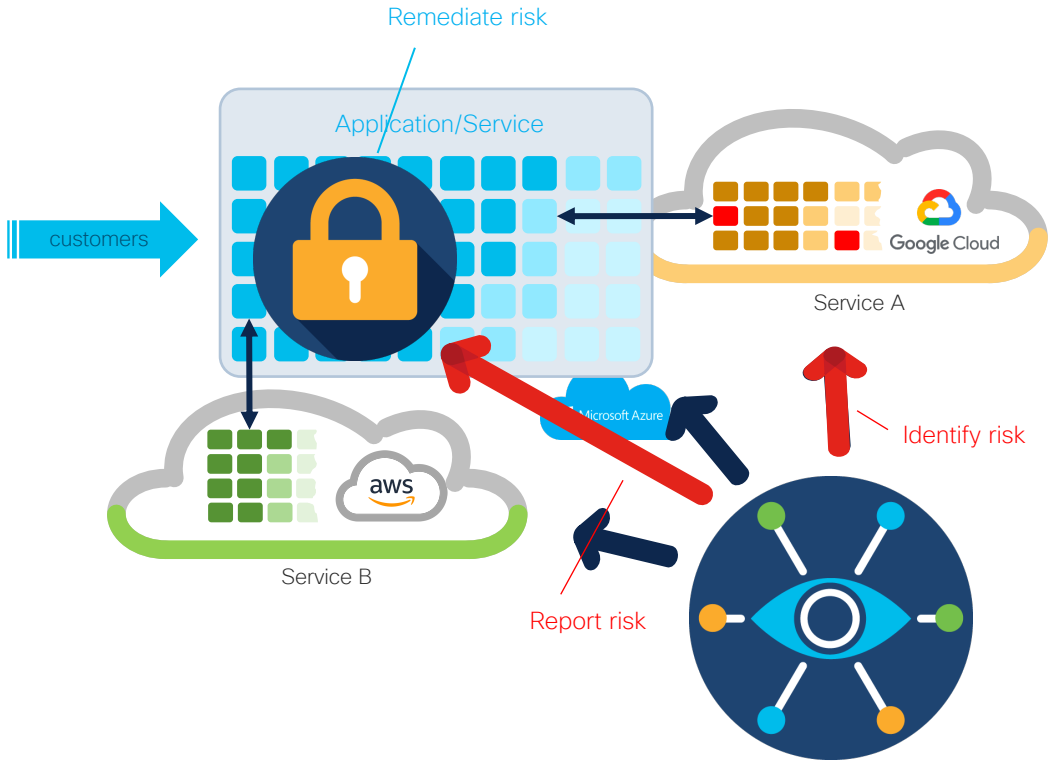


Applications are increasingly assembled with services

The quality and security of remote services is oftentimes unknown, esp. not in the CI/CD cycle itself

Is customer data at risk when interacting with “the application”?

How is API scoring addressing the problems?



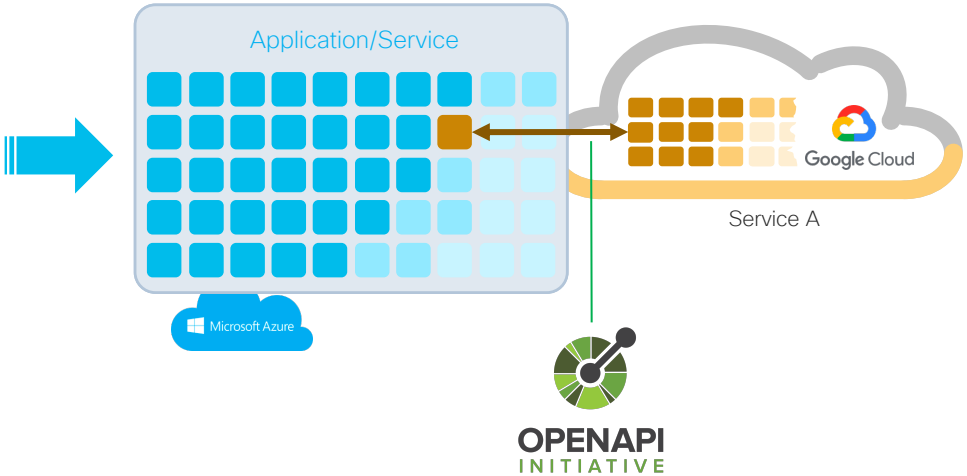
Observe services for issues, CVE violations, poorly implemented interfaces, missing parameters

Report issues before these become an issue and act upon it in the application

Block, quarantine and/or amend calls based on URL of APIs, CRUD and data policies

api.service.com/api/v1/voice vs api.service.com/api/v1/sms

What is the problem API security addresses?



Semantic-level problems, the server not implementing what the specification states

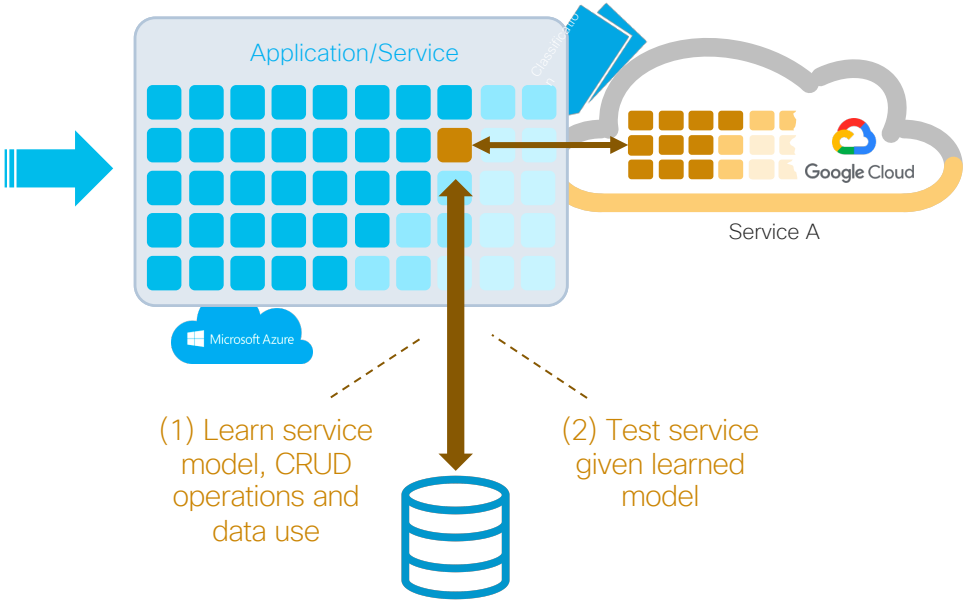
Servers that are susceptible to parameter fuzzing

Poorly implemented servers leading to potential data leakage

Services relying on client-side functions for data filtering

Exposing PII data for poorly rates services

How is API security addressing the issues?



Chaos engineering applied to API services

Learn how a service is really used – if 99.9% of the requests are similar, it is probably the right model

Craft positive and negative tests against a remote service to learn of its quality

Use during staging and optionally during run-time

Use crowd-sourcing for a full picture of the service

Auto-classify parameters and enforce security policies given classifications

Cloud applications security challenges
Functions and serverless containers

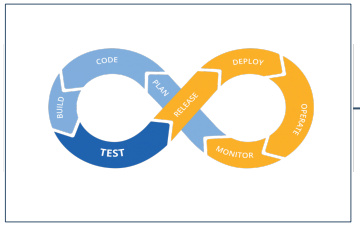
Serverless functions augment application development and require a dedicated security solution distinct from existing security solutions

Observability is a security challenge when deploying functions; serverless functions are based on a new programming paradigm (pub/sub versus RPC)

Integrating security checks into CI/CD automates a significant portion of security checks and discover faults that require a fix prior to their deployment

Provide for observational functions and enforcement functions in the serverless functions; integrate into the cloud-native pipeline

Serverless security function angles
Serverless containers



Scanning of image issues

Enforcement of security policies

Treat images akin to “standard” container deployments with vulnerability scanning (CVEs, code execution and injection issues)

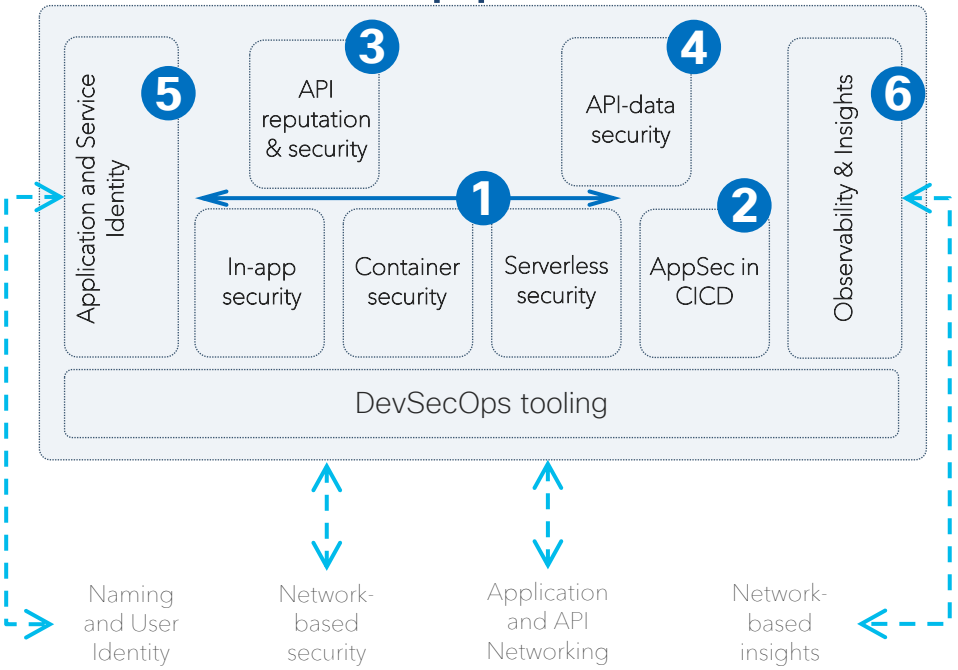
Cover OWASP/CSA top issues

Scan images and configurations before these are deployed (secrets, access keys, passwords) and use least-privileges security approach

Extend App Mesh with agent to enforce security policies

Rich set of security automation tests in CI (vulnerabilities) and CD (keys, passwords, configurations) with dedicated plugins/CLI options

Application and API Security



1. Security for various infrastructure ways of building an app – using traditional monolithic methods to cloud native methods
2. Integration of security insights into the build and deploy pipelines
3. API-layer scoring, reputation and security, across SaaS, public cloud and internally consumable API surfaces
4. Elements of data security related to API-API and API-Data accesses, e.g.:
   - Tokenization
   - Dependency graph visualization
5. Identity for Applications and Services and tie into User Identity and policies
6. Security insights at the App and API layer
7. DevSecOps tooling (CI/CS/CD)

The bridge to possible

Thank you


cisco Live!
TURN
IT
UP
#CiscoLive