Benefits of a Positive Security Model for APIs

+

The Speakers

Colin Domoney

Developer Advocate at 42Crunch

Michael Farnum

CTO at Set Solutions

Set Solutions + 42crunch

Main Points

AppSec Tooling and the Negative Security Model

Issues between Security and Development

Frustration of Developers

Benefits of a Positive Security Model in APIs

Creating empathy for the Developer

Reducing Developer fatigue

Set Solutions + 42 crunch

Developer Frustration and Fatigue

Set Solutions + 42 crunch

API Security vs Web Security

OWASP API Security Top 10	OWASP Top 10
API1:2019 Broken Object Level Authorization	A01:2021-Broken Access Control
API2:2019 Broken User Authentication	A02:2021-Cryptographic Failures
API3:2019 Excessive Data Exposure	A03:2021-Injection
API4:2019 Lack of Resources & Rate Limiting	A04:2021-Insecure Design
API5:2019 Broken Function Level Authorization	A05:2021-Security Misconfiguration
API6:2019 Mass Assignment	A06:2021-Vulnerable and Outdated Components
API7:2019 Security Misconfiguration	A07:2021-Identification and Authentication Failures
API8:2019 Injection	A08:2021-Software and Data Integrity Failures
API9:2019 Improper Assets Management	A09:2021-Security Logging and Monitoring Failures
API10:2019 Insufficient Logging & Monitoring	A10:2021-Server-Side Request Forgery

Set Solutions + 42crunch

Traditional Application Security Tooling

- **SAST** — wasn't designed for API-centric applications. Complex data flow paths or unsupported frameworks reduce the accuracy of a SAST analysis since the model may be incomplete or inaccurate.
- **DAST** — lacks context of APIs. DAST tools can't provide an intelligent assessment of API security.
- **SCA** – useful but not sufficient
- **WAF –** Negative model can miss API-specific attacks

https://thenewstack.io/application-security-tools-are-not-up-to-the-job-of-api-security/

Set Solutions + 42crunch

Positive Model vs Negative Model

Allowlist

- Allowed data types strong defined and enforce in OAS mode
- Data format can be precisely defined
- Operations can be fully specified too

Only allow data conforming to specification — anything else is an error

Only allows “known good”

vs

Blocklist

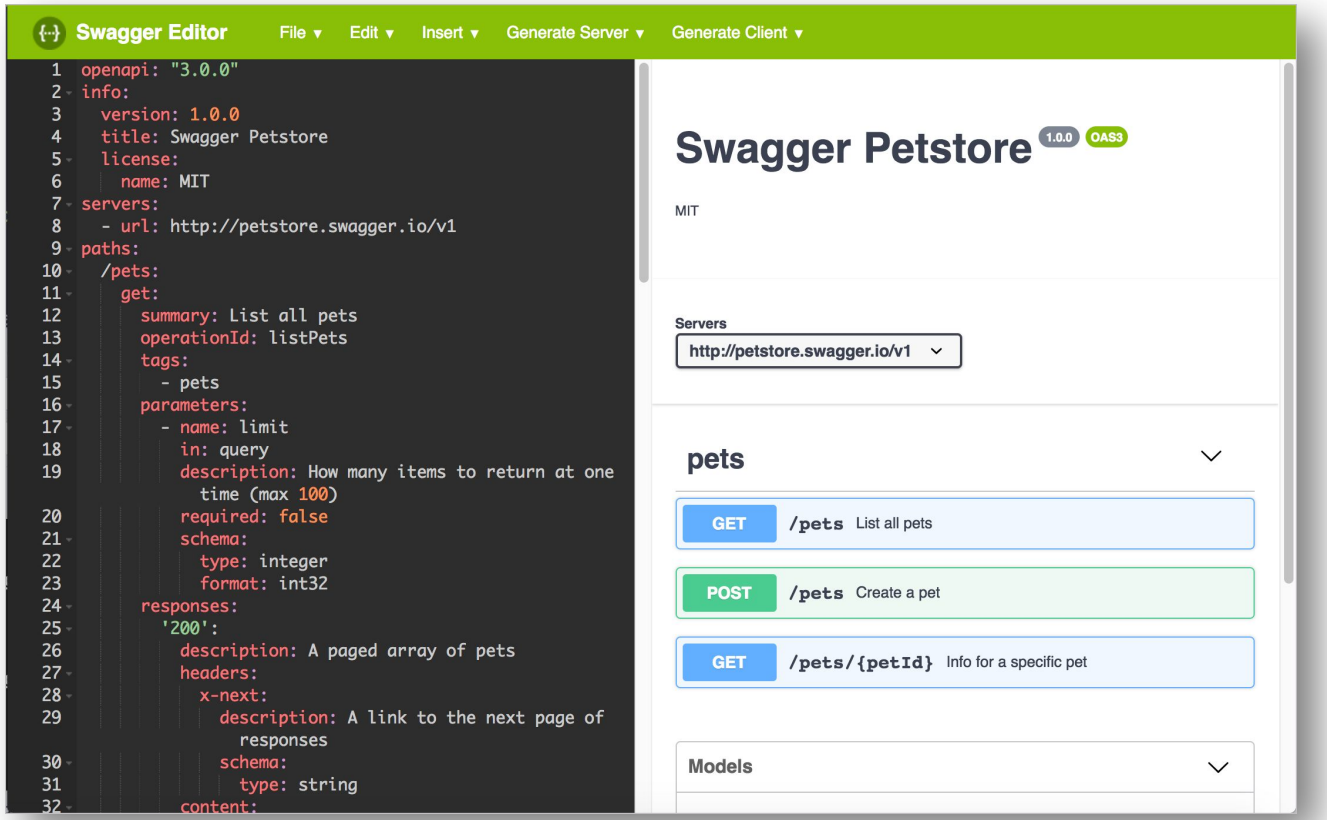
- Attempts to interpret data based on the runtime context i.e., Javascript, HTML
- Attempt to block what shouldn’t be present in a given context
- Can easily be subverted with encoding, etc.

Attempts to block “known bad”

Set Solutions + 42crunch

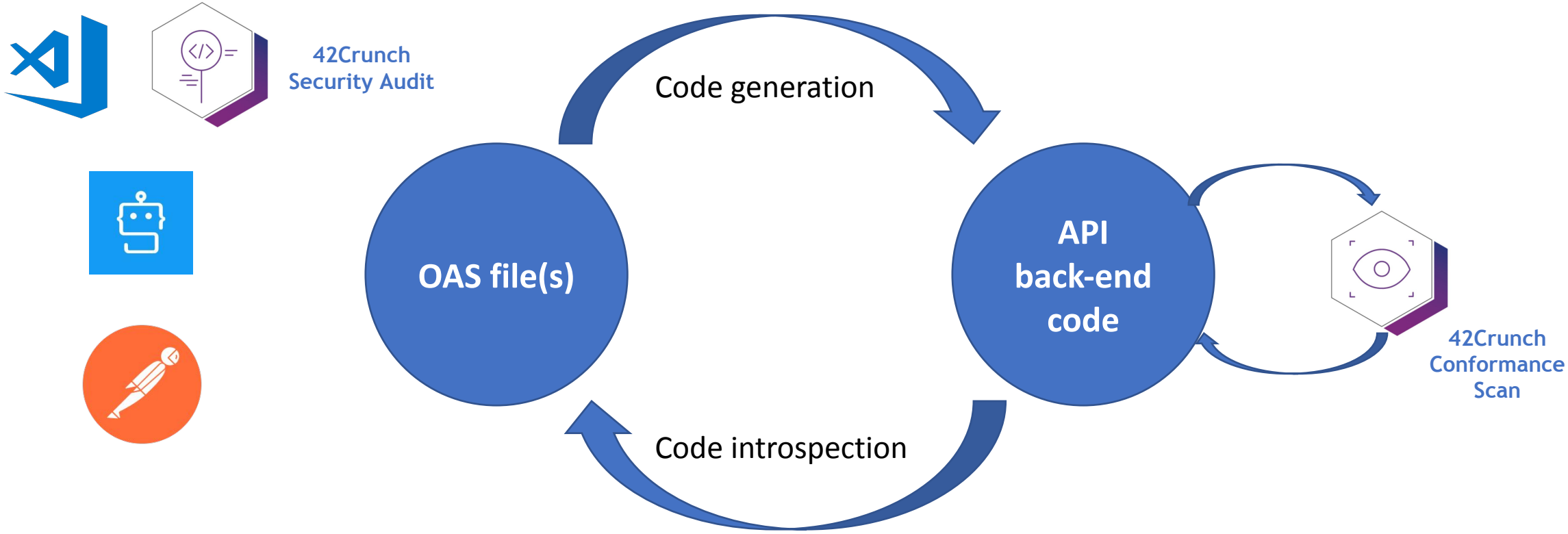
API Definitions/Specifications

- OAS forms a definitive contract for all downstream development
- OAS allows for a precise definition of request and response data types
- OAS allows operations to be tightly specified
- OAS allows security primitives to be specified
- Extensions allow for additional primitives to be included



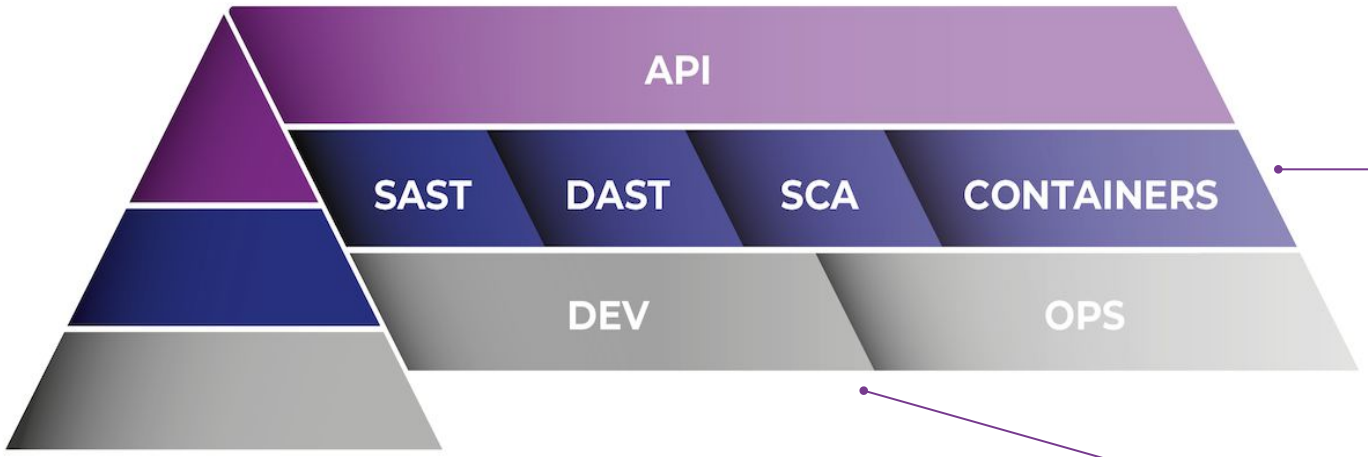
Set Solutions + 42crunch

Using OAS to “Shift Left”



Set Solutions + 42crunch

A Layered Approach to API Security



- Dedicated API security tooling is essential as the final layer of defenses allowing for:
  - Validation of OAS specifications
  - Verification of API implementation
  - Runtime protection of APIs
  - API discovery and inventory
  - Integration into IDEs and CI/CD

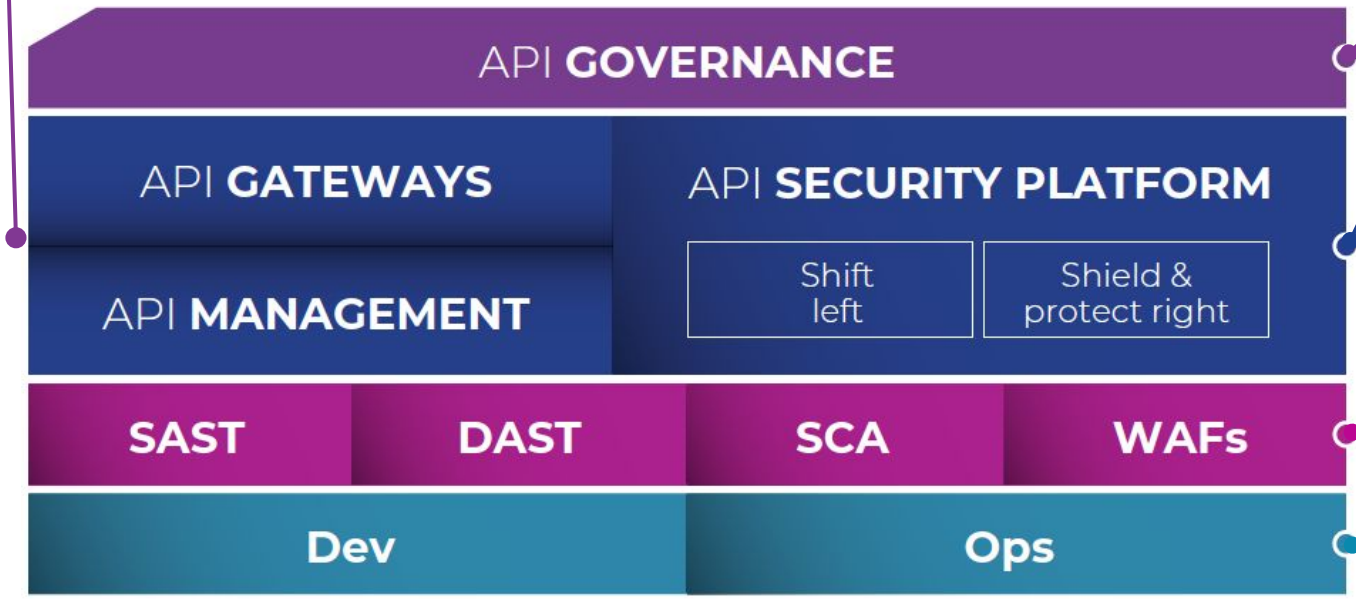
- A well established AppSec process is vital to ensure that basic coding and implementation errors are detected in the build process
- SAST, DAST, and SCA are the tools of the trade for detecting OWASP Top 10 type issues.
- Unfortunately, such tools are not specifically tailored for API development and may lead to gaps in coverage.

- DevOps emphasizes collaboration between Development and Operations teams to ensure high-velocity delivery of quality applications in a repeatable, automated manner.
- Without DevOps, API teams may struggle to produce quality APIs in a repeatable manner which may impact API security.

Set Solutions + 42crunch

A Layered Approach to API Security

API management portals and gateways are essential to the operation of APIs at scale, and **security features should be leveraged to provide another layer of protection**.



A dedicated API Security platform is essential as the ultimate layer of defense allowing for:

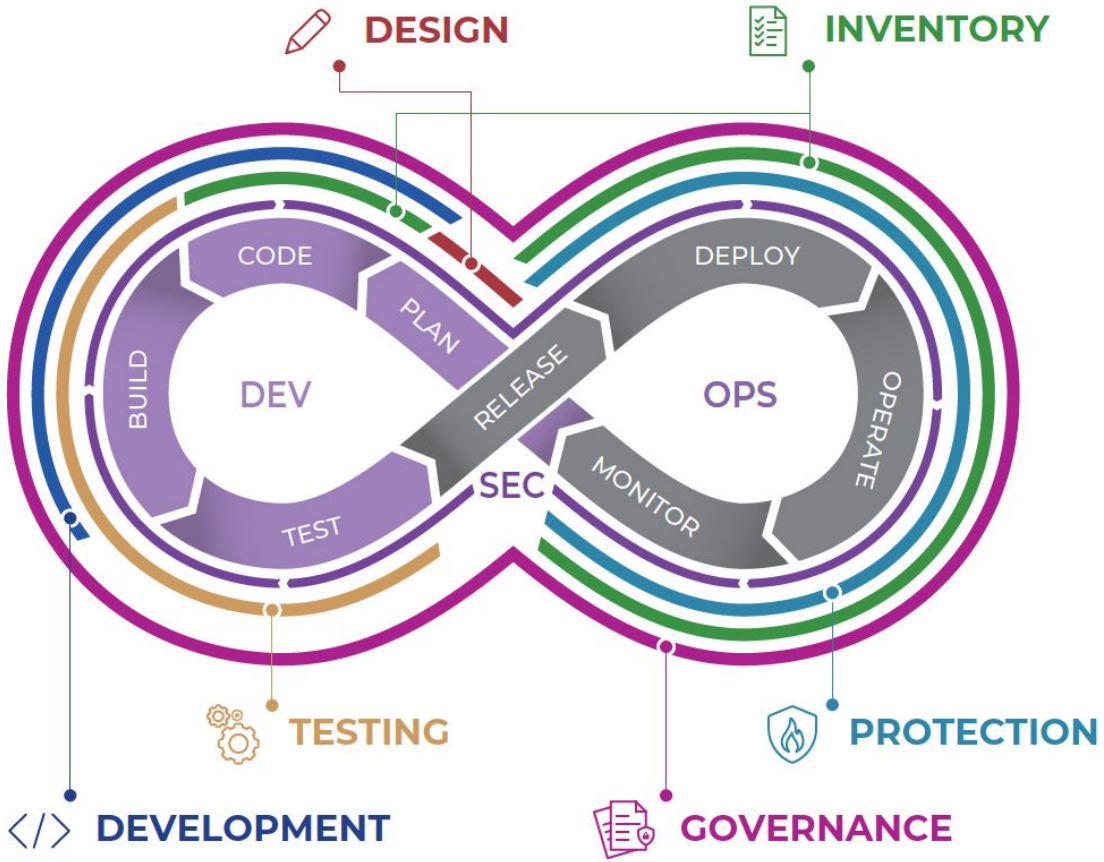
- **Validation** of OAS specifications
- **Verification** of API implementation
- **Runtime protection** of APIs
- **API discovery** and inventory
- **Integration** into IDEs and CI/CD

A well established AppSec process is vital to ensure that basic coding and implementation errors are detected in the build process. Traditional security tools (WAF, SAST, DAST) are not specifically tailored for API development and are likely to lead to **gaps in coverage**.

DevOps emphasizes collaboration between Development and Operations teams to ensure **high-velocity delivery of quality applications in a repeatable, automated manner.**

Set Solutions + 42crunch

Six Domains of API Security



API INVENTORY

Do you understand what APIs you own? Do you track shadow and zombie APIs?

API TESTING

Are you doing automated API testing? Are you considering security in your test strategy?

API DESIGN

Are you doing API-design-first? Do you incorporate security into the design phase?

API PROTECTION

Are you using API protection technology (WAFs, WAAPs, API gateways) in your deployments?

API DEVELOPMENT

Are your developers trained to code securely? Do they understand API security threats and risks?

API GOVERNANCE

Do you control and actively monitor your API estate and environments?

Set Solutions + 42crunch

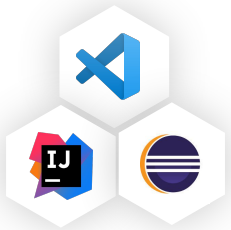
Shift-Left, Shield-Right

SHIFT-LEFT
WITH SECURITY AS CODE

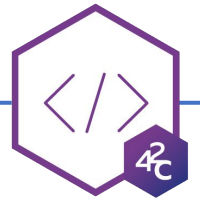
SHIELD-RIGHT
RUNTIME PROTECTION

01 Design

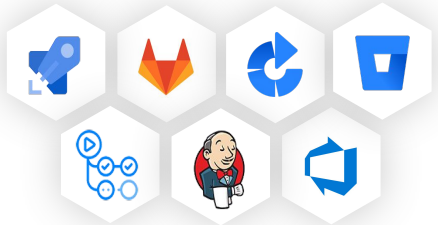
Code in security at design time



02 Develop



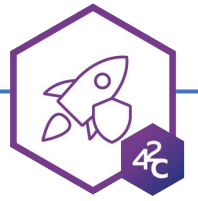
Document & audit API contract



03 Integrate & Test

API vulnerability scanning in CI/CD pipeline

04 Deploy & Protect



Security policy enforced at runtime

Set Solutions + 42 crunch

The Developer-First Approach

SECURITY MANAGEMENT & GOVERNANCE

Visibility & control of security policy enforcement throughout API lifecycle for security teams.

API AUDIT

Lock down your API's definitions to reduce the attack surface and remove potential security gaps.

API SCAN

Dynamic runtime testing of your API to ensure compliance with API Contracts.

API PROTECT

Protect each API with an API micro-firewall to distinguish legitimate traffic from malicious API attacks.

INTEGRATED ACROSS API LIFECYCLE

Continuous security enforcement across IDE, CI/CD and at runtime.

BENEFITS: AUTOMATION - COMPLIANCE - COST SAVINGS - TIME TO MARKET

Set Solutions + 42crunch

Questions?

What about restrictions that are not expressed in the spec?
e.g. account api states how to return account information but not that accounts should only be returned if the end user is the owner of that account?

So what does Set Solutions do, and how might they enhance the 42c experience?

re: BOLA & 42c - what sort of authz works best with 42c and JWT? resource/policy/role/claims?

Set Solutions + crunch

APIsecurity.io

+

https://www.youtube.com/setsolutions

Thank you