AIRLOCK®

# OWASP API Security Top 10

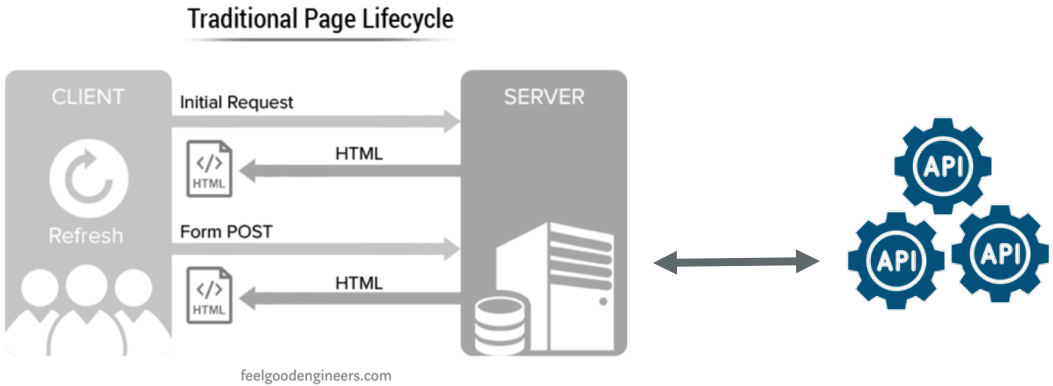
OWASP Switzerland Meeting, 7.10.2020

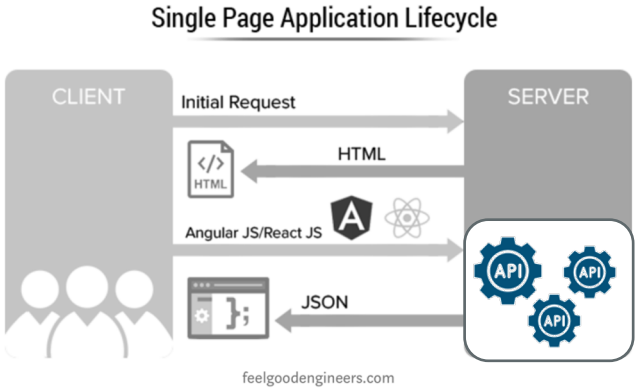
Dr. Martin Burkhart
Head of Product Management Airlock
@ Ergon Informatik AG

https://www.airlock.com

# The Rise of Web APIs

– Multi-page apps
– Application logic on servers
– Server generates HTML
– Full page reloads
– APIs hidden behind server
– SOAP used a lot




– Single-page apps (SPA)
– UI logic in browser (JavaScript)
– Partial updates of page
– **Web APIs accessible from browser!**
– Mostly JSON/REST
– Third-party API providers

By 2021,
**65% of new applications** […]
**communicate via APIs**.

Gartner: Critical Capabilities for Full Life Cycle API Management, May 2018

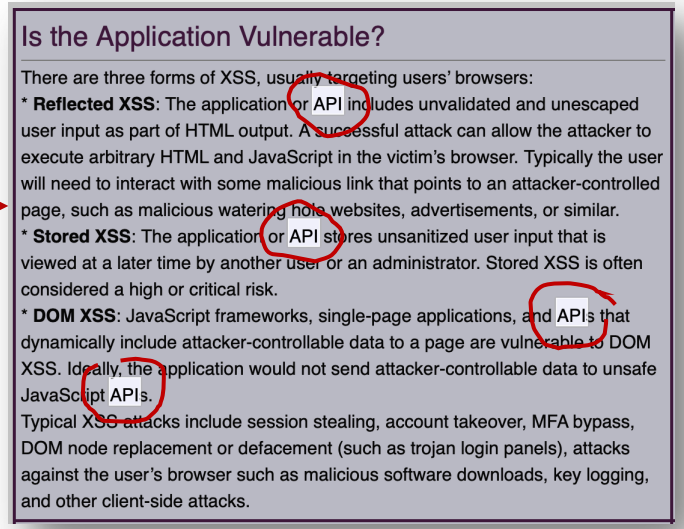
# OWASP Top 10 (Web Applications)

| ID | Name |
|---|---|
| A1 | Injection |
| A2 | Broken Authentication |
| A3 | Sensitive Data Exposure |
| A4 | XML External Entities (XXE) |
| A5 | Broken Access Control |
| A6 | Security Misconfiguration |
| A7 | Cross-Site Scripting (XSS) |
| A8 | Insecure Deserialization |
| A9 | Using Components with Known Vulnerabilities |
| A10 | Insufficient Logging & Monitoring |

– First published: 2003

– Current version: 2017

– See talk by Sven Vetsch, Feb. 2020



Is the Application Vulnerable?

There are three forms of XSS, usually targeting users' browsers:

* **Reflected XSS**: The application or API includes unvalidated and unescaped user input as part of HTML output. A successful attack can allow the attacker to execute arbitrary HTML and JavaScript in the victim's browser. Typically the user will need to interact with some malicious link that points to an attacker-controlled page, such as malicious watering hole websites, advertisements, or similar.
* **Stored XSS**: The application or API stores unsanitized user input that is viewed at a later time by another user or an administrator. Stored XSS is often considered a high or critical risk.
* **DOM XSS**: JavaScript frameworks, single-page applications, and APIs that dynamically include attacker-controllable data to a page are vulnerable to DOM XSS. Ideally, the application would not send attacker-controllable data to unsafe JavaScript APIs.

Typical XSS attacks include session stealing, account takeover, MFA bypass, DOM node replacement or defacement (such as trojan login panels), attacks against the user's browser such as malicious software downloads, key logging, and other client-side attacks.

# OWASP API Security Top 10 (2019)

| ID | Name |
|---|---|
| A1 | Injection |
| A2 | Broken Authentication |
| A3 | Sensitive Data Exposure |
| A4 | XML External Entities (XXE) |
| A5 | Broken Access Control |
| A6 | Security Misconfiguration |
| A7 | Cross-Site Scripting (XSS) |
| A8 | Insecure Deserialization |
| A9 | Using Components with Known Vulnerabilities |
| A10 | Insufficient Logging & Monitoring |

OWASP Top 10 (2017)

| ID | Name |
|---|---|
| A1 | Missing Object Level Access Control |
| A2 | Broken Authentication |
| A3 | Excessive Data Exposure |
| A4 | Lack of Resources & Rate Limiting |
| A5 | Missing Function/Resource Level Access Control |
| A6 | Mass Assignment |
| A7 | Security Misconfiguration |
| A8 | Injection |
| A9 | Improper Assets Management |
| A10 | Insufficient Logging & Monitoring |

OWASP API Security Top 10 (2019)

# Kind of Surprising

A1 → A8

## Injection

- Modern frameworks have better input validation built-in
- SQL injection → NoSQL injection
- But: clients may also be native apps, IOT devices, other APIs
- Don’t rely on client-side validation!

A7 → gone

## Cross-Site Scripting (XSS)

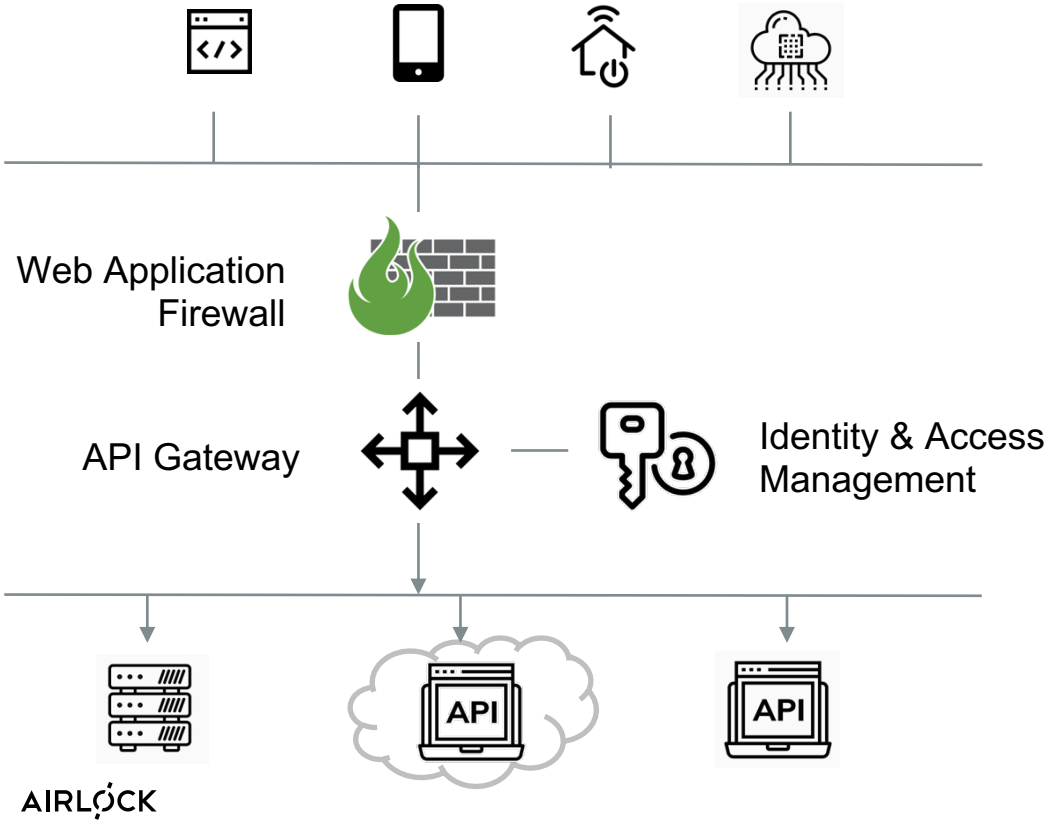
- APIs do not interpret Javascript
- But: Some clients may run in a browser or web view on mobile phone
- Persistent XSS may be an issue!
- Appsec EU 2017: “Bypassing XSS Mitigations Via Script Gadgets”, S. Lekies et al.

A4 → gone

## XML External Entity (XXE)

- SOAP is less common

# Security Infrastructure




## Good at

- Raising bar for successful attacks
- Authentication enforcement (A2)
- Rate Limiting / DOS protection (A4)
- Coarse-grained access authorization (A5)
- Blocking injections and XSS (A8)
- Hiding internal services (A9)
- System-wide logging & monitoring (A10)
- Sorting out suspicious clients
  - Reputation-based (IP blacklists)
  - Behavior-based (anomaly detection, ML)

# Developer


## Good at

– Knowing the business logic

– Knowing ownership of business objects

– Knowing technical stack
(e.g., is there an SQL database?)

## Duties

– Ultimately responsible for code security

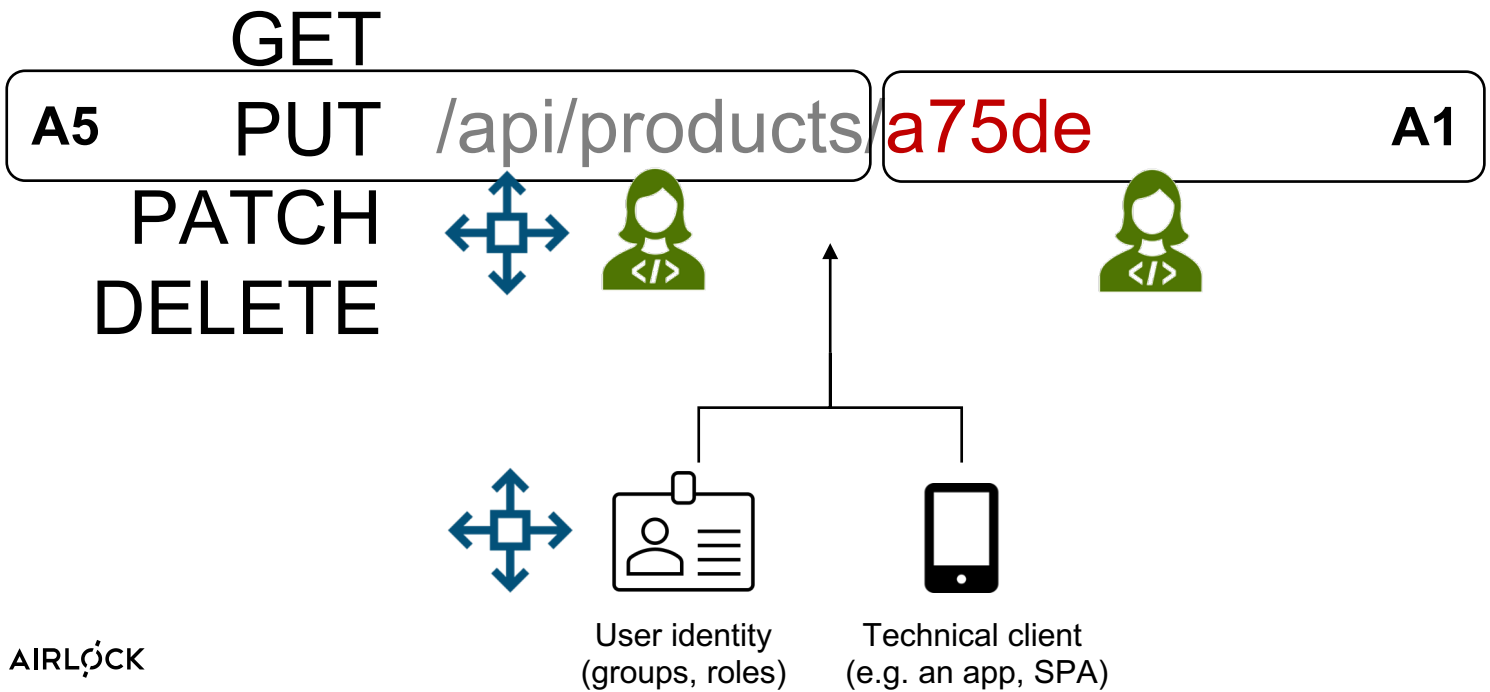
– Security by design

– Continuous security (build pipeline)

– Don’t rely on client!

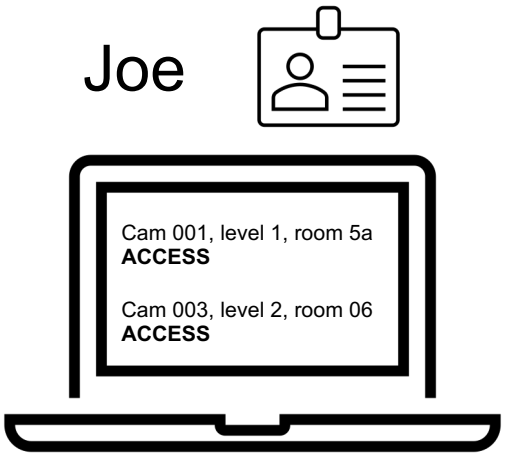
# A1: Missing *Object Level* Access Control

# A5: Missing *Function/Resource Level* Access Control



# A3: Excessive Data Exposure

Joe




Joe's cameras

GET /api/sites/111/cameras

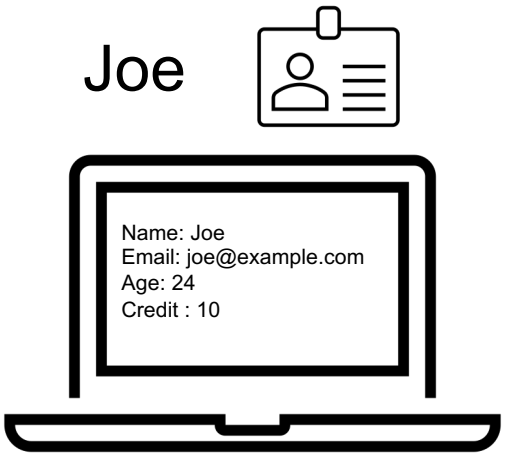
```
{
  "data": [
    { "id":"001", "location" :"01-05a",   "access_token": "..." },
    { "id":"002", "location": "01-floor", "access_token": "..." },
    { "id":"003", "location": "02-06",    "access_token": "..." },
    { "id":"003", "location": "roof-01",  "access_token": "..." }
  ]
}
```

All cameras of site 111

- API implemented in a "generic" way
- Relies on client for data filtering
- "Missing *Attribute Level* Access Control"

# A6: Mass Assignment

Joe

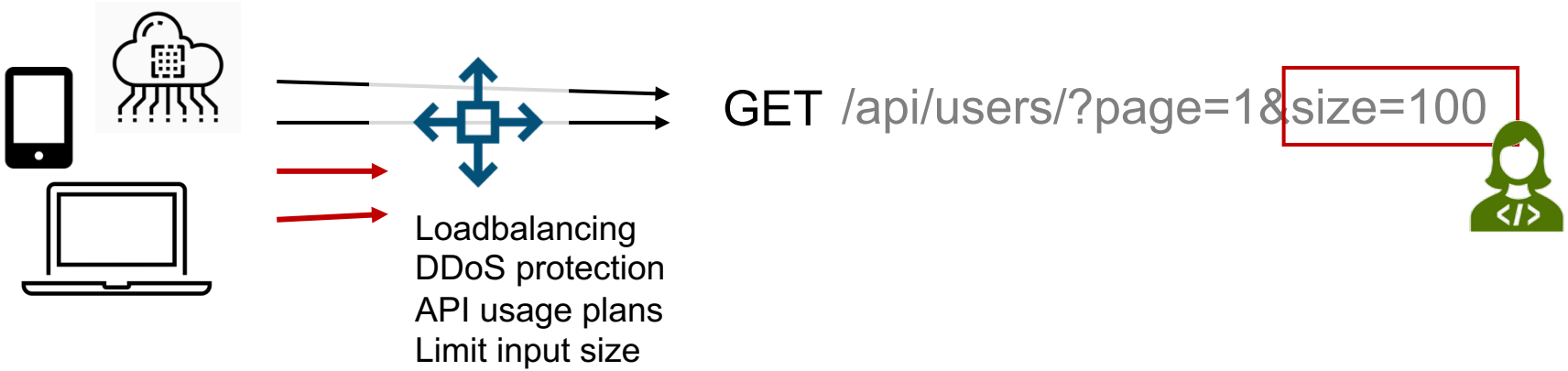



PUT /api/users/me

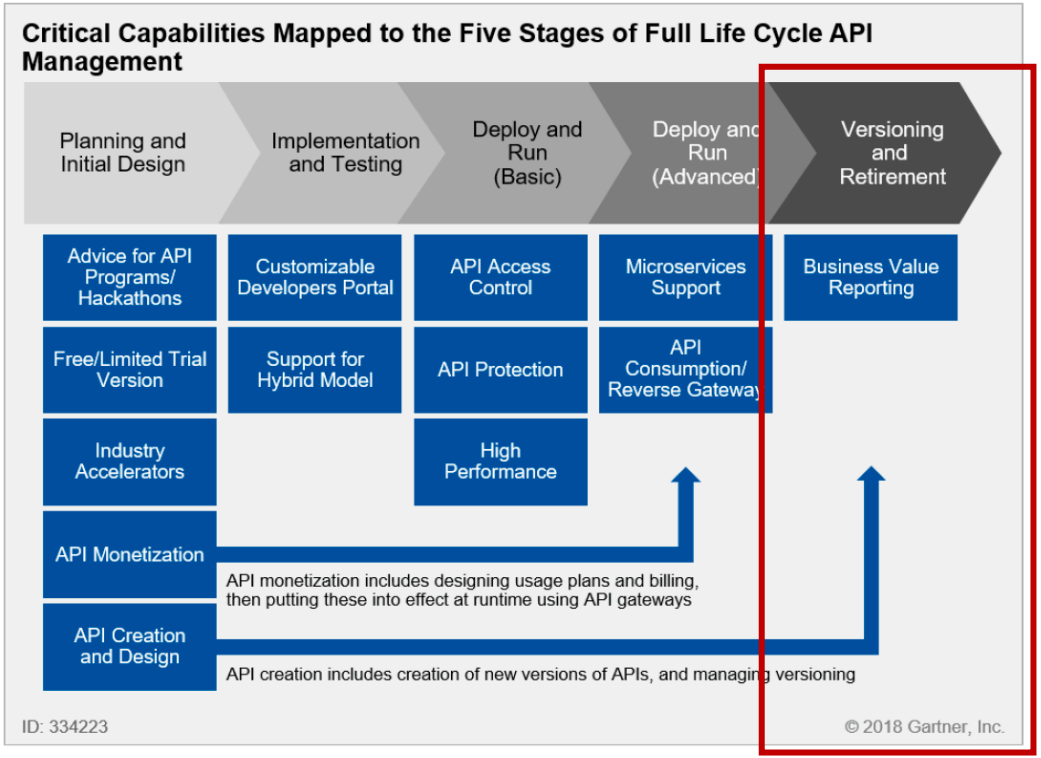
```
{
  “name”: “Joseph”,
  “email”: “joe@example.com”,
  “birthday”: “1996-10-07”,
  “role”: “admin”,
  “credit_balance”:  500
}
```

- Don’t automatically convert client parameters to object attributes
- Check permissions
- Don’t update process-related or internal properties directly

# A4: Lack of Resource and Rate Limiting



# A9: Improper Assets Management




– Lack of service/host inventory
– Old, unpatched API versions still accessible (connected to same DB)
– Internal services exposed to Internet

API Management

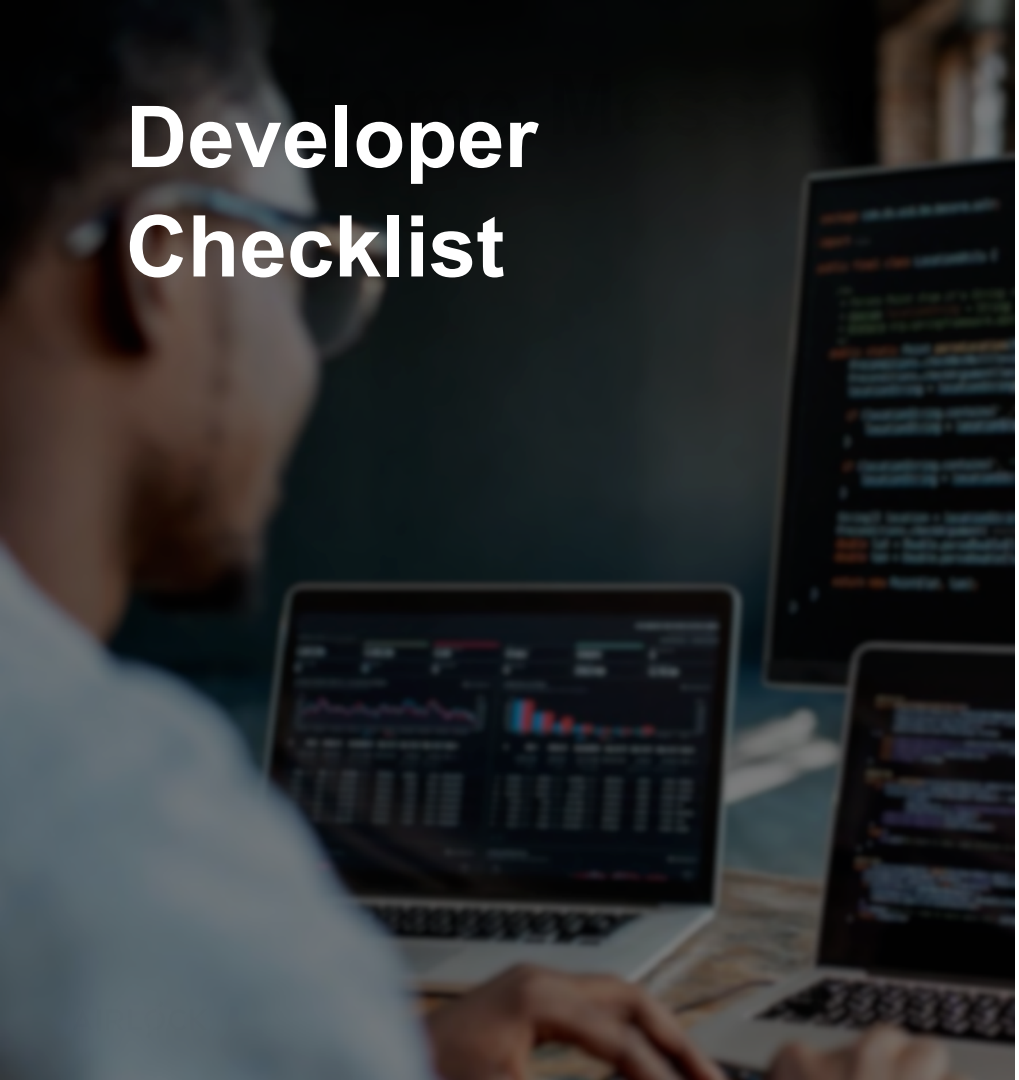
Dev**Ops**

Is there a process for retiring APIs?

# Take-Home Messages

- Web APIs will soon be most important attack surface.
- Web APIs come with specific security risks.
- Security products help a lot, but are limited if business logic is involved.

# Developer Checklist



- ✓ Perform access control on function (A5), object (A1) and attribute level (A3, A6)
- ✓ Handle API keys with care (no public cloud storages!)
- ✓ Don’t trust the client!

ergon

# Further Reading

KNOWLEDGE | SECURITY

OWASP API Security Top 10

## Well protected

Martin Burkhart

Hackers are increasingly setting their sights on APIs. A glance at the OWASP list of vulnerabilities shows the points at which developers are required.

Exposed APIs are becoming the most popular attack surface of web applications. OWASP (Open Web Application Security Project) is responding to this with a new and specialised top ten list for API security.

The original OWASP top ten is a

In this, OWASP emphasises the increasing importance of API security for companies.

Dedicated security products such as web application firewalls (WAF), API Gateways and CIAM systems (Customer Identity Access Management) may have

Modern APIs are typically implemented as RESTful web services. The SPAs call them up directly from the browser, making them equally vulnerable to attack as traditional web applications. Unfortunately, APIs of this kind often exhibit similar or even the same vulnerabilities. Another aggravating factor is that they are located even closer to the sensitive data than was previously the case,

Java Enterprise facad

In a new API

estimates that, by 2

surface in 90 perc

will result

According

abuse o

signi

w

data inc

APIs are being rep

of the US Postal

affected the data

reason for the suc

of underlying acce

Comparison

When comparin

APIs with the

applications fro

clear that seve

same or at least

– Broken Aut

– Security M


heise online

Feburary 2020

https://www.heise.de/hintergrund/Gut-behuetet-OWASP-API-Security-Top-10-4660904.html

→ page 72

English Version:
https://www.airlock.com/fileadmin/content/07_Airlock-PDFs/200305_OWASP-API-Security-en-Heise.pdf

OWASP Top 10 for API Security 2019

AIRLOCK® SECURE ACCESS HUB

Airlock and the OWASP Top 10 API Security 2019
The Ten Most Critical API Security Risks

The following table lists the ten most critical API security risks, as identified by OWASP in the "OWASP Top 10 2019" as of December 2019. It explains how Airlock API addresses each of these risks to protect APIs from these types of attacks and which features are relevant.

| Vulnerability | Description by OWASP | How Airlock API Gateway prevents an exploit | Relevant Airlock Features |
|---|---|---|---|
| A1 – Broken Object Level Authorization | APIs tend to expose endpoints that handle object identifiers, creating a wide attack surface Level Access Control issue. Object level authorization checks should be taken in mind in every function that accesses a data source using an input from the user. | Airlock provides comprehensive access management features for APIs. Therefore, API access in general is controlled by solid authentication and authorization policies.<br><br>Citing from the OWASP document, the main problem occurs "because the server component usually does not fully track the client's state, and instead relies more on parameters like object IDs, that are sent from the client to decide which objects to access." Airlock uses its patented DyVE (Dynamic Value Endorsement) feature to do exactly what's missing: tracking the state of object IDs across client requests. Using DyVE, client requests are restricted to only use valid object IDs that have been offered by the API beforehand in the same client interaction. With this approach, attackers will not be able to inject stolen or crafted object IDs and get unauthorized access. | – API access management<br>– User/client authentication and authorization<br>– Dynamic Value Endorsement (DyVE) |

**About OWASP**
The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organisations to conceive, develop, acquire, operate and maintain applications that can be trusted. All of the OWASP tools, documents, forums and chapters are free and open to anyone interested in improving application security. For more information visit the homepage at www.owasp.org

**About OWASP Top 10**
OWASP started a project for a new top 10 list for API Security raising awareness for this new topic, like it already does for application security. The 10 issues listed represent a broad consensus on what the most critical API security topics are at this time. For more information on the OWASP Top 10, visit https://www.owasp.org/index.php/OWASP_API_Security_Project

https://www.airlock.com/fileadmin/content/07_Airlock-PDFs/OWASP_Top_10-API-2019-eng.pdf

AIRLOCK®

# Thank you for your attention!

Dr. Martin Burkhart

martin.burkhart@airlock.com


AIRLOCK®
SECURE ACCESS HUB