42crunch

Securing an API World

TOP 5 OF POCS ISSUES

ISABELLEMAUNY
ISABELLE@42CRUNCH.COM

WHAT'S IN A PLATFORM?

Design

Developer initiates security work at design time.

Best practices and recommendations are documented.

APIsecurity.io

Develop

Developer documents the API contract with OpenAPI/Swagger.

API Contract security is **audited** from your IDE using 42Crunch plugin.

Integrate & Test

API Contract quality is enforced via CI/CD pipeline. Builds are blocked when minimal security requirements defined by security teams are not met.

API implementation is tested via **Conformance Scan**

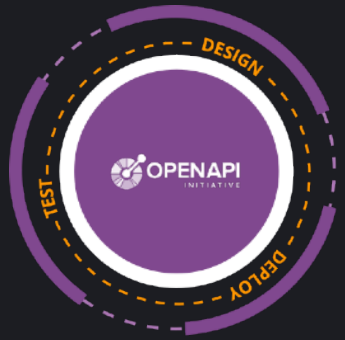
Deploy & Protect

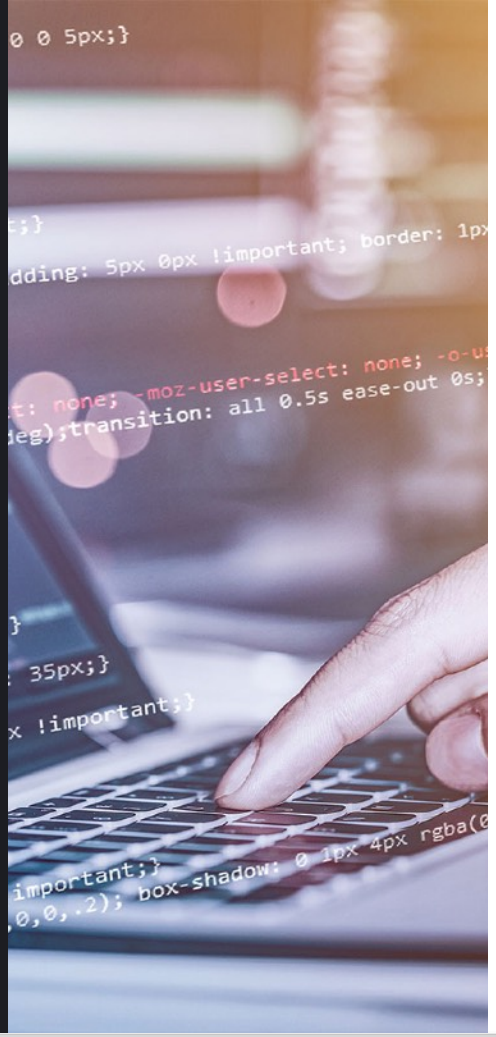
API Firewall is automatically configured from OAS file and deployed in line of traffic.

The firewall can **protect** APIs deployed in containers (such as Docker or Kubernetes) or as reverse proxy in front of API Gateways.

OPENAPI SECURITY AUDITING

DEVELOPERS INITIATE SECURITY AT DESIGN TIME





The **42C Audit service** performs 200+ security checks

- Developers describe the API contract in a language they know
- Audit is available from IDEs and CI/CD pipelines
- Actionable report with **zero false positives**

Key Benefits

- Instant visibility into API security status
- Governance of corporate security standards
- Required security is declared instead of developed/maintained manually across multiple tools/environments

SAMPLE REPORT

API Summary 13 / 100 | Security Audit Report 46 | Conformance Scan Report | Protection | Security Editor

Security Audit

This is the audit score of your OpenAPI file that Security Audit calculated based on more than 200+ checks.

13 out of 100

View checks

Priority Issues | All Issues

Security	0/30				
✗ Authentication	6	0	0	0	0
✓ Authorization	0	0	0	0	0
✗ Transport	0	0	1	0	0

Data validation	13/70				
✗ Parameters	0	1	18	0	0
✓ Response headers	0	0	0	0	0
✗ Response definition	0	0	20	0	0
✓ Schema	0	0	0	0	0

Security in Authentication

15 **Critical issue:** The security section is undefined

The global `security` field of the API has not been defined. This field specifies if your API requires the API consumer to authenticate to use the API.

Go to issue

Data validation in Parameters

13 **Medium risk issue:** String parameter has no maximum length defined

Some string parameters in your API do not have the maximum length specified.

Go to issue

Data validation in Response definition

13 **Medium risk issue:** Response that should contain a body has no schema defined

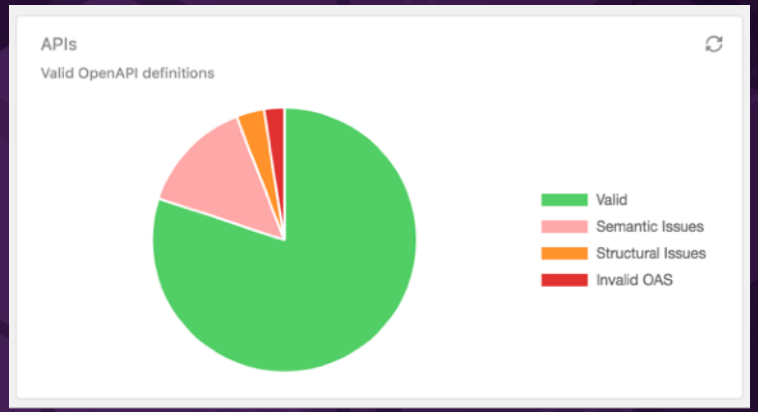
You have not defined any schemas for responses that should contain a body.

Go to issue

42crunch

SOME STATISTICS!

- More than 3000 OpenAPI/ Swagger files tested in POCs!
- **20%** of OAS files are **invalid**
  - ✓ Structural and semantic issues
  - ✓ Not spec compliant
- Average score is **25**
  - ✓ Lowest score was **1**
  - ✓ Highest score was **70**



SECURITY SECTION DEFINITION

- Lack of security definition
  - ✓ There is security but it is not defined in the OAS file
  - ✓ Security is defined but not applied at API or operation level
- Access control mainly via APIkey or similar
  - ✓ JWT used as access token (no OAuth)
  - ✓ Long-lived API keys
- HMAC-based signatures on the rise!
  - ✓ Each request is signed with symmetric/asymmetric key.
  - ✓ Not possible to describe with OAS today, but we are working on it. Vote/Comment here https://github.com/OAI/OpenAPI-Specification/issues/1953 if you're interested!

Severity	Error
Critical	The security section is undefined

Description

The `security` field of the operation has not been defined. This field specifies if your API operation requires the API consumer to authenticate to use it.

For more details, see the OpenAPI Specification.

Severity	Error
Medium	Credentials transported over the network

Description

The API accepts credentials (basic authentication credentials or API keys) transported over the network. The credentials are sent over the network on each API call, over and over again, and are exposed to attack attempts to retrieve them.

WHY THIS MATTERS...

- API key vs. OAuth: Make an informed decision
  - ✓ How sensitive is your API ?
  - ✓ What would be the damage if the APIkey is lost, found in an app or log, stolen via a MITM attack ?
  - ✓ How much does the API key give access to ?
  - ✓ Would a short-lived access token be better suited to the risk?
- If you're adopting OAuth now, remember that authorization_code with PKCE is the grant type you want to use in 95% of cases!

DATA CONSTRAINTS

- Data is poorly constrained
  - ✓ Unbounded array sizes
  - ✓ Undefined strings
  - ✓ Unbounded numbers
- Why this matters ?
  - ✓ Data leakage (API3)
  - ✓ Overflow protection (API4)
  - ✓ Injection protection (API8)
- Base for Input/Output Validation !

```
"properties": {
  "_id": {
    "type": "number",
    "format": "number",
    "example": 1
  },
  "pic": {
    "type": "string",
    "format": "uri",
    "example": 1
  },
  "email": {
    "type": "string",
    "format": "email",
    "example": "email@email.com"
  },
  "password": {
    "type": "string",
    "format": "string",
    "example": "p@ssword1"
  },
```

BUT YOU KNOW THE DATA!

```yaml
parameters:
  - name: uuid
    in: header
    description: >-
      A 128 bit universally unique identifier (UUID) that you generate
      every request and is used for tracking. It is recommended to use
      output from Java UUID class or an equivalent
    type: string
    required: true
```

```json
"properties" : {
  "accountGroup" : {
    "type" : "string",
    "example" : "CHECKING",
    "description" : "Account group is a classification of accounts. Values include:
    CHECKING and SAVINGS"
  },
```

WHAT ABOUT THIS INSTEAD?

```
"name" : "uuid",
"in" : "header",
"description" : "128 bit random UUID generated uniquely for every request",
"required" : true,
"type" : "string",
"maxLength" : 36,
"minLength" : 36,
"pattern" : "[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}"
```

```
"properties" : {
  "accountGroup" : {
    "type" : "string",
    "example" : "CHECKING",
    "enum" : [ "CHECKING", "SAVINGS" ]
  },
```

OUR APIS NEVER FAIL !!
(AND THEY DON'T RETURN DATA...)

```
"paths": {
  "/v1/creditCards/notifications": {
    "post": {
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json"
      ],
      "parameters": […
      ],
      "responses": {
        "200": {
          "description": "Successful operation."
        },
```

OTHER COMMON ISSUES FOUND BY AUDIT

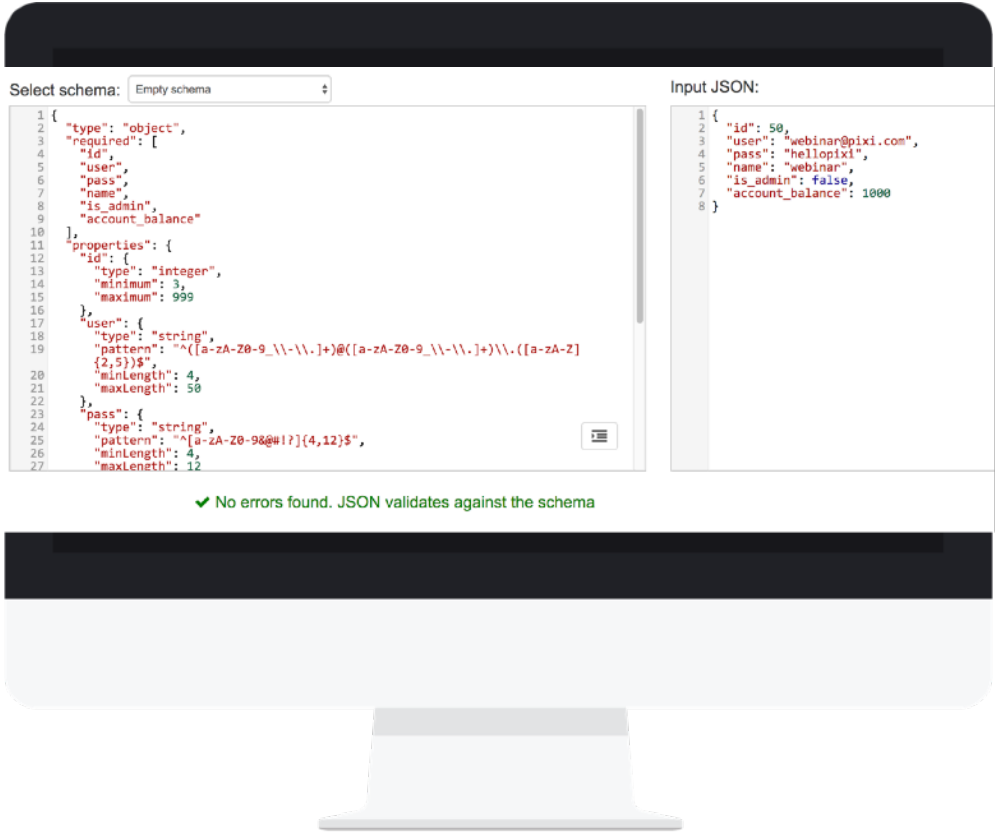
Medium	0.2	415 response should be defined for operations receiving a body (POST, PUT, PATCH)
Medium	0.2	429 response should be defined for all operations
Medium	0.2	No default response defined for the operation
Medium	0.1	If operation has security defined, the 401 response should be defined

RESPONSES MATTER !

- Which responses are you going to return ?
  - ✓ API3 : Data leakage and exception leakage
- Which error codes are valid ?
- Do you control them ?
  - ✓ Do you have tests that can trigger any of those codes ?
- Take ownership of your schemas!
  - ✓ Are they strict enough ?
  - ✓ Where do you validate against them ?
  - ✓ Are you **sure** you are doing that systematically ?
-

DEMO:

SCHEMA VALIDATOR

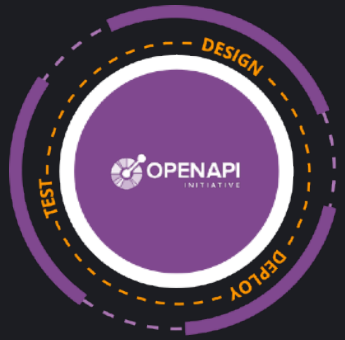


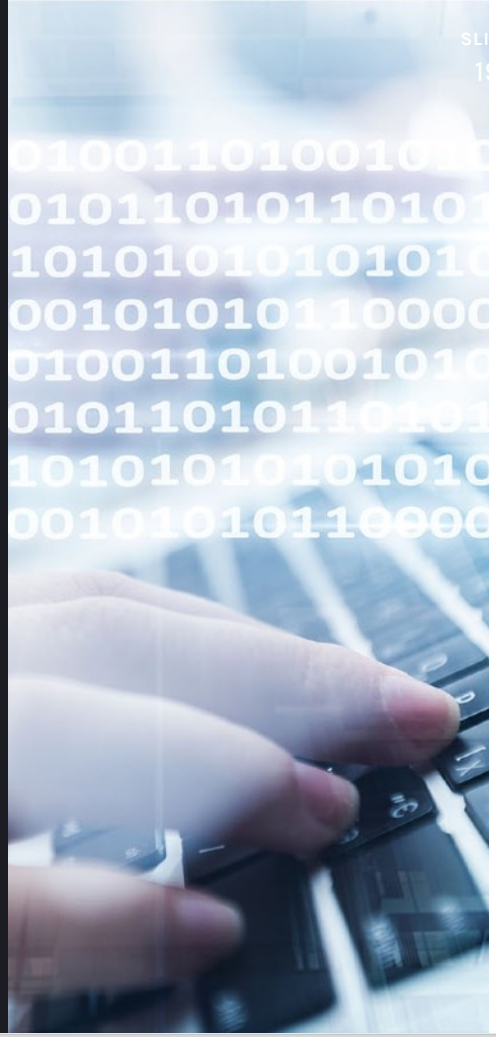
[Demo at : https://www.jsonschemavalidator.net](https://www.jsonschemavalidator.net)

BUT SO DO REQUESTS !

- Schemas on requests need to be strict as well
- Prevents Mass Assignment issues (API6)
- Recommendation: one size does not fit all - Create different APIs per client type/consuming pattern.
  - ✓ Security (do not expose unwanted data)
  - ✓ Performance (reduce network traffic)

SECURITY TEAMS DETECT POTENTIAL ISSUES EARLY





42Crunch conformance scan detects **misconfigurations** and **API vulnerabilities**.

- API Implementation is scanned from API contract
- Ensures conformance to the API contract
- Detects misconfigurations and misbehaviors

Key Benefits

- Early detection of data or exception leakage
- Continuous scans for API vulnerabilities

SAMPLE REPORT

API Summary 88 / 100 | Security Audit Report 11

Conformance Scan

Filter paths
/api/register

All methods | Clear Selection

/api/register 58 TOTAL: 59

- DELETE /api/register 1
- GET /api/register 1
- HEAD /api/register 1
- OPTIONS /api/register 1
- PATCH /api/register 1
- POST /api/register 52
- PUT /api/register 1
- TRACE /api/register ✓

Issue:

The generated value is of the type boolean instead of the type 'object'

The content type '%s' in the received response is not defined in the OpenAPI definition of the API

Url:	https://ds-42c-api.eastus.cloudapp.azure.com/api/register
Status:	400
JSON pointer:	/definitions/UserRegistrationData
Request content type:	application/json
Response time:	172 ms
Response size:	967 bytes
Response content type:	text/html; charset=utf-8

Response body:	<!DOCTYPE html> <html lang="en"> <head> <meta charset ="utf-8"> <title>Error</title> </head> <body> <pre>SyntaxError: Unexpected token t in JSON at position 0
	    at JS ON.parse (<anonymous>)
	    at createStrictS yntaxError (/usr/src/pixi/web/node_modules/body-parser/lib/types/j son.js:158:10)
	    at parse (/usr/src/pixi/web/node_ modules/body-parser/lib/types/json.js:83:15)
	    at /usr/src/pixi/web/node_modules/body-parser/lib/read.js:121:18
	    at invokeCallback (/usr/src/pixi/web/node_modules/r aw-body/index.js:224:16)
	    at done (/usr/src/pixi/ web/node_modules/raw-body/index.js:213:7)
	    at IncomingMessage.onEnd (/usr/src/pixi/web/node_modules/raw-bo dy/index.js:273:7)
	    at emitNone (events.js:111:2 0)
	    at IncomingMessage.emit (events.js:208:7)< br> at endReadableNT (_stream_readable.js:1064:1 2)</pre> </body> </html>

Curl request:	curl -X 'POST' -d 'true' -H 'Content-Type: application/json' -H 'X-Scan-Transactionid: 44f23a8e-29bb-460a-ae95-03d7e0218b6 e' 'https://ds-42c-api.eastus.cloudapp.azure.com/api/register'

Issue 3 of 52 | Previous Issue | Next Issue | Close Window

42crunch

COMMON CONFORMANCE SCAN ISSUES

- Unknown error message types
  - ✓ Bad data triggers unknown response types (text/html for example)
- Unknown response codes
  - ✓ Bad data triggers undocumented responses
- Responses do not conform to schemas
  - ✓ Schemas are not aligned to responses

BUT THE GOOD NEWS IS...

OPENAPI INITIATIVE

OpenAPI Specification (formerly Swagger Specification) is an API description format for REST APIs. An **OpenAPI** file allows you to describe your entire API, including: Available endpoints (/users) and operations on each endpoint (GET /users , POST /users)

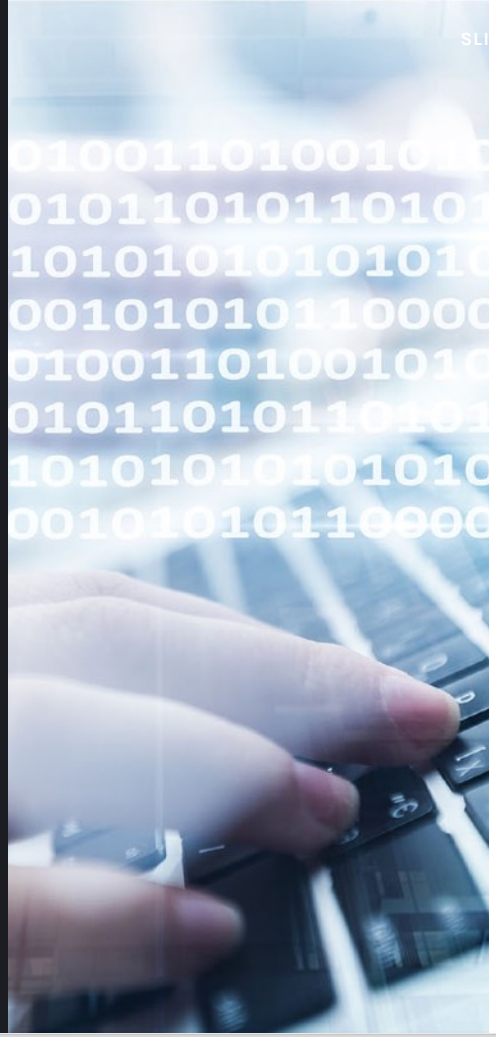
OPEN API = POSITIVE SECURITY MODEL

- Web Application Security is painful because the security is not handled from beginning
- Developers cannot define how the web application is built and designed
- After 20 years of R&D, detection and protection tools have to use AI to understand how the Web Application works...

=> Now we have a worldwide accepted and used API standard: OpenAPI Specification

=> We build a whitelist based on OAS

SECURITY REQUIRES GOVERNANCE!



STRENGTHEN YOUR API CONTRACTS!

- OpenAPI is the perfect format to represent the interface of your APIs
- Make sure it truly represent what the API does
- Use our unique Audit functionality to evaluate the completeness of your API contracts

Start evaluating today from apisecurity.io or using our VSCode IDE extension.

42 crunch

Securing an API World

CONTACT US:

INFO@42CRUNCH.COM

Start testing your APIs today on apisecurity.io!

42CRUNCH RESOURCES

- 42Crunch Website
- Free OAS Security Audit
- OpenAPI VS Code Extension
- OpenAPI Spec Encyclopedia
- OWASP API Security Top 10
- APIsecurity.io