



Securing your API: The OWASP API Top 10

Rob Allen, October 2026



57% of organizations suffered an API-related data breach in the past two years

Traceable 2025 Global State of API Security report



OWASP API Security Top 10



The background of the slide is a dark, moody photograph of a heavy, rusty metal door. A thick metal chain is attached to the door, and a large, weathered padlock is visible in the lower right quadrant. The overall tone is somber and industrial.

Who are you and what can you access?

Authentication and authorisation failures



#2

Broken Authentication

APIs that don't properly verify who you are

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Broken Authentication

APIs that don't properly verify who you are

- Weak/no token validation
- Credential stuffing attacks

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Broken Authentication

Examples

- API accepts JWT without verifying the signature
- Login endpoint allows unlimited password attempts

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Broken Authentication

✓ *Prevention*

- Use established standards (OAuth 2.0, OpenID Connect)
- Implement proper token validation and expiration
- Rate limiting on auth endpoints

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



#5

Broken Function Level Authorisation

Users can access functionality they shouldn't

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Broken Function Level Authorisation

Users can access functionality they shouldn't

- Incorrect authorisation check on a function or resource
- Legitimate calls to endpoints that the user shouldn't have access to
- Undocumented open endpoints

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Broken Function Level Authorisation

Examples

- /debug/dump
- Role unchecked

```
GET /users HTTP/1.1
```

```
Authorization: Bearer <regular user's token>
```

```
HTTP/1.1 200 OK
```

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Broken Function Level Authorisation

✓ *Prevention*

- Deny by default
- Check roles/permissions on every endpoint
- Don't rely on hiding endpoints from documentation

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



#1

Broken Object Level Authorisation

Users can access objects belonging to other users

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Broken Object Level Authorisation

Users can access objects belonging to other users

- User can access another user's resource
- Changing an ID or key allows access to privileged data

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Broken Object Level Authorisation

Examples

- /users/123/orders - change to 124 and see someone else's orders

```
GET /users/124/orders HTTP/1.1
Authorization: Bearer <token for user 123>

HTTP/1.1 200 OK
```

- /accounts/1/documents/999 checks the account, but not the document

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Broken Object Level Authorisation

✓ *Prevention*

- Implement proper authorisation based on user policies
- Check if the user has access to the requested resource
- Check that the operation is also allowed

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



#3

Broken Object Property Level Authorisation

Users can read or modify properties they shouldn't

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Broken Object Property Level Authorisation

Users can read or modify properties they shouldn't

- Sending properties that this user shouldn't see
- Allowing this user to change a property they shouldn't

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Broken Object Property Level Authorisation

Examples

- Profile response includes PII that only admins should see
- User updates profile

```
PATCH /users/123 HTTP/1.1
Content-Type: application/json

{"name": "Rob", "role": "admin"}
```

Broken Object Property Level Authorisation

✓ *Prevention*

- Specifically choose object properties to return
- Explicit allowlists for input properties

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10







Exploiting how your API works

Business logic and resource abuse



#4

Unrestricted Resource Consumption

APIs that can be abused through resource consumption

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Unrestricted Resource Consumption

APIs that can be abused through resource consumption

- Expensive operations without throttling
- Exhausting memory through requests for too much data
- Denial of service

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Unrestricted Resource Consumption

Examples

- `/widgets?page=1&per_page=1000000`
- Resending an OTP; each SMS costs money

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Unrestricted Resource Consumption

✓ *Prevention*

- Rate limiting (per IP, per user, per endpoint)
- Pagination with maximum limits
- Resource quotas / Timeouts

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

#6

Unrestricted Access to Sensitive Business Flows

Critical workflows lack protection against automation

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Unrestricted Access to Sensitive Business Flows

Critical workflows lack protection against automation

- Legitimate calls, but unexpected order
- Excessive access may harm the business

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Unrestricted Access to Sensitive Business Flows

Examples

- Ticket scalping bots
- Inventory hoarding

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Unrestricted Access to Sensitive Business Flows

✓ *Prevention*

- Device fingerprinting
- Behavioural analysis
- Transaction limits

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

#10

Unsafe Consumption of APIs

Your API trusts third-party APIs too much

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Unsafe Consumption of APIs

Your API trusts third-party APIs too much

- Dependency on another's vulnerabilities
- Not accounting for failure

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Unsafe Consumption of APIs

Examples

- Synchronous Geolocation API times out
- Third-party API redirects your request to an attacker's host

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Unsafe Consumption of APIs

✓ *Prevention*

- Validate all external data
- Whitelist redirect URLs
- Implement timeouts

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10





Operational security gaps

Configuration and infrastructure vulnerabilities

#8

Security Misconfiguration

Insecure defaults and missing security hardening

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Security Misconfiguration

Insecure defaults and missing security hardening

- Default configurations
- Missing security updates
- Unnecessary features enabled

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10



Security Misconfiguration

Examples

- Error messages return stack traces
- Public S3 buckets holding data backups or user uploads

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Security Misconfiguration

✓ *Prevention*

- Regular security updates
- Regular audits

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

#9

Improper Inventory Management

Do you know your API?

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Improper Inventory Management

Do you know your API?

- Old API versions still running
- Shadow APIs
- Non-production environments accessible

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Improper Inventory Management

Examples

- v1 API wasn't decommissioned
- Staging environment is public

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Improper Inventory Management

✓ *Prevention*

- Maintain API inventory/catalogue
- Retire old versions with clear timelines

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

#7

Server Side Request Forgery

API fetches remote resources without validation

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Server Side Request Forgery

API fetches remote resources without validation

- User-controlled URLs in API requests
- API fetches a remote resource from a user-supplied URL

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Server Side Request Forgery

Examples

- The API takes an internal URL; calls and returns it

```
POST /avatar HTTP/1.1
Content-Type: application/json

{"url" "http://127.0.0.1:8080/metrics" }

HTTP/1.1 200 OK
Content-Type: text/plain

process_cpu_seconds_total 4817.3
```

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

Server Side Request Forgery

✓ *Prevention*

- Validate and sanitise URLs
- Whitelist for domains & media types, etc
- Disable HTTP redirection where possible
- Don't send raw responses to clients

#1

#2

#3

#4

#5

#6

#7

#8

#9

#10

In closing



OWASP API Security Top 10

Authentication & authorisation failures

- #2 Broken Authentication
- #5 Broken Function Level Authorisation
- #1 Broken Object Level Authorisation
- #3 Broken Object Property Level Authorisation

Business logic & resource abuse

- #4 Unrestricted Resource Consumption
- #6 Unrestricted Access to Sensitive Business Flows
- #10 Unsafe Consumption of APIs

Configuration & infrastructure vulnerabilities

- #8 Security Misconfiguration
- #9 Improper Inventory Management
- #7 Server Side Request Forgery



Security requires

- Defence in depth
- Testing with the mindset of an attacker
- Ongoing attention



Resources

OWASP API Security Project website

<https://owasp.org/projects/api-security-project>

REST Security Cheat Sheet

https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html

OWASP crAPI (Completely Ridiculous API)

<https://github.com/OWASP/crAPI>



Thank you!

