OWASP API SECURITY TOP 10

Christian Dussol

CRITICAL THREATS YOU NEED TO KNOW

83%

of all Internet traffic is API traffic

#1

API attacks will become the main attack vector

94%

Organizations experienced API incidents (2023)

Akamai

Gartner®

SALT SECURITY

API1:2023

CRITICAL

BROKEN OBJECT LEVEL AUTHORIZATION

Problem

- User A can access User B data
- ID manipulation in URLs/requests

GET /api/users/123/transactions
→ Change 123 to 456 = unauthorized access!

Solution

- ✅ Systematic access rights validation
- ✅ Granular authorization controles
- ✅ Automated permission testing

API2:2023

CRITICAL

BROKEN AUTHENTICATION

Common vulnerabilities

- Misconfigured JWT tokens
- Missing key rotation
- Insufficient credential validation

Financial impact

- 🏦 Unauthorized access to customer accounts
- 💳 Transaction fraud
- 📊 Sensitive data theft

Best practices

- ✅ OAuth2/OIDC
- ✅ Multi-factor authentication
- ✅ Token expiration and rotation

API3:2023

HIGH

BROKEN OBJECT PROPERTY LEVEL AUTHORIZATION

The trap: Returning ALL object properties instead of necessary data

Dangerous example

```
{
    "user": {
        "name": "John Doe",
        "email": "john@example.com",
        "ssn": "123-45-6789", ⚠️ Sensitive!
        "salary": 75000, ⚠️ Confidential!
        "role": "user"
    }
}
```

Solution

- ✅ Role-based field filtering
- ✅ Controlled serialization
- ✅ Principle of least privilege

API4:2023

HIGH

UNRESTRICTED RESOURCE CONSUMPTION

The attacks

- ⚡ Denial of Service (DoS)
- ⚡ Server Resource exhaustion
- ⚡ Exploding cloud bills

Real scenarios

- 🔴 100 GB file uploads
- 🔴 Unpaginated requests
- 🔴 Unlimited intensive computations

Protection

- ✅ Intelligent rate limiting
- ✅ Appropriate timeouts
- ✅ Payload size validation
- ✅ Resource monitoring

API5:2023

CRITICAL

BROKEN FUNCTION LEVEL AUTHORIZATION

The risk: A standard user accessing admin functions

Typical example

POST /api/admin/delete-user → Accessible with a normal user token!

In Financial context

💰 Credit limit modifications

📈 Access to confidential reports

🏛️ Customer account administration

Security

✅ Role-based access control (RBAC)

✅ Systematic server-side validation

✅ Clear separation of responsibilities

API6:2023

HIGH

UNRESTRICTED ACCESS TO SENSITIVE BUSINESS FLOW

The threat: Malicious automation of critical processes

- 🤖 Mass creation of fake accounts
- 💸 Automated fraudulent transfers
- 📊 Exchange rate manipulation
- 🎟️ Financial service scalping

Concrete scenario: Playstation 5 scalping, automated high-frequency trading abuse

Defenses

- ✅ Abnormal behavior detection
- ✅ CAPTCHA on sensitive actions
- ✅ User/IP action limitations
- ✅ Behavioral analysis

API7:2023

HIGH

SERVER SIDE REQUEST FORGERY

The attack: Force the server to make unauthorized requests

Dangerous scenario

```
POST /api/webhook {

    "url": "http://internal-admin:8080/delete-db"

} → Access to internal network!
```

Specific risks

🔓 Firewall bypass

💾 Internal database access

☁️ Cloud service compromise

Mitigations

✅ Allowed domain whitelist

✅ Strict URL validation

✅ Network segmentation

API8:2023

HIGH

SECURITY MISCONFIGURATION

Classic mistakes

- HTTPS disabled in production
- Missing security headers
- Overly detailed error messages
- Unchanged default configurations

Impacts

📱 Sensitive information exposure

🥷 Easier fingerprinting for attackers

🔍 Technical data leakage

Security checklist

✅ HTTPS everywhere

✅ Restrictive CORS headers

✅ Generic error handling

✅ Regular configuration audits

API9:2023

Medium

IMPROPER INVENTORY MANAGEMENT

The invisible problem: You cannot secure what you don't know exists!

Risks

- Undocumented "ghost" APIs
- Unmaintained obsolete versions
- Debug endpoints in production
- Unaudited third-party APIs

In financial services

- Test endpoints with real data
- Unsecured partner APIs
- Exposed internal microservices

Solution

- ✅ Automatic API discovery
- ✅ Centralized and up-to-date catalog
- ✅ Controlled versioning and deprecation

API10:2023

HIGH

UNSAFE API CONSUMPTION

The weak link: Blindly trusting third-party APIs

Attack vectors

- Malicious data injection
- Supply chain attacks
- ⚡ Denial of service via third-party API

Critical example

A compromised identity verification API → Massive fraud

Best practices

- ✅ Rigorous data validation
- ✅ Timeout and circuit breakers
- ✅ Third-party API monitoring
- ✅ Continuity plans

SET ACTIONS AND PRIORITIES

This week

Audit critical APIs

This month

Automate security testing

Ongoing

Monitor & Maintain

💡 API security is not a project, it’s a continuous process!

WHAT ABOUT YOU
WHAT'S YOUR PRIORITY?

🎯 **Which OWASP risk** concerns you most?

🛠️ **Which tools** do you use to secure your APIs?

📚 **Which resources** do you recommend?

🛡️ Pro Bonus Tip for Kubernetes

SECURE API WITH KYVERNO

✅ Kubernetes-native policy engine

✅ No proprietary language (YAML)

✅ Policy-as-Code approach

✅ No agent required

🔐 **How Kyverno secures your APIs**

```
# Block privileged containers exposing APIs
apiVersion: kyverno.io/v1
kind: Policy
spec:
  rules:
  - name: check-privileged
    validate:
      message: "Privileged containers are not allowed"
      pattern:
        spec:
          securityContext:
            privileged: "false"
```

Pro Bonus Tip for Kubernetes

SECURE API WITH KYVERNO

Validate
Block non-compliant APIs

Mutate
Auto-add security configs

Generate
Create NetworkPolicies

Perfect for Fintech: Compliance automation, audit trails, multi-tenant isolation

Get started: playground.kyverno.io