Writing Secure APIs: A Look At the OWASP 2023 API Top 10 List

Darylynn Ross
Please don’t hold questions to the end. Discussion is encouraged.
AWARENESS
November 2022 T-Mobile: Hackers exploited an API to steal the data of 37M customers. They have had 8 data breaches in less than 5 years resulting in more than $500M in fines and settlements.

September 2022 Optus (Australian Telecom): Attackers abused a publicly exposed endpoint that didn’t require authentication. They gained highly sensitive PII rich data of 10M customers. To make it easier for attackers, the API used sequential ids. Estimated financial impact was $140M.

July 2022 X (formerly Twitter): exposed the PII of 5.4M user accounts. An API allowed attackers to find user accounts by providing email or phone numbers. They had to pay $250M in fines for the incident.
Everybody’s Got A Story (From Traceable)

- **60%** of organizations have faced an API-related breach in the last two years and **74%** of these endured three or more incidents.

- **57%** stated that traditional security solutions are unable to effectively distinguish genuine from fraudulent API activity.

- **61%** anticipate rising API-related risks in the next two years as they deal with an average of **127 third-party API connections**, with just **33%** confident in managing external API threats.
Why do you like APIs?

What’s important to developers?
Why do you like APIs?

What’s important to product owners?
Why do you like APIs?

What’s important to security engineers?
Why do you like APIs?

What's important to bad actors?
Why Are Our APIs Insecure?

- Easy Access
- Giant Inventory
- No Security in Design
- Zombie APIs
- Shadow APIs
- We Share Tons of Info
- Weak Security Testing
- Testing
OWASP API Top Ten Security Risks
What is OWASP & Why an API Top Ten?

https://owasp.org/API-Security/
We’re #1

Authentication & Authorization

#1 Broken Object Level Authorization

#2 Broken Authentication

#3 Broken Object Property Level Authorization

#5 Broken Function Level Authorization

#6 Unrestricted Access to Sensitive Business Flows

Attacks on Confidentiality
Broken Authentication

Attacker can compromise passwords, keys or session tokens, user account information, and other details.

Systems are unlikely to be able to distinguish attackers from legitimate users.

No Brute Force Controls, Leaking Auth Details, Weak Auth Standards, Poor Validation, Weak Encryption.

What can we do about it?
Broken Authentication Prevention

- Know the program auth flows
- Know how your auth mechanisms work
- Treat recovery endpoints like login
- Require re-auth for sensitive operations
- Implement account lockout controls
Broken Authentication Prevention

- Use Multi Factor Authentication when possible
- Don’t use API Keys for user authentication
- Use anti brute force /cred stuffing controls
- Use strong encryption controls
- Reference OWASP Authentication Cheat Sheet
Every API endpoint that receives an ID of an object, and performs any action on the object, should implement object-level authorization checks. The checks should validate that the logged-in user has permissions to perform the requested action on the requested object.
Property
- Allow sensitive data to be read by the user
- Allow CRUD ops on the value of an object property

Method

Function
- X can access a function that should be exposed only to Y
- Access to Admin Endpoints
- Access CRUD via change in HTTP Method
- An API endpoint is regular or admin based only on the URL

Object

“Every API endpoint that receives an ID of an object, and performs any action on the object, should implement object-level authorization checks. The checks should validate that the logged-in user has permissions to perform the requested action on the requested object.”
Property
- Allow sensitive data to be read by the user
- Allow CRUD ops on the value of an object property

Method

Function
- X can access a function that should be exposed only to Y
- Access to Admin Endpoints
- Access CRUD via change in HTTP Method
- An API endpoint is regular or admin based only on the URL

Object

“Every API endpoint that receives an ID of an object, and performs any action on the object, should implement object-level authorization checks. The checks should validate that the logged-in user has permissions to perform the requested action on the requested object.”
**Property**
- Allow sensitive data to be read by the user
- Allow CRUD ops on the value of an object property

**Method**

**Function**
- X can access a function that should be exposed only to Y
- Access to Admin Endpoints
- Access CRUD via change in HTTP Method
- An API endpoint is regular or admin based only on the URL

**Broken Authorization Symptoms**

“Every API endpoint that receives an ID of an object, and performs any action on the object, should implement object-level authorization checks. The checks should validate that the logged-in user has permissions to perform the requested action on the requested object.”
Broken Authorization Prevention

- Use Proper Authorization Mechanisms
- Check Access for Every DB Action
- Use Random and Unpredictable GUIDS
- Test Authorization Functionality for ALL Roles
- Deny Access to Sensitive Data By Default
- Inherit Admin Authorization Checks from Abstract Controller
API Abuse

Unauthorized Access to Sensitive Business Flows

- Unsanctioned Use
- Unintended Use
- Exploiting Vulnerabilities
Unrestricted Access to Sensitive Business Flows

Identify the business flows that might harm the business if they are excessively used.

Choose the right protection mechanisms to mitigate the business risk.
Intermission
Consumption & Forgery

#4 Unrestricted Resource Consumption

#7 Server-Side Request Forgery

#10 Unsafe Consumption of APIs

Attacks on Integrity & Availability
Server-Side Request Forgery

Occurs when an API is fetching a remote resource without validating the user-supplied URI.

This allows an attacker to coerce the server-side application to make requests to an unintended location, regardless of WAF or VPN protections.

External Resource Access Happens Here
Server-Side Request Forgery

Access to all the files including contents in any bucket

Relayed Metadata Service Requests

(Server-Side Request Forgery Attack)

Misconfigured WAF hosted on AWS

Breach

July 2019
100M Consumer Credit Apps Stolen Settlement & Fines: $270M
Server-Side Request Forgery

- Isolate Resource Fetching
- Use Allow Lists
- Disable HTTP Redirects
- URL Parser
- Validate Client Supplied Data
- Don’t Forward Raw Responses
Unrestricted Resource Consumption
“Here Piggy, Piggy”

- Timeouts
- # of Ops per request
- Max Upload File Size
- Spending Limits
- # of recs to return per page
- Max Alloc Memory
- Max # of Processes

- Containers / Serverless
- Spending Limits / Alerts
- Server-Side Validation
- Max Size Incoming Payloads
- Rate Limiting
- Limit single client # of ops

FIXED
Unsafe Consumption of APIs

What friends should you TRUST?

- Assess 3rd party API service providers
- Validate and sanitize data from other APIs
- Allow list any redirect URIs / don’t blindly follow redirects
- Use TLS
- Limit resources available to 3rd parties
The Rest: Making Life Easier For Attackers

#8 Security Misconfiguration

#9 Improper Inventory Management

Attacks on Integrity
Security Misconfig

You might be vulnerable if ....

- Your error messages contain stack traces
- You have bonus features
- You don’t use TLS
- You aren’t keeping up on patching
- Your Cross Origin Resource Sharing (CORS) policy is missing or improperly set up
- You are clueless about configuring your cloud
- You don’t have a repeatable, automated hardening process
- You don’t use security headers
Improper Inventory Mgmt

Sensitive Data / Dataflow Blind Spots
× No business approval or justification
× No inventory or visibility
× No visibility of what data is being shared

Documentation Blind Spots
× Where it runs?
× Who has access?
× No versioning strategy?
× Old or no documentation
× Old or no inventory data
Improper Inventory Management

Prevention
- Automation
- Security Tooling
- Inventory and Documentation
- Know Sensitive Dataflows
- Be Careful with Data in Non Prod Envs
- Version Strategy
How Do I Use This?
Next Steps

- Talk to Your Security Team
- Assess Your Risk
- Prioritize
- Knock Out the Easy Stuff
- Tapas (Small bites) for the Hard Stuff
- Educate Yourself and Others
- Join the App Sec Conversation
Questions?
Resources

Sources

• Why API attacks are increasing and how to avoid them CSOonline.com
• Firetail API Data Breach Tracker Firetail.io
• https://www.techtarget.com/searchsecurity/news/252467901/Capital-One-hack-highlights-SSRF-concerns-for-AWS Tech Target
• https://www.traceable.ai/ Traceable
• https://portswigger.net/web-security/ssrf Portswigger on Server Side Request Forgery
• https://www.darkreading.com/cyber-risk/even-perfect-apis-can-be-abused Dark Reading on API Abuse
Resources

OWASP Resources:

• [https://owasp.org/API-Security/](https://owasp.org/API-Security/)
• [https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html)