OWASP API Threat Protection with 42Crunch

Part 2: Authentication/Authorisation

ISABELLEMAUNY
ISABELLE@42CRUNCH.COM

API Breaches are on the rise!

- 300+ breaches reported on apisecurity.io since Oct. 2018
- And those are just the public ones!
- Most recurrent causes (combination of):
  - Lack of Input validation
  - Lack of Rate Limiting
  - Data/Exception leakage
  - BOLA/IDOR (Authorization)

Hacking Starbucks and Accessing Nearly 100 Million Customer Records

June 20, 2020 samwcyo

Facebook - 50 million users' personal information was exposed

PayPal - 1.6 million customers at risk of data exposure

T-Mobile - 76 million users' phone numbers and addresses stolen

Instagram - 49 million users' emails and phone numbers exposed

Uber - 57 million riders and drivers accounts were compromised

Justdial - Over 100 million Indian users' personal data at risk

Equifax - 147 million users personal data stolen

Starbucks - 100 million customer records accessed

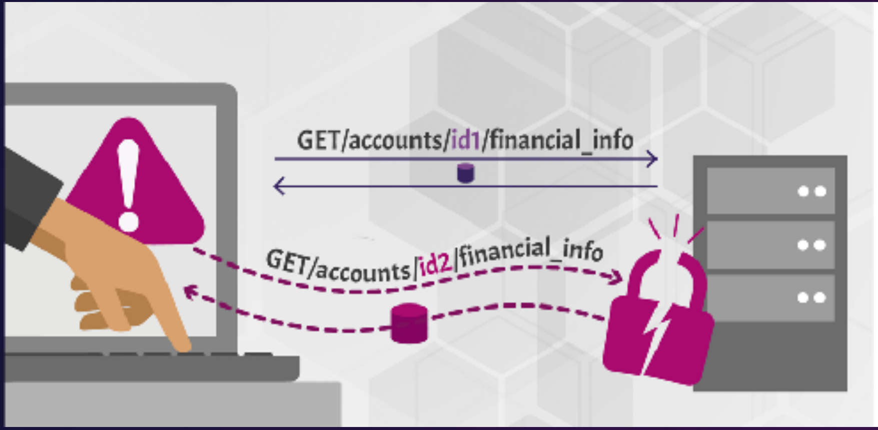
Verizon - 14 million subscribers phone numbers and PINs exposed

OWASP API Security Top 10

APIs have different vulnerabilities

- API1 : Broken Object Level Access Control
- API2 : Broken Authentication
- API3 : Excessive Data Exposure
- API4 : Lack of Resources & Rate Limiting
- API5 : Missing Function Level Access Control
- API6 : Mass Assignment
- API7 : Security Misconfiguration
- API8 : Injection
- API9 : Improper Assets Management
- API10 : Insufficient Logging & Monitoring

API1 : BOLA / IDOR


GET/accounts/id1/financial_info
GET/accounts/id2/financial_info

APIs tend to expose endpoints that handle object identifiers, creating a wide attack surface Level Access Control issue. Object level authorization checks should be considered in every function that accesses a data source using an input from the user.

Uber

New Relic

Google Cloud

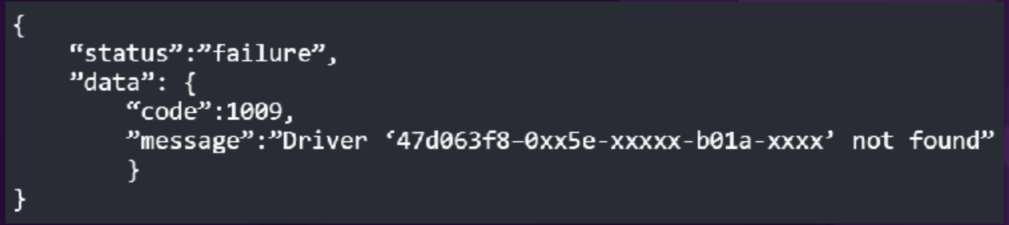
SONICWALL

Online Registration System
Ministry of Electronics & Information Technology
Goverment of India

UBER (SEPT 2019)

https://appsecure.security/blog/how-i-could-have-hacked-your-uber-account

- The Attack
  - ✓ Account takeover for any Uber account from a phone number
- The Breach
  - ✓ None. This was a bug bounty.
- Core Issues
  - ✓ First Data leakage : driver internal UUID exposed through error message!



```
{
    “status”:”failure”,
    ”data”: {
        “code”:1009,
        ”message”:”Driver ‘47d063f8-0xx5e-xxxxx-b01a-xxxx’ not found”
        }
}
```

  - ✓ Hacker can access any driver, user, partner profile if they know the UUID
  - ✓ Second Data leakage via the getConsentScreenDetails operation: full account information is returned, when only a few fields are used by the UI. This includes the **mobile token** used to login onto the account

API1 (BOLA) MITIGATION

- Fine-grained authorisation in **every controller layer**
- Additionally:
  - ✓ Avoid guessable IDs (123, 124, 125…)
  - ✓ Avoid exposing internal IDs via the API
  - ✓ Alternative: GET https://myapis.com/phone/me
- OAuth scopes are not the solution here, as they limit access to an operation and not to a resource.
- **Test this use case!**
- Mitigate potential data scrapping by putting rate limiting in place

How 42Crunch addresses API1

API SECURITY AUDIT (DEVELOPMENT/TEST)	**CONFORMANCE SCAN** (DEVELOPMENT/TEST)	**MICRO-API FIREWALL** (RUNTIME PROTECTION)
• Flag enumeration exposure risks*	• Automatic enumeration of Ids*	• IDOR Policies* • Integration with fine-grained authorization systems like OPA

API2 : Broken Authentication

Authentication mechanisms are often implemented incorrectly, allowing attackers to compromise authentication tokens or to exploit implementation flaws to assume other user's identities temporarily or permanently. Compromising system's ability to identify the client/ user, compromises API security overall.

facebook

Auth0

SIEMENS

AUTHO (APRIL 2020)

https://insomniasec.com/blog/auth0-jwt-validation-bypass

- The Attack
  - ✓ Authentication Bypass
- The Breach
  - ✓ None. Discovered as part of pen-testing.
- Core Issues
  - ✓ The Authentication API prevented the use of **alg: none** with a case sensitive filter. This means that simply capitalising any letter e.g. alg: **nonE**, allowed tokens to be forged.

API2 (BROKEN AUTH) MITIGATION

- No un-authenticated endpoints!
- Use short-lived access tokens and limit their scope
- Use OAuth properly (most likely authorization_code with PKCE)
  - ✓ Financial API Grade profiles as reference (https://openid.net/wg/fapi/)
- Make sure you validate JWTs according to Best Practices (RFC 8725) - https://www.rfc-editor.org/rfc/rfc8725.txt
- Enforce 2FA, captcha
- Use secure storage for credentials
- **Test with all kind of combinations!**

How 42Crunch addresses API2

API SECURITY AUDIT (DEVELOPMENT/TEST)	**CONFORMANCE SCAN** (DEVELOPMENT/TEST)	**MICRO-API FIREWALL** (RUNTIME PROTECTION)
• Flag weak/missing authentication schemes as well as weak transport settings	• Tests automatically for API implementation security issues at early development stages	• JWT Validation according to RFC 8725 • Access tokens/API keys validation from API Contract

API4 : Lack of Resources / Rate Limiting

Quite often, APIs do not impose any restrictions on the size or number of resources that can be requested by the client/user. Not only can this impact the API server performance, leading to Denial of Service (DoS), but also leaves the door open to authentication flaws such as brute force.

facebook

zoom

FACEBOOK (FEB 2018)

https://appsecure.security/blog/we-figured-out-a-way-to-hack-any-of-facebook-s-2-billion-accounts-and-they-paid-us-a-15-000-bounty-for-it

- The Attack
  - ✓ Account takeover via password reset at https://www.facebook.com/login/identify?ctx=recover&lwv=110.
  - ✓ facebook.com has rate limiting, beta.facebook.com does not!
- The Breach
  - ✓ None. This was a bug bounty.
- Core Issues
  - ✓ Rate limiting missing on beta APIs, which allows brute force guessing on password reset code
  - ✓ Misconfigured security on beta endpoints

API4 (RATE LIMITING) MITIGATION

- Protect all authentication endpoints from abuse (login, password reset, OAuth endpoints)
  - ✓ Smart rate limiting : by API Key/access token/user identity/fingerprint
  - ✓ Short timespan
- Not so good example: Instagram, 200 attempts/min/IP for password reset

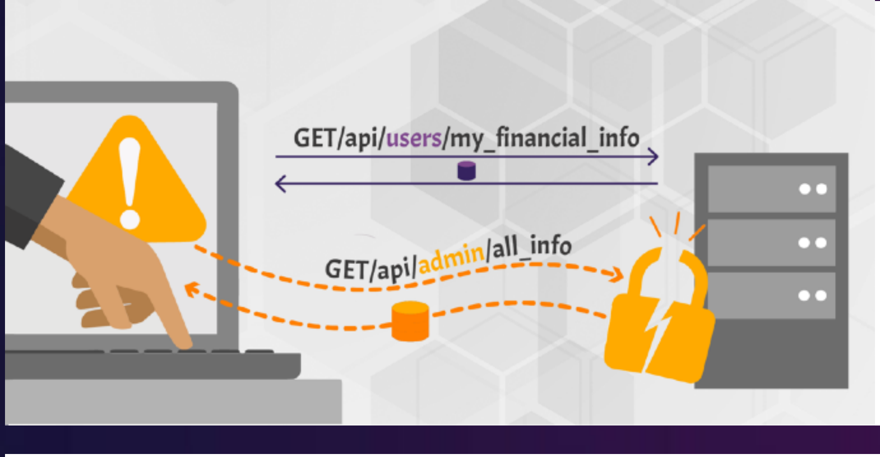
“In a real attack scenario, the attacker needs 5000 IPs to hack an account. It sounds big but that’s actually easy if you use a cloud service provider like Amazon or Google. It would cost around 150 dollars to perform the complete attack of one million codes”

How 42Crunch addresses API4

API SECURITY AUDIT (DEVELOPMENT/TEST)	CONFORMANCE SCAN (DEVELOPMENT/TEST)	MICRO-API FIREWALL (RUNTIME PROTECTION)
• Flag data missing constraints (min/max size) • Flag operations that do not declare 429 responses	• Test data constraints	• Rate Limiting by API and by operation • Blocks overflow type attacks • JSON Parser protection

API 2 + API 4 : Lethal Combination

API5 : Broken Function Level Authorization



Complex access control policies with different hierarchies, groups, and roles, and an unclear separation between administrative and regular functions, tend to lead to authorization flaws. By exploiting these issues, attackers gain access to other users’ resources and/or administrative functions.

הליכוד

Likud Party

LIKUD VOTING APP (NOV 2019)

https://www.zdnet.com/article/netanyahus-party-exposes-data-on-over-6-4-million-israelis/

- The Attack
  - ✓ Data breach using leaked admin userid/passwords
- The Breach
  - ✓ Unknown. Potentially could have leaked the details of 6.4 million Israelis.
- Core Issues
  - ✓ Admin endpoint left open with no authentication - Allows to retrieve all systems users, including the password (in clear). Endpoint was hardcoded in application source code.

API 5 MITIGATION

- Do not mix admin and non-admin operations in the same API
  - Easy to discover via dictionary attacks
- Restrict access to admin APIs, for example:
  - by Mutual TLS
  - by IP Range
  - Do not rely on the client to do that!
- Design properly your authorization policies and test them !
  - OAuth scopes can help here

How 42Crunch addresses API5

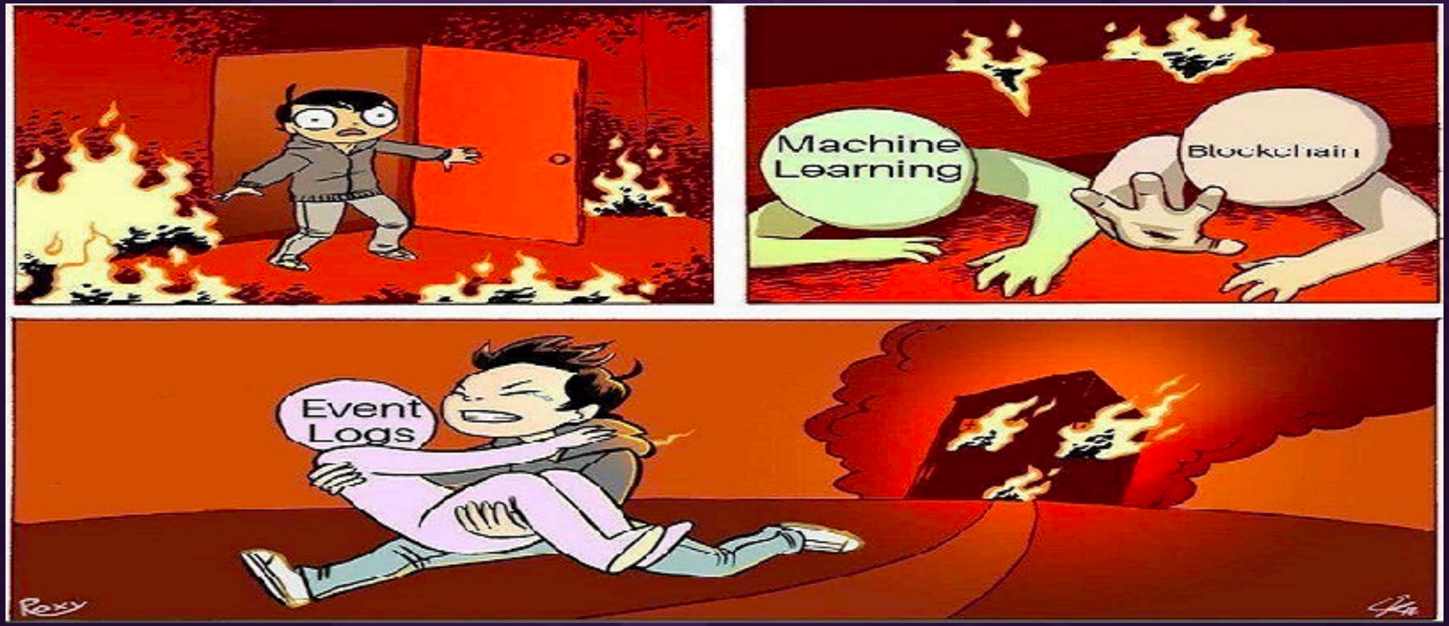
API SECURITY AUDIT (DEVELOPMENT/TEST)	CONFORMANCE SCAN (DEVELOPMENT/TEST)	MICRO-API FIREWALL (RUNTIME PROTECTION)
• Automated audit to discover APIs • Flag missing/invalid OAuth scopes	• Test how API handles unknown requests (verbs, paths, data)	• Block requests with unexpected verbs and paths/subpaths (including path traversal attacks) • Blocks unknown APIs requests

API10 : Logging and Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems to tamper with, extract, or destroy data. Most breach studies demonstrate the time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

A10 : LOGS, LOGS, LOGS!

- Log all API activity
- Pushed to security platforms such as SIEMs for automated Threat detection.



How 42Crunch addresses API10

API SECURITY AUDIT (DEVELOPMENT/TEST)	CONFORMANCE SCAN (DEVELOPMENT/TEST)	MICRO-API FIREWALL (RUNTIME PROTECTION)
N/A	N/A	• All traffic is monitored and logged • Integration with enterprises logging infrastructure • Integration with SIEM

OUR PHILOSOPHY

- We believe good security starts at design time
  - ✓ Encourage good security practices from early days of development
  - ✓ Help developers understand the vulnerabilities that some development practices may lead to.
- We bring tools that easily fit in the development cycle
  - ✓ Fast execution (no burden on productivity)
  - ✓ Actionable reports
- We allow security teams to enforce automatically their requirements along the API lifecycle (dev and ops)
  - ✓ Automation of security testing and deployment of firewall via CI/CD integration
- We use OpenAPI as the single source of truth across Dev, Ops and Security teams.

Thank you!

Contact us | info@42crunch.com | 42crunch.com

Free security tools from 42Crunch

https://42crunch.com/resources-free-tools/