OWASP Top 10 Web App Security Risks

Peter Cosemans - Michiel Olijslagers

Copyright © - Euricom 2024

Organisation

OWASP

- Open Worldwide Application Security Project®
- Web application security
- Open source software projects
- Education, training, awareness,...
- OWASP top 10
- Cheat sheet series
- Samm
- Web security testing guide
- CWE vs CVE

Top 10

OWASP - Top 10

- Awareness document for most common vulnerabilities
- 10 most critical web application security risks
- Get started with security
- 10 categories
  - 8 from data
  - 2 from community survey
- CWE (Common Weakness Enumeration)
  - Weakness type
  - Identification
  - Mitigation
  - Prevention

OWASP - Top 10 - 2021

1. Broken access control
2. Cryptographic failures
3. Injection
4. Insecure design
5. Security misconfiguration
6. Vulnerable and outdated components
7. Identification and authentication failures
8. Software and data integrity failures
9. Security logging and monitoring failures
10. Server-side request forgery

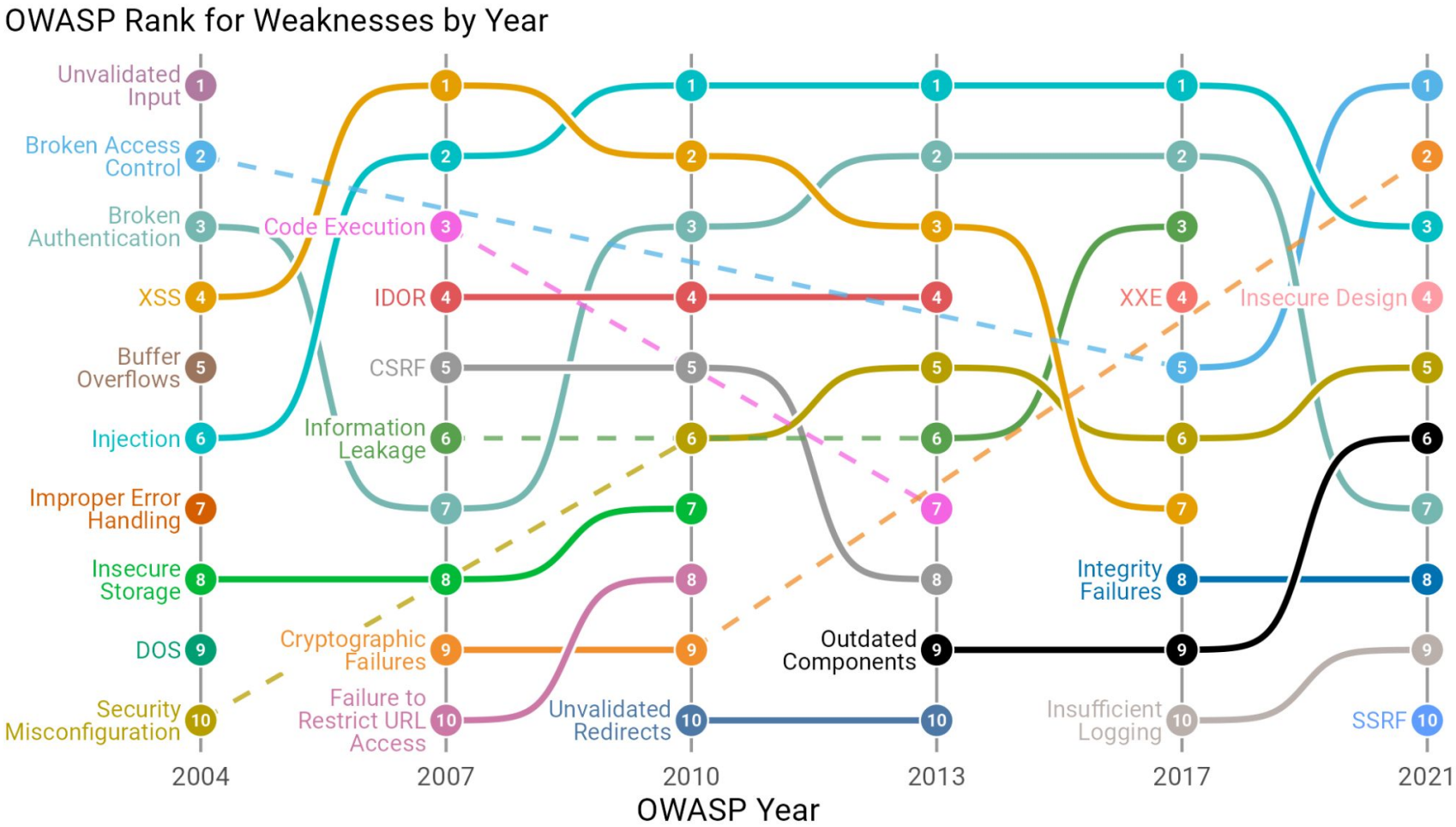
CSRF

Hashing without salt

Improper input validation

Improper neutralization

SSRF



https://www.f5.com/labs/articles/threat-intelligence/the-evolving-cve-landscape

OWASP Top 10 - Data

- **CWEs Mapped:** Number of CWEs in this category
- **Incidence rate:** Percentage of apps vulnerable to that CWE
- **Total occurrences:** Total number of apps found to have CWEs
- **Total CVEs:** Total number of CVEs in the NVD (national vulnerability database)

https://owasp.org/Top10/A00_2021_Introduction/#data-factors

OWASP - A01: Broken access control

CWEs	Incidence rate	Total occurrences	Total CVEs
34	55.97%	318 487	19 013

Interestings CWEs:

- CWE-200 Exposure of Sensitive Information to an Unauthorized Actor
- CWE-201 Insertion of Sensitive Information Into Sent Data
- CWE-352 Cross-Site Request Forgery

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

OWASP - A01: Broken access control

Real life example

2015: Laxman Muthiyah - Facebook business pages

- Assign permissions to your own account for a page

```
POST /<page_id>/userpermissions HTTP/1.1
Host: graph.facebook.com
Content-Length: 245
role=MANAGER&user=<target_user_id>
  &business=<associated_business_id>
  &access_token=<application_access_token>
```

https://thezerohack.com/hacking-facebook-pages

OWASP - A01: Broken access control

Prevention

- Principle of least privilege
- Setup CORS correctly
- Log access control failures, alert admins
- Rate limit API
- Invalidate stateful session identifiers on server after logout

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

OWASP - A02: Cryptographic failures

CWEs	Incidence rate	Total occurrences	CVEs
29	46.44%	233 788	3 075

Interestings CWEs:

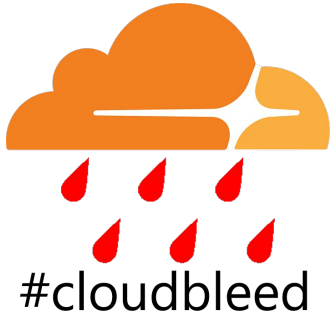
- CWE-259 Use of Hard-coded Password
- CWE-324 Use of a Key Past its Expiration Date
- CWE-327 Broken or Risky Crypto Algorithm
- CWE-760 Use of a One-Way Hash with a Predictable Salt

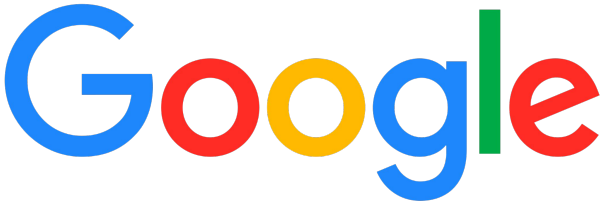
https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

OWASP - A02: Cryptographic failures

Real life example

- Project Zero: Cloudbleed
- Cloudflare
- Assumption of a secure TLS
- Random portion of memory is returned
- Ragel code to parse and modify HTML pages
  - Insert Google analytic tag
  - Rewrite http to https
  - Obfuscate email addresses
  - …





https://bugs.chromium.org/p/project-zero/issues/detail?id=1139
https://blog.cloudflare.com/incident-report-on-memory-leak-caused-by-cloudflare-parser-bug/

OWASP - A02: Cryptographic failures

Real life example

- Ragel generates C code that is compiled
- Buffer overrun

```
/* generated code */
if ( ++p == pe )
    goto _test_eof;
```

Vulnerable code

```
/* generated code */
if ( ++p >= pe )
    goto _test_eof;
```

Solution code

- Incorrect use of ragel
- Returns data in plain text
- Encryption keys, cookies, password, chunks of POST data

https://bugs.chromium.org/p/project-zero/issues/detail?id=1139
https://blog.cloudflare.com/incident-report-on-memory-leak-caused-by-cloudflare-parser-bug/

OWASP - A02: Cryptographic failures

Prevention

- Only store sensitive data when needed -> GDPR
- Encrypt all data at rest
- Force https header (HSTS) (always use forward secrecy)
- Store passwords with salt and hashing function
- Do not use legacy protocols like FTP or SMTP

https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

OWASP - A03: Injection

CWEs	Incidence rate	Total occurrences	CVEs
33	19.09%	274 288	32 078

Interestings CWEs:

- CWE-20 Improper Input Validation
- CWE-79 Improper Neutralization of Input During Web Page Generation (XSS)
- CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

https://owasp.org/Top10/A03_2021-Injection/

OWASP - A03: Injection

~~Real life~~ example

```
queryString=`UPDATE users SET password=${newPassword}
    WHERE userName=${userName}
    AND password=${oldPassword}`
```

- Register as a user with username `admin'---`
- Update password results in the following query

```
UPDATE users SET password=secureNewPassword WHERE
userName=admin-- AND password=somethingrandom
```

OWASP - A03: Injection

Prevention

- Use ORM (Object Relational Mapping) tools
- Positive server side input validation (partial solution)
- Escape characters, using the correct context
- Source code review

https://owasp.org/Top10/A03_2021-Injection/

OWASP - A04: Insecure design

CWEs	Incidence rate	Total occurrences	CVEs
40	24.19%	262 407	2 691

Interestings CWEs:

- CWE-209 Generation of Error Message Containing Sensitive Information
- CWE-256 Unprotected Storage of Credentials
- CWE-501 Trust Boundary Violation
- CWE-522 Insufficiently Protected Credentials

https://owasp.org/Top10/A04_2021-Insecure_Design/

OWASP - A04: Insecure design

Real life example

- Prevented supply of Nvidia GPUs at recommended retail price
- E-commerce sites did not secure against bots that buy up stocks
- Then the cards are resold at cut-throat marked up prices

Scalpers Flip RTX 40 Series Founders Edition GPUs For Fat Profits In China

By Zhiye Liu published June 24, 2023

Scalper pricing is up to 56% higher than MSRP.

Comments (3)

GeForce RTX 40 Series (Image credit: Nvidia)

OWASP - A04: Insecure design

Prevention

- Establish and use a secure development lifecycle
- Establish and use a library of secure design patterns
- Threat modeling should be integrated into refinement sessions
- Write unit and integration tests to validate that all critical flows are resistant to the threat model
- Segregate tenants robustly by design throughout all tiers
- Limit resource consumption by user or service

https://owasp.org/Top10/A04_2021-Insecure_Design/

OWASP - A05: Security misconfiguration

CWEs	Incidence rate	Total occurrences	CVEs
20	19.84%	208 387	789

Interestings CWEs:

- CWE-541 Inclusion of Sensitive Information in an Include File
- CWE-547 Use of Hard-coded, Security-relevant Constants
- CWE-614 Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
- CWE-1004 Sensitive Cookie Without 'HttpOnly' Flag

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

OWASP - A05: Security misconfiguration

Real life example

- Amazon S3 bucket used to store PII (Personal Identifiable Information)
- Public by default

OWASP - A05: Security misconfiguration

Real life example

- Mirai botnet
- Used CCTV cameras, DVRs and routers.
- Trying common passwords

OWASP - A05: Security misconfiguration

Prevention

- Identical config for dev, QA and production
- Remove or uninstall unused features
- Segmented application architecture
- Automated process to verify configurations and settings in all environments

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

OWASP - A06: Vulnerable and outdated components

CWEs	Incidence rate	Total occurrences	CVEs
3	27.96%	30 457	0

Interestings CWEs:

- CWE-1104 Use of Unmaintained Third Party Components
- CWE-1035 2017 Top 10 A9: Using Components with Known Vulnerabilities

https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/

OWASP - A06: Vulnerable and outdated components

Real life example

Log4j2

- Remote Code Execution (RCE) vulnerability reported on 9 December, 2021
- Fixes
  - 2.15.0 06 December, 2021
  - 2.16.0 13 December, 2021
  - 2.17.0 17 December, 2021
  - 2.17.1 27 December, 2021

Vulnerable to DoS attack

https://www.microsoft.com/en-us/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/

OWASP - A06: Vulnerable and outdated components

Prevention

- Remove unused dependencies, unnecessary features, components,...
- Create SBOM + check against the NVD (National vulnerability database)
- Obtain dependencies over secure links
- Monitor for dependencies that lose their maintainers

https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/

OWASP - A07: Identification and authentication failures

CWEs	Incidence rate	Total occurrences	CVEs
22	14.84%	132 195	3 897

Interestings CWEs:

- CWE-290 Authentication Bypass by Spoofing
- CWE-306 Missing Authentication for Critical Function
- CWE-384 Session Fixation
- CWE-798 Use of Hard-coded Credentials

https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

OWASP - A07: Identification and authentication failures

Real life example

- 2012
- Default password set in the authentication layer
- 3.6 million Social Security numbers
- 387 000 credit card numbers

https://www.databreaches.net/sc-3-6-million-social-security-numbers-stolen-from-state-department-of-revenue/

OWASP - A07: Identification and authentication failures

Prevention

- Multi-factor auth prevents credential stuffing, brute force,...
- Do not deploy with any default credentials, particularly for admin users.
- Implement weak password checks, such as testing new or changed passwords against the top 10,000 worst passwords list
  - National Institute of Standards and Technology (NIST) 800-63b's guidelines in section 5.1.1
- Limit or increasingly delay failed login attempts

https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

OWASP - A08: Software and data integrity failures

CWEs	Incidence rate	Total occurrences	CVEs
10	16.67%	47 972	1 152

Interestings CWEs:

- CWE-345 Insufficient Verification of Data Authenticity
- CWE-353 Missing Support for Integrity Check
- CWE-494 Download of Code Without Integrity Check
- CWE-502 Deserialization of Untrusted Data

https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/

OWASP - A08: Software and data integrity failures

Real life example

- Many of which are Fortune 500 clients
- Hackers added malicious code which was undetected
- SolarWinds send out update including malicious code
- Update without signing: Many devices don’t verify update signatures

OWASP - A08: Software and data integrity failures

Prevention

- Use digital signatures to verify the software is from the expected source and has not been altered
- Only consume trusted libraries
- Ensure that a software supply chain security tool, such as OWASP Dependency Check or OWASP CycloneDX, is used to verify that components do not contain known vulnerabilities
- Review process for code and configuration changes

https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/

OWASP - A09: Security logging and monitoring failures

CWEs	Incidence rate	Total occurrences	CVEs
4	19.23%	53 615	242

Interestings CWEs:

- CWE-778 Insufficient Logging
- CWE-117 Improper Output Neutralization for Logs
- CWE-223 Omission of Security-relevant Information
- CWE-532 Insertion of Sensitive Information into Log File

https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/

OWASP - A09: Security logging and monitoring failures

Prevention

- Ensure that you log all login and failed attempts
- Use a log format standard
- Establish an incident response and recovery plan
- Add audit trail for high value transactions

https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/

OWASP - A10: Server side request forgery

CWEs	Incidence rate	Total occurrences	CVEs
1	2.72%	9 503	385

Interestings CWEs:

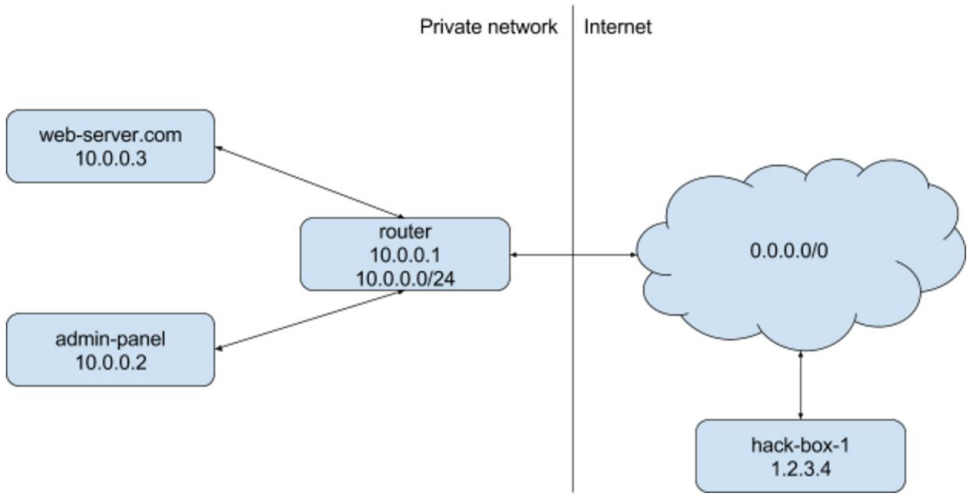
- CWE-918 Server-Side Request Forgery (SSRF)

https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/

OWASP - A10: Server side request forgery

Real life example

- Attacker gained access to a set of AWS access keys via SSRF
- App was behind a WAF



https://blog.appsecco.com/an-ssrf-privileged-aws-keys-and-the-capital-one-breach-4c3c2cded3af
https://www.capitalone.com/digital/facts2019/

OWASP - A10: Server side request forgery

Prevention

Application layer

- Sanitize and validate all input data
- Disable HTTP redirect
- Enforce origin with a positive allow list

Network layer

- Segment remote resource access in separate networks
- Enforce ‘deny by default’ firewall rules

https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/

OWASP Top 10 API

OWASP top 10 api

1. Broken object level authorization
2. Broken authentication
3. Broken object property level authorization
4. Unrestricted resource consumption
5. Broken function level authorization
6. Unrestricted access to sensitive business flows
7. Server side request forgery
8. Security misconfiguration
9. Improper inventory management
10. Unsafe consumption of APIs

https://owasp.org/API-Security/editions/2023/en/0x11-t10/

Key takeaways

Key takeaways

There are a lot of attack scenarios

A small mistake can lead to large hacks

Further reading

OWASP, **2021 owasp top 10** (https://owasp.org/Top10/A00_2021_Introduction)