OWASP

Open Web Application Security Project

OWASP Top 10 2017

Daifallah Alotaibi

Software Consultant

What is OWASP?

- Open Web Application Security Project
- Awareness project and not stander
- Released 2003, 2004, 2007, 2010, 2013, 2017RC
- There are more than 10 ...

OWASP Top 10, (2013 - 2017)

OWASP Top 10 – 2013 (Previous)	OWASP Top 10 – 2017 (New)
A1 – Injection	A1 – Injection
A2 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References - Merged with A7	A4 – Broken Access Control (Original category in 2003/2004)
A5 – Security Misconfiguration	A5 – Security Misconfiguration
A6 – Sensitive Data Exposure	A6 – Sensitive Data Exposure
A7 – Missing Function Level Access Control - Merged with A4	A7 – Insufficient Attack Protection (NEW)
A8 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Components with Known Vulnerabilities	A9 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards - Dropped	A10 – Underprotected APIs (NEW)

A1 — Injection

Description:

Sending the user input as is the interpreter.

- SQL
- LDAP
- Regex
- OS command

Defenses:

- Prepared Statements
- White List Input Validation
- Escaping Input
- Use ORM , Frameworks

A2 – Broken Authentication and Session Management

Description:

Allowing attackers to compromise passwords, keys, or session tokens.

- Session IDs in the URL
- Session without timeout or long
- Change password without old one
- Login with national id and mobile no

Defenses:

- Reduce timeout
- Regenerate Session IDs
- Check user-agent
- Check user IP

A3 – Cross-Site Scripting (XSS)

Description:

Allow attacker to inject some code into your page.

- Java script code
- HTML
- SVG image support embed js
- <object> tag flash , java, ocx
- Unsafe link which do some action

Defenses:

- Escaping input JS, HTML
- Remove tag attributes
- Remove URLs
- Avoid including, loading untrusted data to your page

A4-Broken Access Control (NEW)

Description:

Access unauthorized functionality or data.

- Access to Admin page/functionality/ data
- View sensitive files
- Modify other users' data
- Change access rights
- Show information by user_id or account_no (use Index)

Defenses:

- Check access everywhere
- Use per user or session indirect object references
- Use Index instead of id

A5-Security Misconfiguration (Admin)

Description:

Harden your server.

- Default configuration
- Default software phpmyadmin
- Weak password
- Open ports
- Un-updated packages

Defenses:

- Change default configuration
- Remove default software
- Harden OS, Web server, PHP
- Run each software with different user
- Close un-needed port
- Keep system updated

A6-Sensitive Data Exposure

Description:

Attacker can access sensitive data.

- Data transmitted in clear text
- Data stored in clear text
- No encryption
- No Hashing
- Weak or unsalted hashes (MD5)
- HTTP

Defenses:

- Use HTTPS
- Don’t store unnecessarily sensitive data
- Encryption sensitive data
- Strong Hashing for password bcrypt
- Disable autocomplete on forms
- Disable caching

A7-Insufficient Attack Protection (NEW)

Description:

Application cannot detect, prevent, and respond to both manual and automated attacks.

- Unlimited user login failure
- Too many request
- No logs
-

Defenses:

- Limit login failures
- Don’t store unnecessarily sensitive data
- Encryption sensitive data
- Strong Hashing for password bcrypt
- Disable autocomplete on forms
- Disable caching
- Use captcha

A8 – Cross-Site Request Forgery (CSRF)

Description:

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request

- Using GET method to do action
- Accept request from other site

Defenses:

- Use CSRF token
- Use POST method
- Regenerate CSRF token

A9-Using Components with Known Vulnerabilities

Description:

Using components with known Vulnerabilities.

- Old frameworks
- Old libraries
- Discontinue Project

Defenses:

- Update your framework
- Update composer packages
- Use less Components
- Use new libraries

A10 – Underprotected APIs

Description:

Using components with known Vulnerabilities.

- Using Token to access API
- Using username and passwordd
- Discontinue Project

Defenses:

- Use session
- Use HTTPS (ssl pining)
- Escape input
- Apply the TOP 10 to API

Thank you

Daifallah Alotaibi

@github.com/daif
@twitter.com/daif
@+966556639884

9 September 2017