OWASP Top 10

Name: Ammar Yasser Mohamed
Enterprise Security Department - Summer 2025

Date: August 3, 2025

Overview of the OWASP Top 10

The OWASP Top 10 is a globally recognized awareness document for web application security. It represents a broad consensus about the most critical security risks to web applications, based on data collected from hundreds of organizations and thousands of applications. The list helps developers and security teams understand and mitigate these common threats.

OWASP Top 10 - 2021 Categories

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 Vulnerable and Outdated Components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 (SSRF)

Focus on A02:2021 - Cryptographic Failures

Cryptographic Failures, previously referred to as "Sensitive Data Exposure," focuses on the lack or misimplementation of cryptographic protections. These failures often result in the unintentional exposure of sensitive data and arise from the incorrect use of encryption, poor key management, or insecure protocols.

Common Issues Leading to Cryptographic Failures

- Use of outdated or broken algorithms (e.g., MD5, SHA1).
- Transmission of sensitive data over insecure channels (e.g., HTTP).
- Hard-coded passwords or cryptographic keys.
- Insecure initialization vectors (IVs) or modes (e.g., ECB).
- Lack of authenticated encryption.
- Poor random number generation or predictable seeds.
- Deprecated padding schemes (e.g., PKCS #1 v1.5).
- Storage of passwords without key derivation functions.

What is an Initialization Vector (IV)?

An **Initialization Vector (IV)** is like a *random starting point* used when encrypting data. Its purpose is to make the same plaintext encrypted in different ways, **even if the same key is used**.

Example: Imagine you encrypt the word `"password"` 100 times:

- If IVs are used properly, you'll get 100 **different** ciphertexts.
- If IVs are missing or reused, you might get the **same** ciphertext each time, revealing patterns.

Why ECB Mode is Dangerous:
ECB (Electronic Code Book) does **not** use IVs. This means it encrypts identical blocks into identical ciphertexts, which can expose patterns — especially in structured data like images or repetitive text.

Safer Alternative: Use secure modes like `CBC`, `GCM`, etc., with a **unique random IV for each encryption**.

What is padding schemes and PKCS #1 v1.5?

Real-Life Analogy:
Imagine you're locking messages in a box (encryption), but the box must always be full.
PKCS #1 v1.5 just stuffs the box with predictable filler—easy to guess. OAEP mixes up the filler each time—unpredictable and secure.

Imagine a vending machine that only accepts exactly $1.00 in coins.
If you insert 50 cents, it won't work. You must "pad" with more coins to make it work. RSA is the same—the "machine" needs a full box (fixed-size input), and padding ensures this is done safely.

Scenario 1: No TLS Enforcement

A website supports both HTTP and HTTPS. An attacker uses a public Wi-Fi network to intercept requests, downgrade connections to HTTP, and steal session cookies, gaining unauthorized access to user accounts.

Scenario 2: Weak Password Hashing

A password database uses unsalted SHA-1 hashes. An attacker exploits a file upload vulnerability to extract the database and uses a rainbow table to crack all passwords efficiently.

How to Prevent Cryptographic Failures

- Classify sensitive data and apply required controls (e.g., PCI DSS, GDPR).
- Avoid storing sensitive data unless necessary; discard or tokenize it.
- Encrypt sensitive data at rest and in transit using strong algorithms.
- Use modern protocols (e.g., TLS 1.3 "Transport Layer Security protocol") with forward secrecy and secure settings.
- Enforce HTTPS with HSTS headers. "HTTP Strict Transport Security" header is a security mechanism that tells web browsers to only access a website using HTTPS, automatically converting any HTTP requests to HTTPS.
- Store passwords with strong, salted, adaptive hash functions (e.g., bcrypt, Argon2).
- Use CSPRNGs for generating keys and IVs. "Cryptographically Secure Pseudo-Random Number Generator" is a special type of pseudo-random number generator designed to be suitable for use in cryptography.
- Avoid hardcoded credentials and keys; use secure key management.
- Regularly audit cryptographic configurations and libraries.

Mapped CWE Examples "Common Weakness Enumeration"

- CWE-259: Use of Hard-coded Password
- CWE-327: Use of Broken or Risky Cryptographic Algorithm
- CWE-331: Insufficient Entropy
- CWE-321: Use of Hard-coded Cryptographic Key
- CWE-329: Not Using a Random IV with CBC Mode
- CWE-759: Use of One-Way Hash without a Salt