OWASP Top 10 Cheatsheet

Cryptographic failures

#02

What are they?

Cryptographic failures occur when **encryption, hashing, or other cryptographic methods are misused**, typically leading to sensitive data exposure. These failures **undermine the CIA trifecta** (confidentiality, integrity, and authenticity), making it easier for attackers to steal, alter, or forge data. Also, failing to encrypt sensitive data **can result in compliance violations** under GDPR, PCI DSS, and HIPAA, leading to legal and financial consequences.

There are a **lot of possible causes** (too many to cover here), but common ones include using weak encryption algorithms, reusing or hardcoding cryptographic keys, transmitting sensitive data in plaintext (yep, it still happens), and using insecure encryption modes.

Cryptographic failures aren't just a hypothetical risk - **real-world breaches have exposed billions of records** due to poor encryption practices.

How are they exploited?

Attackers can take advantage of weak cryptography in several ways, including:

- **Intercepting plaintext data:** When sensitive information (e.g. passwords, credit card numbers) is stored or transmitted unencrypted, attackers can steal it in transit, for example by using network sniffing tools.
- **Breaking weak encryption:** Algorithms like MD5 and SHA-1 are vulnerable to collisions, making them easy to crack.
- **Exploiting key mismanagement:** Reusing encryption keys or failing to rotate them makes systems predictable and easier to compromise.
- **Forging digital signatures:** Weak hashing allows attackers to manipulate data while maintaining a valid signature.

Real-world examples

Examples of these in the real-world are many and various. In 2014, an OpenSSL flaw known as **Heartbleed** exposed encryption keys, login credentials, and sensitive data. That followed **Adobe's** exposure of 153 million accounts in 2013 due to improper encryption of customer credentials. More recently, **Facebook** were found to be storing millions of passwords in plain text, accessible to internal employees.

How do you prevent it?

Taking a **Secure by Design approach** you can ensure cryptographic risks are mitigated before attackers can exploit them. Here's a few top tips:

- **Use strong encryption:** Implement AES-256 for data at rest and enforce TLS 1.2/1.3 for data in transit.
- **Enforce proper key management:** Store encryption keys in secure hardware or dedicated key management systems.
- **Use modern password hashing:** Replace SHA-1 with Argon2, bcrypt, or PBKDF2 with salting and stretching.
- **Secure session management:** Implement MFA and short-lived session tokens.
- **Enforce HTTPS and HSTS:** Prevent protocol downgrade attacks by ensuring secure communication.
- **Regularly audit cryptographic implementations:** Conduct penetration testing and automated security scans.