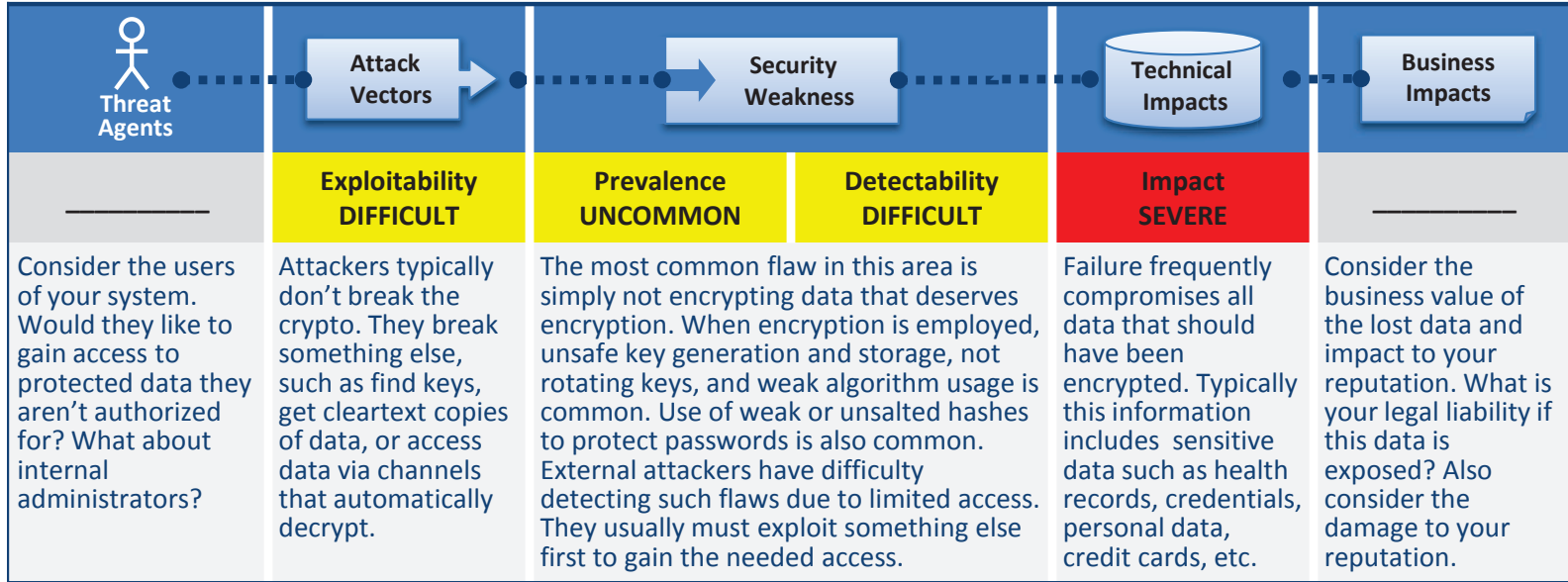


Insecure Cryptographic Storage



Am I Vulnerable?

The first thing you have to determine is which data is sensitive enough to require encryption. For example, passwords, credit cards, health records, and personal information should be encrypted. For all such data, ensure:

1. It is encrypted everywhere it is stored long term, particularly in backups of this data.
2. Only authorized users can access decrypted copies of the data (i.e., access control – See A4 and A8).
3. A strong standard encryption algorithm is used.
4. A strong key is generated, protected from unauthorized access, and key change is planned for.

And more ... For a more complete set of problems to avoid, see the [ASVS requirements on Cryptography \(V7\)](#)

How Do I Prevent This?

The full perils of unsafe cryptography are well beyond the scope of this Top 10. That said, for all sensitive data deserving encryption, do all of the following, at a minimum:

1. Considering the threats you plan to protect this data from (e.g., insider attack, external user), make sure you encrypt all such data at rest in a manner that defends against these threats.
2. Ensure offsite backups are encrypted, but the keys are managed and backed up separately.
3. Ensure appropriate strong standard algorithms and strong keys are used, and key management is in place.
4. Ensure passwords are hashed with a strong standard algorithm and an appropriate salt is used.
5. Ensure all keys and passwords are protected from unauthorized access.

Example Attack Scenarios

Scenario #1: An application encrypts credit cards in a database to prevent exposure to end users. However, the database is set to automatically decrypt queries against the credit card columns, allowing an SQL injection flaw to retrieve all the credit cards in cleartext. The system should have been configured to allow only back end applications to decrypt them, not the front end web application.

Scenario #2: A backup tape is made of encrypted health records, but the encryption key is on the same backup. The tape never arrives at the backup center.

Scenario #3: The password database uses unsalted hashes to store everyone's passwords. A file upload flaw allows an attacker to retrieve the password file. All the unsalted hashes can be brute forced in 4 weeks, while properly salted hashes would have taken over 3000 years.

References

OWASP

For a more complete set of requirements and problems to avoid in this area, see the [ASVS requirements on Cryptography \(V7\)](#).

- [OWASP Top 10-2007 on Insecure Cryptographic Storage](#)
- [ESAPI Encryptor API](#)
- [OWASP Development Guide: Chapter on Cryptography](#)
- [OWASP Code Review Guide: Chapter on Cryptography](#)

External

- [CWE Entry 310 on Cryptographic Issues](#)
- [CWE Entry 312 on Cleartext Storage of Sensitive Information](#)
- [CWE Entry 326 on Weak Encryption](#)