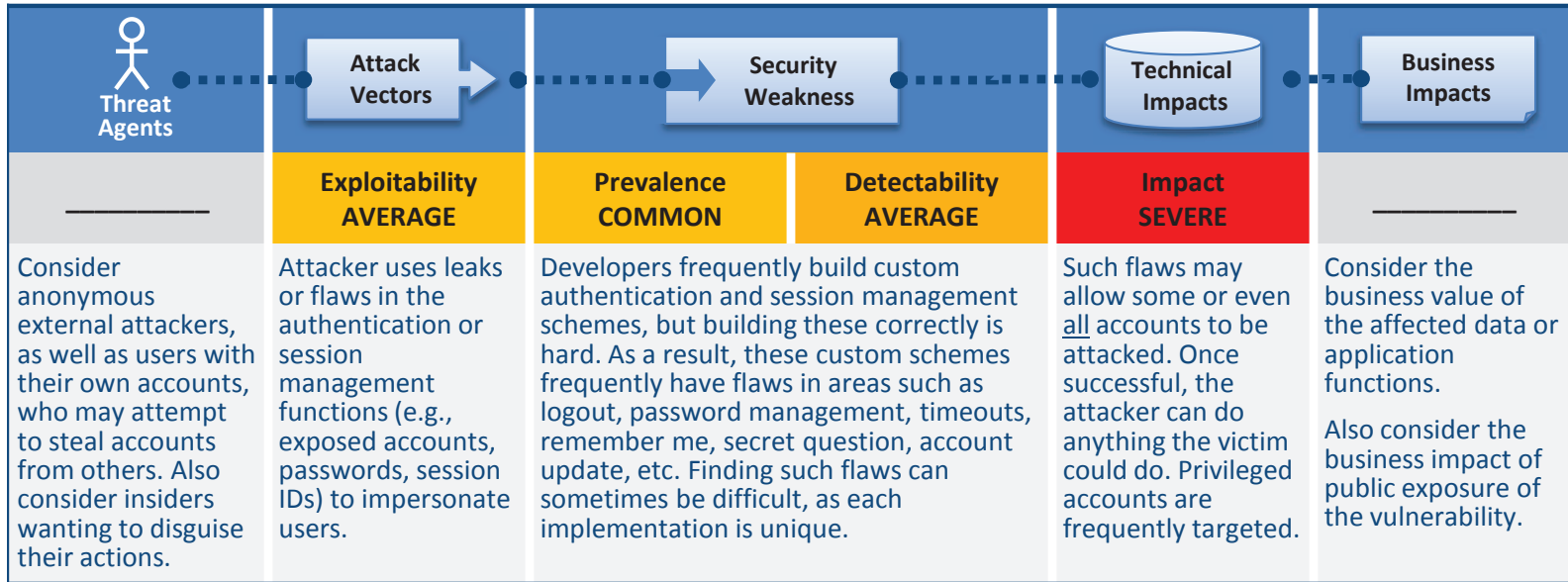


A3

Broken Authentication and Session Management



Am I Vulnerable?

The primary assets to protect are credentials and session IDs.

1. Are credentials always protected when stored using hashing or encryption? See A7.
2. Can credentials be guessed or overwritten through weak account management functions (e.g., account creation, change password, recover password, weak session IDs)?
3. Are session IDs exposed in the URL (e.g., URL rewriting)?
4. Are session IDs vulnerable to session fixation attacks?
5. Do session IDs timeout and can users log out?
6. Are session IDs rotated after successful login?
7. Are passwords, session IDs, and other credentials sent only over TLS connections? See A9.

See the [ASVS](#) requirement areas V2 and V3 for more details.

How Do I Prevent This?

The primary recommendation for an organization is to make available to developers:

1. **A single set of strong authentication and session management controls.** Such controls should strive to:
 - a) meet all the authentication and session management requirements defined in OWASP's [Application Security Verification Standard \(ASVS\)](#) areas V2 (Authentication) and V3 (Session Management).
 - b) have a simple interface for developers. Consider the [ESAPI Authenticator and User APIs](#) as good examples to emulate, use, or build upon.
2. Strong efforts should also be made to avoid XSS flaws which can be used to steal session IDs. See A2.

Example Attack Scenarios

Scenario #1: Airline reservations application supports URL rewriting, putting session IDs in the URL:

<http://example.com/sale/saleitems;jsessionid=2POOC2JDPXM0OQSNDLPSKHJCJUN2JV?dest=Hawaii>

An authenticated user of the site wants to let his friends know about the sale. He e-mails the above link without knowing he is also giving away his session ID. When his friends use the link they will use his session and credit card.

Scenario #2: Application's timeouts aren't set properly. User uses a public computer to access site. Instead of selecting "logout" the user simply closes the browser tab and walks away. Attacker uses the same browser an hour later, and that browser is still authenticated.

Scenario #3: Insider or external attacker gains access to the system's password database. User passwords are not encrypted, exposing every users' password to the attacker.

References

OWASP

For a more complete set of requirements and problems to avoid in this area, see the [ASVS requirements areas for Authentication \(V2\) and Session Management \(V3\)](#).

- [OWASP Authentication Cheat Sheet](#)
- [ESAPI Authenticator API](#)
- [ESAPI User API](#)
- [OWASP Development Guide: Chapter on Authentication](#)
- [OWASP Testing Guide: Chapter on Authentication](#)

External

- [CWE Entry 287 on Improper Authentication](#)