**Root Cause and solution of Session Management and Broken Authentication Vulnerabilities**

**WHAT:** This document is based on the work of Daniel Huluka and Oliver Popov (2012)[[1]](#footnote-1). The main points of their article to identify and combat these vulnerabilities a listed below.

**HOW TO READ THIS DOCUMENT:** Firstly, the causes and solutions of session management are listed. The causes are further dived into hijacking[[2]](#footnote-2) and session fixation[[3]](#footnote-3). Secondly the causes and solutions of broken authentication are listed.

**WHO SHOULD READ THIS DOCUMENT:** This document is created for the person that is responsible for implementing and/or testing fixes for these vulnerabilities.

***Root causes of Session Management Vulnerabilities and Suggested Solutions***

**Root Causes of Session Hijacking**

1. Usage of guessable session ID.

2. Absence of detection mechanism for “repeated guessing trial” either with brute-force or systematic methods.

3. Unable to detect repeated guessing trials while there is a mechanism in place.

4. Weak cryptography: a weakness in the cryptography algorithm or a weakness in the way a strong cryptographic algorithm is used.

5. Limitation of HTTP: the statelessness of the protocol or lack of any inherent or integrated state management mechanism.

6. Insecure session handling methods.

7. Solution misuse: the misconfiguration or improper use of basically strong solutions

8. Weakness in the Inactive session management technique.

**Root Causes of Session Fixation**

1. Permissive Server: a server that accepts client generated session IDs

2. Session management type in use.

3. Reuse of session identifiers: generating same session identifiers twice or more for different sessions of the same or different clients.

***Root Causes of Broken Authentication Vulnerabilities and Suggested Solutions***

1. Lack of metrics: absence of well-developed metrics that can assist in making the right decision in the selection of security mechanisms.

2. Lack of security knowledge among programmers to apply information and communication security mechanisms to their solutions.

3. Wrong decisions or compromises: both designers and programmers are prone to wrong decisions due to lack of metrics and security knowledge.

4. Use of self-developed modules instead of well tested and thoroughly analyzed modules for security services such as authentication.

5. Storing user credentials with other application data.

6. Guessing Attempts: allowing repeated guessing attempts.

7. Level of user data in the system: the level of information the system knows/holds about users.

8. Lack of security awareness among users.

9. Stringent requirements set to strengthen security might be unrealistic and very difficult to be meet by users.

1. Huluka, D., & Popov, O. (2012, June). Root cause analysis of session management and broken authentication vulnerabilities. In *Internet Security (WorldCIS), 2012 World Congress on* (pp. 82-86). IEEE. [↑](#footnote-ref-1)
2. Hijacking sometimes also known as cookie hijacking is the exploitation of a valid [computer session](https://en.wikipedia.org/wiki/Session_(computer_science))—sometimes also called a *session key*—to gain unauthorized access to information or services in a computer system. (source: Wikipedia) [↑](#footnote-ref-2)
3. Session Fixation is specific hijacking attack that permits an attacker to hijack a valid user session. (source: <https://www.owasp.org/index.php/Session_fixation>) [↑](#footnote-ref-3)