Tutorial No. 6

Title: Design and implementation flaws in authentication

Roll No.: 16010423076 **Tutorial No.:6**

Aim: Design and implementation flaws in authentication

Resources : Internet connection, Research papers and articles on authentication flaws

---

Theory:

Design Flaws in Authentication Mechanisms

a. Bad Passwords

Bad passwords are weak, easily guessable, or reused across multiple accounts. Common examples include "123456", "password", and easily guessed personal information. These flaws arise when users are not encouraged or required to create strong, unique passwords.

Example:
In the 2012 LinkedIn data breach, millions of accounts were compromised due to weak and reused passwords. Attackers exploited these flaws to gain unauthorized access to sensitive information.

b. Password Change Functionality

Insecure password change processes can lead to account compromise. A common flaw is not verifying the user's identity adequately during a password reset, allowing attackers to hijack accounts.

Example:
In some websites, a password change request only requires knowing the username and answering easily guessable security questions, making it prone to attacks.

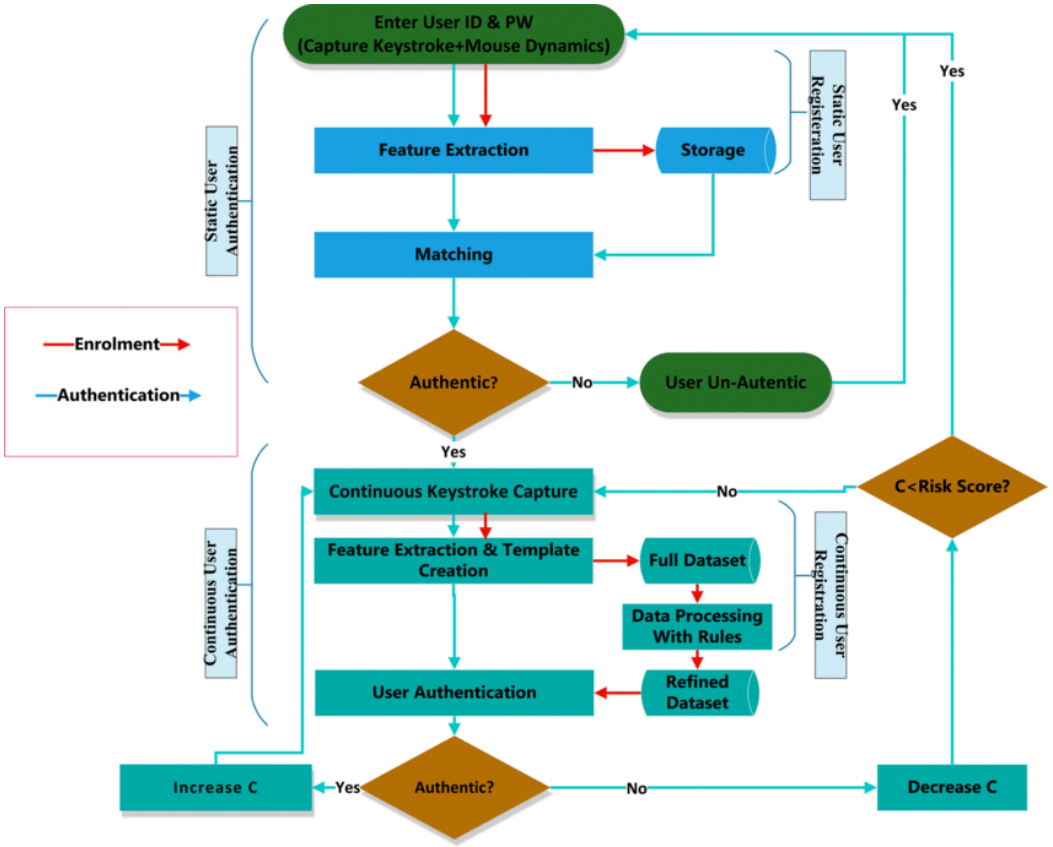
Implementation Flaws in Authentication

Defects in Multistage Login Mechanisms

Multistage login mechanisms, like multi-factor authentication (MFA), are designed to enhance security. However, flaws in implementation can render them ineffective.

Example:
A common issue is when a system does not properly validate MFA tokens during login, allowing attackers to bypass the second authentication stage. In 2021, some MFA systems were exploited by attackers who intercepted SMS-based tokens

Authentication Flow Diagram :



The above diagram illustrates a two-step authentication mechanism that leverages keystroke and mouse dynamics to enhance security. The process begins with the user entering their ID and password while the system captures their typing and mouse movement patterns.

1. **Static User Authentication:**
   - During enrollment, features such as keystroke timing and mouse dynamics are extracted and stored.
   - The system matches the input patterns with the stored data.
   - If the authentication is successful, the user gains access; otherwise, they are marked as unauthentic.

2. **Continuous User Authentication:**

- Once authenticated, the system continuously monitors the user's keystrokes to detect anomalies.
- If the risk score crosses a set threshold, indicating suspicious behavior, the authentication status is re-evaluated.
- The system refines data using processing rules to improve accuracy.
- The user's risk score is adjusted based on the consistency of their typing and mouse dynamics.

This dynamic authentication model aims to reduce the risks associated with static password-based authentication by continuously validating user behavior.

Outcomes: CO1: Realize the premise of vulnerability analysis and penetration testing (VAPT).

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

From this tutorial, I learned that authentication flaws can arise both from poor design choices and faulty implementation. Addressing these flaws requires strong password policies, secure password change processes, and robust multi-factor authentication mechanisms. Implementing best practices can significantly reduce the risk of unauthorized access.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

REFERENCES:

https://owasp.org - OWASP Authentication Cheat Sheet

https://www.ncsc.gov.uk - National Cyber Security Centre Guidance

https://portswigger.net - Web Security Academy

https://auth0.com/blog - Auth0 Authentication Best Practices