Brute-forcing a stay-logged-in cookie in an E-commerce webapp

Ref: CVE-2025-48461

Executive Summary

This report outlined the findings of a security assessment conducted on the session management functionality of the target web application. A vulnerability was discovered in the "stay-logged-in" cookie mechanism, which is susceptible to brute-force attacks. By exploiting predictable cookie structure and hashing behaviour, an attacker can impersonate another user and gain unauthorized access. The severity of this vulnerability is **high**, as it compromises authentication and session integrity.

Introduction

The purpose of this security assessment was to identify weaknesses in the application's session persistence mechanism, specifically focusing on how the “stay logged in” functionality is implemented and whether it can be abused for unauthorized access.

Methodology

1. Logged in as a normal user with the "Stay logged in" option enabled.
2. Analyzed the `stay-logged-in` cookie, which was Base64-encoded.
3. Decoded the cookie to reveal the format: ie.`username:md5(password)`.
4. Confirmed this by hashing the known password using MD5 and reconstructing the cookie.
5. Used Burp Suite Intruder with payload processing to:
   - Hash candidate passwords using MD5.
   - Prefix the hash with `user` and Base64-encode it.
6. Replaced the cookie and request parameter with the victim's username.
7. Detected successful brute-force attempt by grepping for the presence of `Update email` in the HTTP response.

Vulnerability Findings:

- **Type:** Insecure Authentication Token / Predictable Session Cookie

- **Location:** `stay-logged-in` cookie

- **Severity:** High

Description

The application implemented a persistent login mechanism using a cookie formatted as `base64(username:md5(password))`. This format was predictable and weak, as it used a known username and a standard MD5 hash of the password. An attacker could brute-force the password offline using a list of common passwords, then reconstruct a valid cookie to impersonate any user.

Proof of Concept (PoC)

1. Base64 ‘username’

2. Inserted as the value for the `stay-logged-in` cookie in a GET request to `/my-account?id=’username`

3. Server response included the `Update email` button, confirming successful login as ‘username’.

Impact Assessment

This vulnerability allows an attacker to bypass authentication controls and hijack user accounts without interacting with the login form. The exposure of the MD5 hashing structure and Base64-encoding format creates an easily exploitable path for unauthorized access, leading to privacy breaches and potential privilege escalation.

Recommendations:

1. Avoid predictable cookie formats and never store user credentials or hashes in client-side tokens.

2. Use secure, random session tokens for persistent login functionality.

3. Store authentication data server-side and validate it securely on each request.

4. Implement rate limiting and monitoring for suspicious session usage patterns.

5. Deprecate the use of weak hashing algorithms such as MD5 in favour of strong, salted password hashing algorithms like bcrypt or Argon2.

Conclusion:

The discovery of a brute-forceable authentication cookie highlighted the need for strong cryptographic practices and secure session handling. Immediate remediation is necessary to protect user accounts and ensure the integrity of the authentication process.