**Section 4: ASVS Web Application Security Auditing**

|  |  |
| --- | --- |
| **V3: Session Management** | |
| **Overall Maturity Level (L1, L2, L3)** | L1 |
| **Justification: This application has some security measures taken into account, but the session token handling is frankly juvenile, and needs quite a bit of maturing before this application should be used in production.** | |
| **Criteria:** Verify that the application never reveals session tokens in URL parameters | **Status:**  Fail |
| Session tokens appear in GET and POST parameters on the login page. | |
| **Criteria:** Verify that the application generates a new session token on user authentication | **Status:**  Fail |
| Session tokens generated by hashing the current time. | |
| **Criteria:** Verify that session tokens possess at least 64 bits of entropy | **Status:**  Fail |
| Session tokens are generated by hashing the current time. | |
| **Criteria:** Verify the application stores session tokens in the browser using secure methods such as appropriately secured cookies or HTML 5 session storage | **Status:**  Fail |
| Session tokens are not securely stored. | |
| **Criteria:** Verify that the session tokens are generated using approved cryptographic algorithms | **Status:**  Fail |
| Generated by hashing the current time. | |
| **Criteria:** Verify that logout and expiration invalidate the session token, such that the back button or a downstream relying party does not resume an authenticated session, including across relying parties. | **Status:**  Fail |
| Session tokens can be reused by the user if they know how to manipulate session tokens. | |
| **Criteria:** If authenticators permit users to remain logged in, verify that re-authentication occurs periodically both when actively used or after an idle period | **Status:**  Pass |
| Re-authentication occurs every ~5 minutes when a new page is loaded. | |
| **Criteria:** Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. | **Status:**  Fail |
| Application does not provide an option to revoke all active session tokens. | |
| **Criteria:** Verify that cookie-based session tokens have the 'Secure' attribute set. | **Status:**  Fail |
| Session tokens are not cookie-based, and no cookies have this parameter set. | |
| **Criteria:** Verify that cookie-based session tokens have the 'HttpOnly' attribute set. | **Status:**  Fail |
| Session tokens are not stored with any attributes. OWASP ZAP found this. | |
| **Criteria:** Verify that cookie-based session tokens utilize the 'SameSite' attribute to limit exposure to cross-site request forgery attacks. | **Status:**  Fail |
| Session tokens do not use this attribute, and CSRF is possible. | |
| **Criteria:** Verify that cookie-based session tokens use the "\_\_Host-" prefix so cookies are only sent to the host that initially set the cookie. | **Status:**  Fail |
| Cookies do not use any prefixes. | |
| **Criteria:** Verify that if the application is published under a domain name with other applications that set or use session cookies that might disclose the session cookies, set the path attribute in cookie-based session tokens using the most precise path possible. | **Status:**  N/A |
| The application is not published with any others. | |
| **Criteria:** Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. | **Status:**  N/A |
| OAuth cookies are not used in this application and the application does not use trust relationships. | |
| **Criteria:** Verify the application uses session tokens rather than static API secrets and keys, except with legacy implementations. | **Status:**  Pass |
| The application uses a session token based on the current time which is constantly changing. | |
| **Criteria:** Verify that stateless session tokens use digital signatures, encryption, and other countermeasures to protect against tampering, enveloping, replay, null cipher, and key substitution attacks. | **Status:**  Fail |
| Session tokens are unencrypted plaintext and not protected or verified. Session token manipulation is very possible on the login page. | |
| **Criteria:** Verify that Relying Parties (RPs) specify the maximum authentication time to Credential Service Providers (CSPs) and that CSPs re-authenticate the user if they haven't used a session within that period. | **Status:**  N/A |
| CSPs and RPs are not used in this application. | |
| **Criteria:** Verify that Credential Service Providers (CSPs) inform Relying Parties (RPs) of the last authentication event, to allow RPs to determine if they need to re-authenticate the user. | **Status:** N/A |
| CSPs and RPs are not used in this application. | |
| **Criteria:** Verify the application ensures a full, valid login session or requires re-authentication or secondary verification before allowing any sensitive transactions or account modifications. | **Status:**  Unknown |
| The application does not allow password changes. | |