**Attacks:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | ID | Title | Vulnerability | Comment | Example Tests |
| 1 | ST0001\_SQLInjection | Verify the Web application behavior on injecting SQL scripts | SQL Injection | Verify whether SQL injection attack is possible through URLs and Text fields | <https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OWASP-DV-005)> |
| 2. | ST0002\_NoSQLInjection | Check the vulnerabilities in MongoDB by NoSQL injection | No SQL Injection | Verify whether NoSQL injection attack is possible | <https://www.owasp.org/index.php/Testing_for_NoSQL_injection>  Check for vulnerabilities in MongoDB by changing the data in WHERE clause |
| 3. | ST0003\_HTMLInjection | Verify the Web application behavior on injecting HTML scripts | HTML Injection | Verify whether HTML injection attack is possible through URLs and Text fields | <https://www.owasp.org/index.php/HTML_Injection> |
| 4. | ST0004\_CrossSiteScripting | Check the  System behavior if a user provides  scripts as input especially javascript) | Cross site Scripting | Verify whether providing input such as Javascript changes application behavior | <https://www.owasp.org/index.php/Testing_for_Cross_site_scripting> |
| 5. | ST0005\_XSSInjection | Verify the Web application  behavior on injecting  xss scripts. | XSS Scripting | Verify XSS from  Text-fields,  URLs,  Cookies | <https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)>  Using script tags try to read user session or cookie information  and use that using  the web site URL. |
| 6. | ST0006\_BrokenAuthenticationAndSessionManagement | Verify the application  behavior if a user forms a URL with  session id and tries to access it. | Broken Authentication And Session Management(BASM) | Verify BASM from  Different page  URLs | <https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management> |
| 7. | ST0007\_ BrokenAuthenticationAndSessionManagement | Check whether the application  exposes  session ids of user or not. | Broken Authentication And Session Management(BASM) | Verify BASM by monitoring the Http traffic using Fiddler | <https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management> |
| 8. | ST0008\_ BrokenAuthenticationAndSessionManagement | Verify application  Session expires after  certain time interval if user closes the browser | Broken Authentication And Session Management(BASM) | Verify BASM by  Signing in to Channels application and closing the browser and then again the starting session in the same  browser. | <https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management> |
| 9. | ST0009\_InsecureDirectObject | Verify application  behavior if user logs in with one  credentials and provides  url parameter for other user. | Insecure Direct Obje  ct References (IDOR) | Verify IDOR by  Changing the URL parameter value and  providing someone else's parameter value | <https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References> |
| 10. | ST0010\_InsecureDirectObject | Verify IDOR by accessing the URL which user does  not have an access to. | Insecure Direct Obje  ct References (IDOR) | Verify IDOR by accessing the URL  Which user does  not have an access to. | <https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References> |
| 11. | ST0011\_CrossSiteForgery | Verify if the application sends session cookies to  other site if user clicks on any link | Cross Site Forgery  (CRF) | Verify CSRF by injecting a link to third party page and  trying to access the session using the session id | <https://www.owasp.org/index.php/CrossSite_Request_Forgery_(CSRF)> |
| 12. | ST0012\_SecurityMisconfiguration | Verify web server/DB server machines  are updated with latest security patches | Security Misconfiguration | Verify Security  Misconfiguration by signing on to the web server and checking for OS/ software  updates | <https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration> |
| 13. | ST0013\_FailureToRestrictURLAccess | Verify application  behavior if a user  tries to directly access the URL which needs  user login | Failure to restrict UR  L access | Verify URL Access  by forming the URLs which need logging in and hiting them into  browser directly. | <https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References> |
| 14. | ST0014\_CredentialsTransported Over AnEncrypted Channel | Verify if user  credential's are transmitted over https while signing in to the system | Credentials  Transported over an  Encrypted Channel  (CTOEC) | Verify CTOEC by reading the https request using fiddler  and checking if there is any user data in not  encrypted format | Verify CTOEC by  reading the https  request using fiddler and checking if there  is any user data in  not encrypted format |
| 15. | ST0015\_DenialOfService | Verify dashboard,  Web UI and  Contribution Service  behavior on DOS  attack. | Denial Of Service |  |  |
| 16. | ST0016\_BannerDisclosure | Verify whether the server or the application does not disclose its type via the HTTP response headers | Banner Disclosure | Verify whether the server or the application does not disclose its type via the HTTP response headers |  |
| 17. | ST0017\_ErrorMessagesAndStackTraces | Verify that the error messages/stack traces do not show any sensitive and important information | Error messages and stack traces | Verify that the error messages/stack traces do not show any sensitive and important information | Verify all the stack traces generated |