**Mind Alliance Web Security Testing:**

**Scope/Goals:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Interfaces | | |  |
| Vulnerability ID | Vulnerabilities | Front End  Web Application | Web UI  Web  Application | Web Application  Back End | Comments |
| 1 | SQL Injection  (Check whether the application is vulnerable to SQL injection attacks when user provides SQL/HTML directives as input | NA | NA | NA | Will not be applicable since the database used is MongoDB |
| 2 | HTML Tags Injection-  Simply use HTML to modify the page for malicious reasons  (Check whether the application is vulnerable to HTML injection attacks when user provides SQL/HTML directives as input) | NA | YES | NA |  |
| 3. | NoSQL Injection(Check for NoSqL injection as SQL has not been used as backend) | NA | YES | YES | Will be applicable as the database used is mongoDB |
| 4. | Cross site scripting (Check the  System behavior  if a user provides  scripts as input especially javascript) | NA | YES | NA |  |
| 5. | Broken Authentication And Session Manageme  nt(Checking if the  system exposes the session ids or user information in browser) |  | YES | NA |  |
| 6. | Insecure direct object references  (Check for  unauthorized or  broken access  management. Ex- By mistake developer has exposed an object which should  not be exposed  without proper  access validation) | NA | YES |  |  |
| 7. | Denial of Service attacks(Check whether any kind of denial of service attacks does not disrupt the normal functioning of the system) |  | YES |  |  |
| 8. | Cross site request  forgery(Sending  forged/corrupted  http requests  from user's browser to the vulnerable  application so that it can think as if they are legitimate  and process them.) |  | YES | NA |  |
| 9. | Insecure cryptographic storage(Checkin  g for unencrypted sensitive data such as credit cards,user information,account records etc.) | NA | YES | YES | Check for User contact information details |
| 10. | Unvalidated redirects and forwards(Checking validation of redirects and forwards so that user will not be redirected to phishing sites) | NA | YES | NA |  |
| 11. | Credentials Transported over an Encrypted  Channel(Check http requests and verify whether the user credentials are properly encrypted or not) |  |  | YES |  |
| 12. | Weak password policy(Check for system's password policy. For better security password should be strong i.e.  combination of alphanumeric and special characters) | NA | YES | NA |  |
| 13. | Stack/Heap Overflow  (Checking for code  vulnerabilities which could result into stack or heap  overflow so that  they would not  result in Denial of service) | NA | YES | NA |  |
| 14. | File Fuzzing  (Generate fuzzed  files and observe the behavior of the system for fuzzed  input) |  |  |  |  |
| 15. | Banner Disclosure  (The server or the application should not disclose its type via the HTTP response headers) | NA | YES | NA |  |
| 16. | Open Redirect  (Vulnerability where the application takes user input to generate some form of redirection without validating the to-be-redirected-to location.) | NA | YES |  |  |
| 17. | Error messages  (The error messages should not show any important or sensitive information | NA | YES |  | Verify the Stack traces generated |
| 18. | Storage of User credentials in Database  (User credentials should be stored in the database in encrypted format) | NA | NA | YES |  |
| 19. | Lock Account-  (Lock account on multiple wrong attempts for entering password ) | NA | YES | NA |  |
| 20. | Session Cookies-(Cookies issued by the server/application should be marked as “HTTPOnly”) |  | YES | NA |  |
| 21. | Downloading files-  (Verify that files cannot be uploaded or downloaded without signing in) | YES | YES | YES |  |