|  |  |
| --- | --- |
| Name Of The Student | Jaganathan G |
| Internship Project Topic | TCS iON RIO-125: Application of Static Application Security Testing (SAST) Tools – Find Defects in Insecure Web-based Applications |
| Name of the Organization | TCS iON |
| Name of the Industry Mentor | Uma Devi |
| Name of the Institute | Government College of Engineering, Bodinayakkanur, Theni - 625583 |

|  |  |  |
| --- | --- | --- |
| Date | Day # | Hours Spent |
| 14-05-2023 | DAY 24 | 3 Hours |
| Activities done during the day:   1. Browsed the internet to study about topics. 2. Scanned an insecure application named “WebGoat-2023”. The scan result contains.  * Some Critical issues are found: * Bugs: * Change this condition so that it does not always evaluate too "false"   **File Name:** Make sure with Assignment1.java  **Description:** Conditional expressions which are always true or false can lead to dead code. Such code is always buggy and should never be used in production.  **Explanation with Code snippets:**  The src/main/java/org/owasp/webgoat/lessons/challenges/challenge1/Assignment1.java    **Risk / Undesirable impact:**  Because it is easy to extract strings from an application source code or binary, passwords should not be hard-coded. This is particularly true for applications that are distributed or that are open-source.  In the past, it has led to the following vulnerabilities:   * [CVE-2019-13466](http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-13466) * [CVE-2018-15389](http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-15389)   Passwords should be stored outside of the code in a configuration file, a database, or a password management service.  This rule flags instances of hard-coded passwords used in database and LDAP connections. It looks for hard-coded passwords in connection strings, and for variable names that match any of the patterns from the provided list  **Recommendations:**   * Store the credentials in a configuration file that is not pushed to the code repository. * Store the credentials in a database. * Use your cloud provider’s service for managing secrets. * If a password has been disclosed through the source code: change it.   **Solution:**  String username = getEncryptedUser();  String password = getEncryptedPassword();  Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +  "user=" + uname + "&password=" + password); Use try-with-resources or close this "PreparedStatement" in a "finally" clause.File Name: Assignment5.java **Description:** Connections, streams, files, and other classes that implement the Closeable interface or its super-interface, AutoCloseable, needs to be closed after use. Further, that close call must be made in a finally block otherwise an exception could keep the call from being made. Preferably, when class implements AutoCloseable, resource should be created using "try-with-resources" pattern and will be closed automatically.  Failure to properly close resources will result in a resource leak which could bring first the application and then perhaps the box the application is on to their knees. Explanation with Code snippetssrc/main/java/org/owasp/webgoat/lessons/challenges/challenge5/Assignment5.java   **Risk/ Undesirable impact:**  Formatted SQL queries can be difficult to maintain, debug and can increase the risk of SQL injection when concatenating untrusted values into the query  **Recommendations:**   * Some parts of the query come from untrusted values (like user inputs). * The query is repeated/duplicated in other parts of the code. * The application must support different types of relational databases.   **Solution:**   * Use [parameterized queries, prepared statements, or stored procedures](https://cheatsheetseries.owasp.org/cheatsheets/Query_Parameterization_Cheat_Sheet.html) and bind variables to SQL query parameters. * Consider using ORM frameworks if there is a need to have an abstract layer to access data.  1. Visited the digital discussion room and checked for any new announcements. 2. Finally, I wrote the Activity Report | | |