Vulnerability Report for Input Validation Issues – Part 1 in DIVA Application

- **Title:** Vulnerability Report for Input Validation Issues – Part 1 in DIVA Application

- **Severity:** Critical

- **Description:**

  The Diva application is vulnerable to SQL injection in the user details display functionality, which allows an attacker to execute malicious SQL queries, leading to the unauthorised extraction of sensitive user information. The lack of proper input validation exposes the application to this severe security risk.

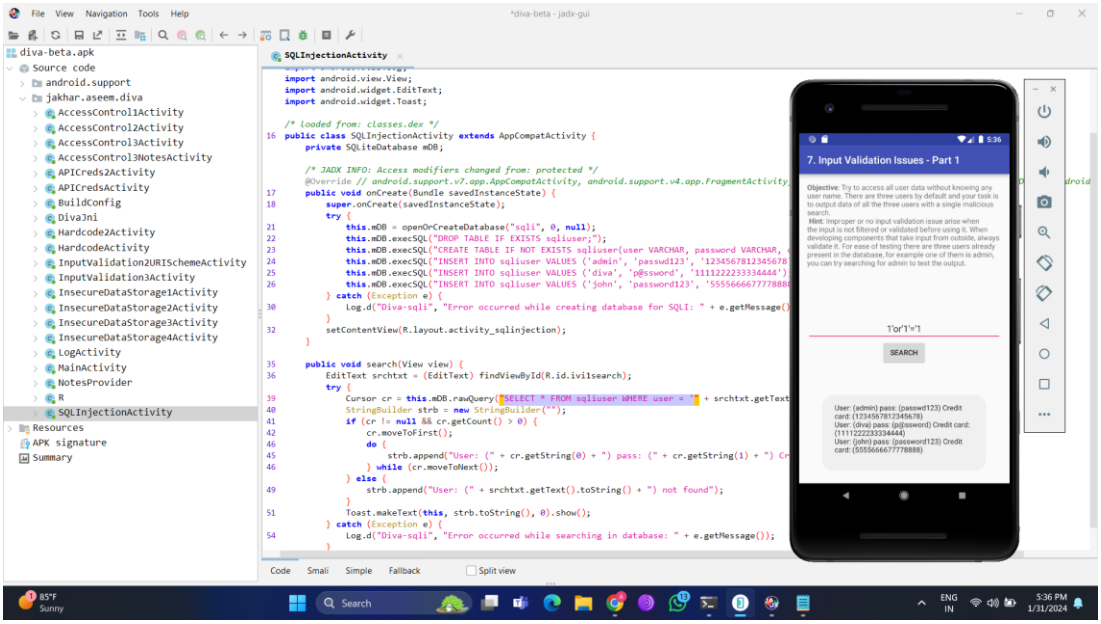
- **Impact:**

  1. Unauthorised Access: Attackers can gain access to confidential user details stored in the database.
  2. Data Theft: Sensitive information, such as usernames, passwords, and personal details, can be extracted.
  3. Privacy Violation: Users' privacy is compromised as their confidential information is exposed.

- **Steps to Reproduce:**

  1. Login to the application.
  2. Click on the “Input Validation Issues – Part 1” option.
  3. The application requires a username stored in the database to display the user details.
  4. Open the “diva-beta.apk” file in the jadx application.
  5. In jadx open the ‘jakhar.assem.diva’ folder, present in the ‘Source code’ folder.
  6. Search for ‘SQLInjectionActivity’ file and open it.
  7. Observe that the code states that the usernames are stored in a database along with its query, hence we can perform SQL injection.
  8. To perform SQL Injection open the application.
  9. Enter the value “ 1’or’1’=’1 ” or “ admin’ ”.
  10. The username and credit card details of all users will be displayed.

- **PoC (Proof of Concept):**



- **Remediation:**

1. Parameterized Queries: Implement parameterized queries or prepared statements to ensure that user inputs are treated as data, not executable code.
2. Input Validation: Implement strict input validation to block malicious input attempts.
3. Least Privilege Principle: Ensure that database accounts used by the application have the minimum required permissions.
4. Regular Security Audits: Conduct regular security audits, including code reviews, to identify and fix vulnerabilities.

- **CWE (Common Weakness Enumeration):**

1. CWE-89: Improper Neutralisation of Special Elements used in an SQL Command ('SQL Injection')
2. CWE-20: Improper Input Validation