Vulnerability Report for Insecure Data Storage - Part 2 in DIVA Application

- **Title:** Vulnerability Report for Insecure Data Storage - Part 2 in DIVA Application
- **Severity:** Critical
- **Description:**

  The Diva application exhibits insecure data storage practices, storing sensitive information in clear text within its databases. This vulnerability exposes confidential data, such as user passwords or personally identifiable information, making it susceptible to unauthorised access.

- **Impact:**

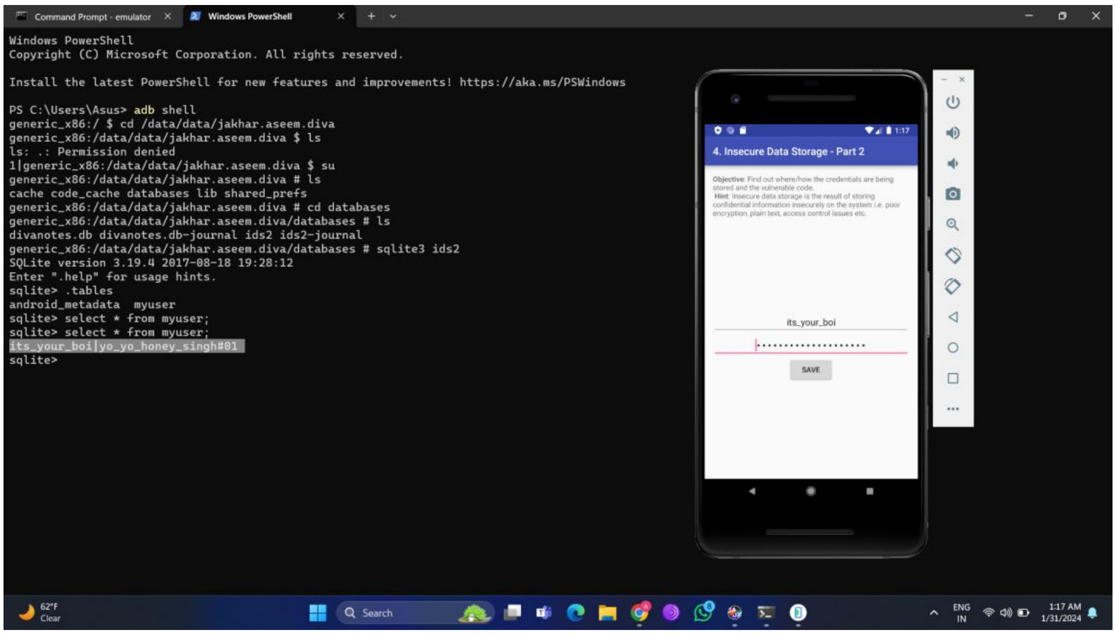
  1. Unauthorised Access: Attackers can gain access to sensitive data directly from the database.
  2. Account Compromise: User credentials stored in clear text are at risk of being used for unauthorised access.
  3. Compliance Violation: Violates data protection regulations and industry best practices.

- **Steps to Reproduce:**

  1. Login to the application.
  2. Click on the “Insecure Data Storage - Part 2” option.
  3. Enter the username and password in the application.
  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 ‘InsecureDataStorage2Activity’ file and open it.
  7. Observe that the code states that the username and password is saved as plain sensitive data in a database named ‘ids2’ and in ‘myuser’ table.
  8. Open the terminal/cmd and type ‘cd data/data/jakhar.aseem.diva’.
  9. List the content of this directory, to do so we need to have root access.
  10. For root access use the command ‘su’.
  11. Use the command ‘cd databases’ to change the directory.
  12. In this directory you will find a database named ‘ids2’.

13. We access and query the database using sqlite3, use the command ‘sqlite3 ids2’ to query the database.
14. Enter ‘.tables’ to get the table names in the database.
15. We get the table name ‘myuser’ which stores the credentials.
16. Enter the query “select * from myuser;” and press enter.

- **<u>PoC (Proof of Concept)</u>:**



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Asus> adb shell
generic_x86:/ $ cd /data/data/jakhar.aseem.diva
generic_x86:/data/data/jakhar.aseem.diva $ ls
ls: .: Permission denied
1|generic_x86:/data/data/jakhar.aseem.diva $ su
generic_x86:/data/data/jakhar.aseem.diva # ls
cache code_cache databases lib shared_prefs
generic_x86:/data/data/jakhar.aseem.diva # cd databases
generic_x86:/data/data/jakhar.aseem.diva/databases # ls
divanotes.db divanotes.db-journal ids2 ids2-journal
generic_x86:/data/data/jakhar.aseem.diva/databases # sqlite3 ids2
SQLite version 3.19.4 2017-08-18 19:28:12
Enter ".help" for usage hints.
sqlite> .tables
android_metadata  myuser
sqlite> select * from myuser;
sqlite> select * from myuser;
its_your_boi|yo_yo_honey_singh#01
sqlite>
```

4. Insecure Data Storage - Part 2

Objective: Find out where/how the credentials are being stored and the vulnerable code.
Hint: Insecure data storage is the result of storing confidential information insecurely on the system i.e. poor encryption, plain text, access control issues etc.

its_your_boi

•••••••••••••••••••••

SAVE

- **<u>Remediation</u>:**

1. Implement Encryption: Utilise strong encryption algorithms (e.g., bcrypt, Argon2) to encrypt sensitive data before storing it in the database.
2. Hash Passwords: Store only hashed and salted versions of passwords in the database to prevent the exposure of actual user credentials.
3. Regularly Update Encryption Methods: Stay current with cryptographic best practices and update encryption methods as needed.
4. Data Purging: Regularly audit and purge unnecessary sensitive information from the database.

- **<u>CWE (Common Weakness Enumeration)</u>:**

1. CWE-256: Plaintext Storage of a Password
2. CWE-261: Weak Cryptography for Passwords
3. CWE-313: Cleartext Storage of Sensitive Information in a Resource that is Accessible by Untrusted Control Sphere