Vulnerability Report for Access Control Issues – Part 2 in DIVA Application

- **Title:** Vulnerability Report for Access Control Issues – Part 2 in DIVA Application

- **Severity:** Critical

- **Description:**

  The Diva application exhibits serious access control issues where credentials can be accessed from outside the application. An attacker can manipulate a boolean variable responsible for displaying credentials, gaining unauthorised access to sensitive information. Additionally, the presence of hardcoded credentials poses a significant security risk, potentially leading to unauthorised access and compromise of sensitive information.

- **Impact:**

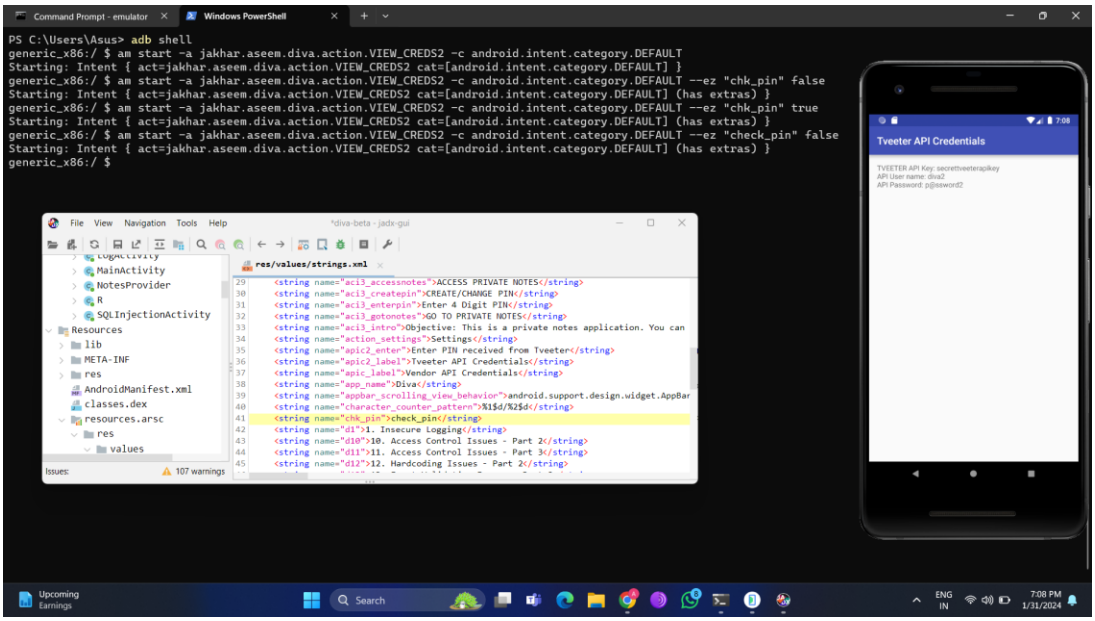
  1. Unauthorised Access: Malicious entities can exploit the access control vulnerabilities to gain unauthorised access to sensitive data.
  2. Credential Compromise: The hardcoded credentials increase the risk of unauthorised users gaining access to the application and its underlying systems.
  3. Data Breach: The combination of access control issues and hardcoded credentials can lead to a severe data breach.
  4. User Impersonation: Malicious actors may impersonate legitimate users due to weak access controls.

- **Steps to Reproduce:**

  1. Login to the application.
  2. Click on the “Access Control Issues – Part 2” option.
  3. Select the “Already Registered” option and click on the “View Tweeter API Credentials” button to view the API credentials.
  4. Close the application.
  5. Open the “diva-beta.apk” file in the jadx application.
  6. In jadx open the ‘jakhar.assem.diva’ folder, present in the ‘Source code’ folder.
  7. Search for ‘APICreds2Activity’ file and open it.
  8. Observe that the API credentials are hardcoded in this file.

9. To access the credentials using terminal/cmd open their “AccessControl2Activity” file in the ‘Source code’ folder.
10. Observe that the code contains the action name “jakhar.aseem.diva.action.VIEW_CREDS2”.
11. Along with the action name, the code contains a boolean variable named “chk_pin”. Credentials can be accessed by changing the value of this variable.
12. Follow the “path “Resources > resources.arsc > res > values > strings.xml” and search the name of chk_pin.
13. The name we got for chk_pin is “.
14. Open the “AndroidManifest.xml” present in the ‘Resources’ folder.
15. Open the terminal/cmd and connect your device using “adb shell”.
16. Now use the command “ am start -a <action name> –ez “check_pin” <true/false> ”.
17. One of the boolean values opens the application automatically and shows the credentials.

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



- **Remediation:**

1. Access Control:
    1) Implement proper role-based access control (RBAC) mechanisms.
    2) Ensure that sensitive resources are protected with proper authentication and authorization checks.

2. Hardcoding Issue:

1) Remove hardcoded credentials from the source code.
2) Utilise secure credential storage mechanisms such as environment variables or dedicated credential stores.
3) Implement dynamic and secure credential retrieval mechanisms.

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

1. CWE-284: Improper Access to Sensitive Information
2. CWE-798: Use of Hard-coded Credentials