Vulnerability Report: Access Control Vulnerability

Title:

Access Control Vulnerability

Severity:

[High]

Description

The application lacks proper access controls, allowing unauthorized users to access or modify sensitive functionality or data. This can enable attackers to elevate privileges or access restricted resources within the application.

Details:

The vulnerability is located in specific components of the application where access control checks are missing. Sensitive actions or data are accessible without authentication or appropriate user permissions.

Impact

his vulnerability allows attackers to:

Access restricted functionality, like viewing or modifying sensitive user data.

Perform privileged actions typically restricted to authenticated users.

The impact of this vulnerability is significant as it could lead to unauthorized access, data exposure, and potential manipulation of the application's data.

Steps to Reproduce

PART 1

- First, run adb shell to get root access of the android device.
- Next, run am start -a jakhar.aseem.diva.action.VIEW_CREDS to start the activity to bypass the "VIEW API CREDENTIALS" button.

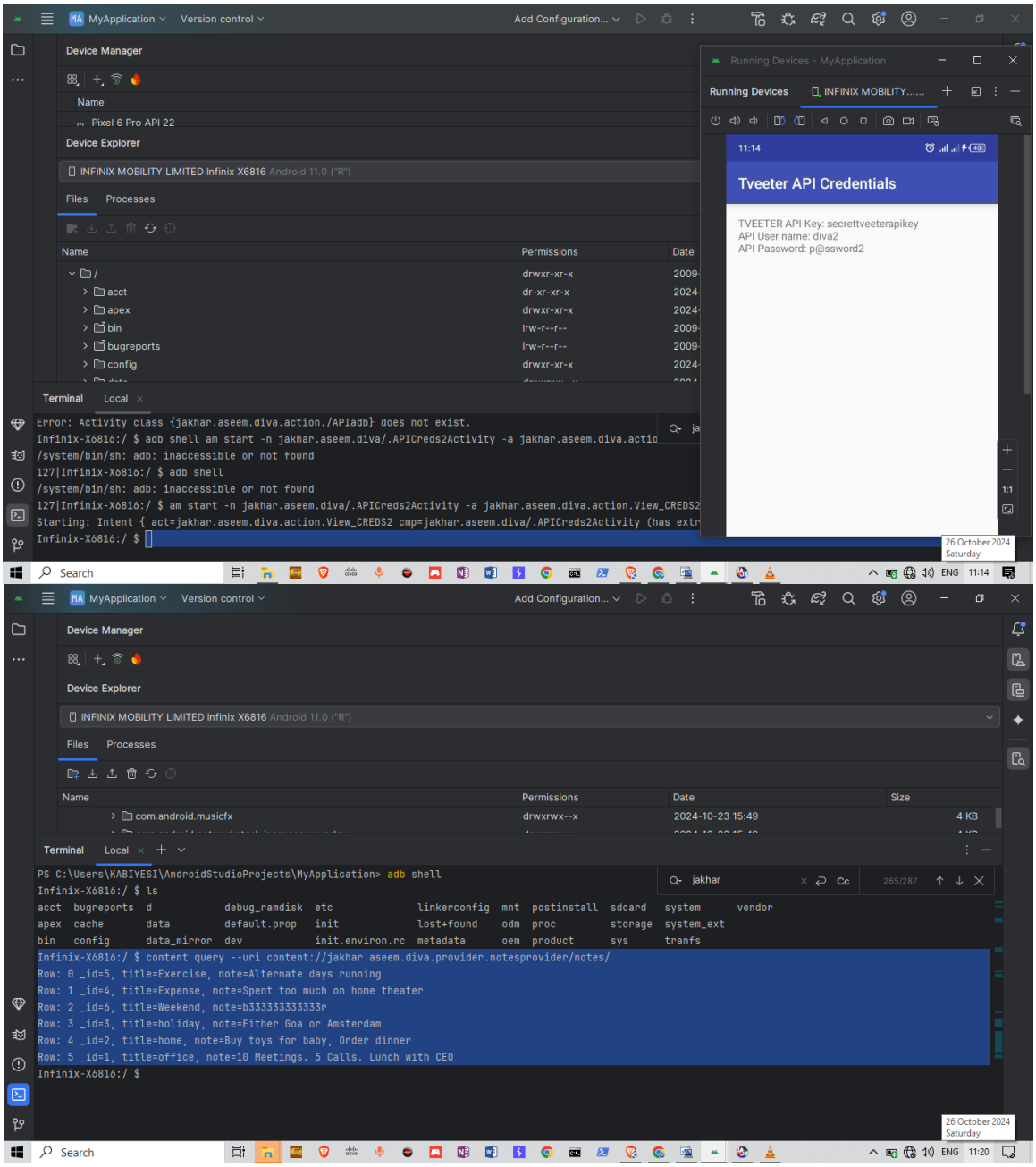
PART 2

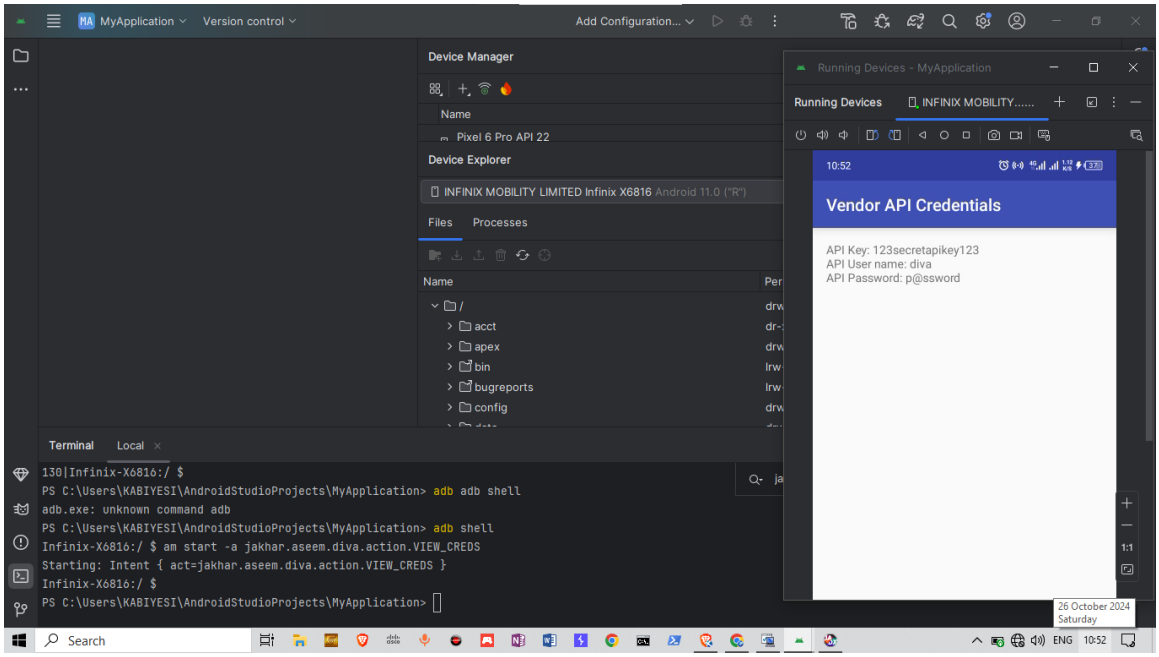
- Run adb shell to get root access to the Android device.
- Next, run am start -n jakhar.aseem.diva/.APICreds2Activity -a jakhar.aseem.diva.action/.VIEW_CREDS2 — ez check_pin false to start the activity that will display the Tveeter API credentials.

PART 3

- Run content query --uri content://jakhar.aseem.diva.provider

Proof of Concept (PoC)





Remediation

To address this vulnerability:

Implement Role-Based Access Control (RBAC):

Ensure that all sensitive actions and resources are protected by access control checks appropriate to the user's role.

Enforce Authentication:

Require users to authenticate before accessing restricted parts of the application.

Regular Security Audits:

Conduct periodic security audits to ensure access controls are functioning as expected.

Implement Logging and Monitoring:

Track access to sensitive areas to detect and respond to unauthorized access attempts.

CWE (Common Weakness Enumeration)

- OWASP Mobile Security Testing Guide: Access Control Testing
- CWE-284: Improper Access Control

Vulnerability Report: Hardcoding Vulnerability

Title:

Hardcoded Keys/Passwords

Severity:

[Medium]

Description

The hardcoded keys/passwords vulnerability is a type of security weakness that arises when sensitive information such as passwords or encryption keys are hardcoded within the source code of an application. This means that the information is embedded in the code itself, making it easily accessible to anyone who can view or obtain the code.

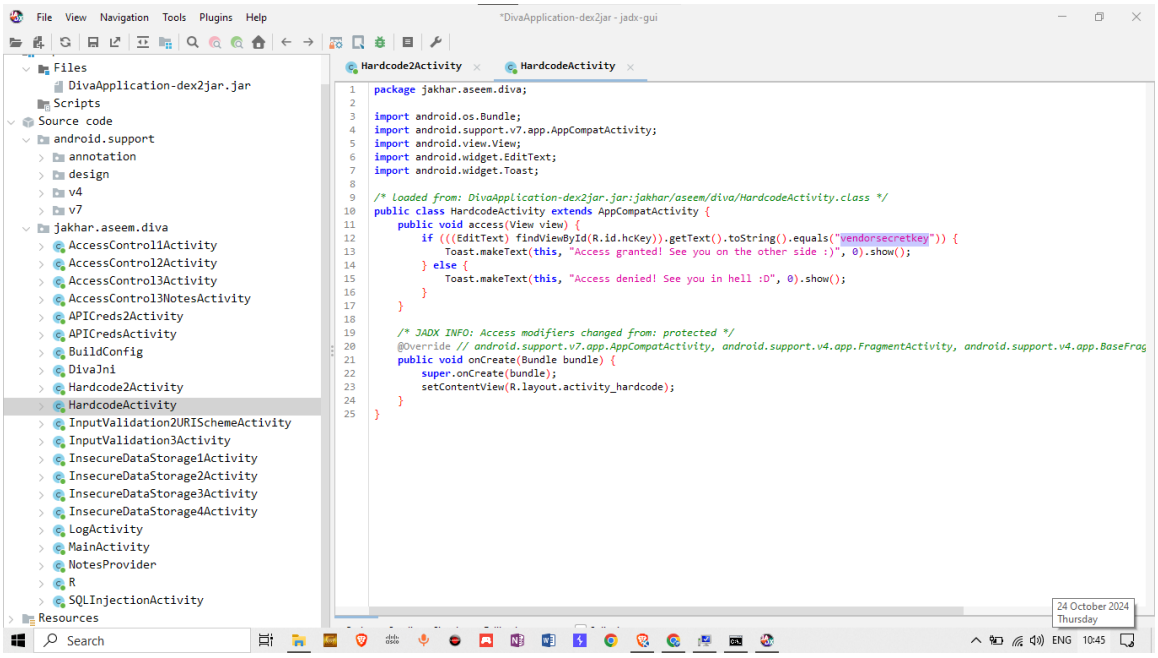
Impact

This vulnerability can lead to a range of security issues, including unauthorized access to sensitive data, compromised user accounts, and data breaches. Attackers can easily obtain these hardcoded keys/passwords by reverse engineering the source code, which can lead to devastating consequences.

Steps to Reproduce

- First convert the APK file into RAR file by only changing the extension.
- Then Extract the RAR file.
- Convert the classes.dex file into jar file with Dex2Jar tool.
- Go to Command Prompt and enter following command: d2j-dex2jar classes.dex
- Now access this JAR file with JD-GUI which is Java Decompiler.

Proof of Concept (PoC)



Remediation

To address this vulnerability, it is recommended that all hardcoded keys/passwords are removed from the source code and replaced with secure and dynamic solutions such as environment variables or secure key stores. Additionally, it is recommended to use strong and unique passwords for all user accounts and regularly rotate them.

CWE (Common Weakness Enumeration)

- CWE-259: Use of Hard-coded Password.

Vulnerability Report: Input Validation Vulnerability

Title:

Lack of Input Validation

Severity:

[Critical]

Description

The application does not properly validate user input, which could allow an attacker to submit malicious data that could be used to execute arbitrary code or disclose sensitive information.

Impact

An attacker could use this vulnerability to execute arbitrary code, disclose sensitive information, or perform other malicious actions. This could lead to a compromise of the affected system, or potentially even the entire network.

Steps to Reproduce

1. Setup Environment:

1. Ensure you have DivaApplication.apk installed on your device/emulator.
2. Launch the DivaApplication.
1. Navigate to different input fields within the application**.**
2. First try to enter single quote (‘) as input and check result.
3. Try to enter single quote twice (‘’) and then check result.
4. You will see the difference in the output of the toast.

2. Input Malicious Data:

In each input field identified, attempt to input various types of malicious data, such as:

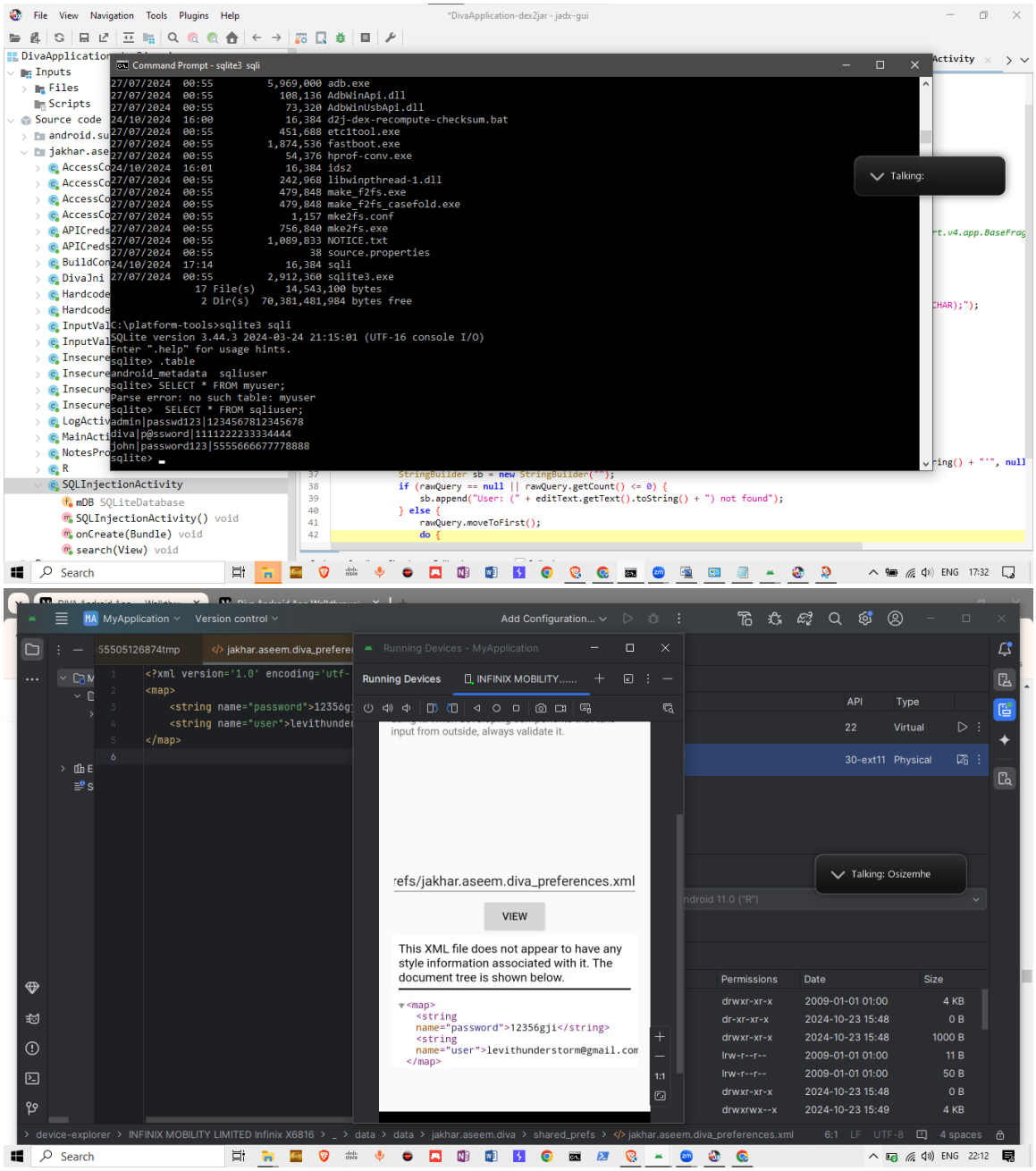
- **Special Characters:**

Input unusual characters, such as %, @, or #, to see how the application handles them.

- **Excessive Length Input:**

Input a very long string (e.g., 1000+ characters) to check for buffer overflow vulnerabilities.

Proof of Concept (PoC)


File View Navigation Tools Plugins Help
*DivaApplication-dex2jar - jadx-gui
Command Prompt - sqlite3 sqli
27/07/2024 00:55 5,969,000 adb.exe
27/07/2024 00:55 108,136 AdbWinApi.dll
27/07/2024 00:55 73,320 AdbWinUsbApi.dll
24/10/2024 16:00 16,384 dzj-dex-recompute-checksum.bat
27/07/2024 00:55 451,688 etc1tool.exe
27/07/2024 00:55 1,874,536 fastboot.exe
27/07/2024 00:55 54,376 hprof-conv.exe
24/10/2024 16:01 16,384 ids2
27/07/2024 00:55 242,968 libwinpthread-1.dll
27/07/2024 00:55 479,848 make_f2fs.exe
27/07/2024 00:55 479,848 make_f2fs_casefold.exe
27/07/2024 00:55 1,157 mke2fs.conf
27/07/2024 00:55 756,840 mke2fs.exe
27/07/2024 00:55 1,089,833 NOTICE.txt
27/07/2024 00:55 38 source.properties
24/10/2024 17:14 16,384 sqli
27/07/2024 00:55 2,912,360 sqlite3.exe
17 File(s) 14,543,100 bytes
2 Dir(s) 70,381,481,984 bytes free
C:\platform-tools>sqlite3 sqli
SQLite version 3.44.3 2024-03-24 21:15:01 (UTF-16 console I/O)
Enter ".help" for usage hints.
sqlite> .table
android_metadata sqliuser
sqlite> SELECT * FROM myuser;
Parse error: no such table: myuser
sqlite> SELECT * FROM sqliuser;
admin|passwd123|1234567812345678
diva|p@ssword|1111222233334444
john|password123|5555666677778888
sqlite>
Talking:
SQLInjectionActivity
mDB SQLiteDatabase
SQLInjectionActivity() void
onCreate(Bundle) void
search(View) void
MyApplication
Version control
Add Configuration...
Running Devices - MyApplication
Running Devices
INFINIX MOBILITY.....
input from outside, always validate it.
refs/jakhar.aseem.diva_preferences.xml
VIEW
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<map>
<string name="password">12356gji</string>
<string name="user">levithunderstorm@gmail.com</string>
</map>
Talking: Osizemhe
API Type
22 Virtual
30-ext11 Physical
Permissions Date Size
drwxr-xr-x 2009-01-01 01:00 4 KB
dr-xr-xr-x 2024-10-23 15:48 0 B
drwxr-xr-x 2024-10-23 15:48 1000 B
lrw-r--r-- 2009-01-01 01:00 11 B
lrw-r--r-- 2009-01-01 01:00 50 B
drwxr-xr-x 2024-10-23 15:48 0 B
drwxrwx--x 2024-10-23 15:49 4 KB
Search
ENG 17:32
ENG 22:12

Remediation

To remediate this vulnerability, input validation should be implemented to ensure that all user input is properly sanitized and validated. This can include using whitelists of acceptable input, or using regular expressions to validate that the input matches a specific format. Additionally, input validation should be performed on the server-side, rather than relying on client-side validation alone. - Parameterized Queries: Use prepared statements for database interactions to avoid SQL injection.

CWE (Common Weakness Enumeration)

CWE ID:

- [CWE ID related to the vulnerability, e.g., CWE-20 (Improper Input Validation), CWE-89 (SQL Injection), CWE-79 (Cross-Site Scripting)]

- CWE-20: Improper Input Validation (https://cwe.mitre.org/data/definitions/20.html)

- CWE: CWE-20: Improper Input Validation.

- CVSS v3.1 Vector: Medium AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N

Vulnerability Report: Insecure Storage Vulnerability

Title:

Insecure Logging Practices

Severity:

[Medium]

Description

The application logs sensitive information, such as usernames, passwords, session tokens, or personal identifiers, in plaintext. This data is written to logs that can be accessed by unauthorized users if they have physical access to the device or gain access via malware.

Details:

- Sensitive data is recorded in the application's logs in plaintext, without any redaction or encryption.
- Example data identified in logs: usernames, authentication tokens

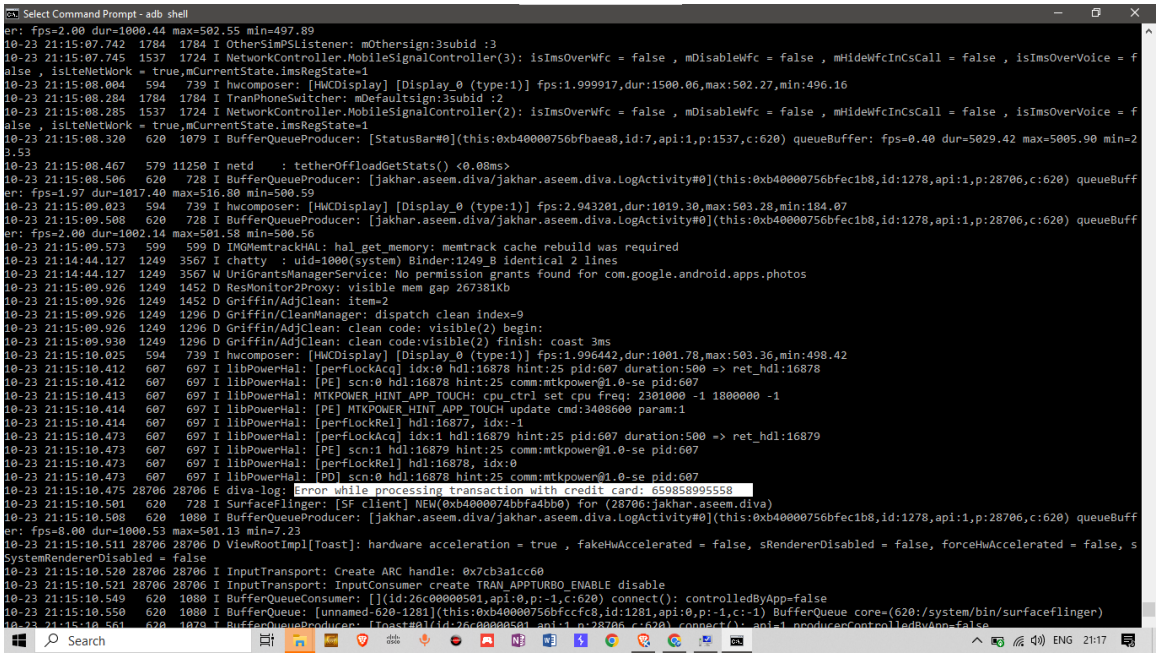
Impact

- An attacker with access to the device or logs can retrieve sensitive information from the log files.
- This could lead to:
- Account compromise if usernames and passwords are stored.
- Disclosure of private information, compromising user privacy.
- This vulnerability poses a risk for data breaches, privacy violations, and potential non-compliance with data protection regulations.

Steps to Reproduce

- execute command adb shell
- Shell will open, type the command: logcat

Proof of Concept (PoC)



Remediation

To remediate this vulnerability:

- **Remove Sensitive Information from Logs:** Avoid logging sensitive data such as passwords or tokens.
- **Implement Log Filtering:** Use a logging framework that allows filtering or redaction of sensitive information.
- **Encrypt Logs if Necessary:** If sensitive data must be logged for debugging, ensure logs are encrypted and stored securely, and limit access to them.
- **Use Secure Storage:** Ensure logs are stored in secure locations with access control, preventing unauthorized access.

CWE (Common Weakness Enumeration)

- CWE-200: Exposure of Sensitive Information
- CWE-259: Use of Hard-coded Credentials

- CWE-522: Insufficient Logging and Monitoring

Vulnerability Report: Insecure Storage Vulnerability

Title:

Insecure Storage of Sensitive Data

Severity:

[High]

Description

The application stores sensitive data (e.g., API keys, user credentials, personal information) in an insecure manner. This data is accessible to unauthorized parties due to lack of proper encryption or secure storage mechanisms.
Sensitive data is found in SharedPreferences, local databases, or files without encryption.

- Example paths where insecure data storage was identified:
- /data/data/com.example.divaapp/shared_prefs/
- /data/data/com.example.divaapp/databases/

Impact

An attacker could exploit this vulnerability to:

Access sensitive user data, leading to account takeovers.

Exfiltrate API keys or secrets that could be used to impersonate the application.

Compromise user privacy by accessing personally identifiable information (PII).

This could lead to significant data breaches, financial loss, and reputational damage.

Steps to Reproduce

PART 1

- Run adb shell to gain root access to the Android device and ls to list all the files in the directory.
- Next, run cd data/data/ to enter the directory and ls to list all the files in the directory.
- Run cd jakhar.aseem.diva to enter the directory and ls to list the files in the directory.
- To access the directory, run cd shared_prefs and ls to list the items in the directory.
- Now, run cat jakhar.aseem.diva_preferences.xml to read the xml file.

PART 2

- First, gain root access to the device using the adb shell. Then, run cd data/data/jakhar.aseem.diva/databases/ to the directory and ls to list the items in the directory.
- Run ls -la to list all the files and items in the directory.
- Run adb pull /data/data/jakhar.aseem.diva/databases/ids2

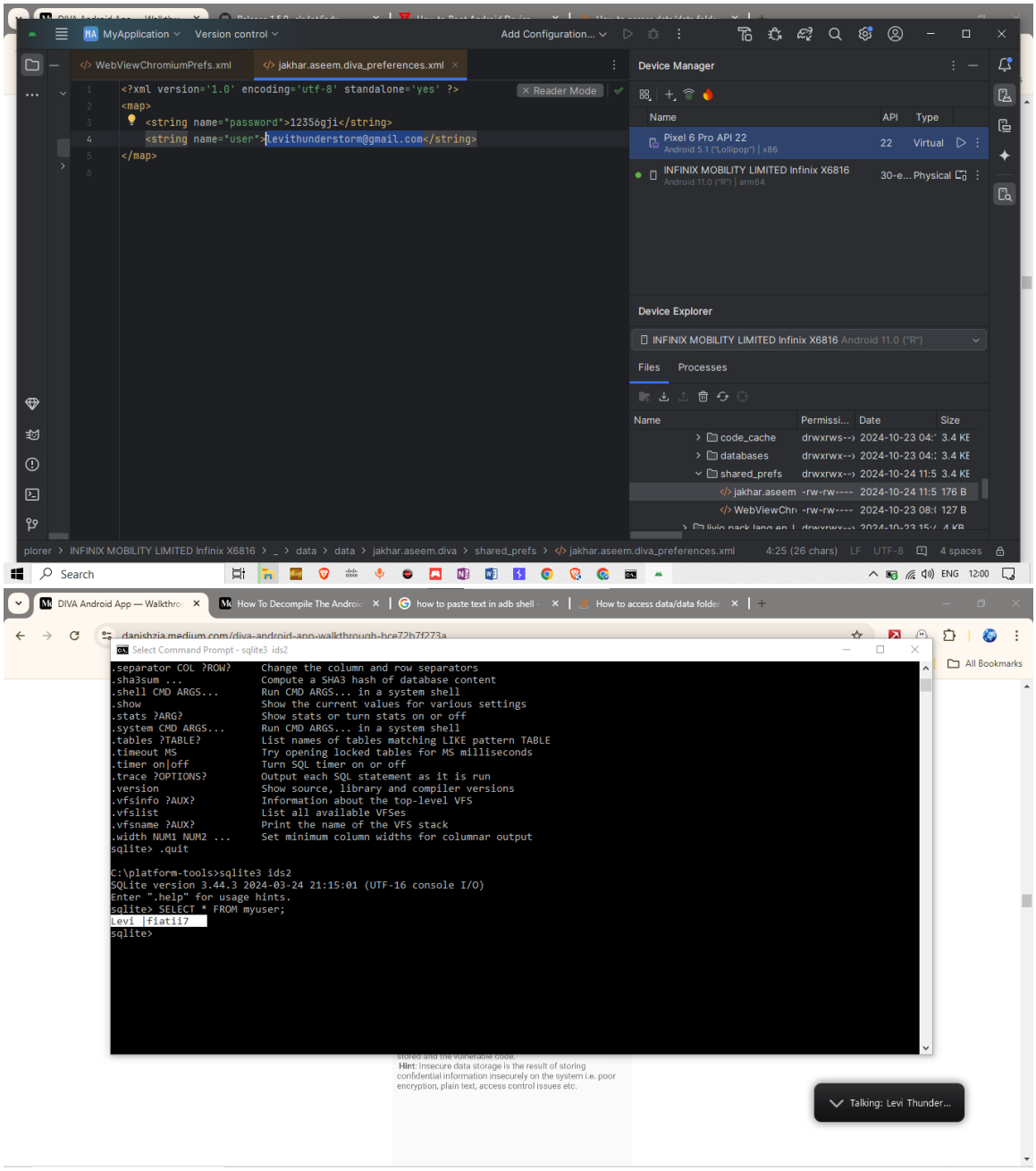
PART 3

- To access the stored credential, I used adb shell to gain root access of the android device.
- Run cd data/data/jakhar.aseem.diva/ to enter the application directory and ls -la to list all the files and directories under the specified directory.
- Next, use cat uinfo44475423tmp to print the file content to a standard output.

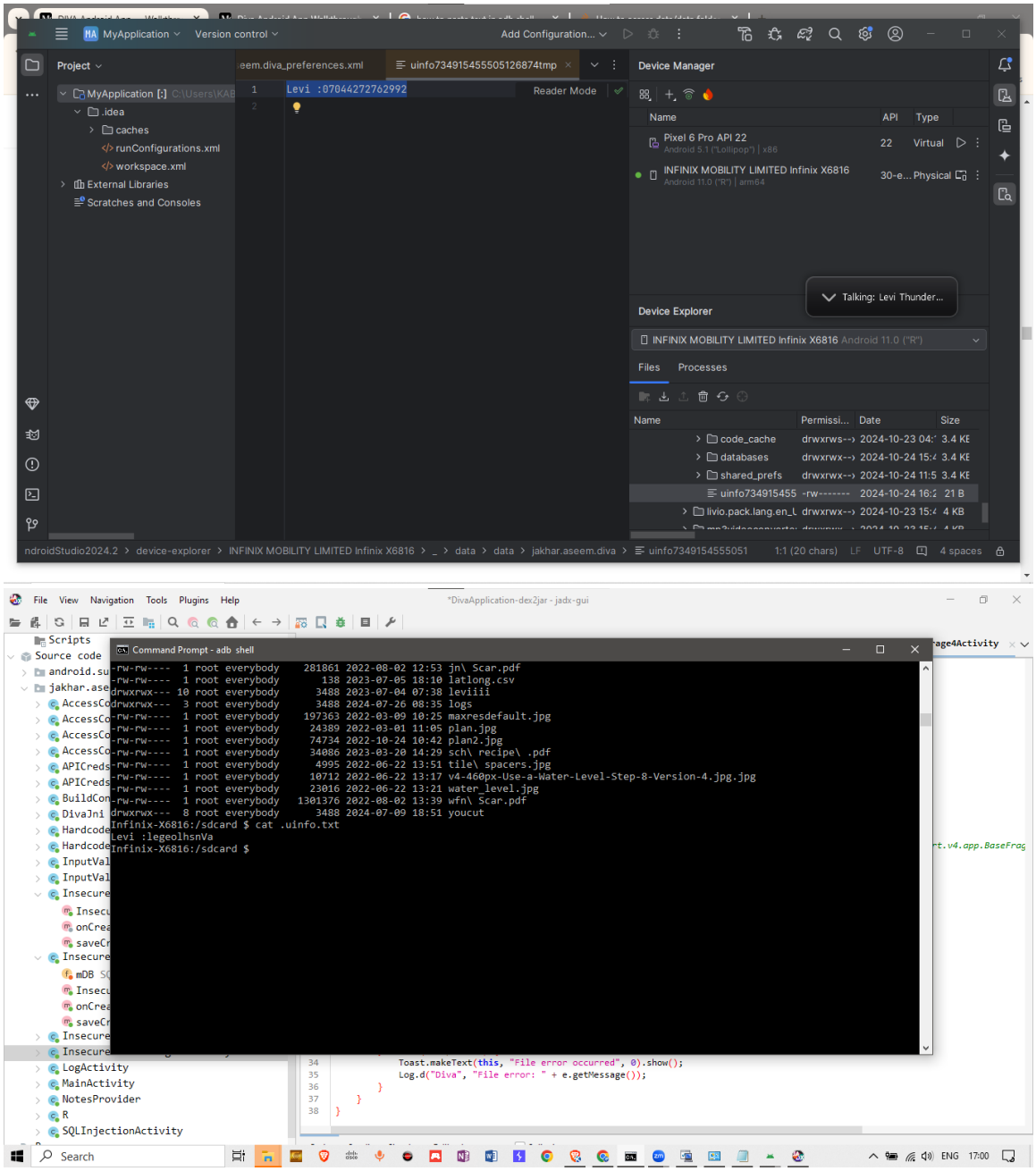
PART 4

- To begin, run adb shell to gain root access of the device and ls to list all the items in the directory
- Run cd mnt to access the external storage on the Android device and ls to list the items in the directory.
- Run cat uinfo.txt to print out the content of the file to a standard output.

Proof of Concept (PoC)


MyApplication
Version control
Add Configuration...
WebViewChromiumPrefs.xml
jakhar.aseem.diva_preferences.xml
Reader Mode
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="password">12356gji</string>
<string name="user">levithunderstorm@gmail.com</string>
</map>
Device Manager
Name
API
Type
Pixel 6 Pro API 22
Android 5.1 ("Lollipop") | x86
22
Virtual
INFINIX MOBILITY LIMITED Infinix X6816
Android 11.0 ("R") | arm64
30-e...
Physical
Device Explorer
INFINIX MOBILITY LIMITED Infinix X6816 Android 11.0 ("R")
Files
Processes
Name
Permissi...
Date
Size
code_cache
databases
shared_prefs
jakhar.aseem
WebViewChr
data
jakhar.aseem.diva
4:25 (26 chars)
LF
UTF-8
4 spaces
Search
ENG
12:00

DIVA Android App — Walkthro
How To Decompile The Androi
how to paste text in adb shell
How to access data/data folder
danishzia.medium.com/diva-android-app-walkthrough-bce72b7f273a
All Bookmarks
Select Command Prompt - sqlite3 ids2
.separator COL ?ROW? Change the column and row separators
.sha3sum ... Compute a SHA3 hash of database content
.shell CMD ARGS... Run CMD ARGS... in a system shell
.show Show the current values for various settings
.stats ?ARG? Show stats or turn stats on or off
.system CMD ARGS... Run CMD ARGS... in a system shell
.tables ?TABLE? List names of tables matching LIKE pattern TABLE
.timeout MS Try opening locked tables for MS milliseconds
.timer on|off Turn SQL timer on or off
.trace ?OPTIONS? Output each SQL statement as it is run
.version Show source, library and compiler versions
.vfsinfo ?AUX? Information about the top-level VFS
.vfslist List all available VFSes
.vfsname ?AUX? Print the name of the VFS stack
.width NUM1 NUM2 ... Set minimum column widths for columnar output
sqlite> .quit
C:\platform-tools>sqlite3 ids2
SQLite version 3.44.3 2024-03-24 21:15:01 (UTF-16 console I/O)
Enter ".help" for usage hints.
sqlite> SELECT * FROM myuser;
Levi |fiat117
sqlite>
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.
Talking: Levi Thunder...


MyApplication
Version control
Add Configuration...
Project
MyApplication [:] C:\Users\KAB
.idea
caches
runConfigurations.xml
workspace.xml
External Libraries
Scratches and Consoles
aeem.diva_preferences.xml
uinfo7349154555051268 74tmp
1 Levi :07044272762992
2
Reader Mode
Device Manager
Name
API
Type
Pixel 6 Pro API 22
Android 5.1 ("Lollipop") | x86
22
Virtual
INFINIX MOBILITY LIMITED Infinix X6816
Android 11.0 ("R") | arm64
30-e...
Physical
Talking: Levi Thunder...
Device Explorer
INFINIX MOBILITY LIMITED Infinix X6816 Android 11.0 ("R")
Files
Processes
Name
Permissi...
Date
Size
code_cache
drwxrws--x
2024-10-23 04:
3.4 KE
databases
drwxrwx--x
2024-10-24 15:
3.4 KE
shared_prefs
drwxrwx--x
2024-10-24 11:5
3.4 KE
uinfo734915455
-rw-------
2024-10-24 16:
21 B
livio.pack.lang.en_L
drwxrwx--x
2024-10-23 15:
4 KB
ndroidStudio2024.2 › device-explorer › INFINIX MOBILITY LIMITED Infinix X6816 › _ › data › data › jakhar.aseem.diva › uinfo7349154555051
1:1 (20 chars)
LF
UTF-8
4 spaces

File View Navigation Tools Plugins Help
*DivaApplication-dex2jar - jadx-gui
Scripts
Source code
android.su
jakhar.ase
AccessCo
AccessCo
AccessCo
AccessCo
APICreds
APICreds
BuildCon
DivaJni
Hardcode
Hardcode
InputVal
InputVal
Insecure
Insecure
Insecure
Insecure
LogActivity
MainActivity
NotesProvider
R
SQLInjectionActivity
Command Prompt - adb shell
-rw-rw---- 1 root everybody 281861 2022-08-02 12:53 jn\ Scar.pdf
-rw-rw---- 1 root everybody 138 2023-07-05 18:10 latlong.csv
drwxrwx--- 10 root everybody 3488 2023-07-04 07:38 leviiii
drwxrwx--- 3 root everybody 3488 2024-07-26 08:35 logs
-rw-rw---- 1 root everybody 197363 2022-03-09 10:25 maxresdefault.jpg
-rw-rw---- 1 root everybody 24389 2022-03-01 11:05 plan.jpg
-rw-rw---- 1 root everybody 74734 2022-10-24 10:42 plan2.jpg
-rw-rw---- 1 root everybody 34086 2023-03-20 14:29 sch\ recipe\ .pdf
-rw-rw---- 1 root everybody 4995 2022-06-22 13:51 tile\ spacers.jpg
-rw-rw---- 1 root everybody 10712 2022-06-22 13:17 v4-460px-Use-a-Water-Level-Step-8-Version-4.jpg.jpg
-rw-rw---- 1 root everybody 23016 2022-06-22 13:21 water_level.jpg
-rw-rw---- 1 root everybody 1301376 2022-08-02 13:39 wfn\ Scar.pdf
drwxrwx--- 8 root everybody 3488 2024-07-09 18:51 youcut
Infinix-X6816:/sdcard $ cat .uinfo.txt
Levi :legeolhsnVa
Infinix-X6816:/sdcard $
Toast.makeText(this, "File error occurred", 0).show();
Log.d("Diva", "File error: " + e.getMessage());
Search
ENG
17:00

Remediation

To remediate this vulnerability:

- **Implement Encryption:**

Use strong encryption to encrypt sensitive data before storage.

- **Use Android KeyStore:**

Store sensitive data such as API keys in the Android KeyStore system, which provides a more secure way to handle cryptographic keys.

- **Minimize Sensitive Data Storage:**

Where possible, avoid storing sensitive information on the device. Use server-side storage with proper access controls.

- **Regular Security Audits:**

Conduct regular security audits and penetration tests to identify and mitigate vulnerabilities in data storage mechanisms.

CWE (Common Weakness Enumeration)

- CWE-327: Use of a Broken or Risky Cryptographic Algorithm
- CWE-326: Inadequate Encryption Strength
- CWE-325: Insufficiently Protected Credentials