



---

# DIVA MOBILE APPLICATION PENETRATION TESTING

---

report



AMJAD AMEEN PP

23-11-24

# Contents

Si no	Description	page no
1	Assessment overview .....	2
2	Components .....	3
2.1	Penetration testing .....	3
2.2	Finding severity ratings .....	3
3	Scope.....	3
4	Executive summary.....	4
5	Attack summary.....	5
5.1	Insecure Logging .....	6
5.2	Hardcoding Issue .....	8
5.3	Sensitive data Stored in a shared preferances.....	11
5.4	Sensitive data Stored in a Local Storage.....	14
5.5	Sensitive data Stored in a Temp file .....	16
5.6	Input Validation (Sql injection).....	18
5.7	Input Validation (URL injection).....	20
6	Conclusion .....	21

# **1.Assessment Overview**

From Nov 18th, 2024 to Nov 24th, 2024, DIVA engaged my company to evaluate the security posture of its infrastructure compared to current industry best practices that included an external penetration test. All testing performed is based on the *OWASP Mobile Application Security Testing Guide (MASTG)*,

Phases of penetration testing activities include the following:

## 1. Reconnaissance and Information Gathering

- Objective: Collect information about the app, its platform, and backend to plan the test.

## 2. Static Analysis

- Objective: Analyze app code (source or decompiled) to find vulnerabilities without running it.

## 3. Dynamic Analysis

- Objective: Test the app's behavior during execution to detect vulnerabilities.

## 4. Backend/API Testing

- Objective: Assess APIs and backend systems for security weaknesses.

## 5. Device-Level Testing

- Objective: Check for insecure data storage or improper interactions with the device.

## 6. Reverse Engineering and Tampering

- Objective: Evaluate app resilience to reverse engineering and unauthorized modifications.

## 7. Reporting

- Objective: Document findings with remediation steps mapped to the OWASP Mobile Top 10.

# **2.Components**

## 2.1 Penetration Test

Emulate an attacker attempting to exploit vulnerabilities in a mobile application to identify risks to sensitive data and systems. This includes gathering information about the app, analyzing its security posture, and identifying exploitable vulnerabilities.

## 2.2 Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

CVSS Score	Severity Level
0.0	None
0.1–3.9	Low
4.0–6.9	Mediam
7.0–8.9	High
9.0–10.0	Critical

### Summary of Key References:

- **OWASP Mobile Top 10:** Provides a list of the most common mobile app risks. Used to categorize vulnerabilities.
- **CVSS v3.1:** Standardized scoring system for vulnerabilities. Helps map security risk to severity.
- **Security Impact Analysis:** Used to estimate the potential impact of vulnerabilities on the security of the mobile application.

## 3. Scope

### Target Application

- Apk : diva-beta.apk
- Domain: mobile application

## 4. Executive Summary

The penetration testing of the **Diva Mobile Application** was conducted using **Genymotion**, **ADB Shell**, and **JADX CLI**, focusing on the **OWASP Top 10 Mobile Risks**. The assessment identified several security issues, including insecure data storage, weak encryption, and flaws in authentication mechanisms. Key vulnerabilities were found in areas such as code tampering and reverse engineering, which could potentially expose sensitive data or allow unauthorized access. The report provides detailed findings and recommendations to enhance the security of the application, including strengthening encryption, improving authentication practices, and addressing code vulnerabilities.

### Tools Used:

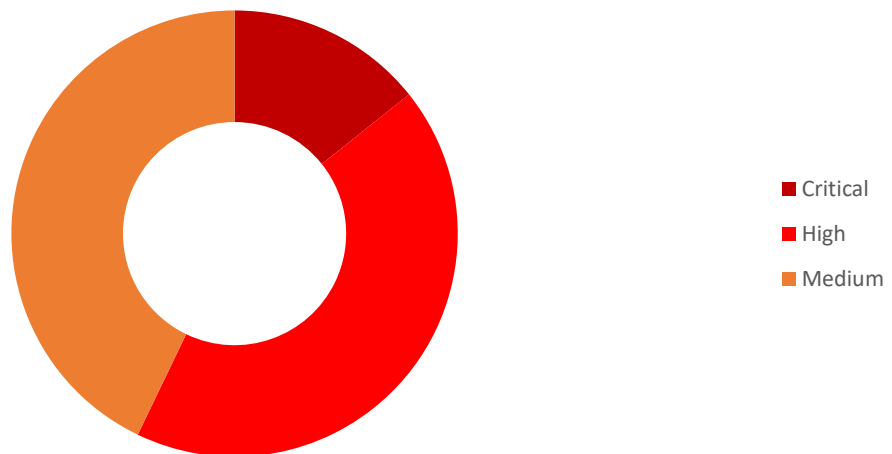
- **Genymotion Simulator:** To emulate various mobile environments for testing vulnerabilities.
- **ADB Shell:** For interacting with the application on the Android device and performing commands to analyze behavior.
- **JADX CLI:** Used to decompile and analyze the mobile application's APK, searching for code weaknesses, insecure configurations, and reverse engineering potential.

## 5. Attack summary

SI NO	Vulnerability	Description	Severity
1	Insecure Logging	The application fails to properly log sensitive data, potentially exposing information in logs that can be accessed by attackers.	Medium
2	Hardcoding Issue	Sensitive information such as API keys, passwords, or tokens are hardcoded in the application code, making them easily exploitable.	High
3	Sensitive Data Stored in Shared Preferences	Sensitive information is stored insecurely in Shared Preferences, which can be easily accessed by attackers with physical access to the device.	High
4	Sensitive Data stored in local storage	Sensitive information is stored insecurely in local storage, which can be easily accessed by attackers with physical access to the device.	Medium
5	Sensitive Data Stored in Temporary Files	Sensitive data is stored in temporary files, making it vulnerable to unauthorized access, especially if the device is compromised.	Medium
6	Input validation issue (sql injection)	this vulnerability occurs when user input is directly concatenated into SQL queries without proper sanitization or parameterization, enabling attackers to manipulate database queries.	Critical

7	Input validation issue (web URL)	This vulnerability occurs when a web application fails to properly validate or sanitize user input, specifically in URL parameters. Malicious users can manipulate the URL to inject unwanted characters, commands, or harmful data, leading to potential security risks such as unauthorized access, data leakage, or execution of malicious code.	High

Total



## 5.1 Insecure Logging

<b>Description</b>	Insecure logging happens when sensitive information (like passwords, tokens, or personal data) .in here we found credit card number
<b>Impacts</b>	Technical impacts: <ul style="list-style-type: none"> <li>Data Exposure: Sensitive information (e.g., passwords, API keys, personal data, session tokens) can be logged in plaintext and accessed by unauthorized users or attackers, leading to data breaches.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Privilege Escalation:</b> Attackers gaining access to logs can obtain valuable information such as authentication tokens or system configurations, potentially escalating privileges or bypassing security controls.</li> </ul> <p>User impacts</p> <ul style="list-style-type: none"> <li>• <b>Data Breach:</b> Sensitive information such as passwords, session tokens, or personal data may be exposed in logs, increasing the risk of a data breach if the logs are accessed by unauthorized users.</li> <li>• <b>Unauthorized Access:</b> Attackers gaining access to logs could extract authentication credentials or other sensitive data, enabling them to impersonate users or gain unauthorized access to accounts or resources.</li> <li>• <b>Privilege Escalation:</b> Attackers can use information from logs (e.g., error messages, stack traces) to escalate their privileges, bypass security mechanisms, or exploit vulnerabilities in the application.</li> </ul>
<b>Mitigations</b>	<ul style="list-style-type: none"> <li>• Avoid logging sensitive information (e.g., passwords, tokens, personal data).</li> <li>• Mask or encrypt sensitive data before logging.</li> <li>• Restrict log file access with proper access controls.</li> <li>• Use secure logging mechanisms with encryption.</li> </ul>

## Proof of concept :

Our first task in the Damn Insecure and Vulnerable Application (DIVA) is capturing sensitive data in logs.

First of all I have performed the adb devices to check whether the device is connected or not

commands are :

adb shell connection checking : **adb shell**

connecting device : **adb connect \$ip**

Identifying Running Processes(pid) : **adb shell ps | grep -i diva**

Identifying Sensitive Information in logcat : **adb logcat |grep -i (pid)**

```

amjad@highfly: ~
$ adb shell
* daemon not running; starting now at tcp:5037
* daemon started successfully
adb: no devices/emulators found

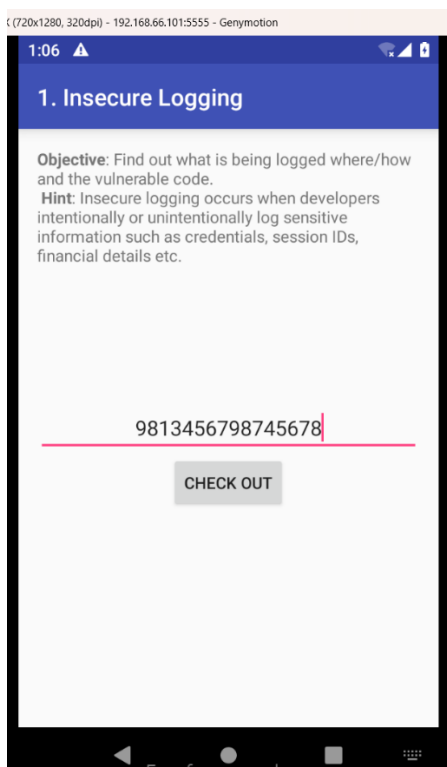
amjad@highfly: ~
$ adb connect 192.168.66.101:5555
connected to 192.168.66.101:5555

amjad@highfly: ~
$ adb shell ps | grep -i diva
uid_349      4250      406 1810416 113388  ep_poll      e7a3bb9b 5 jakhar.aseem.diva

amjad@highfly: ~
$ adb logcat | grep -i 4250
11-20 08:46:08.775 406 406 D Zygote : Forked child process 4250
11-20 08:46:08.790 4250 4250 I Zygote : seccomp disabled by setenforce 0
11-20 08:46:08.799 723 756 I ActivityManager: Start proc : jakhar.aseem.diva/uba49 for activity {jakhar.aseem.diva/jakhar.aseem.diva.MainActivity}
11-20 08:46:08.801 4250 4250 I jakhar.aseem.diva: late-enabling -Xcheckjni
11-20 08:46:08.794 4250 4250 W main : type=1400 audit(0.0:247): avc: granted { read } for name="uioobject_r:net_dns_prop:s0" dev="tmpfs" ino=8183 scontext=u:r:untrusted_app_25:s0:c512,c768 tcontext=u:object_r:net_dns_prop:s0 tclass=fi
le app:jakhar.aseem.diva
11-20 08:46:08.941 4250 4250 E jakhar.aseem.diva: Unknown bits set in runtime_flags: 0x8000
11-20 08:46:08.955 4250 4250 W jakhar.aseem.diva: Unexpected CPU variant for X86 using defaults: x86
11-20 08:46:09.322 4250 4250 I jakhar.aseem.diva: The ClassLoaderContext is a special shared library.
11-20 08:46:09.834 4250 4250 I RenderThread: type=1400 audit(0.0:248): avc: denied { write } for name="property_service" dev="tmpfs" ino=9232 scontext=u:r:untrusted_app_25:s0:c512,c768 tcontext=u:object_r:property_socket:s0 tclass=sock
file permissive=1 app:jakhar.aseem.diva
11-20 08:46:09.854 4250 4250 I RenderThread: type=1400 audit(0.0:249): avc: denied { connecto } for path="/dev/socket/property_service" scontext=u:r:untrusted_app_25:s0:c512,c768 tcontext=u:init:s0 tclass=unix_stream_socket permissiv
e=1 app:jakhar.aseem.diva
11-20 08:46:09.856 4250 4271 D libEGL : Emulator has host GPU support, qemu.gles is set to 1.
11-20 08:46:09.856 4250 4271 D libEGL : loaded /vendor/lib/egl/libEGL_emulation.so
11-20 08:46:10.000 4250 4271 D libEGL : loaded /vendor/lib/egl/libGLESv1_CM_emulation.so
11-20 08:46:10.054 4250 4271 D libGGL : loaded /vendor/lib/egl/libGLESv2_emulation.so
11-20 08:46:10.102 4250 4250 W jakhar.aseem.diva: Accessing hidden method Landroid/view/View;~>computeFitSystemWindows(Landroid/graphics/Rect;Landroid/graphics/Rect;)Z (greylist, reflection, allowed)
11-20 08:46:10.109 4250 4250 W jakhar.aseem.diva: Accessing hidden method Landroid/view/ViewGroup;~>makeOptionalFitsSystemWindows(LJ (greylist, reflection, allowed)
11-20 08:46:10.309 4250 4269 D HostConnection: HostConnection: get() New Host Connection established 0xe74f3eab, pid 4250, tid 4269
11-20 08:46:10.332 4250 4269 D HostConnection: HostComposition ext ANDROID_EMU_sync_buffer_data ANDROID_EMU_async_unmap_buffer ANDROID_EMU_sync_buffer_data GL_OES_egl_image_external_essl3 GL_OES_vertex
_array_object GL_KHR_texture_compression_astc_ldr ANDROID_EMU_host_side_tracing ANDROID_EMU_async_frame_commands ANDROID_EMU_gles_max_version_3_1
11-20 08:46:10.332 4250 4269 E : open_verbose:32: Could not open '/dev/goldfish_pipe': No such file or directory
11-20 08:46:10.332 4250 4269 W : Process pipe failed
11-20 08:46:10.585 4250 4269 W OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...
11-20 08:46:10.639 4250 4269 D EGL_emulation: eglCreateContext: 0xe74f3eab: maj 3 min 1 rcv 4
11-20 08:46:10.848 4250 4269 E : open_verbose:32: Could not open '/dev/goldfish_pipe': No such file or directory
11-20 08:46:10.853 4250 4269 D EGL_emulation: eglMakeCurrent: 0xe74f3eab: ver 3 1 (tinfo 0xdc10f7b0) (first time)
11-20 08:46:10.871 4250 4269 W Gralloc3: mapper 3.x is not supported
11-20 08:46:10.893 4250 4269 D HostConnection: createimique: call
11-20 08:46:10.893 4250 4269 D HostConnection: HostConnection: get() New Host Connection established 0xdc107b40, pid 4250, tid 4269

```

Then we add a value in insecure logging page on diva





The credit card number displayed in logcat

```
813898418985, PerformTraversalsStart=8813898877982, DrawStart=8813918005273, SyncQueued=8813918465667, SyncStart=8815939002914, IssueDrawCommandsStart=8815939002914, QueueBufferDuration=3338000,
11-21 13:01:22.941 3487 3538 I OpenGLRenderer: Davey! duration=1393ms; Flags=0, IntendedVsync=8826512179360, Vsync=8826562179358, OldestInputEvent=9223372036854775807, PerformTraversalsStart=8826566859525, DrawStart=8826626659584, SyncQueued=8826626872460, SyncStart=8826662801607, IssueDrawCommandsStart=8826662801607, QueueBufferDuration=11795000,
11-21 13:01:24.631 3487 3538 I OpenGLRenderer: Davey! duration=1854ms; Flags=0, IntendedVsync=8826995512674, Vsync=8827145512668, OldestInputEvent=9223372036854775807, PerformTraversalsStart=8827159391258, DrawStart=8827159689899, SyncQueued=8827159920655, SyncStart=8827941856068, IssueDrawCommandsStart=8827941856068, QueueBufferDuration=20833000,
11-21 13:01:45.906 3487 3538 I OpenGLRenderer: Davey! duration=1359ms; Flags=0, IntendedVsync=8849545615268, Vsync=8849545615268, OldestInputEvent=9223372036854775807, PerformTraversalsStart=8849548869806, DrawStart=8849549273489, SyncQueued=8849549586098, SyncStart=8849550967838, IssueDrawCommandsStart=8849550967838, QueueBufferDuration=2746000,
11-21 13:01:47.466 3487 3487 I Choreographer: Skipped 33 frames! The application may be doing too much work on its main thread.
11-21 13:01:48.845 3487 3487 I Choreographer: Skipped 42 frames! The application may be doing too much work on its main thread.
11-21 13:01:48.900 3487 3538 I OpenGLRenderer: Davey! duration=760ms; Flags=0, IntendedVsync=8853137185836, Vsync=8853837185808, OldestInputEvent=9223372036854775807, PerformTraversalsStart=8853847039266, DrawStart=8853847147939, SyncQueued=8853856481807, SyncStart=8853859207013, IssueDrawCommandsStart=8853859207013, QueueBufferDuration=514000, QueueBufferDuration=4312000,
11-21 13:01:50.614 3487 3487 W ActivityThread: handleWindowVisibility: no activity for token android.os.BinderProxy@5553034
11-21 13:01:58.618 3487 3487 E diva-log: Error while processing transaction with credit card: 9813456798745678
```

## 5.2 Hardcoding issue

Description	Hardcoding occurs when sensitive information, such as API keys, passwords, encryption keys, or other confidential data, is directly embedded in the mobile application's source code. This makes the information accessible to anyone who can decompile or reverse-engineer the app. In here using jadx-cli tool for tool used to decompile Android APK files and view the source . in here we take the encryption key from source code
Impacts	<p>Technical impacts :</p> <ul style="list-style-type: none"> <li>• <b>Data Leakage:</b> Sensitive data can be extracted from decompiled source code.</li> <li>• <b>Unauthorized Access:</b> Attackers can misuse hardcoded credentials to gain access to systems.</li> <li>• <b>Security Risks:</b> Increases vulnerability to attacks like credential stuffing or brute force.</li> <li>• <b>Reputation Damage:</b> Exposing sensitive data can damage the app's reputation and user trust.</li> </ul> <p>User impacts :</p> <ul style="list-style-type: none"> <li>• <b>Privacy Breach:</b> Exposes users' personal data.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Account Compromise:</b> Attackers can access user accounts.</li> <li>• <b>Data Loss:</b> Sensitive user data may be deleted or leaked.</li> <li>• <b>Loss of Trust:</b> Users may abandon the app due to security concerns.</li> </ul>
Mitigations	<ul style="list-style-type: none"> <li>• <b>Avoid Hardcoding:</b> Do not embed sensitive information like credentials or API keys directly in the code.</li> <li>• <b>Use Secure Storage:</b> Store sensitive data in secure environments, such as encrypted storage or secure vaults.</li> <li>• <b>Fetch Dynamically:</b> Retrieve sensitive data dynamically from a secure server or API during runtime.</li> <li>• <b>Apply Encryption:</b> Encrypt sensitive data at rest and in transit to prevent unauthorized access.</li> </ul>

## Proof of concept :

Firstly we have jadx-CLI : <https://github.com/skylot/jadx/releases/tag/v1.5.1>

Then ,

```

amjad@highfly: ~/jadx-1.5.1/bin
$ ls
diva-beta.apk  jadx  jadx.bat  jadx-gui  jadx-gui.bat

amjad@highfly: ~/jadx-1.5.1/bin
$ ./jadx diva-beta.apk
Picked up JAVA_OPTIONS: -Dant.useSystemAAFontSettings=on -Dswing.aatext=true
INFO - Loading ...
INFO - processing ...
INFO - done

amjad@highfly: ~/jadx-1.5.1/bin
$ ls
diva-beta  diva-beta.apk  jadx  jadx.bat  jadx-gui  jadx-gui.bat

amjad@highfly: ~/jadx-1.5.1/bin
$ cd diva-beta

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta
$ ls
resources  sources

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta
$ cd sources

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta/sources
$ ls
android  jakhar

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta/sources
$ cd jakhar

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta/sources/jakhar
$ ls
aseem

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta/sources/jakhar
$ cd aseem

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta/sources/jakhar/aseem
$ ls
diva

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta/sources/jakhar/aseem
$ cd diva

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta/sources/jakhar/aseem/diva
$ ls
AccessControlActivity.java  AccessControlNotesActivity.java  BuildConfig.java  HardcodeActivity.java  InsecureDataStorage1Activity.java  InsecureDataStorage4Activity.java  NotesProvider.java
AccessControl2Activity.java  APICreds2Activity.java  DivaJaki.java  InputValidation2IRISchemeActivity.java  InsecureDataStorage2Activity.java  LogActivity.java  R.java
AccessControl3Activity.java  APICreds3Activity.java  Hardcode2Activity.java  InputValidation3Activity.java  InsecureDataStorage3Activity.java  MainActivity.java  SQLInjectionActivity.java

```

Take all source code and check it ,

```

[amjad@highfly]~/sources/jahkar/aseem/diva
$ ls
AccessControl1Activity.java  AccessControl3NotesActivity.java  BuildConfig.java  HardcodeActivity.java  InsecureDataStorage1Activity.java  InsecureDataStorage4Activity.java  NotesProvider.java
AccessControl2Activity.java  APICreds2Activity.java  DivaJni.java  InputValidation2URISchemeActivity.java  InsecureDataStorage2Activity.java  LoginActivity.java  R.java
AccessControl3Activity.java  APICredsActivity.java  Hardcode2Activity.java  InputValidation3Activity.java  InsecureDataStorage3Activity.java  MainActivity.java  SQLInjectionActivity.java

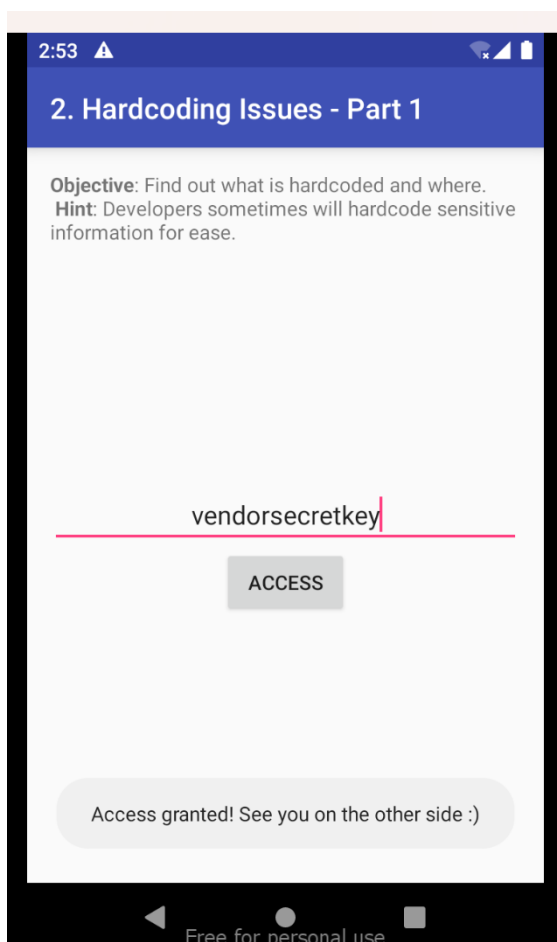
[amjad@highfly]~/sources/jahkar/aseem/diva
$ cat HardcodeActivity.java
package jahkar.aseem.diva;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

/* loaded from classes.dex */
public class HardcodeActivity extends AppCompatActivity {
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.BaseFragmentActivity$DoNot, android.app.Activity
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hardcode);
    }

    public void access(View view) {
        EditText hkey = (EditText) findViewById(R.id.hkey);
        if (hkey.getText().toString().equals("vendorsecretkey")) {
            Toast.makeText(this, "Access granted: see you on the other side :)", 0).show();
        } else {
            Toast.makeText(this, "Access denied! See you in hell :D", 0).show();
        }
    }
}

```



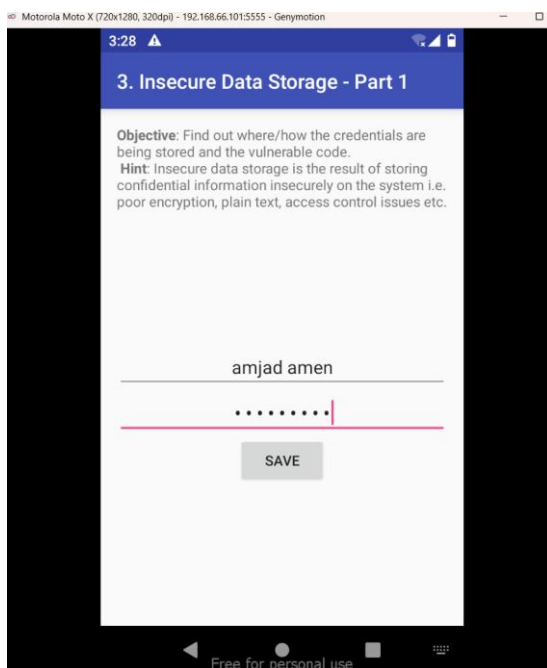
## 5.3 Sensitive Data stored in shared preferences

Description	This vulnerability arises when sensitive information, such as user credentials, tokens, or personal data, is stored insecurely on the device. Local storage mechanisms like Shared Preferences, SQLite databases, or temporary files .in here we take credentials in shared preferences
Impacts	<p>Technical impact:</p> <ul style="list-style-type: none"> <li>• <b>Data Leakage:</b> Sensitive information (e.g., credentials, tokens, personal data) can be easily accessed by attackers through reverse engineering or device compromise.</li> <li>• <b>Unauthorized Access:</b> Exposed credentials or tokens can allow attackers to gain unauthorized access to backend systems or user accounts.</li> <li>• <b>Privilege Escalation:</b> Attackers may misuse stored data to escalate privileges within the application or system.</li> </ul> <p>User impacts:</p> <ul style="list-style-type: none"> <li>• <b>Privacy Breach:</b> Users' sensitive information, such as personal details or financial data, can be exposed.</li> <li>• <b>Account Compromise:</b> Attackers can gain unauthorized access to user accounts.</li> <li>• <b>Identity Theft:</b> Exposed data may be misused for fraudulent activities like identity theft.</li> </ul>
Mitigations	<ul style="list-style-type: none"> <li>• <b>Avoid Storing Sensitive Data Locally:</b> Only store non-sensitive information on the device if possible.</li> <li>• <b>Use Secure Storage APIs:</b> Utilize platform-provided secure storage mechanisms, such as Android's</li> </ul>

	<p>Encrypted Shared Preferences or iOS Keychain.</p> <ul style="list-style-type: none"><li>• <b>Encrypt Stored Data:</b> Implement strong encryption for all sensitive data stored locally, including files, databases, and preferences.</li></ul>
--	--

## Proof of concept :

Fill the values ,



adb shell command : **adb shell**

```

amjad@highfly: ~/jdk-15.1/bin/diva-beta/sources/jakhar/aseem/diva
--(amjad@highfly) [~/sources/jakhar/aseem/diva]
$ adb shell
vbox86p:/ # cd data
vbox86p:/data # ls
adb_app  app-lib  backup  dalvik-cache  gsi  media  misc_ce  ota  preloads  rollback  rollback-observer  ss  system_de  user  vendor_ce
bzip  app-ephemeral  app-staging  cache  dm  lost+found  misc  nfc  per_boot  resource-cache  server_configurable_flags  system_ce  unencrypted_vendor
vbox86p:/data # cd data
vbox86p:/data/data # ls
android  com.android.externalstorage  com.android.gms  com.android.theme.color.green  com.android.webview
com.android.filemanager  com.android.galeriad  com.android.pacprocessor  com.android.theme.color.ocean  com.app.damvulnerablebank
com.android.backupconfirm  com.android.hotspot2  com.android.permissioncontroller  com.android.theme.color.orchid  com.example.android.livcubes
com.android.bips  com.android.htmlviewer  com.android.phone  com.android.theme.color.purple  com.genymotion.genyid
com.android.bluetooth  com.android.inputmethod  com.android.printservice.recommendation  com.android.theme.color.space  com.genymotion.settings
com.android.bluetoothhidiservice  com.android.inputmethod.latin  com.android.printspooler  com.android.theme.font.nerofontsource  com.genymotion.superuser
com.android.bookmarkprovider  com.android.internal.display.cutout.emulation.corner  com.android.providers.blockednumber  com.android.theme.icon.roundedrect  com.genymotion.systempatcher
com.android.calendar  com.android.internal.display.cutout.emulation.double  com.android.providers.calendar  com.android.theme.icon.squircle  com.genymotion.tasklocker
com.android.calllogbackup  com.android.internal.display.cutout.emulation.tall  com.android.providers.contacts  com.android.theme.icon.teardrop  com.google.android.apps.restore
com.android.camera2  com.android.internal.systemui.navbar.gestural  com.android.providers.downloads  com.android.theme.icon_pack.circular.android  com.google.android.backuptransport
com.android.captiveportallogin  com.android.internal.systemui.navbar.gestural_extra_wide_back  com.android.providers.downloads.ui  com.android.theme.icon_pack.circular.launcher  com.google.android.carriersetup
com.android.carrierconfig  com.android.internal.systemui.navbar.gestural_narrow_back  com.android.providers.media  com.android.theme.icon_pack.circular.settings  com.google.android.configdator
com.android.carrierdefaultapp  com.android.internal.systemui.navbar.gestural_wide_back  com.android.providers.settings  com.android.theme.icon_pack.circular.systemui  com.google.android.ext.services
com.android.cellbroadcastreceiver  com.android.internal.systemui.navbar.threebutton  com.android.providers.telephony  com.android.theme.icon_pack.circular.themepicker  com.google.android.ext.shared
com.android.certinstaller  com.android.internal.systemui.navbar.twobutton  com.android.providers.userdictionary  com.android.theme.icon_pack.filled.android  com.google.android.feedback
com.android.companiondevicemanager  com.android.keychain  com.android.proxyhandler  com.android.theme.icon_pack.filled.launcher  com.google.android.gms
com.android.contacts  com.android.launcher3  com.android.quicksearchbox  com.android.theme.icon_pack.filled.settings  com.google.android.gsf
com.android.cts.ctsshim  com.android.localtransport  com.android.se  com.android.theme.icon_pack.filled.systemui  com.google.android.launcher.layouts.genymotion
com.android.cts.priv.ctsshim  com.android.location.fused  com.android.server.telecom  com.android.theme.icon_pack.filled.themepicker  com.google.android.oneinitializer
com.android.customlocale2  com.android.managedprovisioning  com.android.settings  com.android.theme.icon_pack.rounded.android  com.google.android.packageinstaller
com.android.deskclock  com.android.messaging  com.android.settings.intelligence  com.android.theme.icon_pack.rounded.launcher  com.google.android.partnersetup
com.android.development_settings  com.android.mms.service  com.android.sharesetupbackup  com.android.theme.icon_pack.rounded.settings  com.google.android.projection.gearhead
com.android.dialer  com.android.modulemetadata  com.android.shell  com.android.theme.icon_pack.rounded.systemui  com.google.android.setupwizard
com.android.documentui  com.android.mtp  com.android.smapdialog  com.android.tracur  com.google.android.syncadapters.calendar
com.android.dreams.basic  com.android.music  com.android.statementservice  com.android.vending  com.google.android.syncadapters.contacts
com.android.dreams.photostable  com.android.musicfx  com.android.storagemanager  com.android.wallpaperbackup  com.google.android.tts
com.android.dynsystem  com.android.networkstack  com.android.systemui  com.android.wallpapercropper  com.nekki.vector
com.android.egg  com.android.networkstack.permissionconfig  com.android.theme.color.black  jakhar.aseem.diva
com.android.email  com.android.nfc  com.android.theme.color.cinnamon  org.chromium.webview_shell
com.android.emergency  com.android.netinetinitializer  com.android.theme.color.cinnamon  com.android.wallpapericker
vbox86p:/data/data # cd jakhar.aseem.diva/
vbox86p:/data/data/jakhar.aseem.diva # ls
cache  code  cache  databases  lib  shared_prefs
vbox86p:/data/data/jakhar.aseem.diva # cd shared_prefs/
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs # ls
jakhar.aseem.diva.preferences.xml
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs # cat jakhar.aseem.diva.preferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name='password'>Amjj34567</string>
  <string name='user'>amjad ameen </string>
</map>
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs #

```

Take a xml file and read it ,use cat command

```

vbox86p:/data/data # cd jakhar.aseem.diva/
vbox86p:/data/data/jakhar.aseem.diva # ls
cache  code  cache  databases  lib  shared_prefs
vbox86p:/data/data/jakhar.aseem.diva # cd shared_prefs/
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs # ls
jakhar.aseem.diva.preferences.xml
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs # cat jakhar.aseem.diva.preferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name='password'>Amjj34567</string>
  <string name='user'>amjad ameen </string>
</map>
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs #

```

## 5.4 Sensitive Data Stored in local storage

Descriptions	This vulnerability occurs when sensitive information, such as user credentials, tokens, or personal data, is stored insecurely in local storage mechanisms like Shared Preferences, SQLite databases, or temporary files. Without proper encryption or security controls, this data becomes accessible to attackers through reverse engineering or device compromise.in here we found the credentials in a databases
Impacts	Technical impacts: <ul style="list-style-type: none"> <li><b>Data Exposure:</b> Sensitive information, like credentials or tokens, can be easily extracted by attackers.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Unauthorized Access:</b> Exposed data may allow attackers to gain access to user accounts or backend systems.</li> <li>• <b>Privilege Escalation:</b> Attackers can use the exposed data to escalate their access rights within the application or system.</li> </ul>
Mitigations	<ul style="list-style-type: none"> <li>• <b>Use Encrypted Shared Preferences:</b> Leverage Android's Encrypted Shared Preferences to securely store sensitive data.</li> <li>• <b>Avoid Storing Sensitive Data Locally:</b> Store sensitive information, such as tokens or credentials, on secure servers instead of local storage.</li> <li>• <b>Implement Strong Encryption:</b> Encrypt all sensitive data before storing it in Shared Preferences.</li> </ul>

## Proof of concept :

```

amjad@highfly: ~/jadx-1.5.1/bin/diva-beta/sources/jakhar/aseem/diva
vbox86p:/data/data/jakhar.aseem.diva # ls
cache code_cache databases lib shared_prefs
vbox86p:/data/data/jakhar.aseem.diva # cd shared_prefs/
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs # ls
jakhar.aseem.diva_preferences.xml
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs # cat jakhar.aseem.diva_preferences.xml
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<map>
  <string name="password">Amjad@aseem</string>
  <string name="user">Amjad@aseem</string>
</map>
vbox86p:/data/data/jakhar.aseem.diva/shared_prefs # cd ..
vbox86p:/data/data/jakhar.aseem.diva # cd ..
/system/bin/sh: cd: ..: inaccessible or not found
127|vbox86p:/data/data/jakhar.aseem.diva # ls
cache code_cache databases lib shared_prefs
vbox86p:/data/data/jakhar.aseem.diva # ls -la
total 64
drwxr-x--x  0 u0_a49 u0_a49    4096 2024-11-20 11:24 .
drwxrwx--x 151 system system 22288 2024-11-19 00:10 ..
drwxrwx--x  2 u0_a49 u0_a49     4096 2024-11-19 00:10 cache
drwxrwx--x  2 u0_a49 u0_a49     4096 2024-11-19 00:10 code_cache
drwxrwx--x  2 u0_a49 u0_a49     4096 2024-11-19 00:10 databases
lrwxrwxrwx  1 root  root       62 2024-11-20 00:30 lib -> /data/app/jakhar.aseem.diva-NzBMQU3jYHPAw6LxZAMDA=/lib/x86
drwxrwx--x  2 u0_a49 u0_a49     4096 2024-11-20 11:48 shared_prefs
vbox86p:/data/data/jakhar.aseem.diva # cd databases/
vbox86p:/data/data/jakhar.aseem.diva/databases # ls -la
total 68
drwxrwx--x  2 u0_a49 u0_a49    4096 2024-11-19 00:48 .
drwxrwx--x  0 u0_a49 u0_a49     4096 2024-11-20 11:24 ..
-rw-rw----  1 u0_a49 u0_a49 20480 2024-11-19 00:10 divanotes.db
-rw-rw----  1 u0_a49 u0_a49    0 2024-11-19 00:10 divanotes.db-journal
-rw-rw----  1 u0_a49 u0_a49 16384 2024-11-20 12:17 id2
-rw-rw----  1 u0_a49 u0_a49    0 2024-11-20 12:17 id2-journal
vbox86p:/data/data/jakhar.aseem.diva/databases # sqlite3 id
id2
id2-journal
vbox86p:/data/data/jakhar.aseem.diva/databases # sqlite3 id2
SQLite version 3.22.0 2018-12-19 01:30:22
Enter ".help" for usage hints.
sqlite> select * from myuser;
...> select * from myuser;
sqlite> select * from myuser;
Amjad@aseem@567
Amjad@aseem (Amjad@aseem@567)
sqlite>

```

In this stage we are exploiting the sqlite and take the credentials ,

## 5.5 Sensitive Data Stored in Temporary Files

<b>Description</b>	During the penetration testing of the DIVA (Damn Insecure and Vulnerable App) mobile application, sensitive credentials were discovered in temporary files. This vulnerability arises when the application improperly stores sensitive data, such as login credentials or tokens, in temporary directories without adequate security measures.
<b>impacts</b>	<ul style="list-style-type: none"> <li>• <b>Credential Compromise:</b> Attackers can access sensitive credentials (e.g., username/password) from temporary files.</li> <li>• <b>Privilege Escalation:</b> Leaked admin credentials could give attackers full control over the app's backend.</li> <li>• <b>Data Leakage:</b> Sensitive user or application data is exposed, increasing security risks.</li> <li>• <b>Exploitation by Malware:</b> Malware could target and extract sensitive data from temporary files.</li> </ul>
<b>mitigation</b>	<ul style="list-style-type: none"> <li>• <b>Avoid Temporary File Storage:</b> Keep sensitive data in memory instead of writing it to disk.</li> <li>• <b>Use Secure Storage:</b> Utilize secure APIs like Android Keystore or iOS Keychain for sensitive data.</li> <li>• <b>Encrypt Data:</b> Encrypt sensitive data before storing it in temporary files if storage is necessary.</li> <li>• <b>Set File Permissions:</b> Apply strict access controls to prevent unauthorized access to temporary files.</li> </ul>



# Proof of concept :

```

amjad@highfly:~$ adb shell
vbox@p:/ # whoami
root
vbox@p:/ # cd data
vbox@p:/data # ls
adb_app  app-lib  backup  dalvik-cache  gsi  media  misc-ce  ota  preloads  rollback  rollback-observer  ss  system  system_de  user  vendor_ce
anr_app-asec  app-private  bootchart  data  local  mediadm  misc_de  ota_package  property  resource-cache  server_configurable_flags  system  system_ce  unencrypted  vendor_de
vbox@p:/data # cd data
vbox@p:/data/data # ls
android
com.amaze.filesmanager
com.android.backupconfirm
com.android.bips
com.android.bluetooth
com.android.bluetoothhidservice
com.android.bookmarksprovider
com.android.calendar
com.android.calllogbackup
com.android.camera2
com.android.captiveportallogin
com.android.carrierconfig
com.android.carrierdefaultapp
com.android.cellbroadcastreceiver
com.android.certinstaller
com.android.companiondevicemanager
com.android.contacts
com.android.cts.ctsshin
com.android.cts.priv.ctsshin
com.android.customlocalization
com.android.deskclock
com.android.development_settings
com.android.dialer
com.android.documentsui
com.android.dreams.basic
com.android.dreams.phototable
com.android.dynsystem
com.android.egg
com.android.email
com.android.emergency
vbox@p:/data/data # cd jakhar.aseem.diva/
vbox@p:/data/data/jakhar.aseem.diva # ls
cache  code  cache  databases  lib  shared_prefs  userinfo2441522020427344029tmp  userinfo5458653638369407880tmp  userinfo7534009624354127tmp
vbox@p:/data/data/jakhar.aseem.diva # ls -la
total 88
drwxr-x--x 6 u0_a49 u0_a49 4096 2024-11-21 00:15 .
drwxr-x--x 151 system system 12288 2024-11-19 00:10 ..
drwxr-x--x 2 u0_a49 u0_a49 4096 2024-11-19 00:10 cache
drwxr-x--x 2 u0_a49 u0_a49 4096 2024-11-19 00:10 code
drwxr-x--x 2 u0_a49 u0_a49 4096 2024-11-19 00:10 code cache

```

```

vbox@p:/data/data # cd data
vbox@p:/data/data # ls
android
com.amaze.filesmanager
com.android.backupconfirm
com.android.bips
com.android.bluetooth
com.android.bluetoothhidservice
com.android.bookmarksprovider
com.android.calendar
com.android.calllogbackup
com.android.camera2
com.android.captiveportallogin
com.android.carrierconfig
com.android.carrierdefaultapp
com.android.cellbroadcastreceiver
com.android.certinstaller
com.android.companiondevicemanager
com.android.contacts
com.android.cts.ctsshin
com.android.cts.priv.ctsshin
com.android.customlocalization
com.android.deskclock
com.android.development_settings
com.android.dialer
com.android.documentsui
com.android.dreams.basic
com.android.dreams.phototable
com.android.dynsystem
com.android.egg
com.android.email
com.android.emergency
vbox@p:/data/data # cd jakhar.aseem.diva/
vbox@p:/data/data/jakhar.aseem.diva # ls
cache  code  cache  databases  lib  shared_prefs  userinfo2441522020427344029tmp  userinfo5458653638369407880tmp  userinfo7534009624354127tmp
vbox@p:/data/data/jakhar.aseem.diva # ls -la
total 88
drwxr-x--x 6 u0_a49 u0_a49 4096 2024-11-21 00:15 .
drwxr-x--x 151 system system 12288 2024-11-19 00:10 ..
drwxr-x--x 2 u0_a49 u0_a49 4096 2024-11-19 00:10 cache
drwxr-x--x 2 u0_a49 u0_a49 4096 2024-11-19 00:10 code
drwxr-x--x 2 u0_a49 u0_a49 4096 2024-11-19 00:10 code cache
drwxr-x--x 1 root root 4096 2024-11-21 00:18 databases
drwxr-x--x 1 u0_a49 u0_a49 4096 2024-11-21 00:18 lib -> /data/app/jakhar.aseem.diva-NzBX0U5JyHfPmXpLzZMDA=/lib/x86
-rw-r--r-- 1 u0_a49 u0_a49 23 2024-11-21 00:15 userinfo2441522020427344029tmp
-rw-r--r-- 1 u0_a49 u0_a49 23 2024-11-21 00:19 userinfo5458653638369407880tmp
-rw-r--r-- 1 u0_a49 u0_a49 22 2024-11-20 12:17 userinfo7534009624354127tmp
vbox@p:/data/data/jakhar.aseem.diva # cat userinfo2441522020427344029tmp
cat: userinfo2441522020427344029tmp: No such file or directory
vbox@p:/data/data/jakhar.aseem.diva #

```

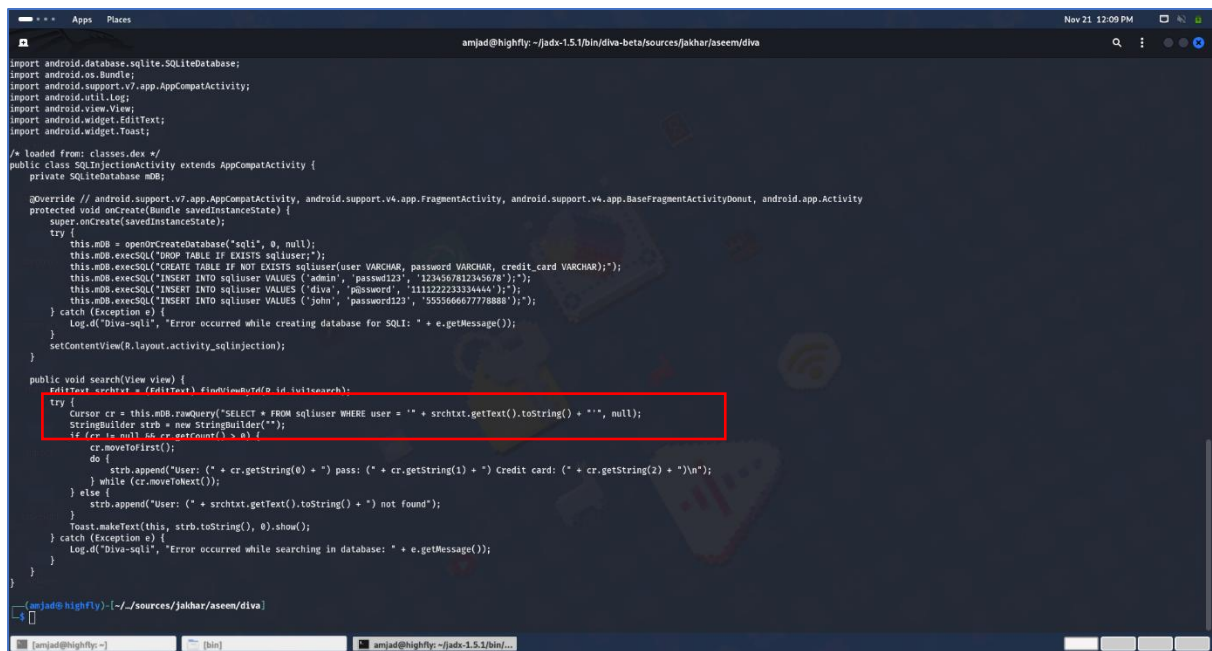
In here we take the credentials in a temp file and compromise ,

## 5.6 input validation issue (sql injection )

<b>Description</b>	During penetration testing of the DIVA (Damn Insecure and Vulnerable App) mobile application, an SQL Injection vulnerability was identified due to inadequate input validation. User input was found to be directly concatenated into SQL queries, allowing attackers to manipulate the database and extract sensitive information. Credentials were retrieved by exploiting this vulnerability. In here we using 'OR '1'='1 query for compromise it .
<b>Impacts</b>	<p>Technical impacts :</p> <ul style="list-style-type: none"> <li>• <b>Unauthorized Database Access:</b> Attackers can gain unauthorized access to the database and sensitive information.</li> <li>• <b>Data Manipulation:</b> Attackers can modify, delete, or add records in the database.</li> <li>• <b>Data Leakage:</b> Sensitive user data (e.g., credentials) can be exposed to attackers.</li> <li>• <b>Privilege Escalation:</b> Exploitation may lead to unauthorized access to higher privilege levels.</li> <li>• <b>System Compromise:</b> Attackers could execute arbitrary commands or escalate attacks via database vulnerabilities.</li> </ul> <p>User Impacts :</p> <ul style="list-style-type: none"> <li>• <b>Credential Theft:</b> User credentials, such as usernames and passwords, may be stolen.</li> <li>• <b>Account Takeover:</b> Attackers could impersonate users and access their accounts.</li> <li>• <b>Loss of Privacy:</b> Sensitive personal information could be exposed or leaked.</li> </ul>
<b>Mitigations</b>	<ul style="list-style-type: none"> <li>• <b>Use Parameterized Queries:</b> Always use prepared statements with parameterized queries to prevent SQL Injection.</li> </ul>

- **Validate and Sanitize Inputs:** Ensure all user inputs are validated and sanitized to allow only expected data types and formats.
- **Escape Special Characters:** Properly escape special characters (like ', ", ;) in user input to prevent query manipulation.

## Proof of concept :



```

import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

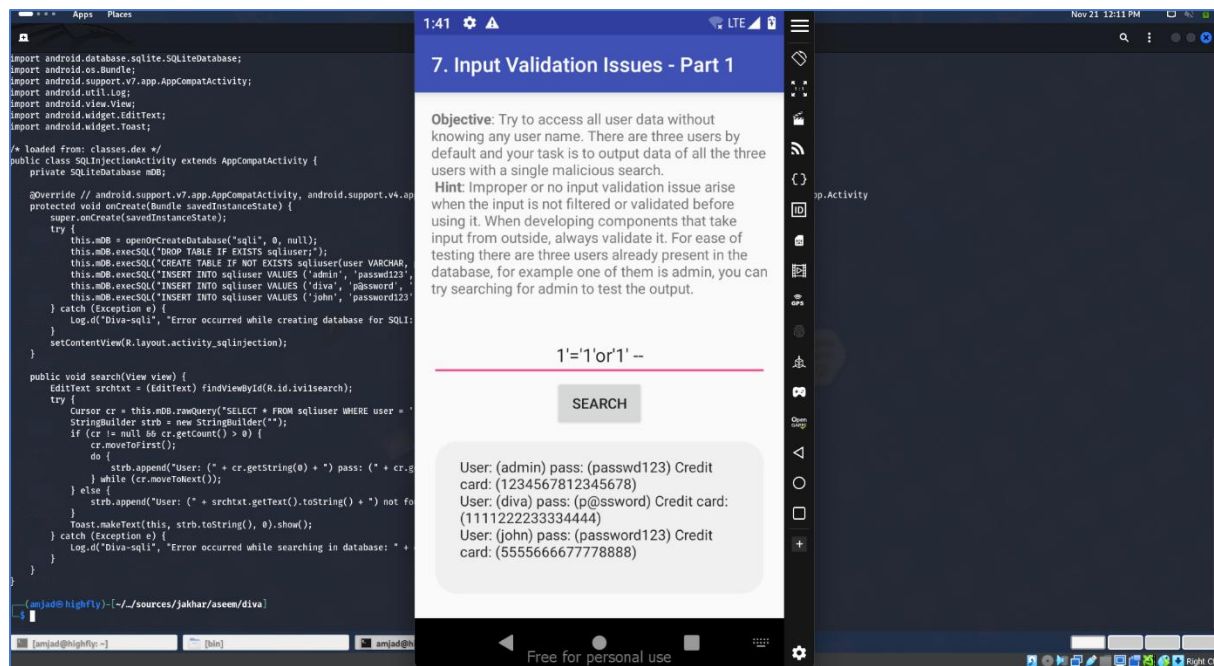
/* loaded from: classes.dex */
public class SQLInjectionActivity extends AppCompatActivity {
    private SQLiteDatabase mDB;

    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.BaseFragmentActivity$Donut, android.app.Activity
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try {
            this.mDB = openOrCreateDatabase("sql", 0, null);
            this.mDB.execSQL("DROP TABLE IF EXISTS sqluser;");
            this.mDB.execSQL("CREATE TABLE IF NOT EXISTS sqluser(user VARCHAR, password VARCHAR, credit_card VARCHAR);");
            this.mDB.execSQL("INSERT INTO sqluser VALUES ('admin', 'password123', '1234567812345678');");
            this.mDB.execSQL("INSERT INTO sqluser VALUES ('diva', 'password', '1111222233334444');");
            this.mDB.execSQL("INSERT INTO sqluser VALUES ('john', 'password123', '5555666677778888');");
        } catch (Exception e) {
            Log.d("Divia-sql", "Error occurred while creating database for SQL: " + e.getMessage());
        }
        setContentView(R.layout.activity_sqlinjection);
    }

    public void search(View view) {
        EditText srchtxt = (EditText) findViewById(R.id.txtsearch);
        try {
            Cursor cr = this.mDB.rawQuery("SELECT * FROM sqluser WHERE user = '" + srchtxt.getText().toString() + "'", null);
            StringBuilder strb = new StringBuilder("");
            if (cr.moveToFirst()) {
                do {
                    strb.append("User: (" + cr.getString(0) + ") pass: (" + cr.getString(1) + ") Credit card: (" + cr.getString(2) + ")\n");
                } while (cr.moveToNext());
            } else {
                strb.append("User: (" + srchtxt.getText().toString() + ") not found");
            }
            Toast.makeText(this, strb.toString(), 0).show();
        } catch (Exception e) {
            Log.d("Divia-sql", "Error occurred while searching in database: " + e.getMessage());
        }
    }
}

```

Then inject the query for search bar , then



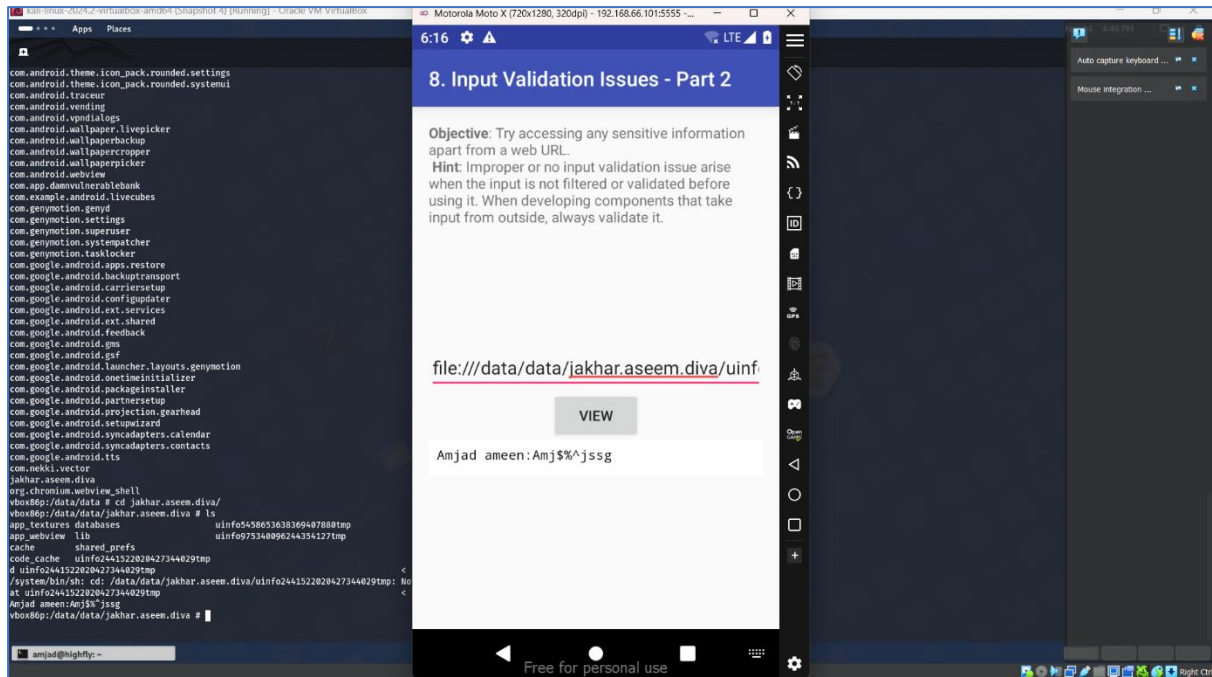
Take all usernames , passwords and sensitive informations

## 5.7 input validation issue ( URL injection )

<b>Description</b>	<p>An URL Injection vulnerability was identified. This issue occurs due to insufficient input validation, allowing attackers to manipulate URL parameters. By injecting malicious URL inputs, attackers could access unauthorized resources or perform actions like retrieving sensitive data, including user credentials. In here we take the credentials</p>
<b>Impacts</b>	<p>Technical impacts:</p> <ul style="list-style-type: none"> <li>• <b>Unauthorized Data Access:</b> Attackers can manipulate URLs to access restricted application resources.</li> <li>• <b>Sensitive Data Exposure:</b> Credentials and other sensitive information may</li> </ul>

	<p>be leaked or stolen from temporary files.</p> <ul style="list-style-type: none"> <li>• <b>Application Integrity Compromise:</b> URL manipulation can alter the application's expected behavior, potentially leading to data corruption or unauthorized actions.</li> <li>• <b>System Vulnerability:</b> An unpatched URL injection vulnerability can allow further attacks, like code execution or privilege escalation.</li> </ul> <p>User Impacts:</p> <ul style="list-style-type: none"> <li>• <b>Credential Theft:</b> User credentials, including usernames and passwords, can be exposed.</li> <li>• <b>Account Compromise:</b> Attackers can impersonate legitimate users and access sensitive data.</li> <li>• <b>Privacy Violations:</b> Users' personal information may be leaked or accessed without permission.</li> </ul>
Mitigations	<ul style="list-style-type: none"> <li>• <b>Validate and Sanitize Inputs:</b> Ensure all URL parameters are validated and sanitized.</li> <li>• <b>Use Whitelisting:</b> Only accept known safe values for URL parameters.</li> <li>• <b>Encode URL Inputs:</b> Properly encode user inputs before adding them to URLs.</li> <li>• <b>Avoid Storing Sensitive Data in Temp Files:</b> Do not store credentials or sensitive data in temporary files.</li> </ul>

**Proof of concept :**



Use URL : <file:///data/data/jakhar.aseem.diva/uinfo2441522020427344029tmp>

## 6. Conclusion

The penetration testing of the DIVA mobile application uncovered seven vulnerabilities, including one critical (SQL Injection), three high (insecure logging, sensitive data stored in shared preferences and local storage), and three medium (sensitive data in temporary files, hard-coded credentials, and URL injection). These vulnerabilities pose significant risks to user data and application security. Immediate remediation is recommended, focusing on secure data storage, proper input validation, removal of hard-coded sensitive information, and reviewing logging practices to strengthen the application's security posture and protect against exploitation.