# External Application Penetration Testing Technical Report For King Saud University

Item	Description
Document Title	Mobile application penetration testing
Requestor	King Saud University   College of Computer and Information Sciences   Information Technology department   IT 371: Application security   Doctor Nourah madi
Author/s	Jumanah alDawsari – Nouf Alsadhan
Date Created	23/5/2023

# Contents

1	EXE	CUTIVI	E SUMMARY	4
	1.1	Introd	duction	4
	1.2		e	
	1.3	Risk	Rating	4
	1.4		at Security Level	
	1.5		nary Table	
	1.6		nary Graph	
	1.7		Finding	
2	Con	CLUSIO	ON	6
3	ME	HODO	LOGY	8
4	DET	AILED	FINDINGS	12
	4.1	Limit	tations	12
	4.2	Techi	nical Description of Findings	12
		4.2.1	Debug Enabled For App	
		4.2.2	Clear text traffic is Enabled For App	
		4.2.3	Insecure WebView Implementation	14
		4.2.4	The App logs information	
		4.2.5	Files may contain hard coded sensitive information	16
Αī	PPEND	IX A : A	AROUT THE TEAM	17

#### 1 Executive Summary

#### 1.1 Introduction

In this project we implemented black box pentesting using some tools such as mobSF, jadx, and ADB.

Blackbox pen testing is security testing where the tester simulates an external attacker with no prior knowledge of the system being tested in order to identify potential vulnerabilities.

#### 1.2 Scope

The specific scope of this project includes performing the Application Penetration test for the specified duration on the below mentioned applications.

We used Samsung android phone as emulator.

Jadx: we used this tool to decompile the APK and read the source code.

MobSF: we used this tool to find the vulnerabilities.

ADB: we used this tool to communicate with the Android phone through terminal and to find emulator logs.

<b>Application Name</b>	Platform	Version	Environment	Approach
InsecureShop	Android	1.0	Windows	Black box Penetration testing

#### 1.3 Risk Rating

The risk rating for the issues and their impact on the operation of the organization is explained in the table 1 below. The overall risk rating reported will be based on vulnerability identification with its potential to be exploited by adversaries.

In general, the following factors were considered to arrive at the risk rating for vulnerability:

- ➤ Technical Impact: The extent to which an attacker may gain access to a system and the severity of it on the application. This metric will take the security triad CIA (Confidentiality, Integrity and Availability) values into account.
- ➤ Likelihood: This metric will take the Popularity and Simplicity of an exploit into consideration.
  - Popularity describes the existing or potential frequency of exploitation of the vulnerability.
  - Simplicity is the amount of effort required to exploit the vulnerability.

	ty			
Technical Impact	HIGH	MEDIUM	HIGH	CRITICAL
(Confidentiality, Integrity,	MEDIUM	LOW	MEDIUM	HIGH

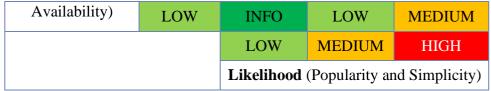


Table 1 Risk Severity

#### 1.4 Threat Security Level

Vulnerabilities are categorized as Critical, High, Medium, Low and Informational.

**Critical:** Severe Impact on the affected application. They require immediate attention and resolution. Successful exploitation may provide the attacker **access to critical data**.

**High:** Severe Impact on the affected application. They require immediate attention. They are relatively easy for attackers to exploit and may provide them with **full control of the affected application.** 

**Medium:** Moderate impact on the affected application. They are often **harder to exploit** and may not provide the same access to affected application.

Low: Limited impact on the affected application. They provide information to attackers that may assist them in mounting **subsequent attacks on the affected applications.** These should also be fixed in a timely manner, but are not as urgent as the other vulnerabilities.

**Informational:** It exposes information that target stake holders simply need to be aware of. These are for findings that are very difficult to exploit in practice.

#### 1.5 Summary Table

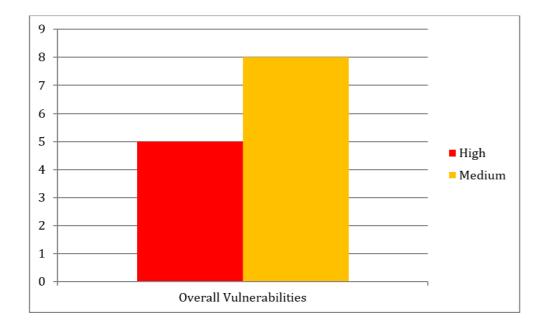
The table below shows the summary of vulnerabilities disclosed during the Penetration Testing.

**Mobile Application Penetration Testing** 

Critical	High	Medium	Low
-	5	8	-

#### 1.6 Summary Graph

The following bar graph highlights the total number of vulnerabilities discovered during the penetration testing.



**Figure 1 Application Penetration Testing** 

#### 1.7 Key Finding

No.	Vulnerabilities Discovered	Platform	Severity Level
1	Debug Enabled For App [android:debuggable=true]	Android	High
2	Clear text traffic is Enabled For App [android:usesCleartextTraffic=true]	Android	High
3	Insecure WebView Implementation	Android	High
4	Files may contain hard coded sensitive information like usernames, password, keys etc.	Android	Medium
5	The App logs information	Android	Info

#### 2 Conclusion

We discovered five vulnerabilities while implementing penetration testing. To enhance the security of the application, we recommend prioritizing and implementing our advice on mitigating these vulnerabilities. This will help prevent attackers from easily exploiting or leveraging the vulnerabilities to their advantage.

- **Debug Enabled For App:** Setting the android:debuggable flag to true enables an attacker to debug the application, making it easier for them to gain access to parts of the application that should be kept secure. Always make sure to set the android:debuggable flag to false when shipping your application [1].
- Clear text traffic is Enabled For App: When android:usesCleartextTraffic is set to true, the application will allow outgoing requests over HTTP, resulting in a potential data leakage or a Man-In-The-Middle (MITM) attack. By setting the value to false, the application will refuse the app's requests to use cleartext traffic [2].
- **Insecure WebView Implementation:** The insecure WebView implementation vulnerability refers to a situation where an app is using a WebView component to display web content, but the WebView is not configured securely. Specifically, this vulnerability occurs when the WebView is set to ignore SSL certificate errors or to accept any SSL certificate, making the app vulnerable to Man-in-the-Middle (MITM) attacks. Always use HTTPS to load web content in WebView. in addition, WebView should always validate SSL certificates presented by the server [3].
- Files may contain hard coded sensitive information: Hard-coded credentials typically create a significant hole that allows an attacker to bypass the authentication that has been configured by the product administrator. This hole might be difficult for the system administrator to detect. Even if detected, it can be difficult to fix, so the administrator may be forced into disabling the product entirely.
  - store passwords, keys, and other credentials outside of the code in a strongly-protected, encrypted configuration file or database that is protected from access by all outsiders, including other local users on the same system. Properly protect the key (CWE-320). If you cannot use encryption to protect the file, then make sure that the permissions are as restrictive as possible. In Windows environments, the Encrypted File System (EFS) may provide some protection [5].
- The App logs information: The vulnerability reported by MOBSF, "The App logs information. Sensitive information should never be logged," means that the app is logging sensitive information, such as usernames, passwords, or other confidential data, in a way that is not secure. Logging is an important part of app development for debugging and troubleshooting, but it's essential to ensure that sensitive information is not being logged. Developers should ensure that sensitive data is not being written into logs, and if it is necessary to log such data, it should be done in a secure and encrypted manner [4].

### 3 Methodology

Our methodology on Mobile penetration testing is based on Mobile Open Web Application Security Project (OWASP); our assessment methodology to carry out the mobile penetration testing includes 2 phases or any preparations:

1- In order to start working on the project, one of the steps is getting a real Android phone or create an emulated one. We already have a Samsung phone that we are using to work in our graduation project, so this step is already ready and we can start working. We can run the InsecureShop.apk by downloading it in our Samsung phone and it worked correctly.



2- we had to install some tools:

#### 2.1 Jadx[6]

Jadx is an open-source decompiler used for reverse engineering Android apps. It is designed to convert Android app binaries into a human-readable format, making it easier for developers and security researchers to analyze and understand the code. With Jadx, we can view the source code of an Android app and explore its structure, classes, and methods. This can be useful for a variety of purposes, such as debugging, modifying, or extending an app's functionality, or identifying security vulnerabilities.

We downloaded Jadx, and opened the Jadx-GUI in order to decompile and read the source code of InsecureShop.apk.

```
🚳 *New Project - jadx-gui
 a 🕫 🛍 🔓 💲 🖨 🥕 🔍 💠 😂 🕒 🥕
                ⊕ G CartListActivity
                                                                             mWebViewClient 🗶 📵 com.insecureshop.ChooserActivity 🧝 😉 com.insecureshop.util.Prefs χ 📵 com.insecureshop.ResultActivity 💥 📳
                 G ChooserActivity
                                                                            @ Re package com.insecureshop;
                    ⊕ CustomReceiver

    DataBinderMapper

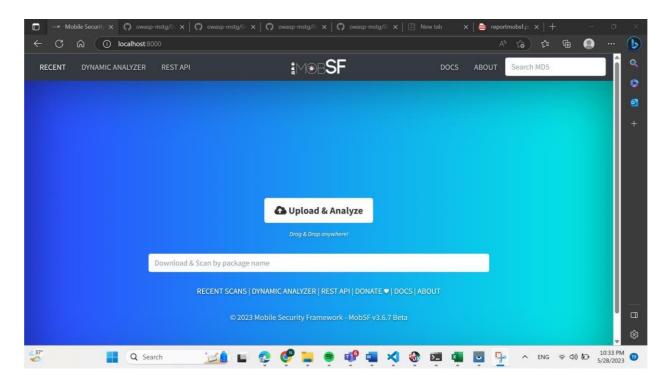
                   ⊕ DataBindingInfo
                                                                                            import android.os.Bundle;
                                                                                           import android.view.View;
import android.view.View;
import android.vapccompata.apr.AppCompatActivity;
import java.util.HashMap;
import kotlin.Metadata;
                 ⊕ ⊕ InsecureShopApp
                 ⊕ G LoginActivity
                  - G PrivateActivity
                 ⊕ ⊕ ProductAdapter
⊕ ⊕ ProductAdapter$∢
                                                                                            @Metadata(bv = {1, 0, 3}, d1 = {"\u0000\u0018\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u0002\n\u
                  - @ ProductAdapter$c
                  @ ProductDetail
                                                                                                     public void _$_clearFindViewByIdCache() {
                     @ ProductListActiv
                                                                                                                                 hashMap = this._$_findViewCache;
hMap != null) {
                  - GR
                                                                                                             if (hashMap != null)
hashMap.clear();
                             s findViewCa
                             _ $_clearFindV:
                            • $ findCached
                                                                                                     public View _$_findCachedViewById(int i) {
   if (this._$_findViewCache == null) {
      this._$_findViewCache = new HashMap
                               onCreate(Bund
                     ⊕ SendingDataViaA
                    → → WebView2Activity
                  ⊕ ⊕ WebViewActivity
                                                                                                             // View view = (View) this._$_findViewCache.get(Integer.valueOf(i));
if (view != null) {
         m kotlin
     🖶 🌐 kotlinx.android
     net.gotev.uploadserv
                                                                                                                View findViewById = findViewById(i);
     org
                                                                                                               this. $_findViewCache.put(Integer.valueOf(i), findViewById);
return findViewById;
   Resources
           annotation
                                                                                                     /* access modifiers changed from: protected */
@Override // androidx.activity.ComponentActivity, androidx.core.app.ComponentActivity, androidx.appcompat.app.AppCompatActivity
            collections
                                                                                                                   <u>소설</u> 🖫 💀 🧬 📜 📵 🐠 🚾 💋 🚳 💌 🚾 💟 🏪 ^ ENG 🗟 네) 🗗 5/28/2023 💿
```

#### 2.2 MobSF[7]

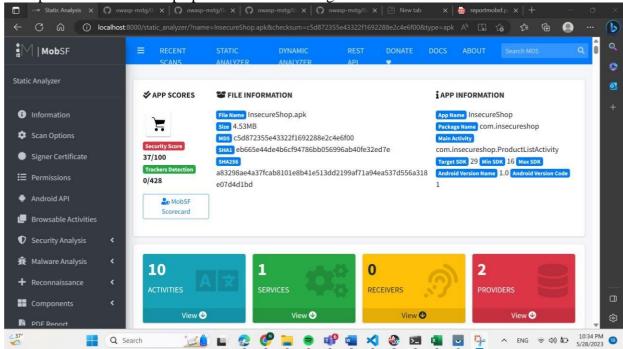
MobSF (Mobile Security Framework) is a free, open-source mobile application security testing framework used for analyzing the security of Android and IOS apps. It provides a variety of dynamic and static analysis tools to identify vulnerabilities and security issues in mobile apps, such as insecure data storage, usage of insecure libraries, and insecure network communication. MobSF can be used both as a command-line tool and as a web-based interface, and supports a range of file formats, including APK, IPA, and APPX. MobSF also integrates with other security tools and frameworks, making it a powerful and flexible solution for mobile app security testing.

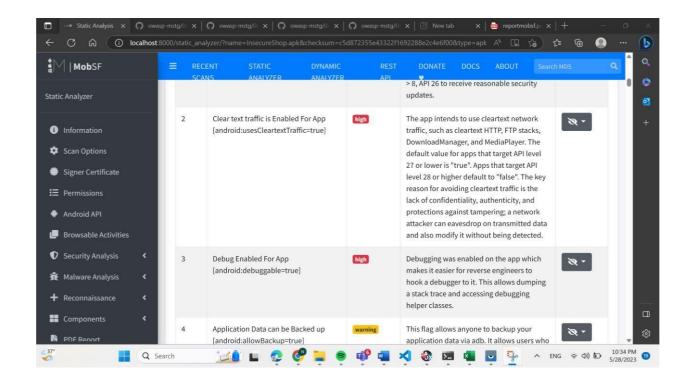
In order to download MobSF, we had to download Java, Python, Visual studio, Git, Open SSL and WkHTMLtox. After downloading all the previous mentioned tools, we started downloading MobSF by writing "git clone" in the command line followed by the link of the github repository, and then we wrote setup.bat and waited until it finished.

After it finished we opened localhost:8000.



We uploaded InsecureShop.apk and started looking for vulnerabilities.

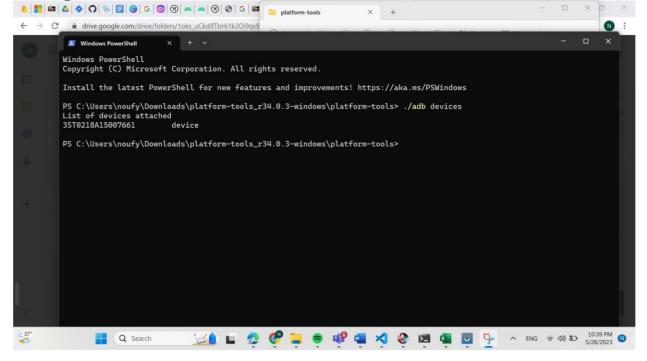




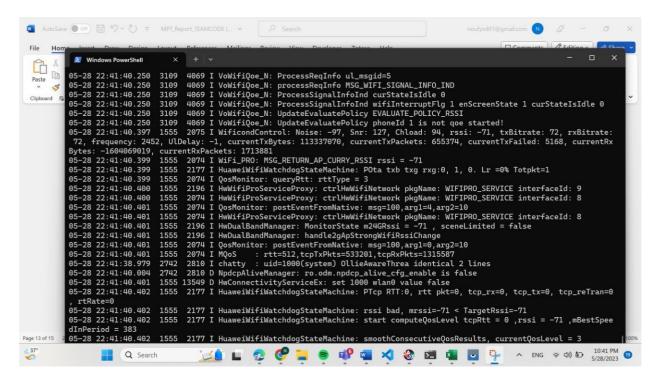
#### 2.2 Android Debug Bridge (ADB)[8]

Android Debug Bridge (ADB) is a command-line tool that allows developers to communicate with an Android device or emulator. ADB is a part of the Android Software Development Kit (SDK) and provides a variety of features for debugging and testing Android apps, such as installing and uninstalling apps, transferring files between a device and a computer, and running shell commands on a device.

After downloading ADB and apk platform tool, we opened the command line to search for our Samsung device, and it showed with its id.



We then started previewing the Android logs by running the command "adb logcat -v threadtime", and we noticed that every click and movement in the phone it directly shows in the logs.



## 4 Detailed Findings

#### 4.1 Limitations

Following were the limitations that we encountered:

- -MobSF requires many requirements(python, jdk, etc..)
- -Installing all the tools consumed a significant amount of memory space.
- -Running the APK file on real phone Emulator require many steps.

## 4.2 Technical Description of Findings

This section explains the details of the identified vulnerability along with technical impact, Proof of Concept, recommendations, and references related to the vulnerability.

#### 4.2.1 Debug Enabled For App

Severity Level	High							
Technical	High	Likelihood		Low				
Impact	High	Popularity	Low	Simplicity	High			
Observation	We noticed Del a significant sec	· ·	ne MobSF in ana	lysis application	which can pose			
Impact		The debug enabled can introduce serious security vulnerabilities that could enable attackers to gain unauthorized access and access sensitive information.						
Platform	Android							

	Proof of Concept							
Q Manifest Analysis  */> Code Analysis  Binary Analysis	3	Debug Enabled For App [android:debuggable=true]	high	Debugging was enabled on the app which makes it easier for reverse engineers to hook a debugger to it. This allows dumping a stack trace and accessing debugging helper classes.				
Recommend ation	Setting the android:debuggable flag to true enables an attacker to debug the application, making it easier for them to gain access to parts of the application that should be kept secure. Always make sure to set the android:debuggable flag to false when shipping your application [1].							
OWASP Reference		https://owasp.org/www-project-mobile-top-10/2016-risks/m10-extraneous-functionality						
Reference	https://de	eveloper.android.com	m/topic/se	ecurity/risks/android-debuggable				

# 4.2.2 Clear text traffic is Enabled For App

Severity Lev	vel High					
Technical	High		Likelihood		High	
Impact			Popularity	High	Simplicity	High
Observation				s and we obser	rve clear text	traffic which
		erious security				
Impact			-	and read sens		
	the netwo		and other conn	idential data, th	iat is being trai	iisiiiittea ovei
Platform	Android					
		P	roof of Conce	pt		
Security Analysis	*			араасы.	_	
A Network Securi	Ly	Clear text traffic is Enabled F [android:usesCleartextTraffi	SALAN CALLED	The app intends to use clear traffic, such as cleartext H		*_
Certificate Anal				DownloadManager, and M default value for apps tha		
Q Manifest Analys	is			27 or lower is "true". Apps that target API level 28 or higher default to "false". The key		
Code Analysis				reason for avoiding clear lack of confidentiality, au		
Binary Analysis				protections against tamp attacker can eavesdrop o		
NIAP Analysis File Analysis				and also modify it withou		
■ File Analysis			<u> </u>			
Recomme	When androi	d:usesCleartex	xtTraffic is set	to true, the ap	plication will	allow
ndation				n a potential d	•	
				ne value to fals	e, the applicat	ion will
0.551.05		<del>-</del>	use cleartext t			
OWASP Reference		•	ject-mobile-to	p-10/2016-risl	ks/m3-insecure	e-
	communicati		rrior com/recir	oes/scw%3Aar	droid%2 A dia	ablad
	nups://senser.s cleartext	securecode war	inor.com/recij	<u> </u>	iui0iu%3Auls	auteu-
	CICUITOAL					

# **4.2.3** Insecure WebView Implementation

Severity	High						
Level							
Technical	hnical High Likelihood			Medium			
Impact		Popularity	High	Simplicity	High		
Observatio	We observed through the use of jadx and MobSF that the implementation of an						
n	insecure WebView in the application which poses a significant security risk.						
Impact	The impact of an insecure Wel	oView in an a	pplication ca	n be significa	nt, as it can		
	leave the application vulneral	ole to various	types of at	tacks, such a	s cross-site		
	scripting (XSS), and data the			can allow a	attackers to		
	execute steal user credentials, a	and inject mal	icious code				
Platform	Android						
	Proof	of Concept					
	łew Project - jado-gui /lew Navigation Tools Help			- o ×			
면 IT	secureshop.util.customkeby: secureshop.util.dustomkeby: android.support.v4 android.support.v6 apport.support.v6 android.support.v6 apport.support.v6 android.support.v6 apport.support.v6 android.support.v6 android.support.v6 android.support.v6 android.support.v6 android.support.v6 android.support.v	\$\n\u0002\u0018\u0002\n\u0002\u0018\u0002\\n\u0002\u0018\u0002\n\u0002\\u0002\u0002\u0002\u0002\\u0002\\u0002\\u0002\u0002\\u0002\\u0002\u0002\\u0002\\u0002\\u0002\u0002\\u0002\u0002\\u0002\u0002\u0002\u0002\u0002\u0002	18\w8002\n\u8002\n\u8002\n\u00				
Dagamman	Code Small	nantation vuln	anahility nafa	ma to a situatio	on ryhana an		
Recommen dation	The insecure WebView implemapp is using a WebView comp						
uativii	not configured securely. Specif						
		-	-				
	is set to ignore SSL certificate errors or to accept any SSL certificate, making the app vulnerable to Man-in-the-Middle (MITM) attacks. Always use HTTPS to load						
	web content in WebView. in addition, WebView should always validate SSL						
	certificates presented by the se						
OWASP	https://owasp.org/www-project	-mobile-top-1	0/2016-risks/	m3-insecure-			
Reference	communication/						
Reference	https://developer.android.com/	guide/webapp	s/webview#s	ecurity			

# **4.2.4** The App logs information

Severity	Info				
Level	TT' 1	T '1 .1'1 1		TT' 1	
Technical Impact	High	Likelihood	High	High	High
Observatio	In command mannet we wand	Popularity  ADD common	High	Simplicity	High
n	In command prompt we used after trying to login with a pass				_
•	logs same as the password ente		isername, the	pussword sin	owed in the
Impact	Can potentially expose sensitive information to unauthorized parties, making it				
	easier for attackers to exploit v	ulnerabilities	and compron	nise the syster	n.
Platform	Android				
	Proof	of Concept			
<b>⊗</b> File	*New Project - jadx-gui View Nevigation Tools Help			- 0	×
hop	■ Windows PowerShell × + ∨	100	2.10	×	( <u>4  </u>
oid.	-5 05-27 23:37:24.234 1555 2074 I MQoS : rtt=1088,tcp × 05-27 23:37:24.234 1555 2074 I QosMonitor: resetRtt:	rttType = 3		app.F	rag
ann	05-27 23:37:24.235 1555 2177 I HuaweiWifiWatchdogStat	(5),wifi-link;use signalL teMachine: PTcp RTT:0, rt	vl:5;RAT:4G t pkt=0, tcp_rx=0, tcp_	erraceid: o	
gif. loa	de 05-27 23:37:24.235 1555 2177 I HuaweiWifiWatchdogStat	teMachine: rssi bad, mrss teMachine: start computeQ	i=-72 < TargetRssi=-69 osLevel tcpRtt = 0 ,rss	i = -72 ,mBestSpee uestM	ana <sub>t</sub>
@ A	dInPeriod = 1212				
⊕ c ⊕ c ⊜ p	95-27 23:37:24, 285 925 921 InputDispatcher: styl. 65-27 23:37:24, 283 1555 2656 InputDispatcher: styl. 65-27 23:37:24, 283 1555 2656 InputDispatcher: styl. 65-27 23:37:24, 285 16815 16815 V AudioManager: querySo.	us touch up, set lcd even us touch up, set lcd even	t 0 t 0	_	
Q E	05-27 23:37:24.285 16815 16815 D userName: nouf@gmail.c 05-27 23:37:24.285 16815 16815 D password: 1111111	COM	G BOCHOO T.		
⊕ L ⊝ N ⊝ R	1 00-27 23:37:24.265 10543 10543 U password: 1111111 105-27 23:77:24.265 1555 1572 I PackageManager: getIns 1031, callingPid: 16815 105-27 23:77:24.304 1555 1572 I WificondControl: Noise 104, frequency: 2452, UlDelay: -1, currentTxBytes: 95 104, frequency: 2452, UDelay: -1, currentTxBytes: 95 104, frequency: 2452, UDelay: -1, currentTxBytes: 95	e: -97, Snr: 127, Chload:	183, rssi: -72, txBitr	ate: 58, rxBitrate	
@ R @ R	05-27 23:37:24.304 3109 4069 I VoWifiQoe_N: ProcessRe	eqInfo ul_msgid=5		ed: 5128, currentR	
9 s	05-27 23:37:24.304 3109 4069 I VoWifiQoe_N: processRe a 05-27 23:37:24.304 3109 4069 I VoWifiQoe_N: ProcessSi u 05-27 23:37:24.304 3109 4069 I VoWifiQoe_N: ProcessSi	egInfo MSG_WIFI_SIGNAL_IN	FO_IND le 0 tFlg 1 enScreenState 1 :	curStateIsIdle 0	
pro req	v: 05-27 23:37:24.304 3109 4069 I VoWifiQoe_N: UpdateEva ue 05-27 23:37:24.304 3109 4069 I VoWifiQoe_N: UpdateEva 05-27 23:37:24.307 1555 1577 I Windowshapacon; addition	aluatePolicy EVALUATE_POL aluatePolicy phoneId 1 is	ICY_RSSI not goe started!		1
sig uti Gen	"T-line(fulfile) decid Died-ma-di01}	Jowronen. Gispenyia.o bin	del . dildi 014 . 03 . 51 lidei get	Jacoba Concil. William	
Recommen	The vulnerability reported by	MORSE "7	The Ann log	s information	Sensitive
dation	information should never be				
	information, such as usernames				_
	is not secure. Logging is an im				•
	troubleshooting, but it's essent				_
	logged. Developers should ensure that sensitive data is not being written into logs,				
	and if it is necessary to log such data, it should be done in a secure and encrypted				
OWACD	manner [4].	1.4 4- /0017	VA 10 2017		
OWASP Reference	https://owasp.org/www-project	_	/A10_201/-		
Reference	Insufficient Logging%2526Mo https://owasp.org/www-project		0/2016_rieles	m2-insecure	data_
	storage	-moone-wp-1	. U/ 2U 1 U-118K8/	1112-11135CU1C-	uata-
Reference	https://developer.android.com/	topic/security	 /data		

# 4.2.5 Files may contain hard coded sensitive information

Severity	Medium				
Level	Wediam				
Technical	High	Likelihood		Medium	
Impact	6	Popularity	High	Simplicity	High
Observation	By jadx and MobSF we n	_ ,		1 0	
Impact	Can make it easier for at				
Impact	example, if the password				
	easily gain access to it.	8		,,	
Platform	Android				
	Pı	roof of Conce	pt		
*New Project - jadx-gui File View Navigation Tools He	do.				- o ×
					2
⊕ ⊕ com ⊕ ⊕ bumptech.glide	eshop.Cart 💥 🕒 com.insecureshop.Priv.	ateActivity 💥 🕒 com.i	nsecureshop.InsecureShopAp	p 🗶 🕒 com.bumptech.gl	ide.load.Option 🗶 🔞
# # annotation # # disklrucach # # gifdecoder	e import com.bumptech.glide.util.Pre				1
a load ⊕ ∰ data	<pre>import java.security.MessageDigest  45 public final class Option<t> {</t></pre>	:3			- 1
engine model	private static final CacheKeyl /* class com.bumptech.glid			Object>() {	- 1
# # resource # 0 DataSourc # 0 DecodeFor				:) {	- 1
⊕ ① Encoder	); ateg private final CacheKeyUpdatero	<pre><t> cacheKeyUpdater;</t></pre>			- 1
⊕ ⊕ HttpExcep ⊕ ⓒ ImageHead		es;			-
⊕ ⊕ ImageHead ⊕ ⊕ Key	erPa: public interface CacheKeyUpdat	ter <t> {</t>	ligast):		1
⊕ ⊕ MultiTran ⊕ ⊕ <mark>Option</mark> ⊕ ⊕ Options	)		and the second		
@ Preferred					
⊕ O ResourceE ⊕ O Transform	FO public static (T) Option(T) wo				
manager module	1				
# # provider # # request # # signature	71 public static <t> Option<t> di 72 return new Option&lt;&gt;(key2,</t></t>		uater(1) Cachekeyopdater2	, (	
⊕ ⊕ util ⊕ <b>Q</b> GeneratedAp	pGlic 85 public static <t> Option<t> di return new Option&lt;&gt;(key2,</t></t>			cacheKeyUpdater2) {	
GenericTran		, , , , , , , , , , , , , , , , , , , ,			
	ı				
Recommend ation  Hard-coded credentials typically create a significant hole that allows are attacker to bypass the authentication that has been configured by the product administrator. This hole might be difficult for the system administrator detect. Even if detected, it can be difficult to fix, so the administrator m forced into disabling the product entirely.				e product ator to or may be	
store passwords, keys, and other credentials outside of the code in a strong protected, encrypted configuration file or database that is protected from as by all outsiders, including other local users on the same system. Properly protect the key (CWE-320). If you cannot use encryption to protect the file make sure that the permissions are as restrictive as possible. In Windows environments, the Encrypted File System (EFS) may provide some protect [5].					from access operly the file, then adows protection
OWASP	https://owasp.org/www-p	project-mobile-	-top-10/2014-r	risks/m2-insec	ure-data-
Reference	storage	/1 6	0011		
Reference	https://cwe.mitre.org/data	a/definitions/7	98.html		

## **Appendix A: About the Team**

Team Code:	
Student Name	Role
Nouf Alsadhan	Work done in her device, Methodology and multiple parts in the report.
Jumanah alDawsari	Detailed findings, conclusion and multiple parts in the report.

#### **References:**

- [1] <a href="https://developer.android.com/topic/security/risks/android-debuggable">https://developer.android.com/topic/security/risks/android-debuggable</a>
- [2] https://sensei.securecodewarrior.com/recipes/scw%3Aandroid%3Adisabled-cleartext
- [3] https://owasp.org/www-project-mobile-top-10/2016-risks/m3-insecure-communication
- [4] https://developer.android.com/topic/security/data
- [5] https://cwe.mitre.org/data/definitions/798.html
- [6] https://github.com/skylot/jadx/releases/tag/v1.2.0.
- [7] https://github.com/MobSF/Mobile-Security-Framework-MobSF
- [8] https://www.xda-developers.com/install-adb-windows-macos-linux/