Android Application Hacking

Who am i?

Gabrielle Botbol

@Gabrielle_BGB

/in/gabriellebotbol

https://csbygb.github.io/

Cyber Security by Gabrielle B.

From blogger to pentester

«Apprenance» is:

«a lasting set of dispositions… favourable to the act of learning… in all situations: formal or informal, experiential or didactic, self-directed or not, intentional or accidental».

Philippe Carré, 2005.

Program in 6 steps

Conferences

MOOC

Volunteering

CTF

Summer Schools

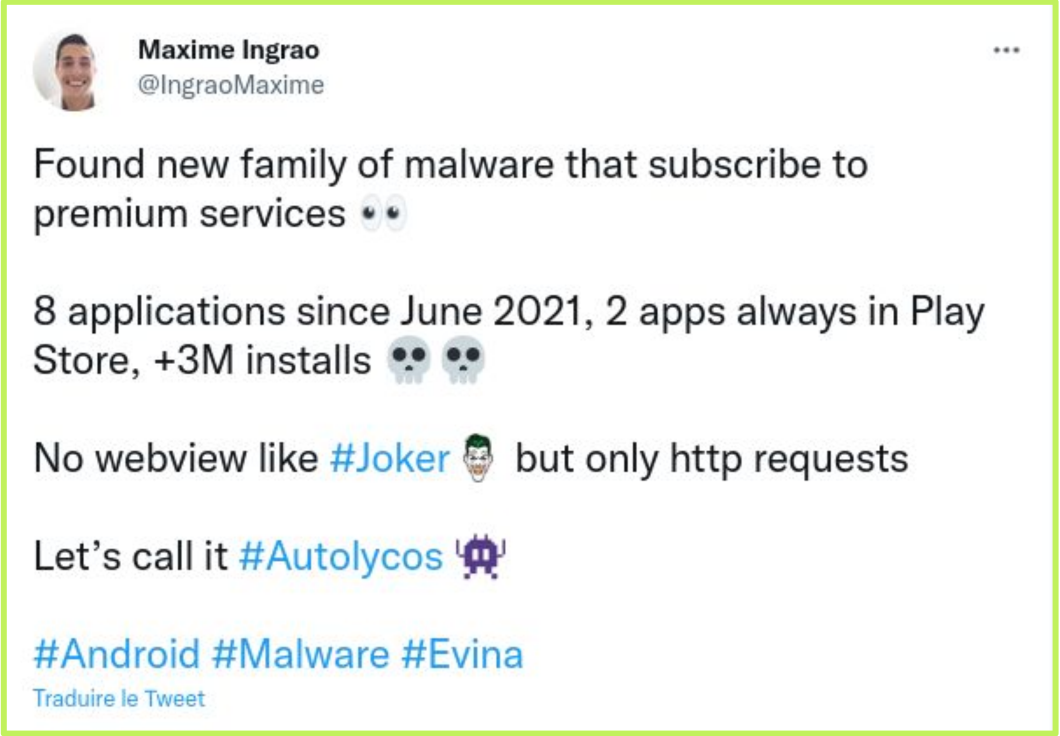
Internship

What is pentest?

Pentesting is a **risk-oriented approach** that requires a good understanding of the **business issues** and **business context** of the targeted system.

Why Android App Pentest?

July 13th 2022



Maxime Ingrao
@IngraoMaxime

Found new family of malware that subscribe to premium services 👀

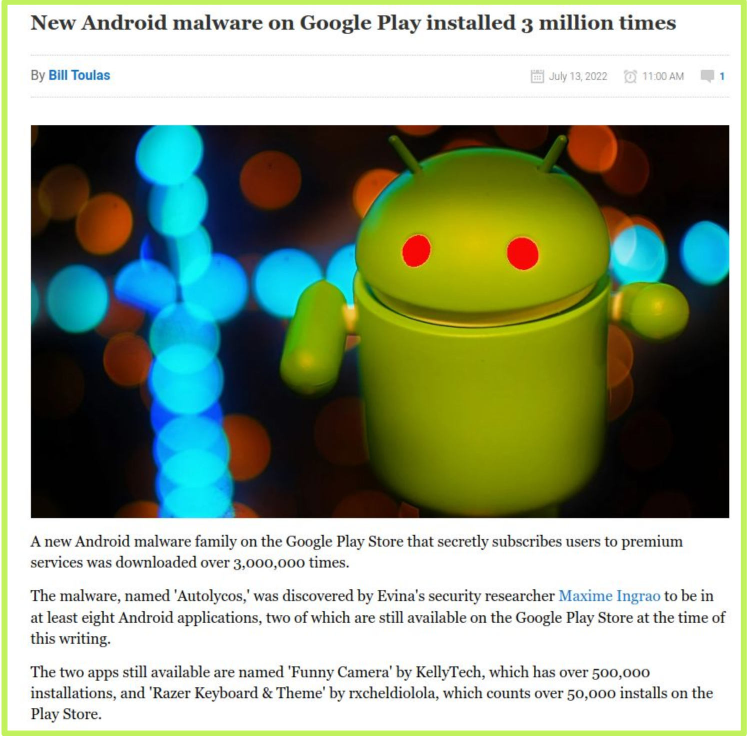
8 applications since June 2021, 2 apps always in Play Store, +3M installs 💀💀

No webview like #Joker 🤡 but only http requests

Let's call it #Autolycos 👾

#Android #Malware #Evina

Traduire le Tweet



New Android malware on Google Play installed 3 million times

By Bill Toulas July 13, 2022 11:00 AM 1

A new Android malware family on the Google Play Store that secretly subscribes users to premium services was downloaded over 3,000,000 times.

The malware, named 'Autolycos,' was discovered by Evina's security researcher Maxime Ingrao to be in at least eight Android applications, two of which are still available on the Google Play Store at the time of this writing.

The two apps still available are named 'Funny Camera' by KellyTech, which has over 500,000 installations, and 'Razer Keyboard & Theme' by rxcheldiolola, which counts over 50,000 installs on the Play Store.

Some figures

2,034,217+

New Mobile Malware Samples Detected in the Wild in 2021

466%

Increase in Exploited, Zero-Day Mobile Vulnerabilities

10M+

Mobile Endpoints Impacted By Threats

42%

Enterprises Reported Mobile Devices and Web Apps Led To A Security Incident

75%

Phishing Sites Specifically Targeted Mobile Devices

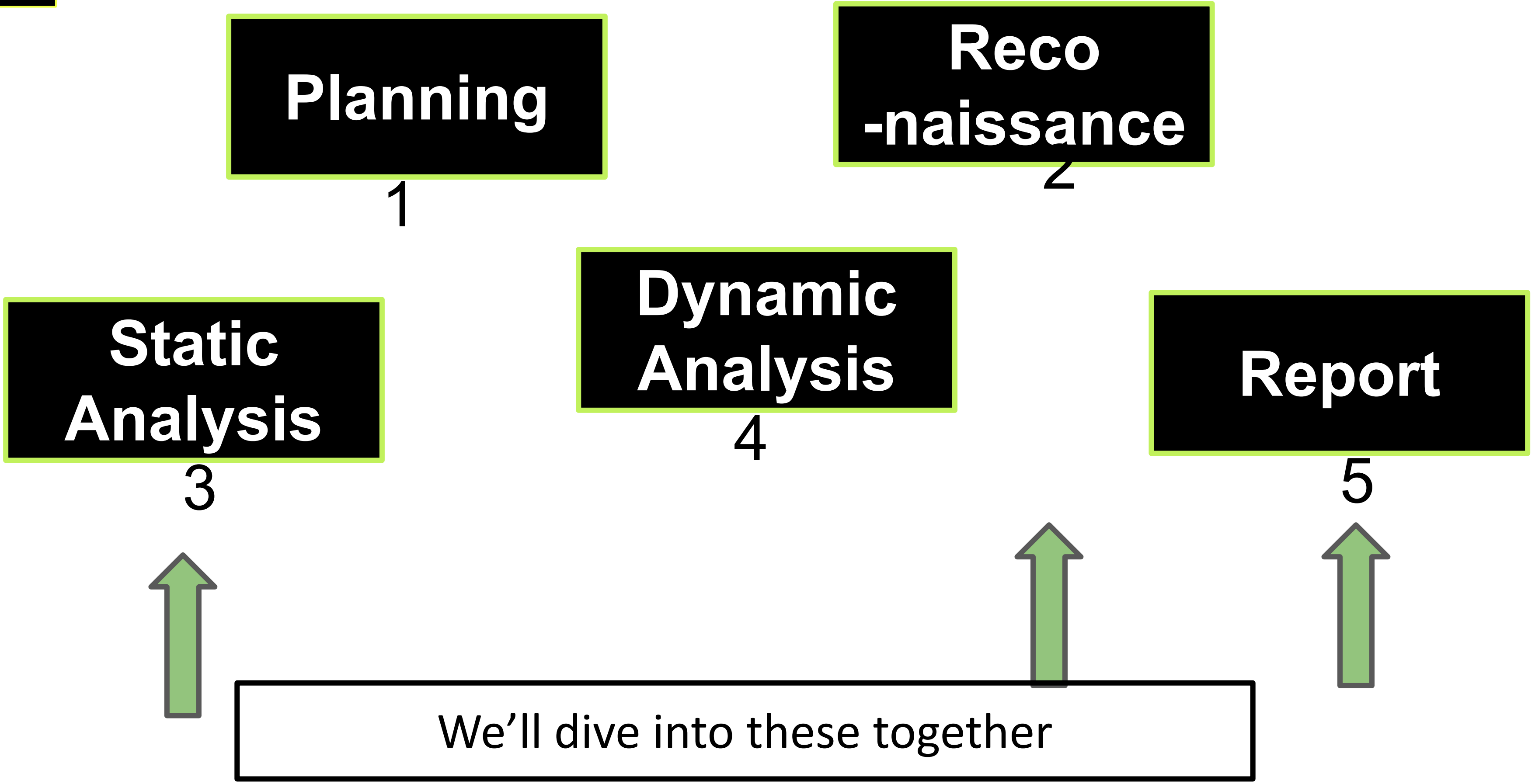
23%

Of Mobile Devices Encountered Malicious Applications Worldwide

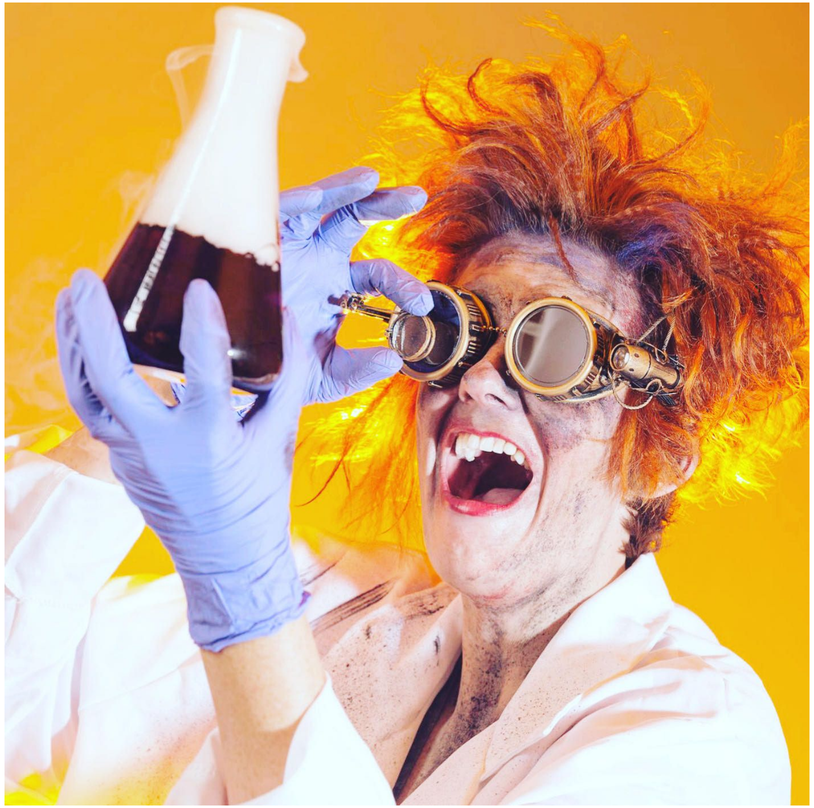
Source: https://www.zimperium.com/global-mobile-threat-report/

What is an Android App Pentest?

Android App pentest process



The importance of the lab



What you will need

Tools:

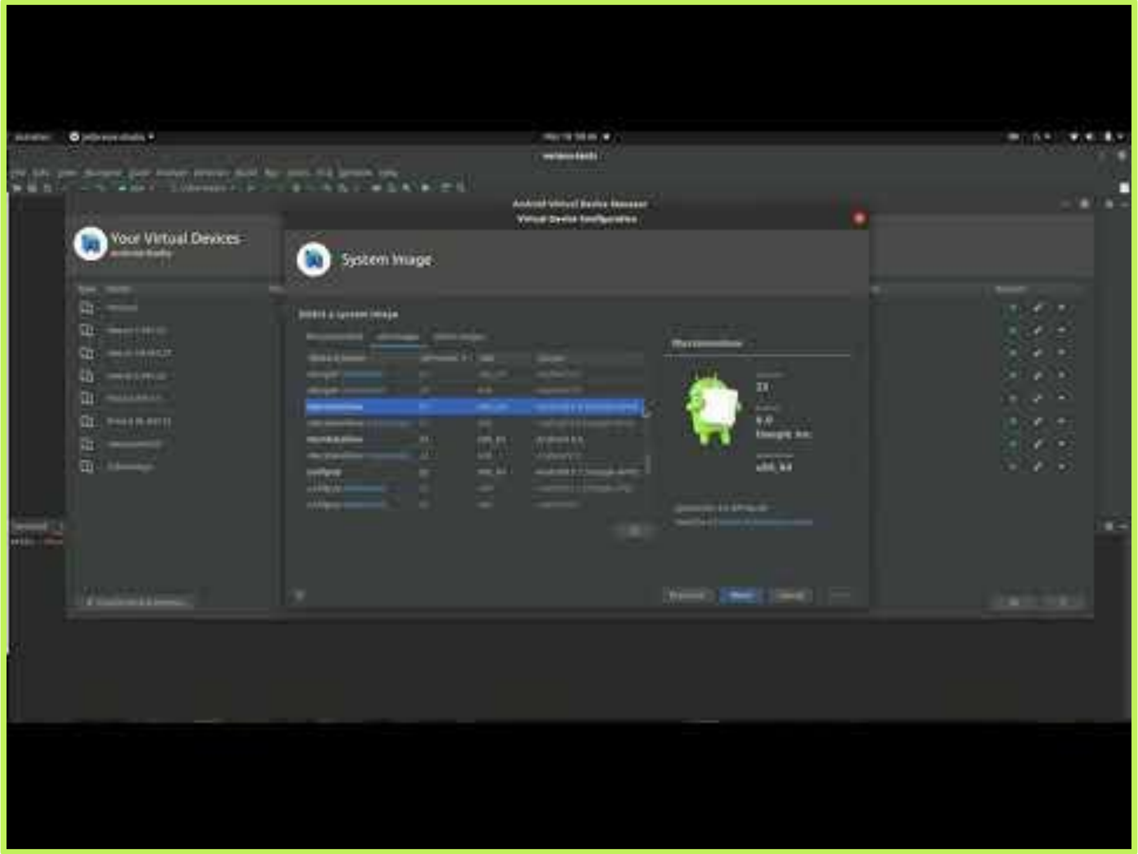
- Jadx
- ADB
- Android Studio
- Burp Suite

Set up the lab - Installs

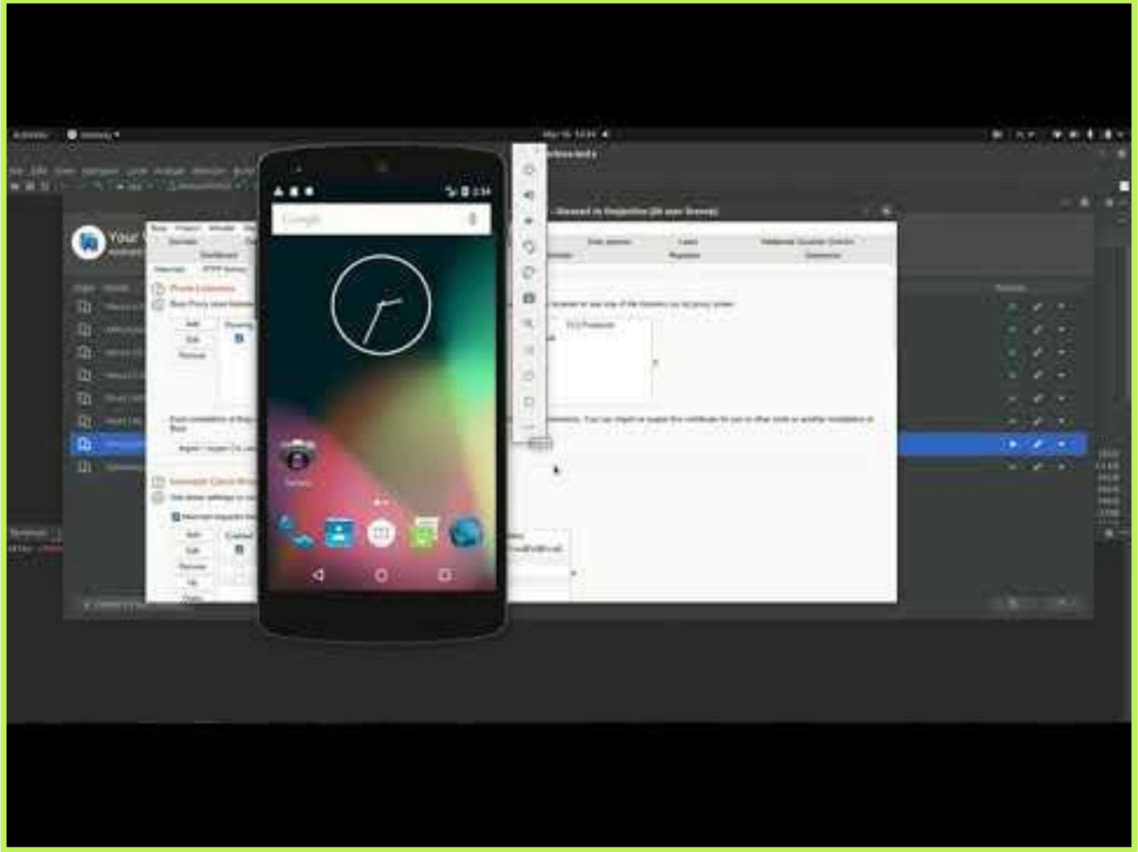
Install Jadx	`sudo apt install default-jdk`
	`sudo apt install jadx`
	`./jadx-gui`

Install adb	`sudo apt-get install adb`
Install Android Studio	**Download** https://developer.android.com/studio
Install Burp Suite	**Download** and **install** the version according to your system here https://portswigger.net/burp/releases/professional-community-2021-12-1?requestededition=community
For more info on these installs	- JADX https://github.com/skylot/jadx - ADB https://www.xda-developers.com/install-adb-windows-macos-linux/

Set up the lab - Create an emulator



Set up the lab - Configure burp



Vuln Apps used for the examples

Get PIVAA here:
https://github.com/HTBridge/pivaa
Purposefully Insecure and Vulnerable Android Application.

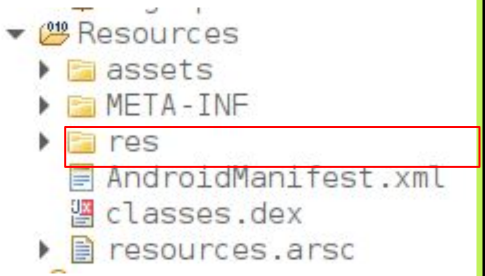
Get InjuredAndroid here:
https://github.com/B3nac/InjuredAndroid/releases/tag/v1.0.12

Static Analysis

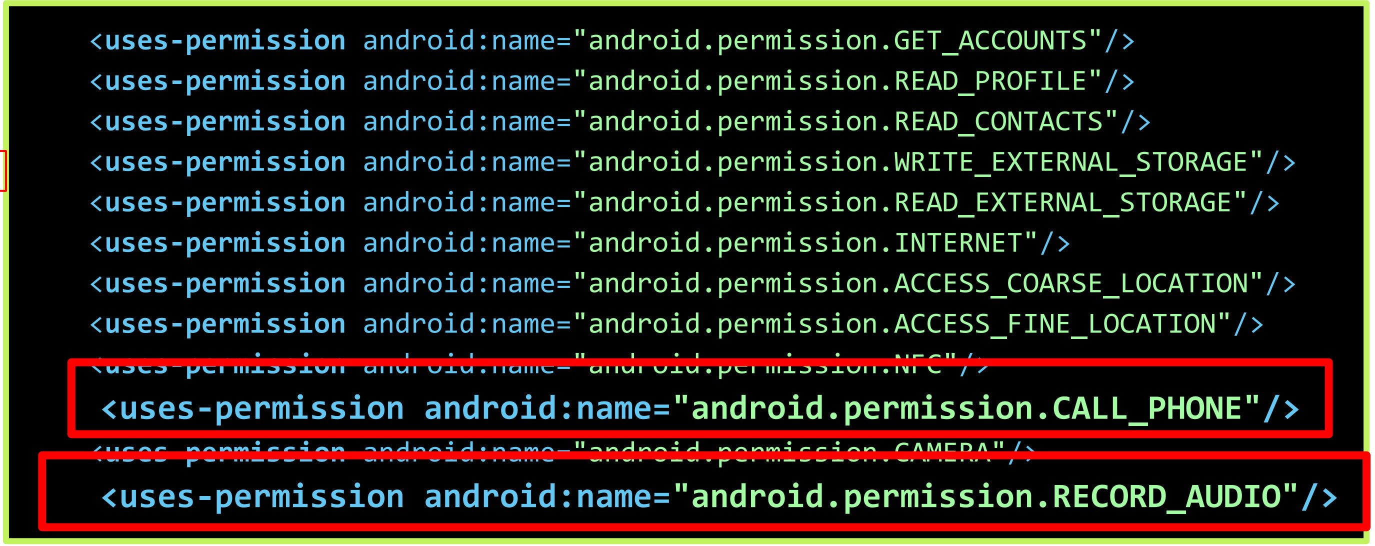
What to check:

- AndroidManifest.xml
- Strings.xml
- Enumerate Database
- Search for secrets and sensitive data

Example PIVAA - AndroidManifest 1



```
▾ Resources
  ▸ assets
  ▸ META-INF
  ▸ res
    AndroidManifest.xml
    classes.dex
  ▸ resources.arsc
```



```
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.READ_PROFILE"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.NFC"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

- List of permissions here: https://developer.android.com/reference/android/Manifest.permission

Example PIVAA - AndroidManifest 2

```
android:allowBackup="true"
```

(ON by default)

OWASP MSTG-STORAGE-8:

https://github.com/OWASP/owasp-mstg/blob/8d67a609ecd095d1bb00aa6a3e211791af5642e8/Document/0x05d-Testing-Data-Storage.md#static-analysis-7

```
android:debuggable="true"
```

(OFF by default)

OWASP MSTG-CODE-2:

https://github.com/OWASP/owasp-mstg/blob/53ebd2ccc428623df7eaf2361d44b2e7e31c05b9/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#testing-whether-the-app-is-debuggable-mstg-code-2

Example PIVAA - AndroidManifest 3

```
<service android:name="com.htbridge.pivaa.handlers.VulnerableService"
android:protectionLevel="dangerous" android:enabled="true" android:exported="true"/>
        <receiver android:name="com.htbridge.pivaa.handlers.VulnerableReceiver"
android:protectionLevel="dangerous" android:enabled="true" android:exported="true">
            <intent-filter>
                <action android:name="service.vulnerable.vulnerableservice.LOG"/>
            </intent-filter>
        </receiver>
        <provider
android:name="com.htbridge.pivaa.handlers.VulnerableContentProvider"
android:protectionLevel="dangerous" android:enabled="true" android:exported="true"
android:authorities="com.htbridge.pivaa" android:grantUriPermissions="true"/>
```

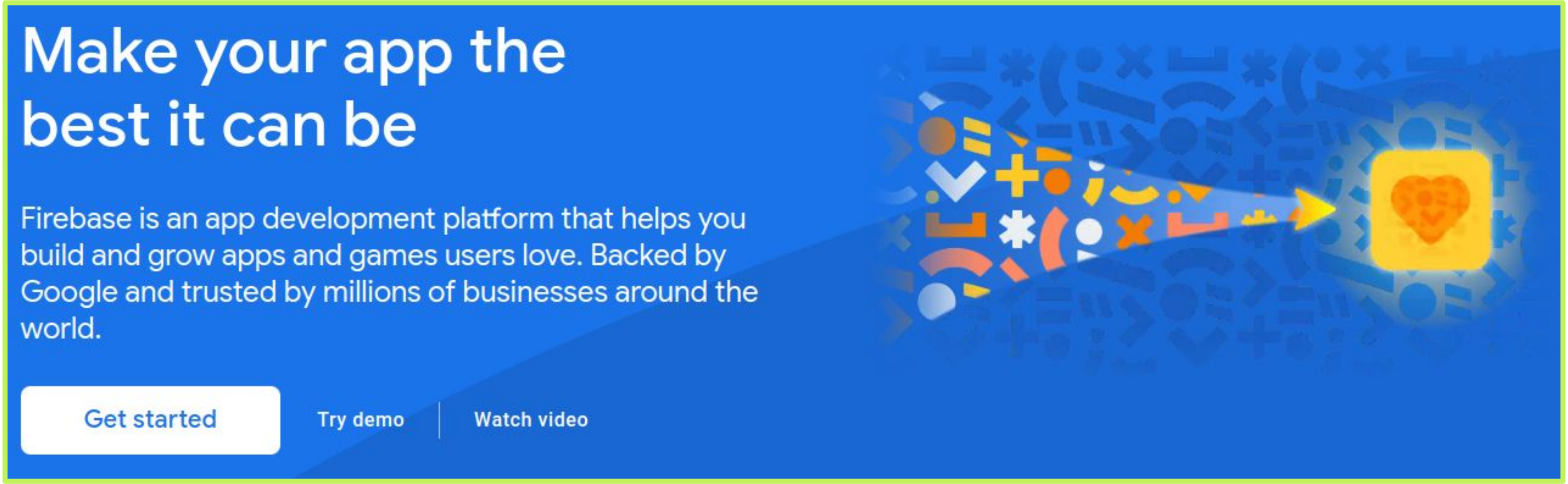
Exportable Activities in PIVAA

Example with InjuredAndroid - Strings

/res/values/strings.xml

```
<string
name="google_api_key">AIzaSyCUImEIOSvqAswLqFak75xhskkB6illd7A</string>
<string
name="google_app_id">1:430943006316:android:d97db57e11e42a1a037249</st
ring>
<string
name="google_crash_reporting_api_key">AIzaSyCUImEIOSvqAswLqFak75xhskkB
6illd7A</string>
<string
name="google_storage_bucket">injuredandroid.appspot.com</string>
```

Example - What is Firebase



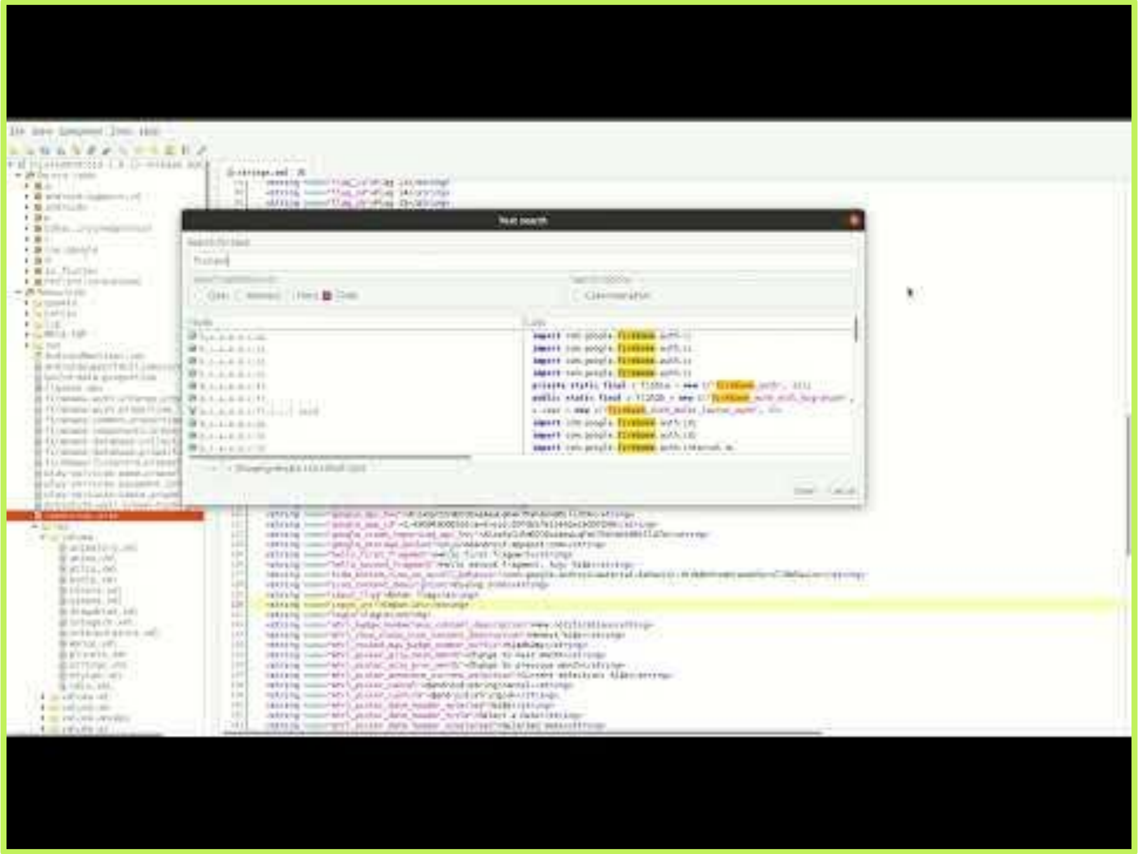
Make your app the best it can be

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

Example InjuredAndroid - Firebase Endpoints

```xml
<string name="fab_transformation_sheet_behavior">com.google.android.material.transformation.FabTransformationSheetBehavior</string>
<string name="firebase_database_url">https://injuredandroid.firebaseio.com</string>
<string name="first_fragment_label">First Fragment</string>
```

General Tips for static analysis



Tools for static analysis

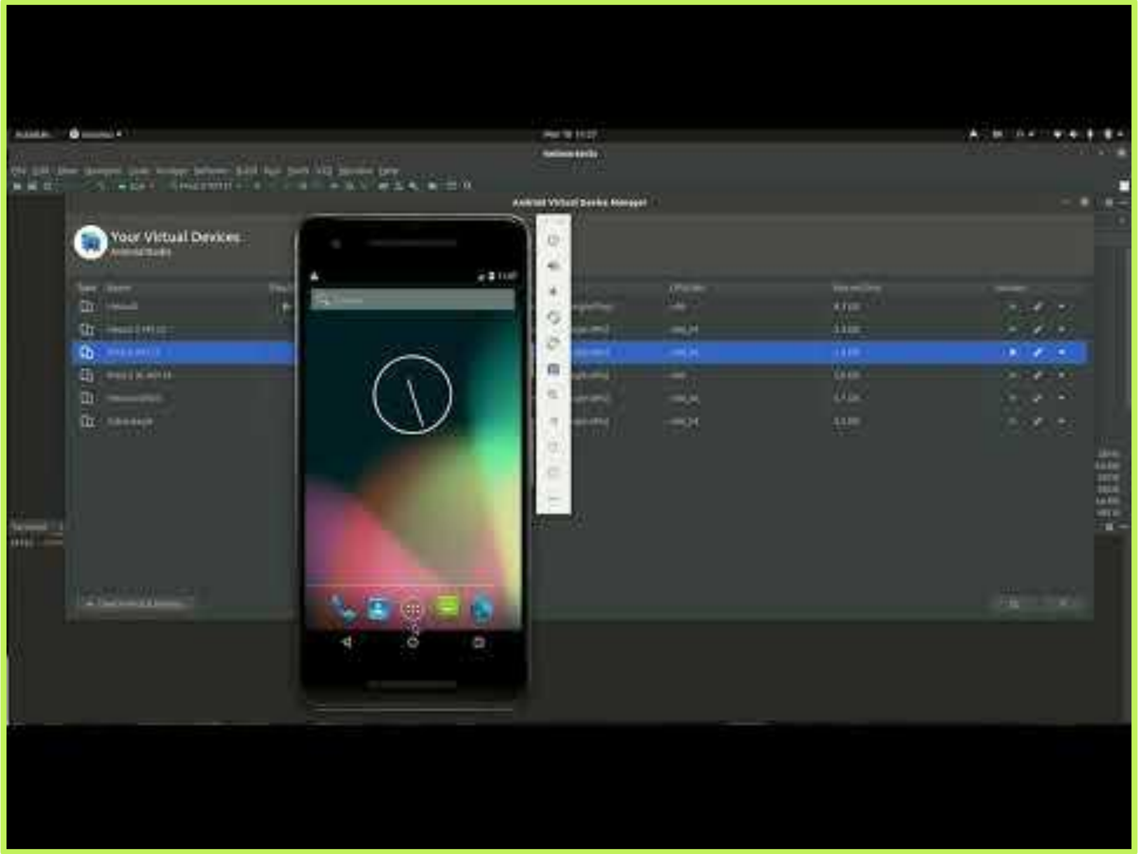
- Firebase Enum Github: https://github.com/Sambal0x/firebaseEnum
- FireBaseScanner: https://github.com/shivsahni/FireBaseScanner
- Cloud Enum https://github.com/initstring/cloud_enum

Dynamic Analysis

What to check:

- Tapjacking
- Can you capture screens with sensitive data
- OWASP Top 10
- Analyse traffic with burp to find odd things

Example with PIVAA - BG capture

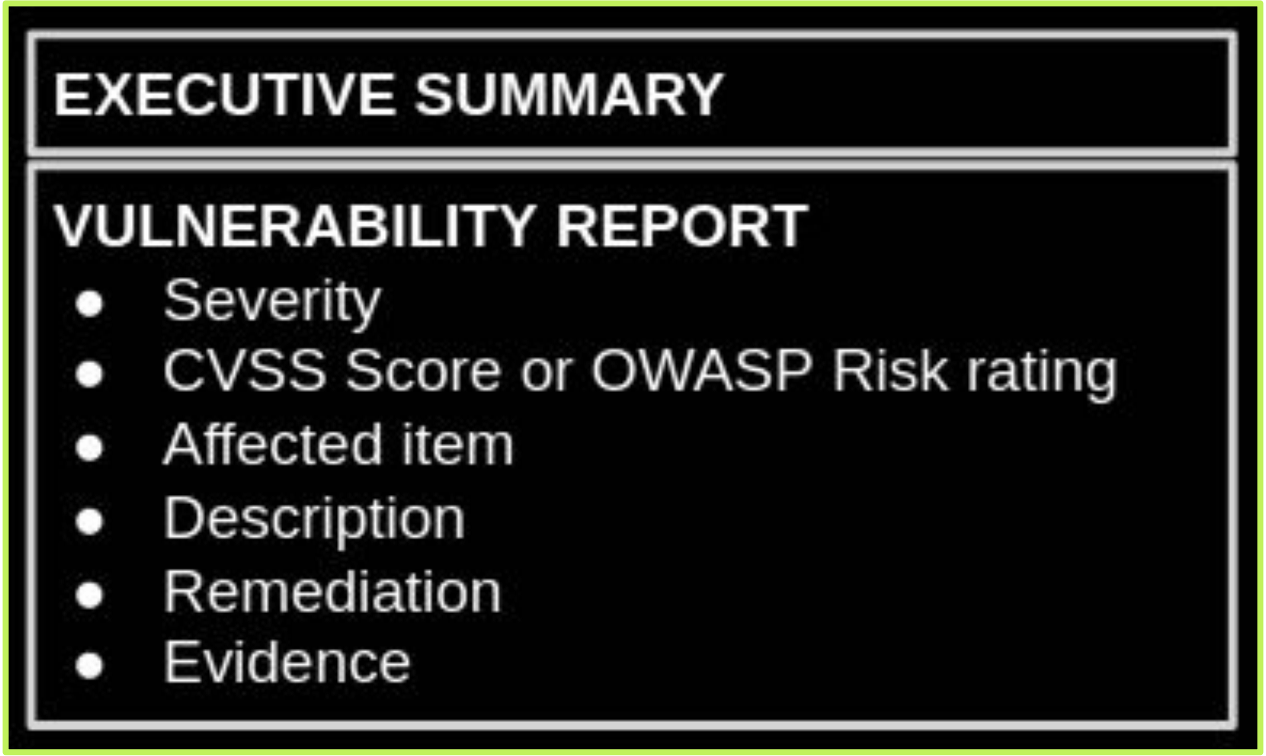


Automatic tools

- MobSF https://github.com/MobSF/Mobile-Security-Framework-MobSF

- Qark https://github.com/linkedin/qark

How to report



EXECUTIVE SUMMARY

VULNERABILITY REPORT

- Severity
- CVSS Score or OWASP Risk rating
- Affected item
- Description
- Remediation
- Evidence

How to report - Example

Information Disclosure Through Backgrounded Screens of the App

Severity: Low

CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N

Description

In order to provide visual transitions in the interface, Android records a screenshot of the current application screen.

This happens when an application is suspended, when the button that leads to the main menu is pressed, when receiving a call, or any other event that may interrupt the application.
These captures can contain user information or other sensitive data.
This way, the attack surface of the mobile application is expanded.
Confidential information can be persisted on the mobile device through this mechanism.

How to report - Example

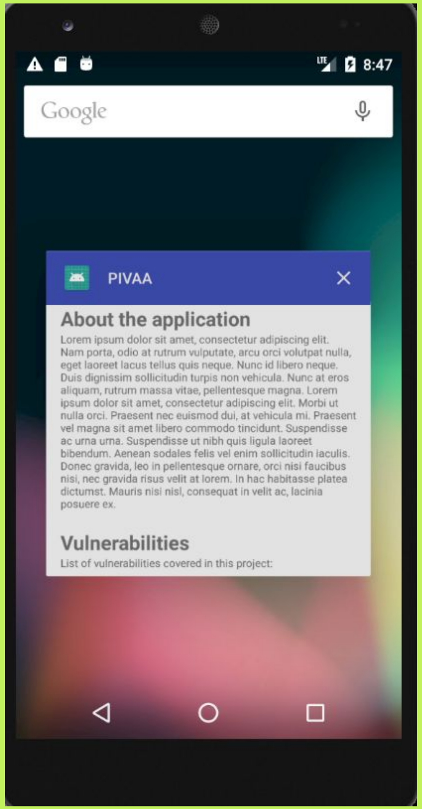
Information Disclosure Through Backgrounded Screens of the App

Remediation

Using the "FLAG_SECURE" flag in screens with sensitive data will prevent their contents from being in backgrounded screenshots.

Resource

https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06d-Testing-Data-Storage.md#testing-auto-generated-screenshots-for-sensitive-information-mstg-storage-9



About the application

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam porta, odio at rutrum vulputate, arcu orci volutpat nulla, eget laoreet lacus tellus quis neque. Nunc id libero neque. Duis dignissim sollicitudin turpis non vehicula. Nunc at eros aliquam, rutrum massa vitae, pellentesque magna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut nulla orci. Praesent nec euismod dui, at vehicula mi. Praesent vel magna sit amet libero commodo tincidunt. Suspendisse ac urna urna. Suspendisse ut nibh quis ligula laoreet bibendum. Aenean sodales felis vel enim sollicitudin iaculis. Donec gravida, leo in pellentesque ornare, orci nisi faucibus nisi, nec gravida risus velit at lorem. In hac habitasse platea dictumst. Mauris nisi nisl, consequat in velit ac, lacinia posuere ex.

Vulnerabilities

List of vulnerabilities covered in this project:

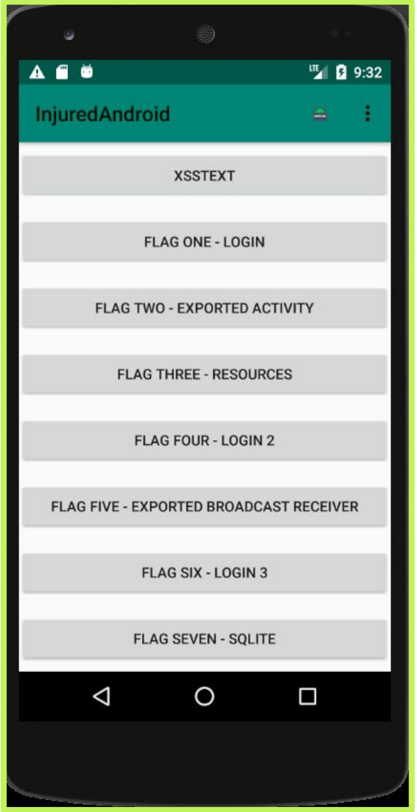
Resources - Practice

Vulnerable Apps to practice on:

- https://thedarksource.com/vulnerable-android-apps/
- https://pentester.land/cheatsheets/2018/10/12/list-of-Intentionally-vulnerable-android-apps.html
- ExploitMe Mobile Android Labs: https://securitycompass.github.io/AndroidLabs/setup.html
- Android Insecure Bank: https://github.com/dineshshetty/Android-InsecureBankv2

Hackthebox

- Track: Introduction to Android Exploitation: https://app.hackthebox.com/tracks/Introduction-to-Android-Exploitation



Resources - Courses & Misc

- Mobile Hacking crash Course on Hackerone https://www.hacker101.com/sessions/mobile_crash_course
- TCM Security Academy, Course Mobile App Pentesting (29,99 US$) https://academy.tcm-sec.com/p/mobile-application-penetration-testing
- Nahamsec Beginner Bounty hunters Mobile Hacking: https://github.com/nahamsec/Resources-for-Beginner-Bug-Bounty-Hunters/blob/master/assets/mobile.md
- A notion site about mobile https://start.me/p/OmxRqE/mobile

Resources - References & Reads

- OWASP MSTG: https://github.com/OWASP/owasp-mstg/
- Android Hackerone disclosed reports and other resources: https://github.com/B3nac/Android-Reports-and-Resources
- HTB Intro to Mobile Peneration testing https://www.hackthebox.com/blog/intro-to-mobile-pentesting
- HackTricks Android: https://book.hacktricks.xyz/mobile-apps-pentesting/android-checklist

Resources - Tools

- Nahamsec tools for mobile hacking: https://github.com/nahamsec/Resources-for-Beginner-Bug-Bounty-Hunters/blob/master/assets/tools.md#Mobile-Hacking
- Qark: https://github.com/linkedin/qark
  - Good tutorial for Qark: https://resources.infosecinstitute.com/topic/android-penetration-tools-walkthrough-series-qark/
- MOBSF: https://mobsf.github.io/docs/#/

Resources - To go further

- Understand certificate pinning: https://littlemaninmyhead.wordpress.com/2020/06/08/understanding-certificate-pinning/

- How to Bypass SSL Pinning: https://youtu.be/SEySgg3vQjg

Get these slides and all the resources

https://csbygb.gitbook.io/

Welcome to CSbyGB's Pentips

CSbyGB

$ whoami /priv

Ethical Hacker 🏳 | 🏆 Pentest Ninja Award W.S Cyberjutsu | 🇨🇦 Top 20 Women in Cybersecurity #DoWeLookLikeHackers 🏳️‍🌈

Quiz to go

Check out the quiz about this presentation here:

https://forms.gle/He2MyhxRoDcWSwgR8

Many Thanks