MCC WORKSHOP:

ANDROID AGAINST HUMANITY - THE FUNDAMENTALS OF ANDROID

WHOAMI

- KELVIN
- PENTESTER
- FOUNDER OF EQCTF
- MCC ALUMNI & CREW
- CTF PLAYER

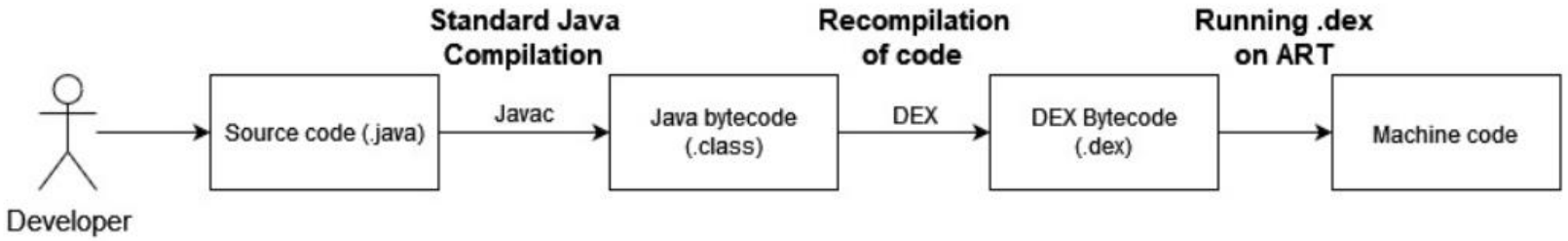
WHOAMI

- KS
- PENTESTER
- ADVISOR OF EQCTF
- MCC ALUMNI & CREW
- CTF PLAYER


WHAT IS
ANDROID?

UNDERSTANDING APK

- Android Package Kit (APK) is an Android binary file.
- The .apk a file format for android that contains all the necessary elements required to function properly within android devices.
- APK files is generally a type of JAR (Java Archive) file, which is also a type of zip file, hence when its decompiled, you can see the contents with specific file structures.



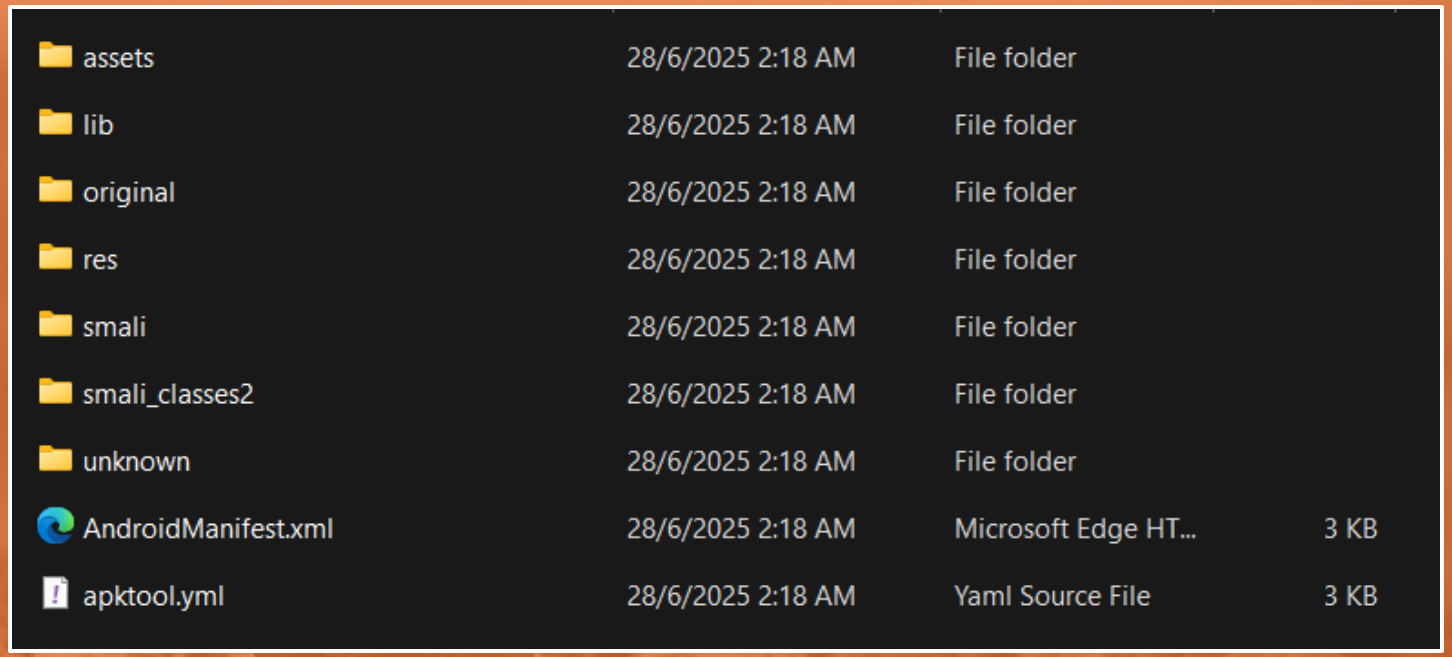
Reference: NUS Greyhat

APK STRUCTURES

APK typically consist of following contents:

- **AndroidManifest.xml**
- **META-INF/**
  - **MANIFEST.MF**
  - **CERT.SF**
- **classes.dex**
- **lib/**
  - **x86_64**
  - **x86**
  - **arm64-8a**
  - **armeabi-v7a**
- **assets/-resources.arsc**
- **source/com/application-name**
- **res/values/strings**

EXAMPLE OF DECOMPILE APK:



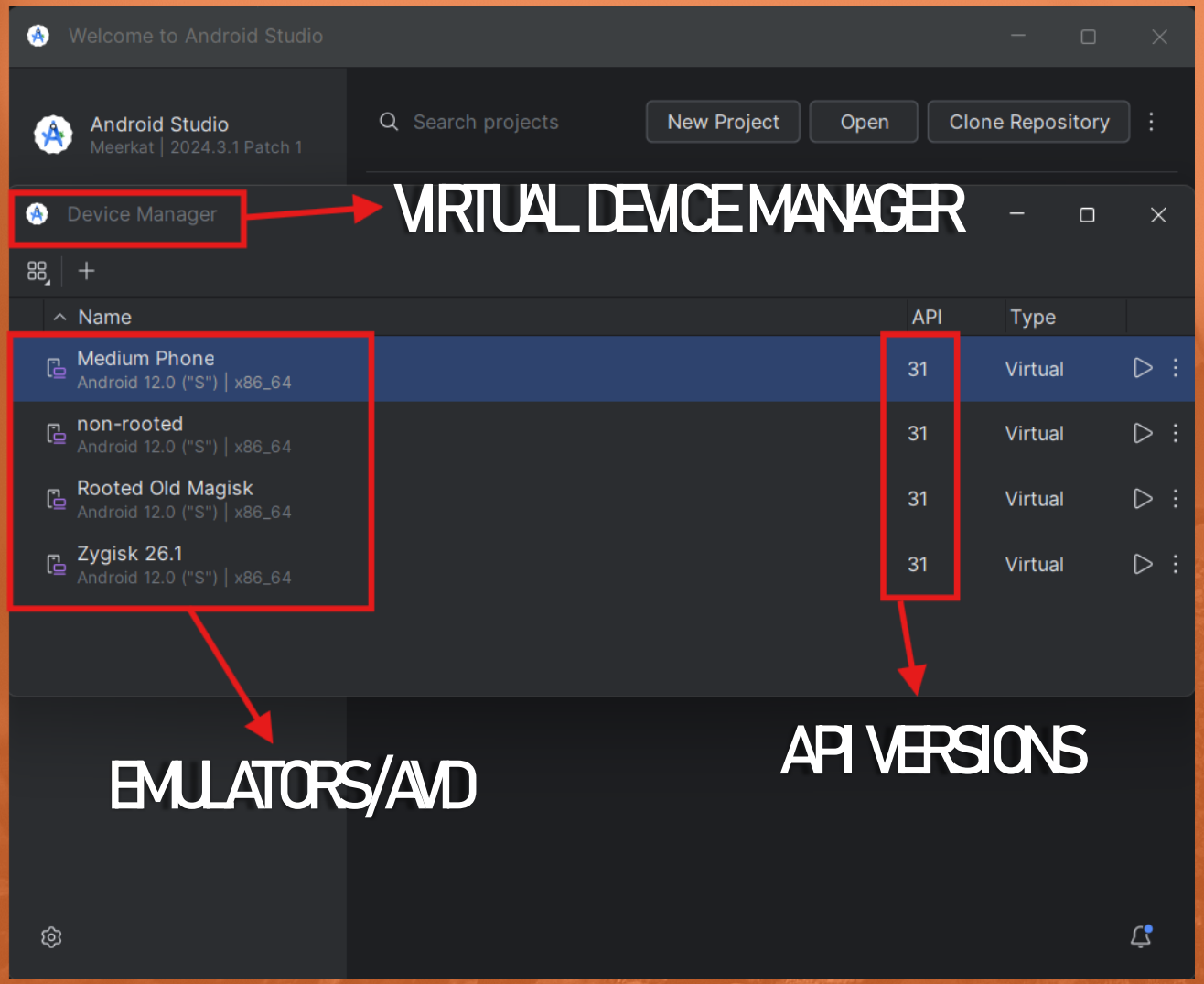
TOOLS:

- Android Virtual Device (AVD)
- JADX
- APKTool
- Frida
- Portswigger Burpsuite

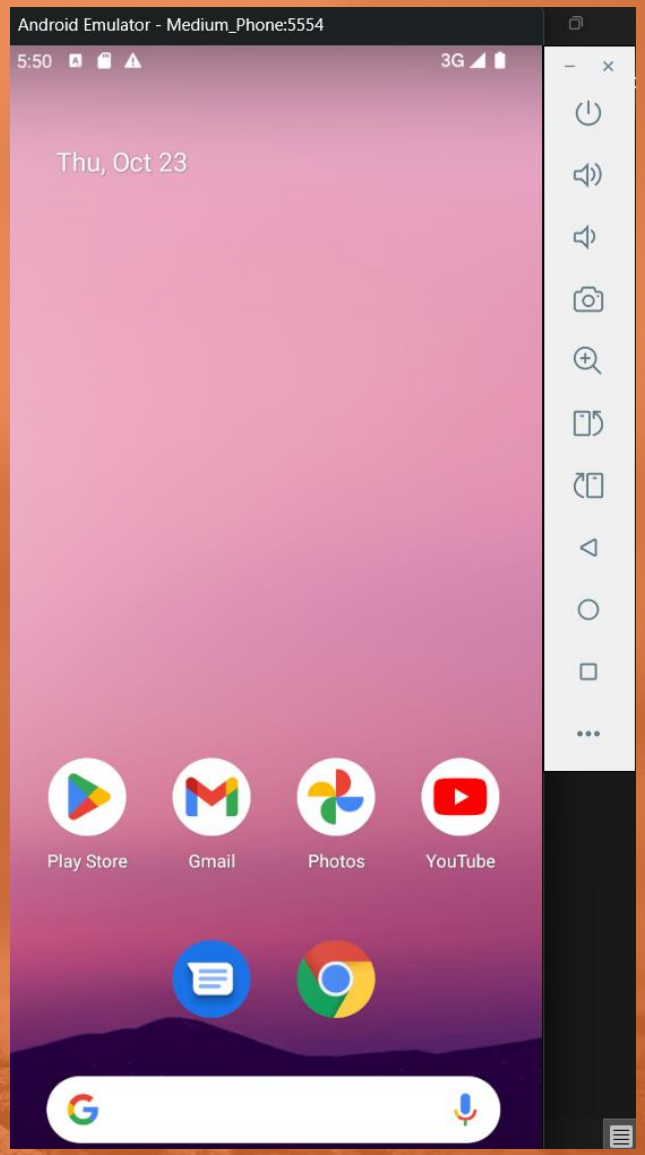
WHAT ARE AVDs

- Android Virtual Device (VM for Android System)
- Used for Development, Testing, or even Security Researching
- Common AVD Tools/Platforms:
  - Android Studio
  - Genymotion
  - MEMU

ANDROID STUDIO



AVD



Why do we root AVDs

- Testing Application within AVDs
- Allows the installation and execution of custom scripts
- Intercepting Network Traffic
- Not everyone has an old phone to root

Limitation of non-rooted device:

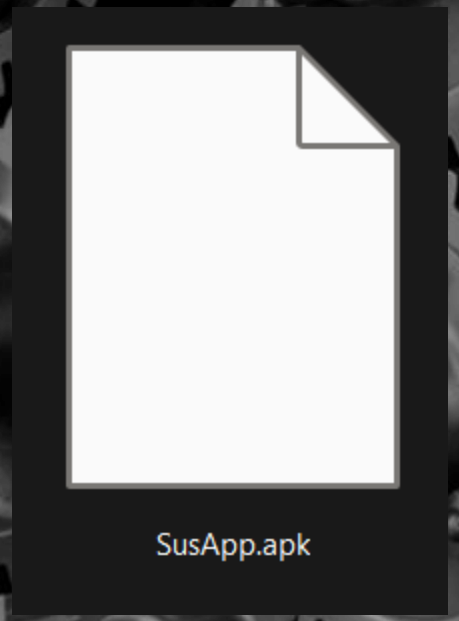
- Only trust third party certificate at a user-level
- Limited capability to test applications

Allowing Burp to Intercept Traffic

- Pre-requisites:
  - Setup Android Studio
  - Install Burp CA on AVD
  - Set proxy to burp
- Why do this?:
  - Intercept Network Traffic
  - Ease dynamic analysis – Test APK like how you test a web application

REVERSING APK

How we turn this:

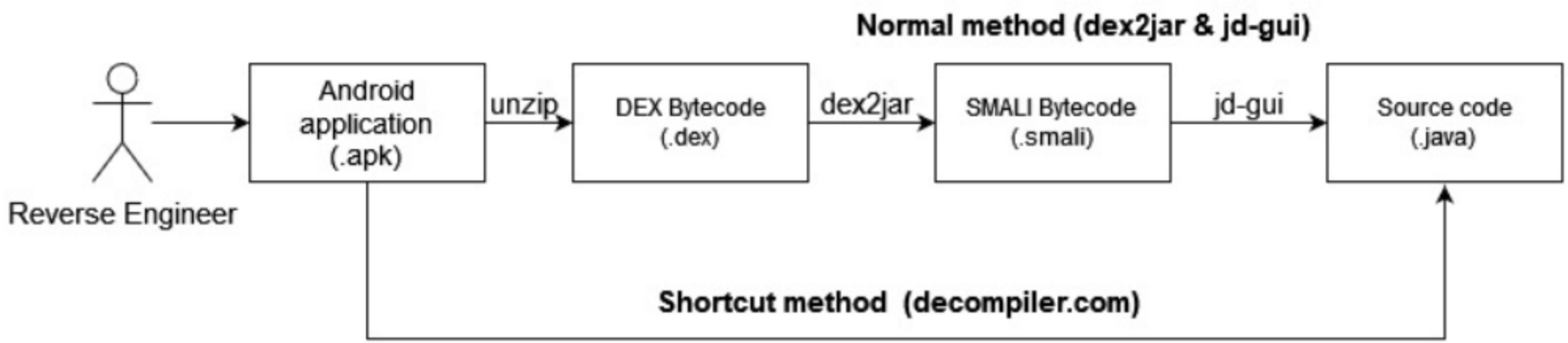


Into this:

assets	28/6/2025 2:18 AM	File folder	
lib	28/6/2025 2:18 AM	File folder	
original	28/6/2025 2:18 AM	File folder	
res	28/6/2025 2:18 AM	File folder	
smali	28/6/2025 2:18 AM	File folder	
smali_classes2	28/6/2025 2:18 AM	File folder	
unknown	28/6/2025 2:18 AM	File folder	
AndroidManifest.xml	28/6/2025 2:18 AM	Microsoft Edge HT...	3 KB
apktool.yml	28/6/2025 2:18 AM	Yaml Source File	3 KB

Reversing APK

- Reversing the apk means decompiling the file, converting the .apk file back to source code.



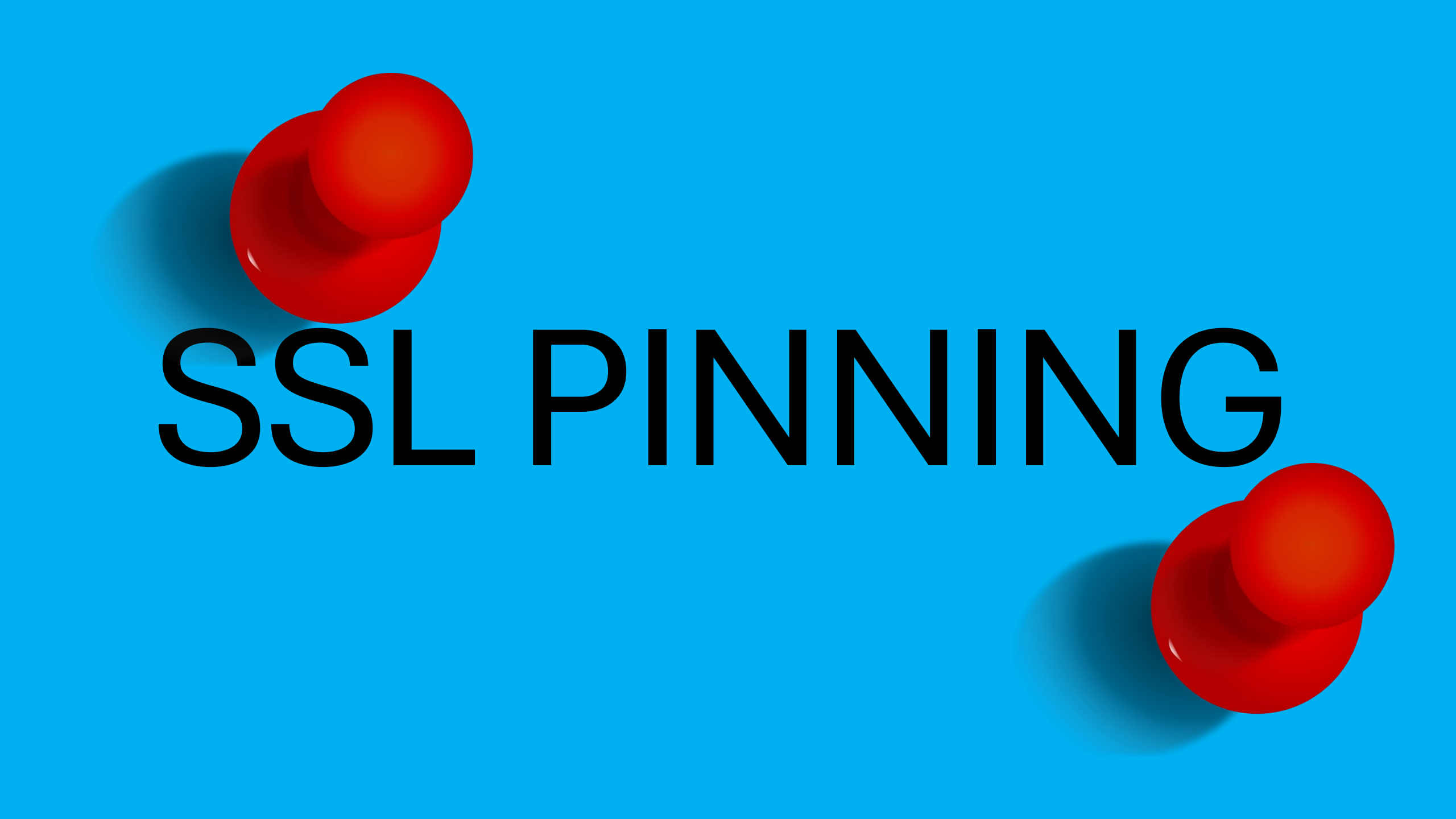
Reference: NUS Greyhat

JADX VS APKTOOL

JADX	APKTOOL
CONVERTS DEX FILES TO SOURCE CODE	DECOMPILES RESOURCES AND MANIFEST FILES INTO READABLE FORM
SPIT OUT FULL SOURCE CODE (JAVA/KOTLIN)	SMALI AND XML FILES
USED FOR SOURCE CODE REVIEWING	USED FOR PATCHING AND REBUILDING APKs
EG: UNDERSTANDING APPLICATION LOGIC	EG: READ AND MODIFY ANDROIDMANIFEST FILE

JAVA CODE VS SMALI VS DEX

	JAVA/KOTLIN	SMALI	DEX
READABILITY	HIGH LEVEL SOURCE CODE	LOW-LEVEL CODE	BYTE CODE (BINARY)
TOOL	JADX	APKTOOL	UNZIP?
USE CASE	READ THE CODE	PATCHING	U DECOMPILE THIS SHIT


SSL PINNING

WHAT IS SSL PINNING?

a security technique used in mobile apps (includes Android and iOS) to ensure the app only trusts a specific server certificate or public key when establishing HTTPS connections.

WHY APP HAVE SSL PINNING?

PREVENT MITM: INTERCEPT, MODIFY, INJECTION

MAKE SURE APP ONLY COMMUNICATES WITH INTENDED SERVER

STATIC SSL PINNING

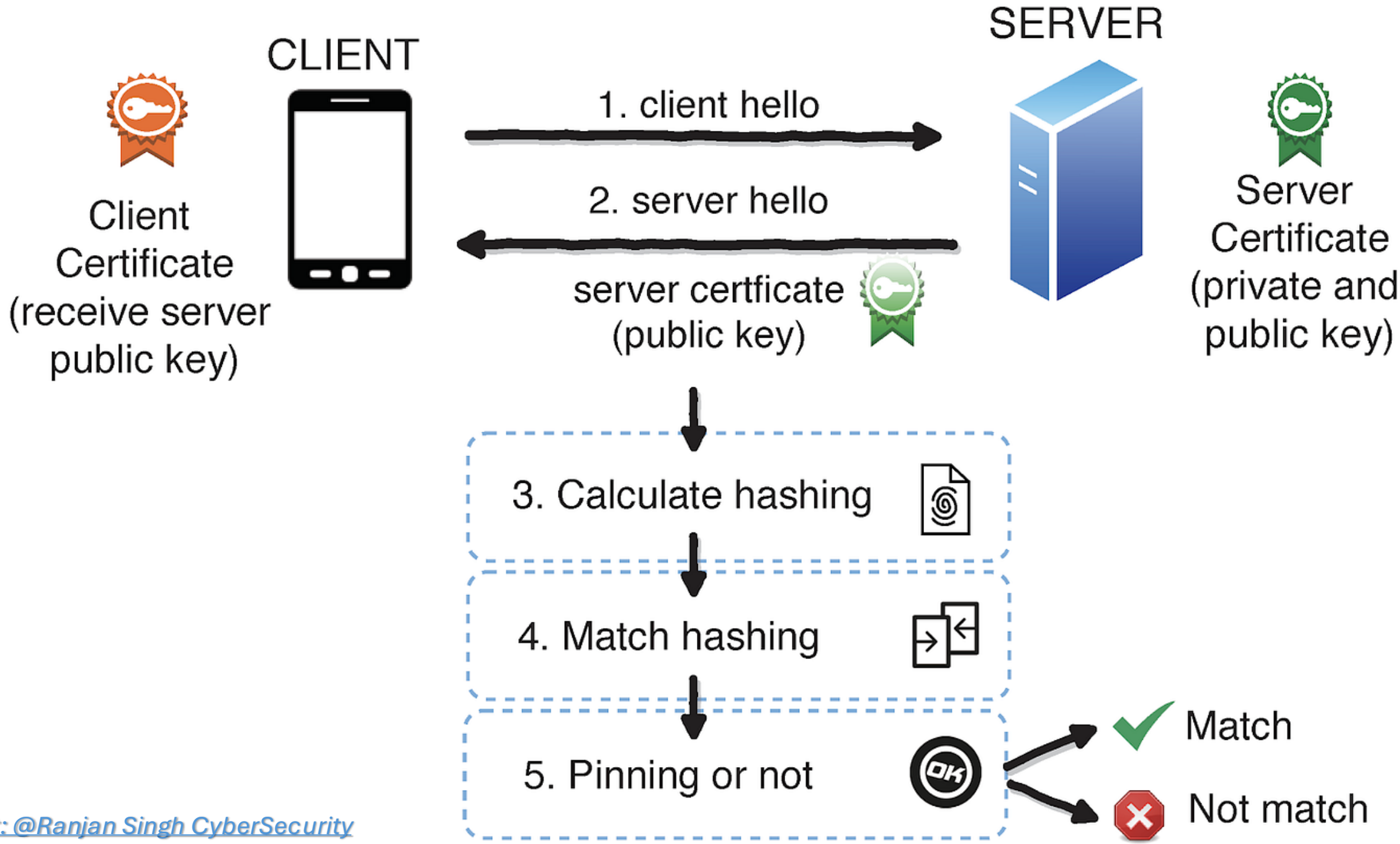
The certificate / public key (the pin) is stored within the apk itself

- Its hardcoded = easier to be extracted

DYNAMIC SSL PINNING

The certificate / public key is retrieved from the server from initial connection and cached it locally

- Can be vulnerable to MITM itself



Credit: @Ranjan Singh CyberSecurity

What is Frida

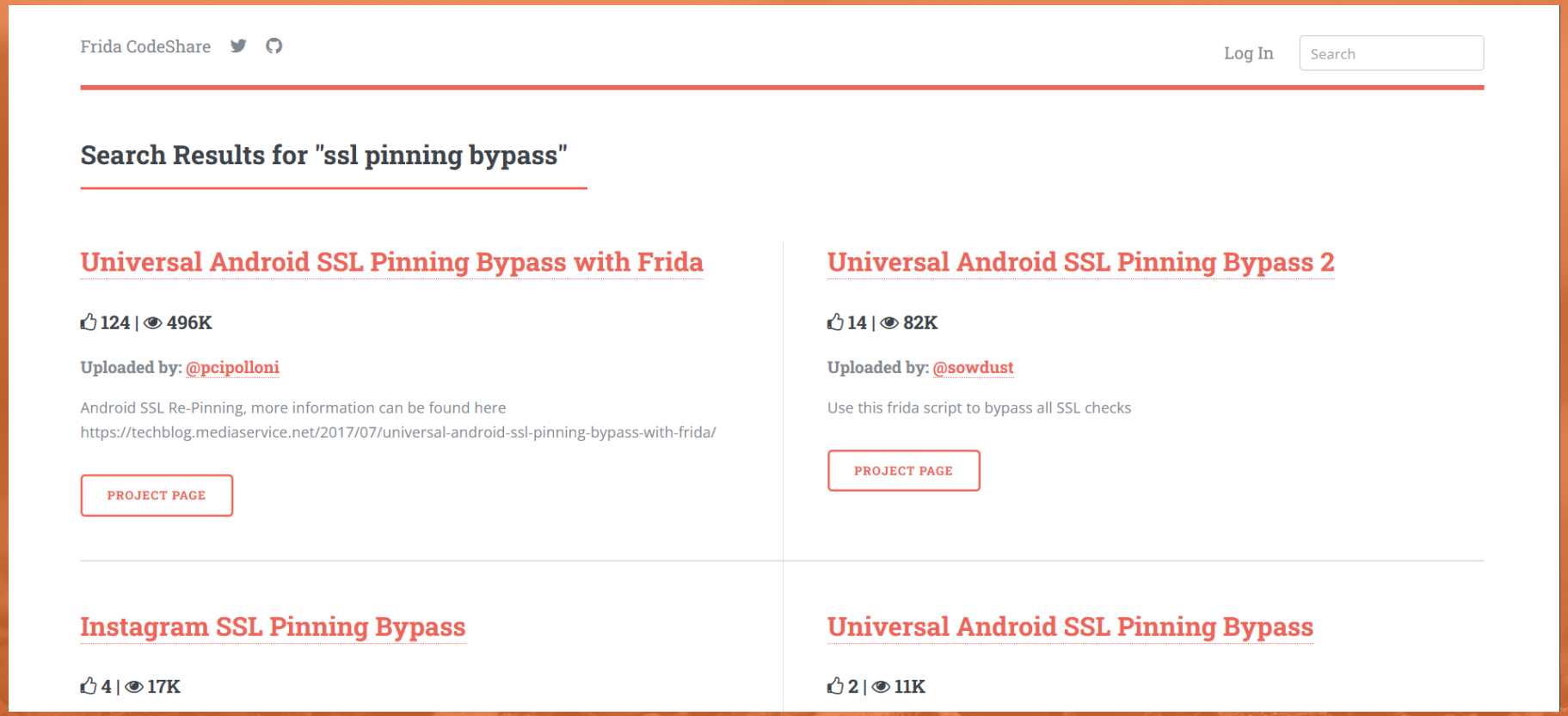
- An open-source framework that can be used to manipulate application's behaviour
- Use case:
  - Hook and modify Java methods at runtime
  - Bypass security controls for testing
  - Inspect internal application logic

Frida GitHub: https://github.com/frida/frida

pip3 install frida-tools

Bypassing SSL Pinning

- Your best friend is Frida codeshare (is like github for Frida codes):



What you need to bypass SSL Pinning

- Connected your AVD/Android device with PC (everything run in terminal)
- APK file binary – you can retrieve it by using ADB
- Frida-server on your android
- Depending on the implementation static or dynamic, develop your scripts, then you can inject scripts simultaneously loading the APK
- Normally with codeshare you will run this on your terminal:

```
frida –U --codeshare SCRIPT-CREATOR/PROJECT -f YOUR_APK_BINARY
```

Sample hook.js:

```
Java.perform(function () {
    var TargetedClass = Java.use("com.example.app.LoginManager");

    TargetClass.validatePassword.implementation = function (input) {
        console.log("[*] validatePassword called with → " + input);

        var result = this.validatePassword(input);

        console.log("[*] Original result: " + result);
        return result; // return original value (no alteration)
    };
});
```

DEMO TIME

PREPARE YOUR ENVIRONMENT

APK FOR DEMOSTRATION: frida-exercise.apk

- REBUILD APK
- FRIDA EXAMPLE
- SIMPLE SSL PINNING

CHALLENGE TIME

PREPARE YOUR ENVIRONMENT

APK FOR CHALLENGE: chall.zip
PASSWORD: TBA

THERE SHOULD BE 3 FLAG IN TOTAL
SUBMIT THE FLAG TO SHROOMISHIE IN DISCORD

THANKS FOR JOINING THE SESSION