Exploiting Esoteric Android Vulnerabilities

NotSoSecure part of claranet cyber security

Sanjay Gondaliya

Principal Security Consultant

@NotSoSecure

- 8+ Years of Experience in Information Technology
- Researcher and vulnerable app developer @Notsosecure
- Consulting experience involves large organizations across different sectors network and application security

Specialization

- Web Application
- Mobile Application
- Desktop Application
- External Infrastructure

GitHub Repositories

- Blacklist3r
- Android Application Analyzer
- Serialized Payload Generator

NotSoSecure part of claranet cyber security

SHARAN PANEGAV

Sr. Security Consultant

@NotSoSecure

- 4+ Years of Experience in Information Security
- Researcher and vulnerable app developer @Notsosecure
- Consulting experience involves Large organizations across different sectors network, system and application security

Specialization

- Web Application
- Mobile Application
- AWS Cloud Configuration Review
- External Infrastructure

Lab Setup

- Please visit the google doc link below and follow the instructions.
  - https://github.com/realsanjay/BesidesAHD2021

Android Application Penetration Testing

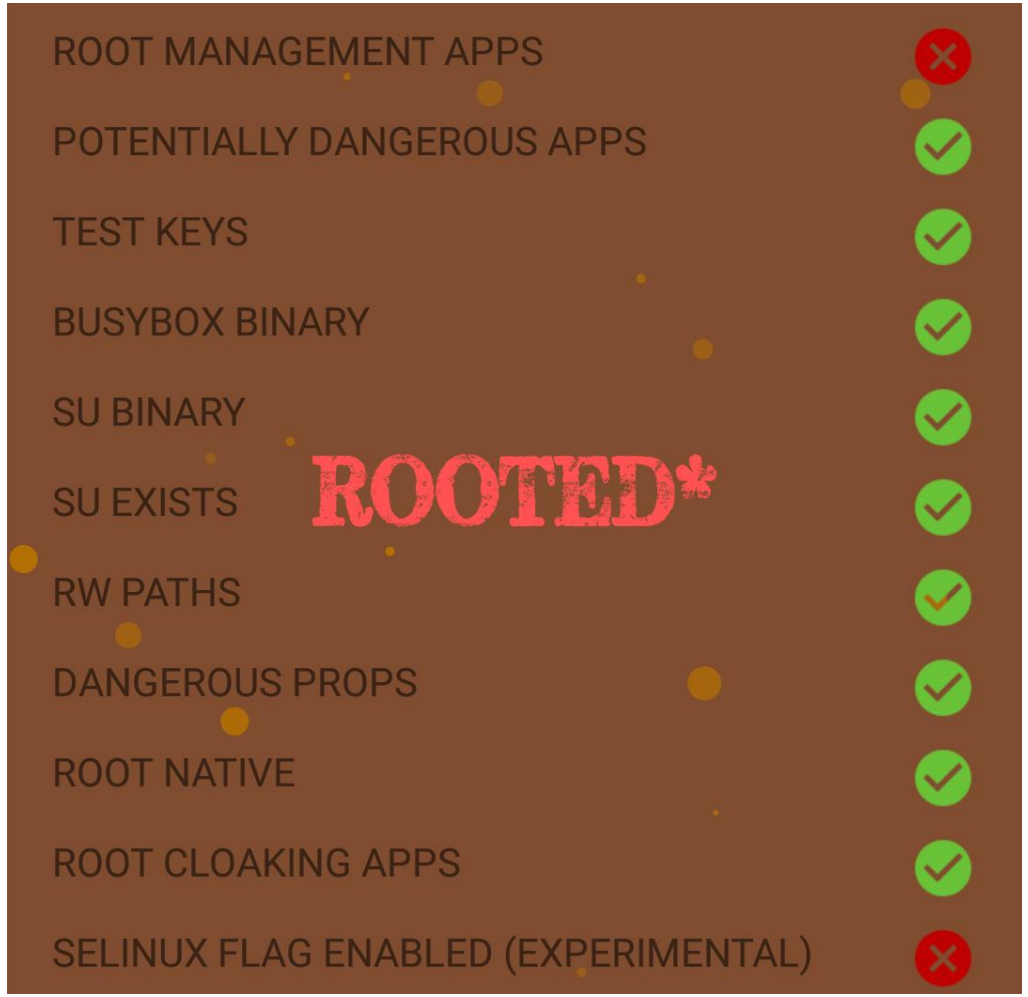
- Android Application Penetration Testing
  - Static Analysis
    - APK Analysis
    - Bypass client-side checks (Root, Emulator, Integrity, SSL Pinning Checks)
    - Application Sandbox Analysis
    - Esoteric Vulnerabilities
  - Dynamic Analysis
    - OWASP Web Top 10

What will we be Looking at ?

- Android Application Penetration Testing
  - Static Analysis
    - APK Analysis
      - MobSF
      - Drozer – Exported Components (Activity, Services, Content Providers, Broadcast Receivers)
    - Methods to Bypass client-side checks (Root, Emulator, Integrity, SSL Pinning Checks)
      - Smali code modification + Android Application Analyzer
      - Frida Hooking + Android Application Analyzer
    - Application Sandbox Analysis
      - Android Application Analyzer
    - Esoteric Vulnerabilities
      - WebView Attacks
      - Remote Debugging in Android
      - App link v/s Deep link
      - Exploiting Android File Picker Misconfiguration
      - Exploiting Mobile Passcodes using bash
  - Dynamic Analysis
    - OWASP Web Top 10

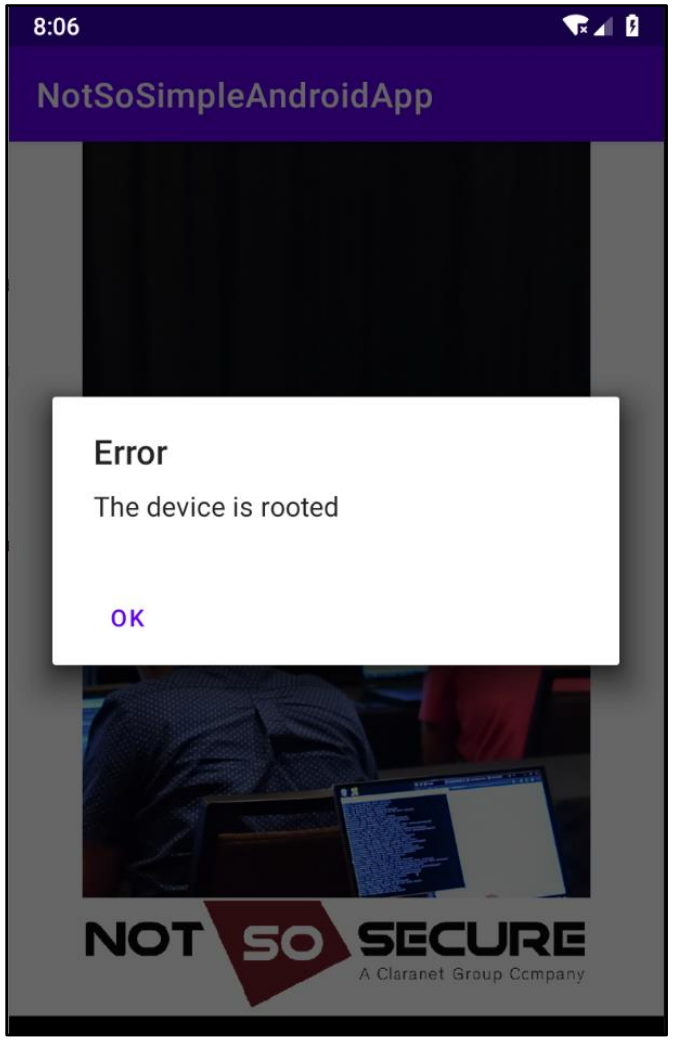
Root Detection

- Third Party Library
  - RootBeer - isRooted
  - Firebase – CommonUtils.isRooted
- Custom Implementation
  - Custom Function



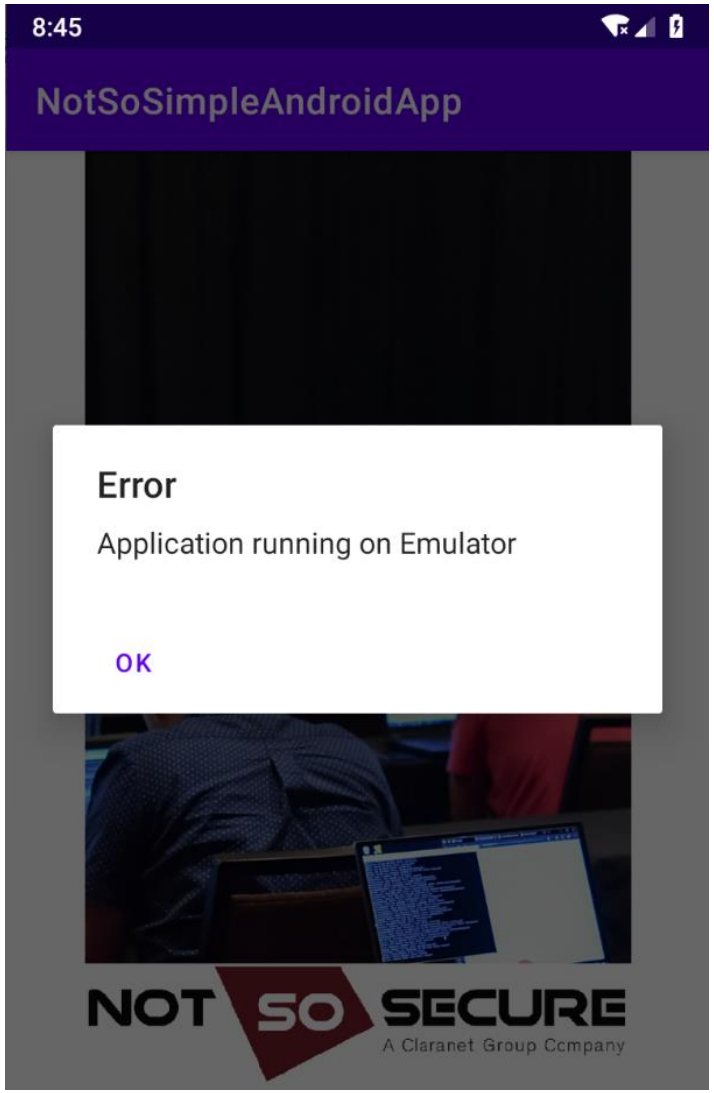
Root Detection

- Third Party Library
  - RootBeer - isRooted
  - Firebase – CommonUtils.isRooted
- Custom Implementation
  - Custom Function



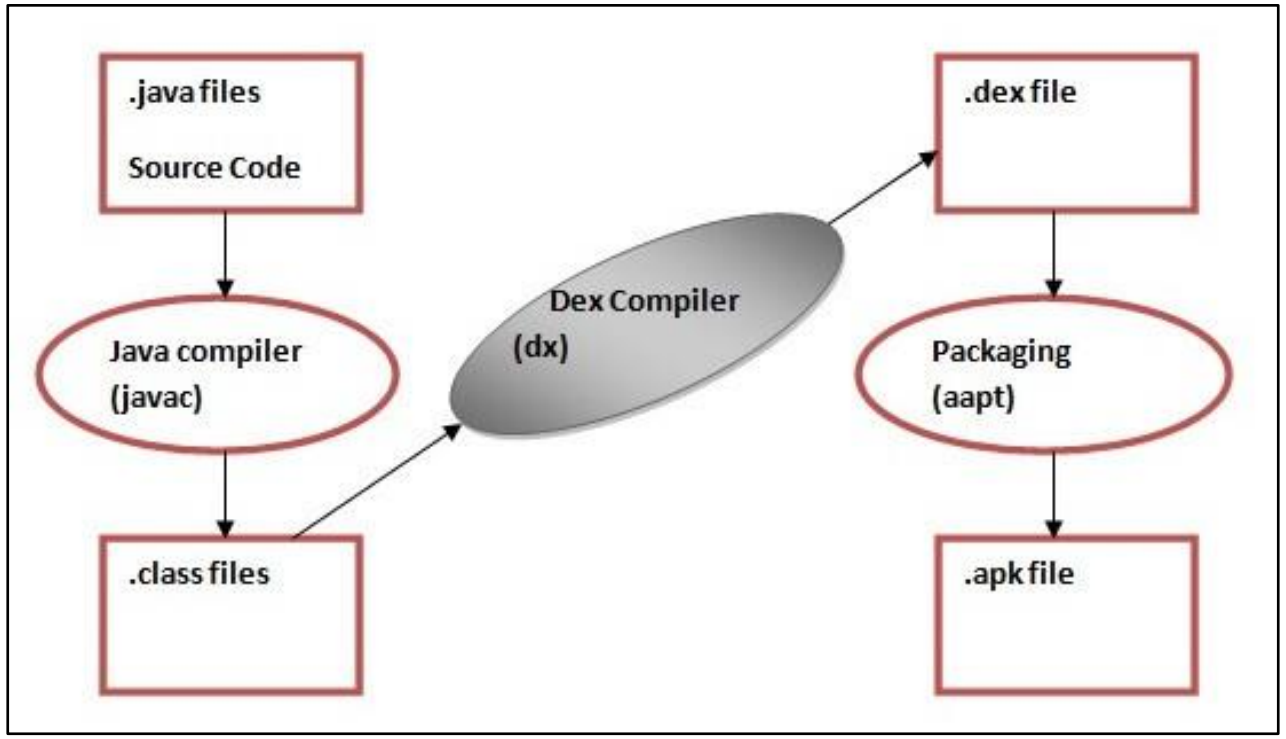
Emulator Detection

- Check the following information
  - Build
  - Brand
  - Device
  - Product
  - Based on keywords
    - Generic
    - Unkonwn
    - Google_sdk
    - Emulator
    - Genymotion
    - Generic



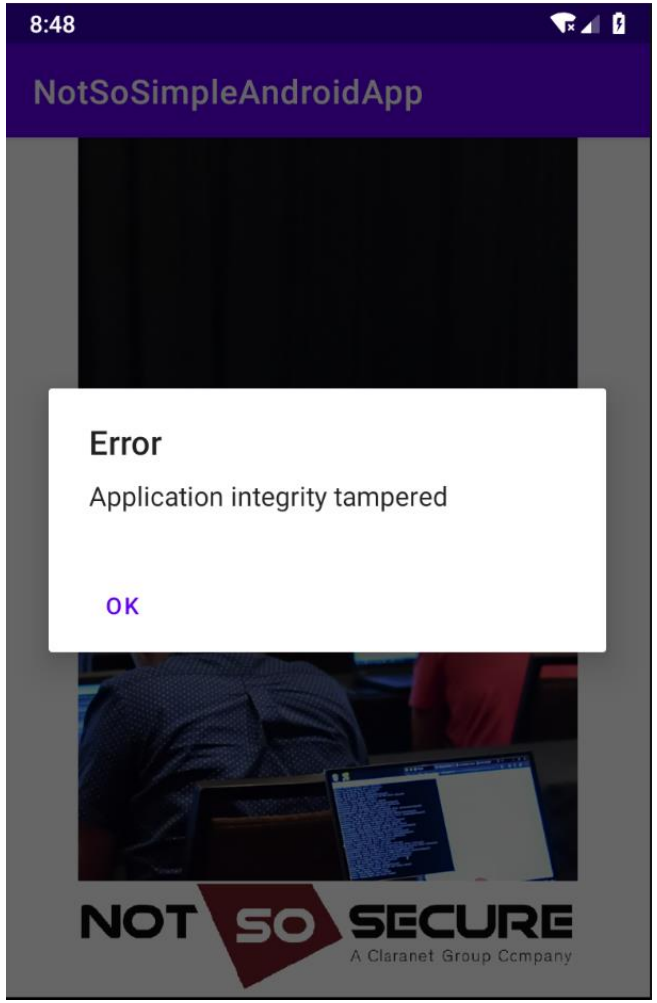
Application Integrity Check

- Common Method
  - Verify checksum of classes.dex



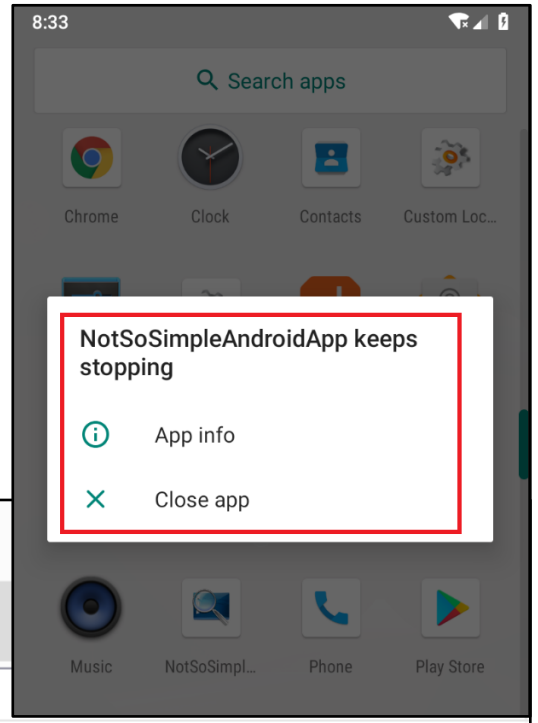
Integrity Check

- Common Method
  - Verify checksum of Classes.dex



SSL Pinning

- Third Party Library
  - OkHttp – CertificatePinner
  - Conscrypt – TrustManagerImpl
- Custom Implementation
  - Custom Function



Event log

Filter Critical Error Info Debug

Type	Source	Message
Error	Proxy	[5] The client failed to negotiate a TLS connection to instantmessaging-pa.googleapis.com:443: Received fatal alert: certificate_...
Error	Proxy	[11] The client failed to negotiate a TLS connection to www.notsosecure.com:443: Received fatal alert: certificate_unknown
Error	Proxy	[13] The client failed to negotiate a TLS connection to www.google.com:443: Received fatal alert: certificate_unknown
Info	Extender	JS Link Finder: BurpJSLinkFinder Passive Scanner enabled
Info	Proxy	Proxy service started on 127.0.0.1:8080

SSL Pinning

- Third Party Library
  - OkHttp – CertificatePinner
  - Conscrypt – TrustManagerImpl
- Custom Implementation
  - Custom Function

Checks SHA256 Hash in the Certificate

It’s Demo time!

Root Detection Bypass

Emulator Detection Bypass

Integrity Check Bypass

SSL Pinning Bypass

Root Detection Bypass

Emulator Detection Bypass

Integrity Check Bypass

SSL Pinning Bypass

Frida

- Dynamic instrumentation toolkit for
  - Developers
  - Reverse-engineers
  - Security researchers.
- Allows
  - Inject your own scripts into black box processes
  - Hook any function of the application

Frida - Installation

- Client
  - pip install frida-tools

```
C:\Users\Sanjay>pip install frida-tools
Requirement already satisfied: frida-tools in c:\users\sanjay\appdata\local\programs\python\python3
9\lib\site-packages (10.4.1)
Requirement already satisfied: colorama<1.0.0,>=0.2.7 in c:\users\sanjay\appdata\local\programs\pyt
hon\python39\lib\site-packages (from frida-tools) (0.4.4)
Requirement already satisfied: frida<16.0.0,>=15.0.0 in c:\users\sanjay\appdata\local\programs\pyth
on\python39\lib\site-packages (from frida-tools) (15.1.10)
Requirement already satisfied: prompt-toolkit<4.0.0,>=2.0.0 in c:\users\sanjay\appdata\local\progra
ms\python\python39\lib\site-packages (from frida-tools) (3.0.18)
Requirement already satisfied: pygments<3.0.0,>=2.0.2 in c:\users\sanjay\appdata\local\programs\pyt
hon\python39\lib\site-packages (from frida-tools) (2.9.0)
Requirement already satisfied: setuptools in c:\users\sanjay\appdata\local\programs\python\python39
\lib\site-packages (from frida<16.0.0,>=15.0.0->frida-tools) (56.0.0)
Requirement already satisfied: wcwidth in c:\users\sanjay\appdata\local\programs\python\python39\li
b\site-packages (from prompt-toolkit<4.0.0,>=2.0.0->frida-tools) (0.2.5)
WARNING: You are using pip version 21.1.1; however, version 21.3.1 is available.
You should consider upgrading via the 'c:\users\sanjay\appdata\local\programs\python\python39\pytho
n.exe -m pip install --upgrade pip' command.
```

- Server
  - Android Device

github.com/frida/frida/releases

frida-qml-15.1.11-windows-x86_64.exe	14.1 MB
frida-server-15.1.11-android-arm.xz	6.46 MB
frida-server-15.1.11-android-arm64.xz	13.7 MB
frida-server-15.1.11-android-x86.xz	13.8 MB
frida-server-15.1.11-android-x86_64.xz	27.7 MB
frida-server-15.1.11-linux-arm64.xz	7.01 MB

```
C:\Users\Sanjay>adb shell
vbox86p:/ # cd /data/local/tmp
vbox86p:/data/local/tmp # ls
frida-server-15.1.10-android-x86
vbox86p:/data/local/tmp # ./frida-server-15.1.10-android-x86 &
```

Frida - Installation

```
C:\Users\Sanjay>frida-ps -U
  PID  Name
-----  ----------------------------------------------
14789  Chrome
14748  Files
12642  Gallery
11940  Google Play Store
  241  adbd
  503  android.hardware.audio@2.0-service
  504  android.hardware.camera.provider@2.4-service
  505  android.hardware.cas@1.0-service
  506  android.hardware.configstore@1.1-service
  507  android.hardware.drm@1.0-service
  508  android.hardware.gnss@1.0-service
  509  android.hardware.graphics.allocator@2.0-service
  510  android.hardware.graphics.composer@2.1-service
  511  android.hardware.health@2.0-service.genymotion
  175  android.hardware.keymaster@3.0-service
  512  android.hardware.light@2.0-service
  513  android.hardware.memtrack@1.0-service
  514  android.hardware.power@1.0-service
```

Root Detection Bypass

Integrity Check Bypass

Emulator Detection Bypass

SSL Pinning Bypass

Root Detection Bypass

Integrity Check Bypass

Emulator Detection Bypass

SSL Pinning Bypass

Root Detection Bypass

Integrity Check Bypass

Emulator Detection Bypass

SSL Pinning Bypass

Esoteric Vulnerabilities

- WebView and its vulnerabilities
- Remote Debugging
- Exploiting WebView Interface
- Deep link Vs App link
- Exploiting File Picker misconfiguration
- Exploiting Mobile passcodes using bash

Introduction to WebView

- Allows to display web content as part of your activity layout.
- Third Party Application Integration
- Advertisement
- Native App Extensions

How to Detect WebView?

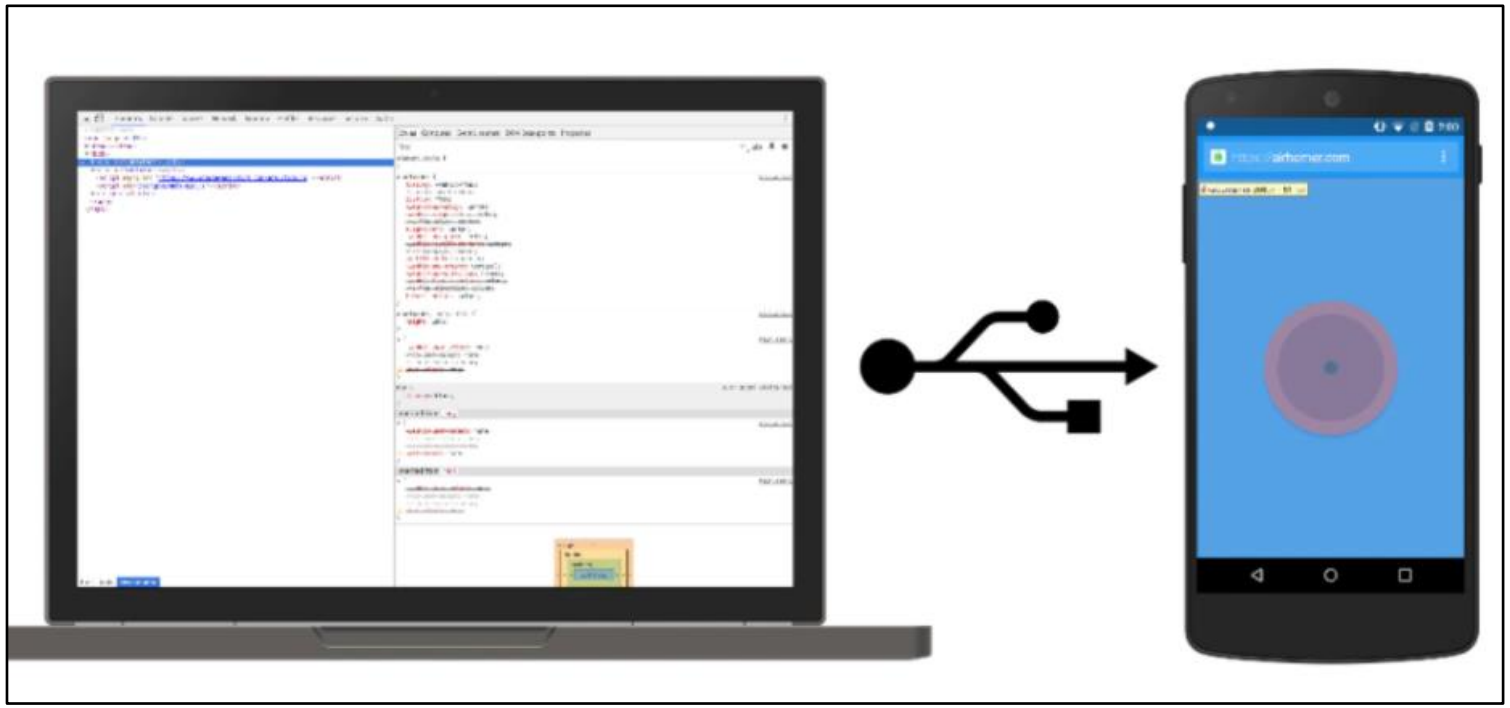
```
import android.net.Uri;
import android.os.Bundle;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import androidx.appcompat.app.AppCompatActivity;

public class WebViewActvitiy extends AppCompatActivity {
    private Uri data;
    WebView webView;

    /* access modifiers changed from: protected */
    @Override // androidx.activity.ComponentActivity, androidx.core.app.ComponentActivity, androidx.appcompat
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(C0431R.layout.activity_web_view_actvitiy);
        WebView webView2 = (WebView) findViewById(C0431R.C0434id.webview2);
        this.webView = webView2;
        webView2.setWebViewClient(new WebViewClient());
        this.webView.getSettings().setLoadsImagesAutomatically(true);
        this.webView.getSettings().setJavaScriptEnabled(true);
        this.webView.setScrollBarStyle(0);
        this.webView.addJavascriptInterface(new WebAppInterface(this), "MyInterface");
        WebView.setWebContentsDebuggingEnabled(true);
        Bundle extras = getIntent().getExtras();
        if (extras != null) {
            this.webView.loadUrl(extras.getString("URL"));
            return;
        }
        this.webView.loadUrl(getIntent().getData().getQueryParameter("URL"));
    }
}
```

Common Vulnerabilities In WebView

- Exported WebView
- JavaScript Interface
- Universal File access from file is enabled for WebView
- Cross-site Scripting (Universal XSS on that Application)
- Exploitation Content Provider

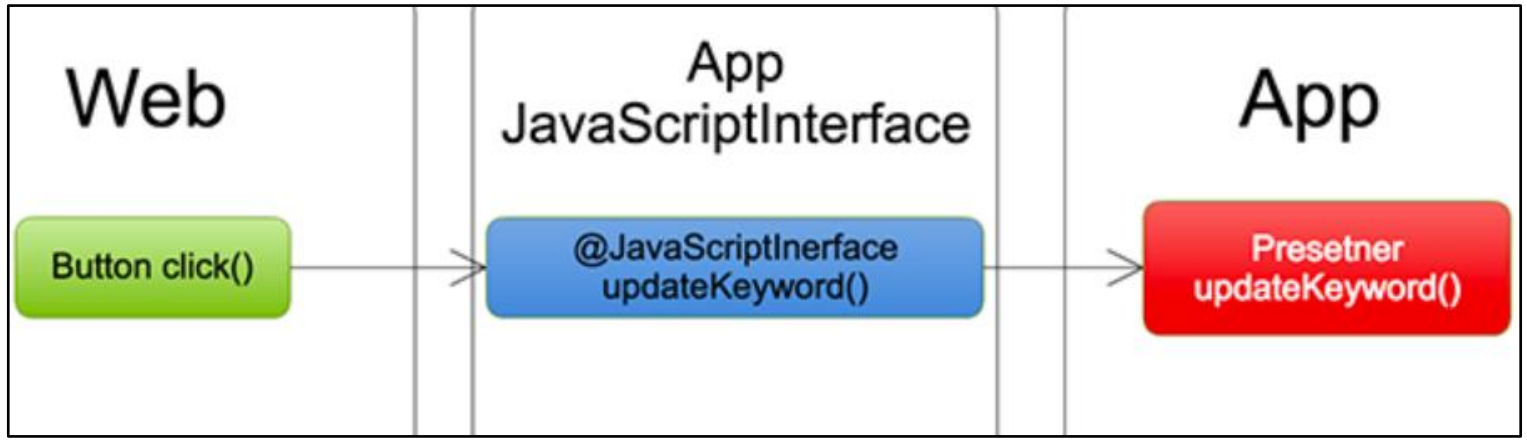
Remote debugging in WebView

- Need of Remote Debug
- Debug live content on
  - Android device from
    - Windows
    - Mac
    - Linux computer
- Inspect and debug live content of Android device
- DOM manipulations .

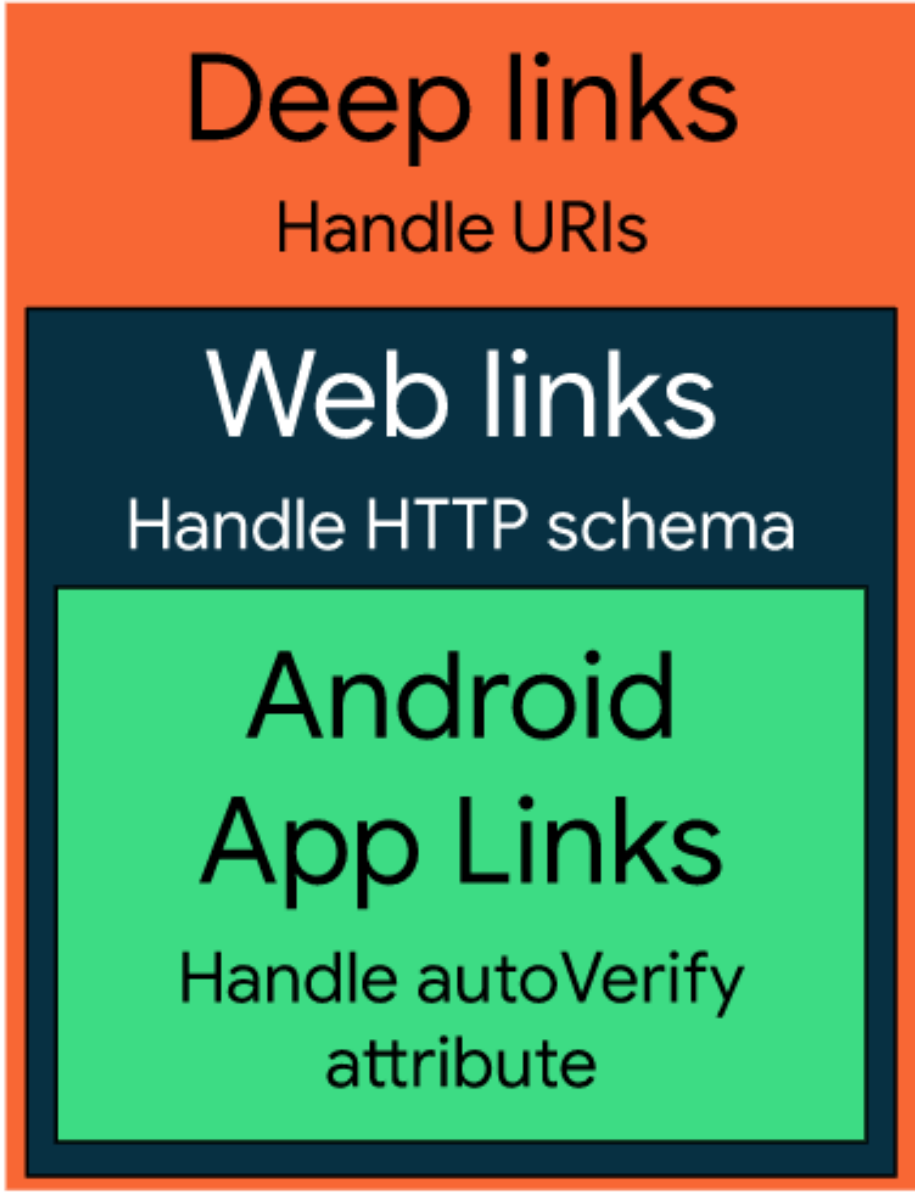


JavaScript Interface

- What is JavaScript Interface
- Java methods on Android application
  - Data return
  - Perform actions

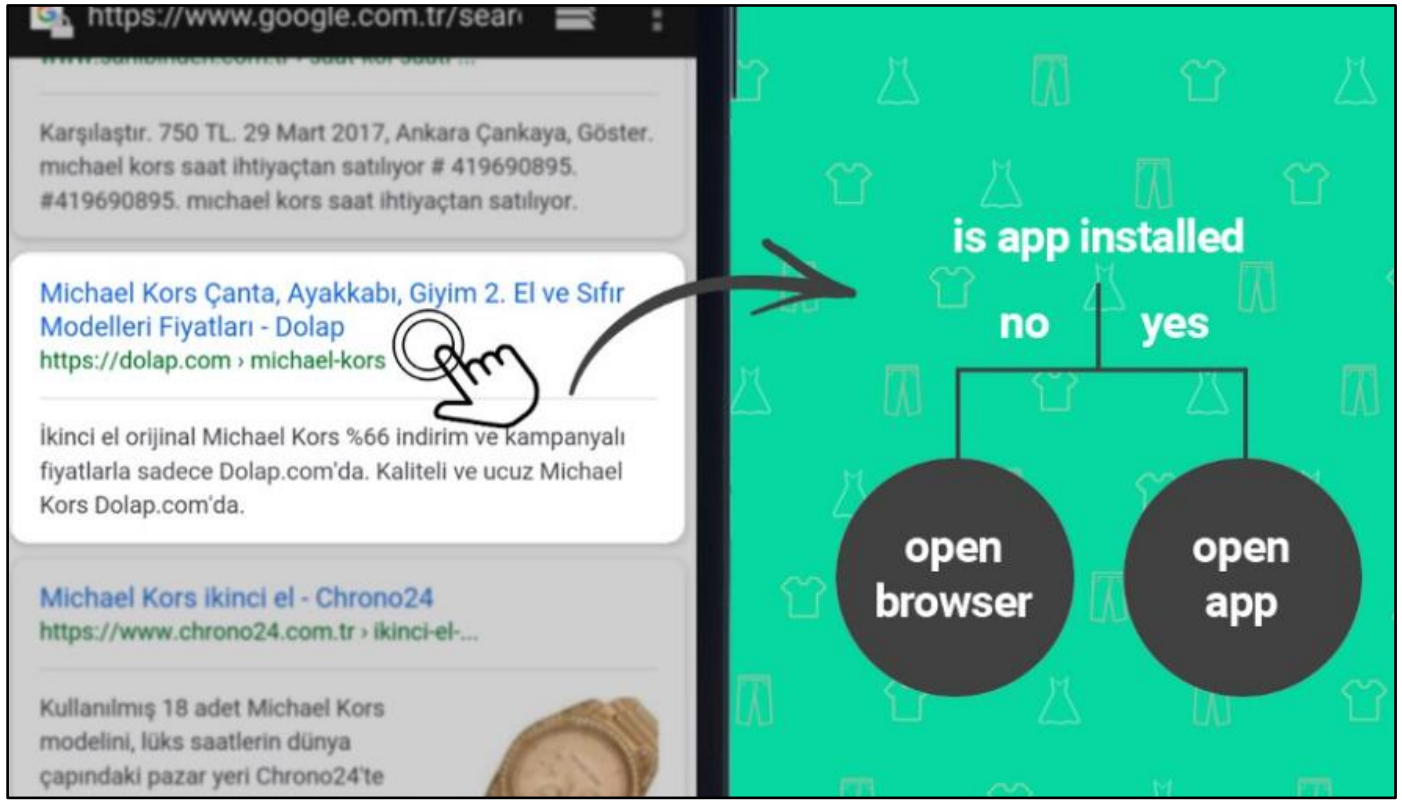


Deep Links VS App Links



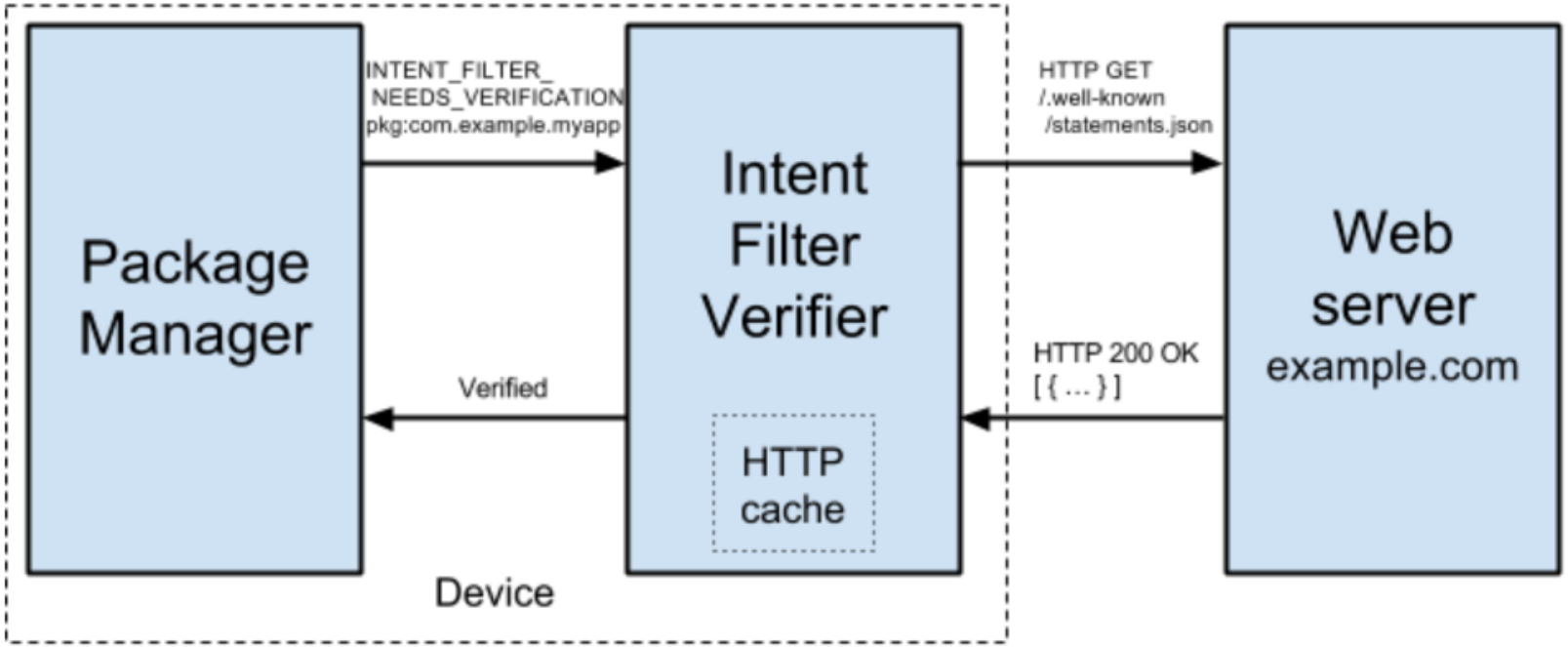
Android Deep links

- Open the user's preferred app that can handle the URI, if one is designated.
- Open the only available app that can handle the URI.
- Allow the user to select an app from a dialog.



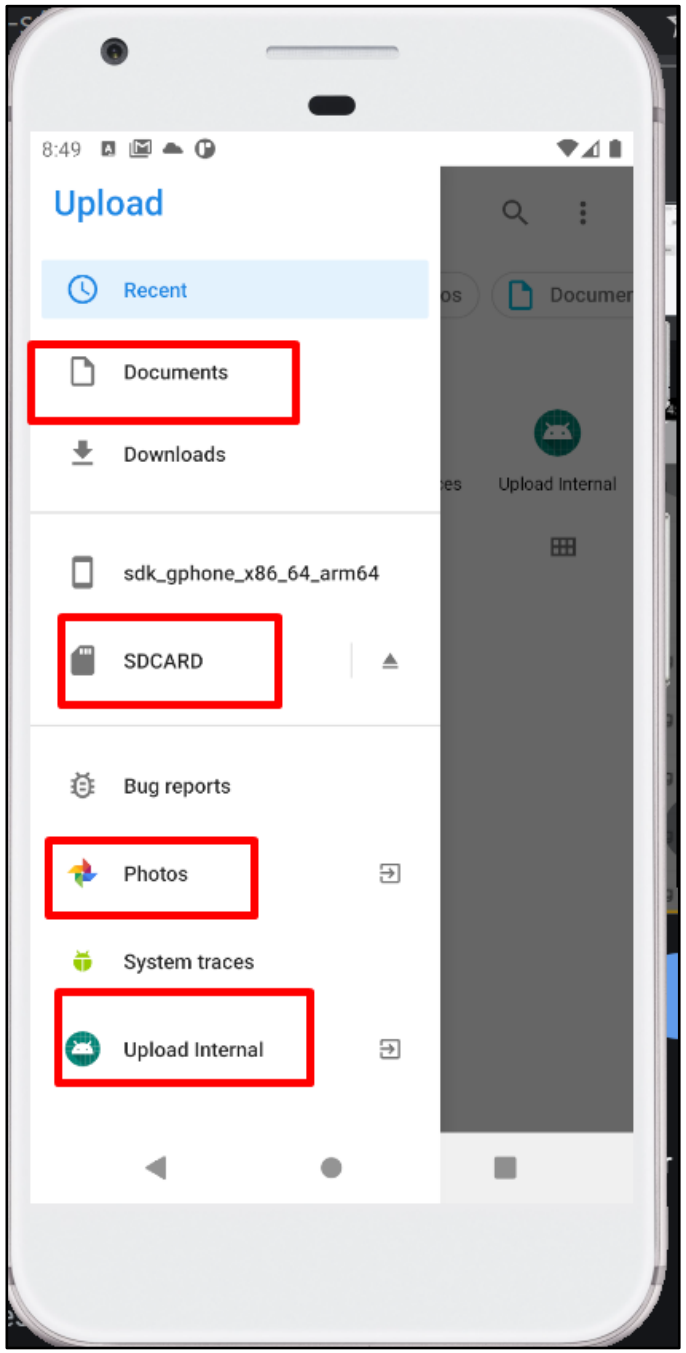
Android App links Verification

- An Android App Link is a deep link based on website URL that has been verified to belong to your website.
- So, clicking one of these immediately opens app if it's installed—the disambiguation dialog does not appear



Android File Picker

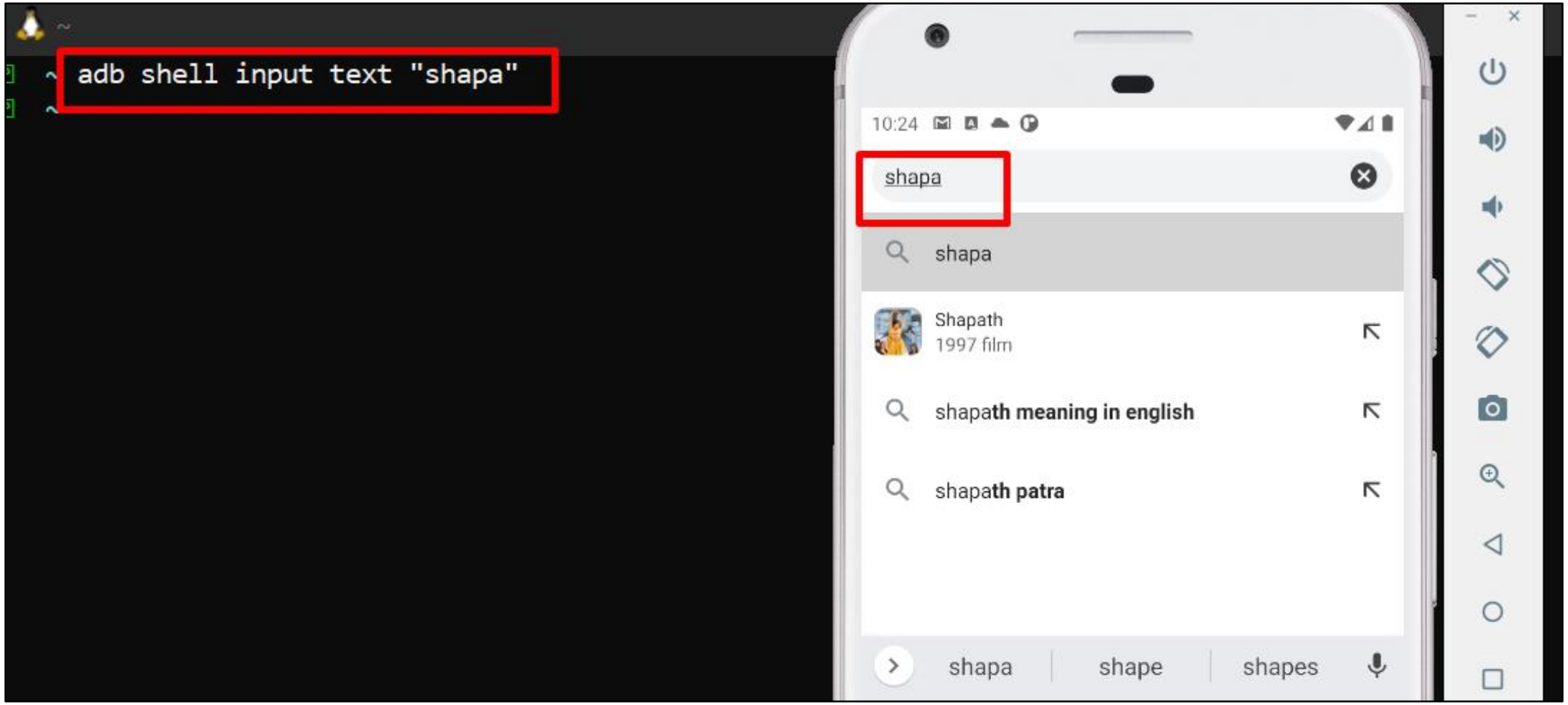
- A file picker which allows to select images and videos with flexibility.
- It supports selection of files by specifying its file type.



Android Passcode

- Setting a passcode on mobile device is very basic security step.
- This will help keep unauthorized users away from your device.
- Input Based Passcode
- Touch Based Passcode

Android Input and Touch Events



Key Workshop Takeaways

- There are multiple ways and techniques to bypass the application level checks.
- Tools and techniques to exploit esoteric vulnerabilities in android application.
- How we can secure android application from remote attacks on deep links
- How to implement Secure deep links and WebView

Thank You

END PRESENTATION