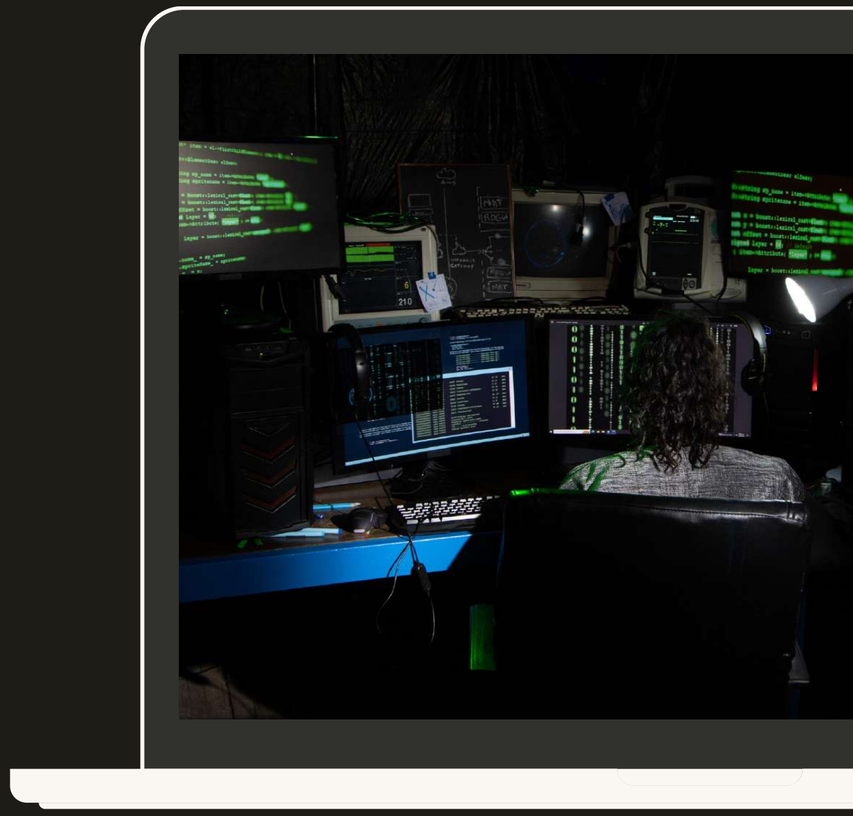
MOBILE APPLICATION SECURITY

BY NAINA SHARMA



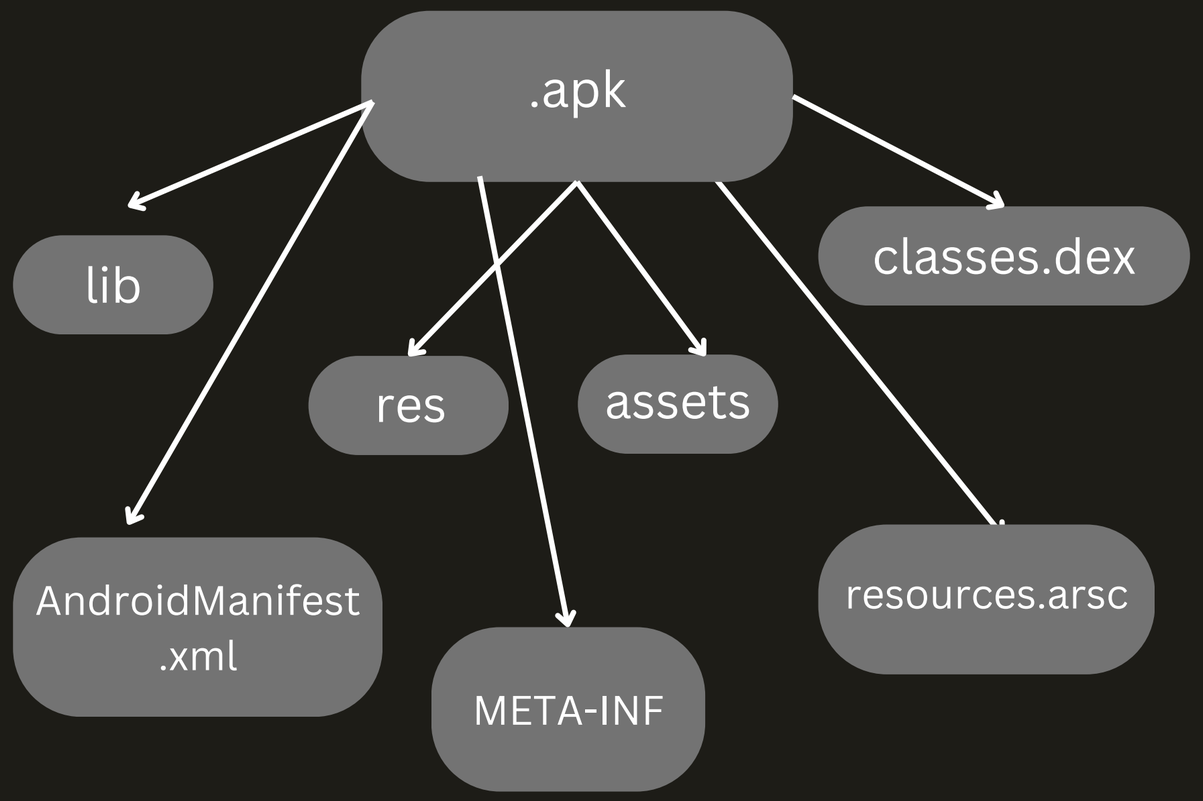
BugBase

AGENDA

- Android Application Security

BugBase

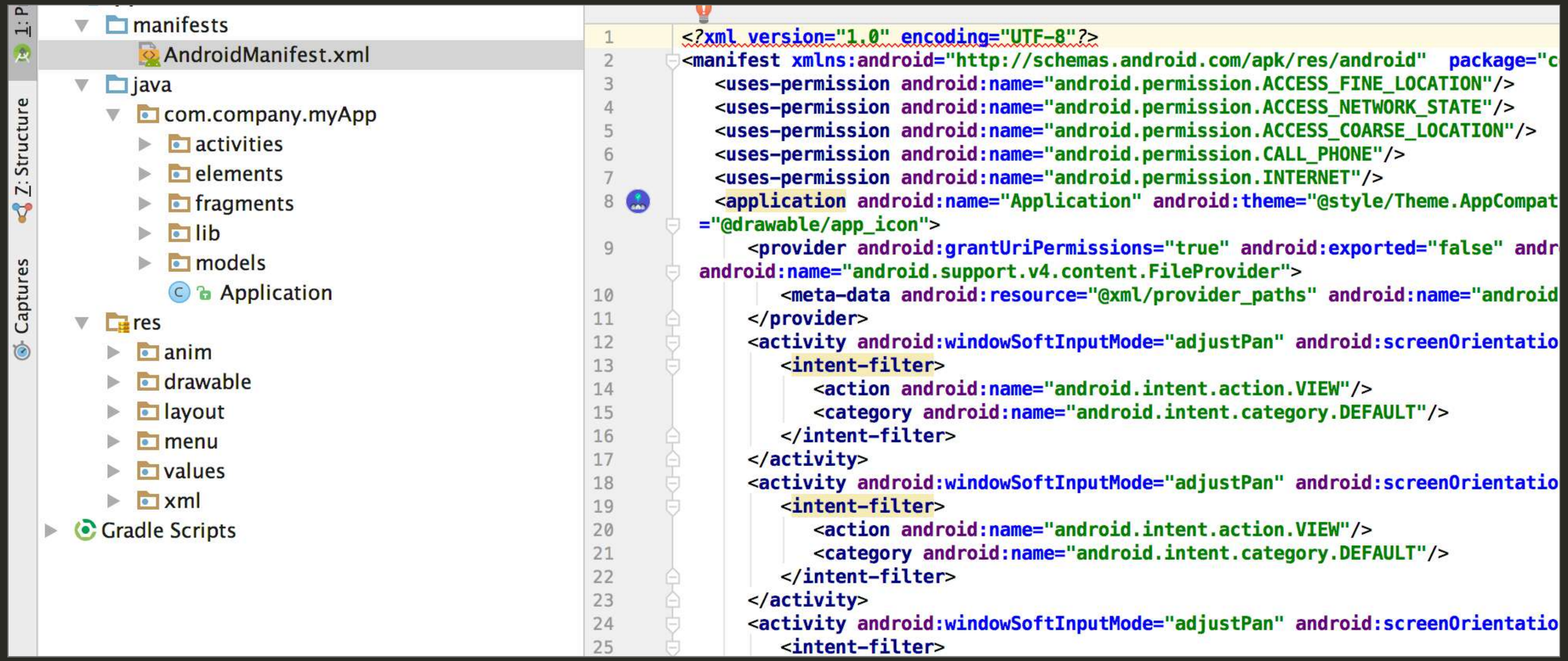
Application Structure



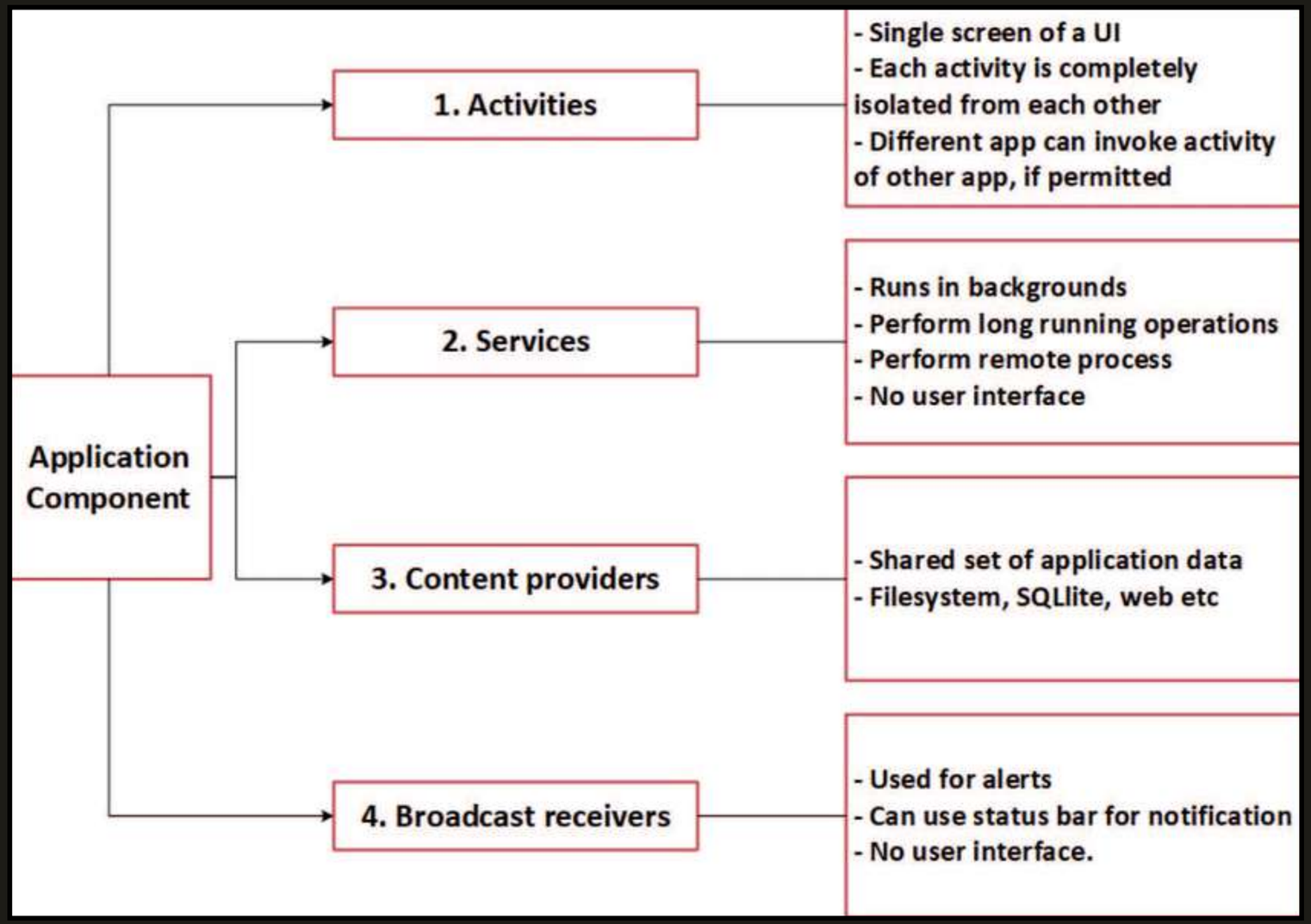
ANDROID

BugBase

AndroidManifest.xml



Application Components



Common misconfigurations:

- Exported Components
- Improper Intent Filter

BugBase

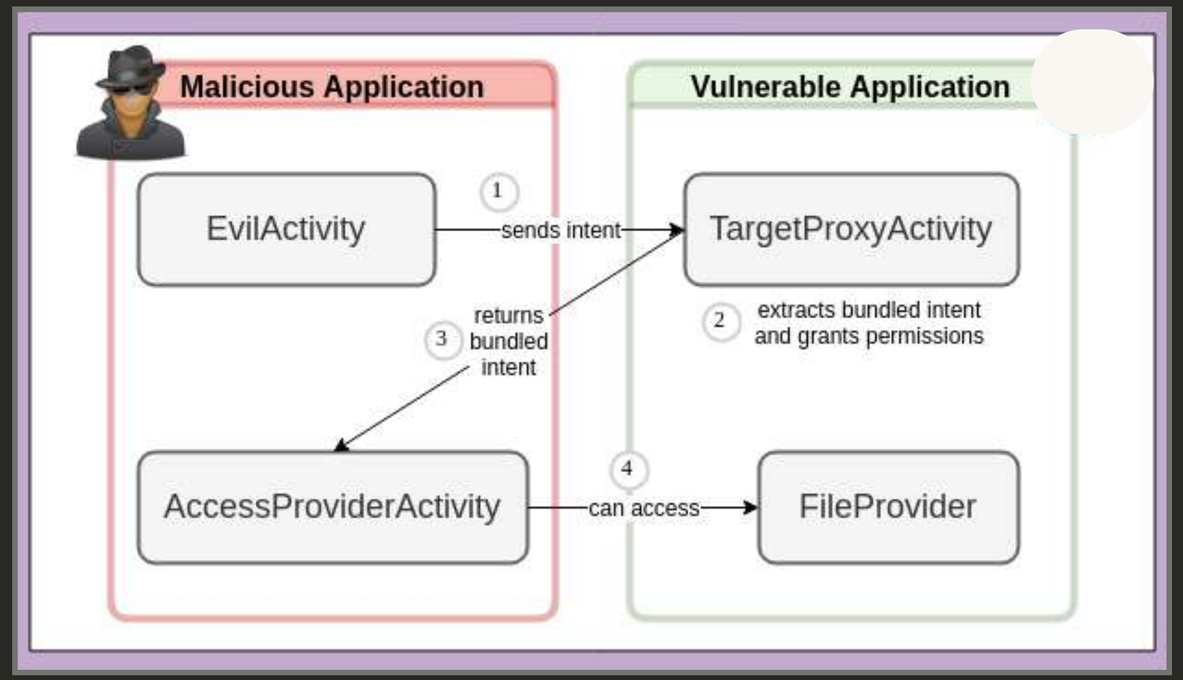
Intents

Intents

Intents are the means of communication that acts as a facilitator when the exchange of message occurs between different components within the same application or from one application to another.

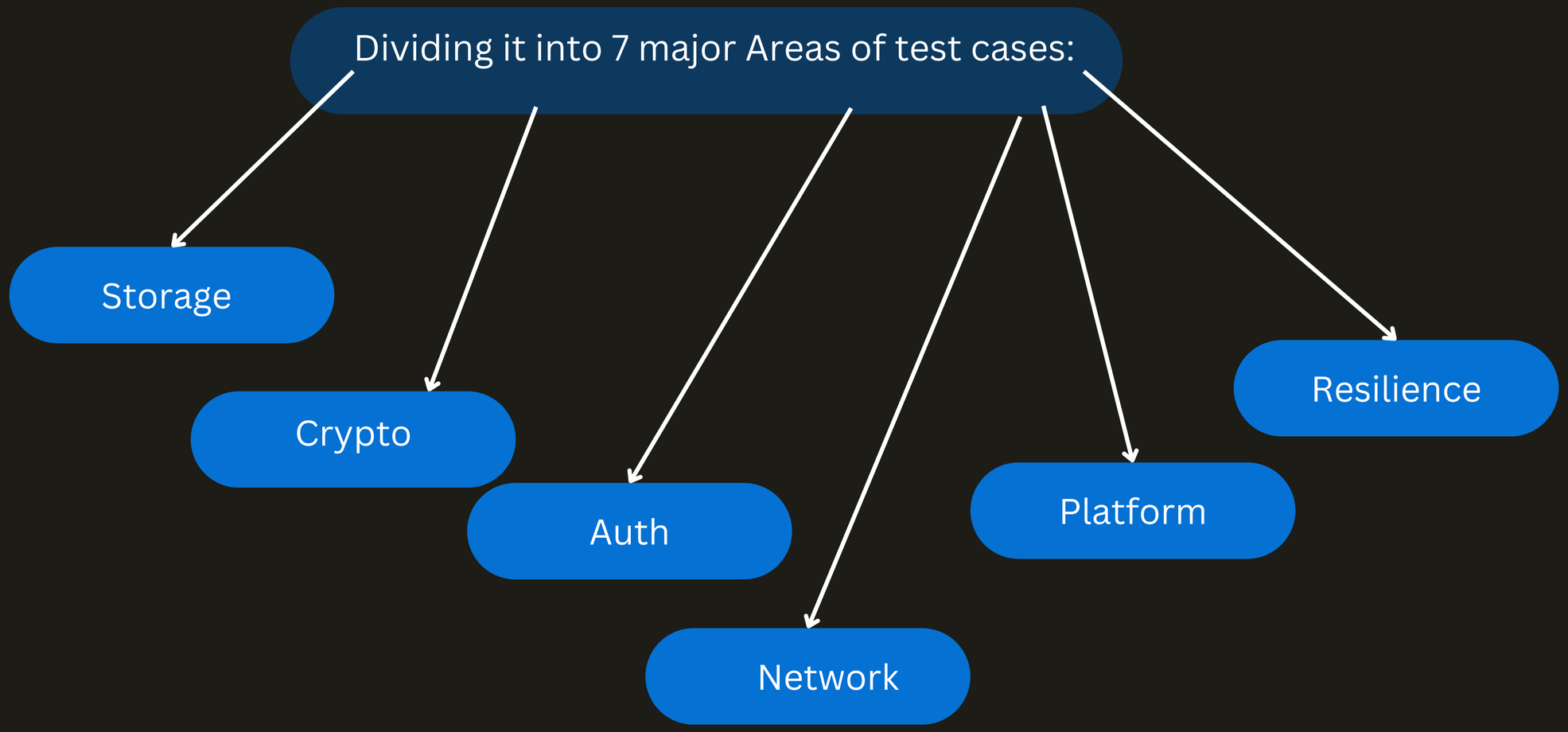
Intent Filters

To advertise which implicit intents your app can receive, declare one or more intent filters for each of your app components with an <intent-filter> element in your manifest file. Each intent filter specifies the type of intents it accepts based on the intent's action, data, and category.



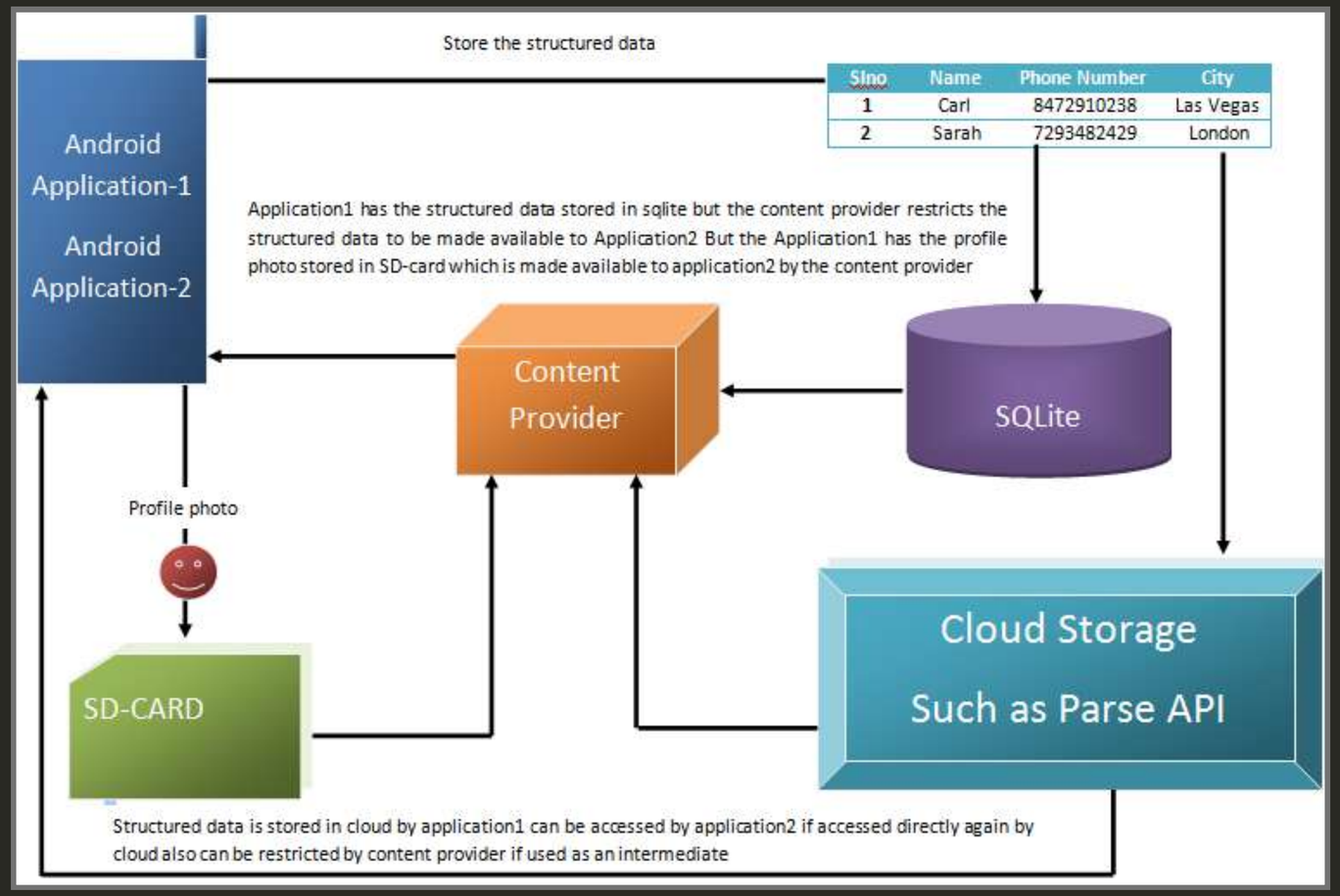
BugBase

Performing a Security Assessment on Mobile Apps



BugBase

Storage



Sensitive data can be found in:

Local Storage -

- Ex: SharedPreferences, internal and external storage, SQLiteDatabase, cached files.
- SQLite databases: /data/data/<package-name>/databases
- SharedPreferences: /data/data/<package-name>/shared_prefs
- File Permissions: /data/data/<package-name>
- Realm Database: /data/data/<package-name>/files/

BugBase

Storage

Side Channel Data Leakage:

- Logs: adb logcat | grep "$(adb shell ps | grep <package-name> | awk '{print $2}')"
- Data Shared with third parties via embedded services: Check all requests to external services for embedded sensitive information.
- Keyboard cache or text input fields: <EditText android:id="@+id/KeyBoardCache" android:inputType="textNoSuggestions" />
- Backup: adb backup "-apk -nosystem <package-name>"
- Memory: Fridump, Objection

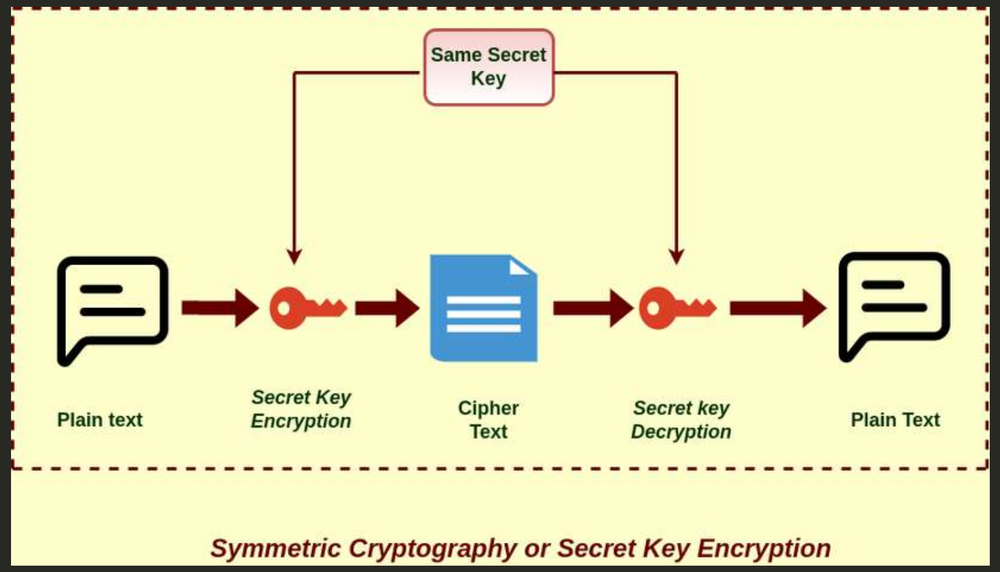
BugBase

Crypto

The following test cases need to be looked into:

- Testing Symmetric Cryptography:
  - symmetric algorithms (such as DES, AES, etc.)
  - grep -r "SecretKeySpec"

- Random Number Generation:
  - Check for the availabiliy of java.util.Random
- Configuration of Cryptographic Standard Algorithms:
  - classes Cipher, Mac, MessageDigest, Signature
  - interfaces Key, PrivateKey, PublicKey, SecretKey
  - functions getInstance, generateKey
  - exceptions KeyStoreException, CertificateException, NoSuchAlgorithmException
  - classes which uses java.security.*, javax.crypto.*, android.security.* and android.security.keystore.* packages.



BugBase

Auth

- Biometric Authentication check:
  - Fingerprint bypass: Frida scripts to bypass authentication when the CryptoObject is not used in the authenticate method of the BiometricPrompt class are available openly. The authentication implementation relies on the callback onAuthenticationSucceded being called.

  - Fingerprint bypass via exception handling: Frida scripts to attempt to bypass authentication when the CryptoObject is used, but used incorrectly. A detailed explanation can be found in the section "Crypto Object Exception Handling" in the blog post.

- Testing confirm Credentials: Validate the duration of time (seconds) for which the key is authorized to be used after the user is successfully authenticated. This is only needed if setUserAuthenticationRequired is used.

BugBase

Network

- Data Encryption on the network: Check network security config file
- TLS Settings: Report usage of TLS v1.0 and TLS v1.1

```
<application android:networkSecurityConfig="@xml/network_security_config"
```

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <base-config cleartextTrafficPermitted="false" />
    <domain-config cleartextTrafficPermitted="true">
        <domain>localhost</domain>
    </domain-config>
</network-security-config>
```

BugBase

Network

- Endpoint identity verification:
- STATIC ANALYSIS
  - Checking custom trust anchors
  - Verifying server certificate
  - Webview Server Certificate Verification
  - Hostname verification
- DYNAMIC ANALYSIS
  - Self Signed Certificate
  - Accepting certificates with an untrusted CA
  - Accepting incorrect hostnames:

```xml
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config>
        <domain includeSubdomains="false">owasp.org</domain>
        <trust-anchors>
            <certificates src="system" />
            <certificates src="user" />
        </trust-anchors>
    </domain-config>
</network-security-config>
```

```
$ keytool -export -rfc
  -keystore your-upload-keystore.jks
  -alias upload-alias
  -file output_upload_certificate.pem
```

Command to sign a certiifcate with the help of keytool

BugBase

Network

Custom Certificate Stores and Certificate Pinning:

DYNAMIC

- SSL Pinning Bypass:
  - Universal Frida Script
  - Objection
  - Smalli Patching

```
.method public checkServerTrusted([Ljava/security/cert/X509Certificate;Ljava/lang/String;)V
    .locals 6

    array-length v0, p1

    const/4 v1, 0x0

    const/4 v2, 0x0

    return-void

    :goto_0
    if-ge v2, v0, :cond_0

    aget-object v3, p1, v2

    const-string v4, "Chain-Certs "

    invoke-static {v4}, Lb/b/a/a/a;->a(Ljava/lang/String;)Ljava/lang/StringBuilder;

    move-result-object v4

    invoke-virtual {v3}, Ljava/security/cert/X509Certificate;->getIssuerDN()Ljava/security/Principal;

    move-result-object v5

    invoke-interface {v5}, Ljava/security/Principal;->toString()Ljava/lang/String;

    move-result-object v5
```

Smalli code

BugBase

Platform

- Sensitive Data Exposed through IPC Mechanisms:
  - Exported Providers
  - SQL Injection in Content Providers

STATIC

```
<provider
      android:authorities="com.mwr.example.sieve.DBContentProvider"
      android:exported="true"
      android:multiprocess="true"
      android:name=".DBContentProvider">
    <path-permission
```

DYNAMIC

```
dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Passwords/ --ver
_id: 1
service: Email
username: incognitoguy50
password: PSFjqXIMVa5NJFudgDuuLVgJYFD+8w== (Base64 - encoded)
email: incognitoguy50@gmail.com
```

BugBase

Platform

SQL injection by manipulating the projection and selection fields that are passed to the content provider

DYNAMIC

```
dz> run scanner.provider.injection -a com.mwr.example.sieve
Scanning com.mwr.example.sieve...
Injection in Projection:
  content://com.mwr.example.sieve.DBContentProvider/Keys/
  content://com.mwr.example.sieve.DBContentProvider/Passwords
  content://com.mwr.example.sieve.DBContentProvider/Passwords/
Injection in Selection:
  content://com.mwr.example.sieve.DBContentProvider/Keys/
  content://com.mwr.example.sieve.DBContentProvider/Passwords
  content://com.mwr.example.sieve.DBContentProvider/Passwords/
```

```
$ adb shell content query --uri content://com.owaspomtg.vulnapp.provider.CredentialProvider/
Row: 0 id=1, username=admin, password=StrongPwd
Row: 1 id=2, username=test, password=test
...
```

BugBase

Platform

Attacking Exported Components

Through Activity Manager

am start -n PackageName/.ActivityName

Through Drozer

run app.activity.start --component com.android.insecurebankv2 com.android.insecurebankv2.PostLogin

Through ADB

adb shell am start -n com.example.demo/com.example.test.MainActivity

DYNAMIC

```
dz> run app.package.attacksurface com.example.root
Attack Surface:
  3 activities exported
  1 broadcast receivers exported
  1 content providers exported
  1 services exported
    is debuggable
dz>
```

BugBase

Platform

App Permissions:

```
$ adb shell dumpsys package com.google.android.youtube
...
declared permissions:
  com.google.android.youtube.permission.C2D_MESSAGE: prot=signature, INSTALLED
requested permissions:
  android.permission.INTERNET
  android.permission.ACCESS_NETWORK_STATE
install permissions:
  com.google.android.c2dm.permission.RECEIVE: granted=true
  android.permission.USE_CREDENTIALS: granted=true
  com.google.android.providers.gsf.permission.READ_GSERVICES: granted=true
...
```

BugBase

Platform

Deep Link Misconfigurations

Deep links are a type of link that send users directly to an app instead of a website or a store. They are used to send users straight to specific in-app locations, saving users the time and energy locating a particular page themselves

STATIC

- Check for deeplink Usage
  - Use the Android "App Link Verification" Tester ↗ script to list all deep links (list-all) or only app links (list-applinks):
- Check for Website Association - adb shell pm get-app-links com.example.package
- Check for Subdomains - can be checked in Manifest.xml
- Check the handler method and URL , hostname validation and whiteliting.

```
// snippet edited for simplicity
public final class WebViewActivity extends AppCompatActivity {
    private ActivityWebViewBinding binding;

    public void onCreate(Bundle savedInstanceState) {
        Uri data = getIntent().getData();
        String html = data == null ? null : data.getQueryParameter("html");
        Uri data2 = getIntent().getData();
        String deeplink_url = data2 == null ? null : data2.getQueryParameter("url");
        View findViewById = findViewById(R.id.webView);
        if (findViewById != null) {
            WebView wv = (WebView) findViewById;
            wv.getSettings().setJavaScriptEnabled(true);
            if (deeplink_url != null) {
                wv.loadUrl(deeplink_url);
            ...
```

BugBase

Platform

DYNAMIC

- Frida hooking
  - the script "Android Deep Link Observer" ↗ from Frida CodeShare to monitor all invoked deep links triggering a call to Intent.getData.
- Invoking Deep Links

```
adb shell am start -W -a android.intent.action.VIEW -d "deeplinkdemo://load.html/?message=ok#

Starting: Intent { act=android.intent.action.VIEW dat=deeplinkdemo://load.html/?message=ok }
Status: ok
LaunchState: WARM
Activity: com.mstg.deeplinkdemo/.WebViewActivity
```

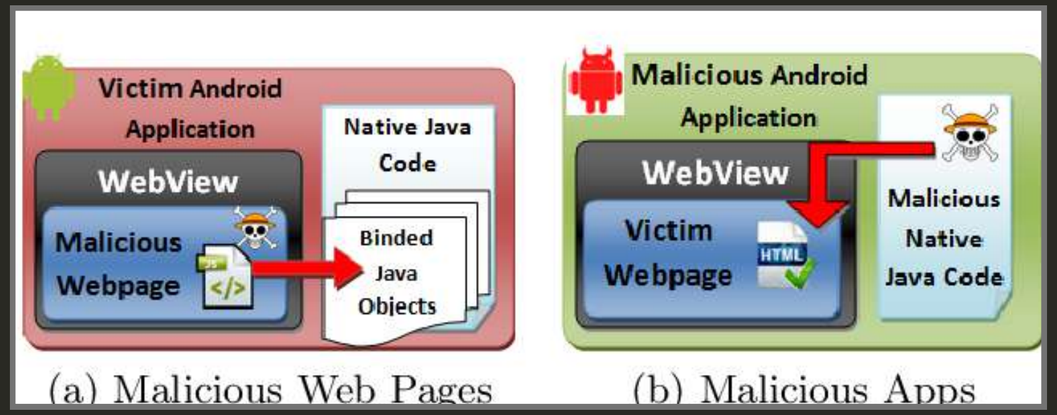
```
[*] Intent.getData() was called
[*] Activity: com.mstg.deeplinkdemo.WebViewActivity
[*] Action: android.intent.action.VIEW

[*] Data
- Scheme: deeplinkdemo://
- Host: /load.html
- Params: message=ok
- Fragment: part1

[*] Stacktrace:

android.content.Intent.getData(Intent.java)
com.mstg.deeplinkdemo.WebViewActivity.onCreate(WebViewActivity.kt)
android.app.Activity.performCreate(Activity.java)
...
com.android.internal.os.ZygoteInit.main(ZygoteInit.java)
```

Platform

Webview Misconfigurations

Disabled Javascript

Javascript is disabled by default in a webview



Disabled Access to Resources:

→ Access to content providers (setAllowContentAccess)

- setAllowContentAccess, default value=true i.e enabled by default

→ Access to local files (setAllowFileAccess)

- setAllowFileAccess, starting Android 11, default value=false, disabled by default

→ Access to local files from external sources

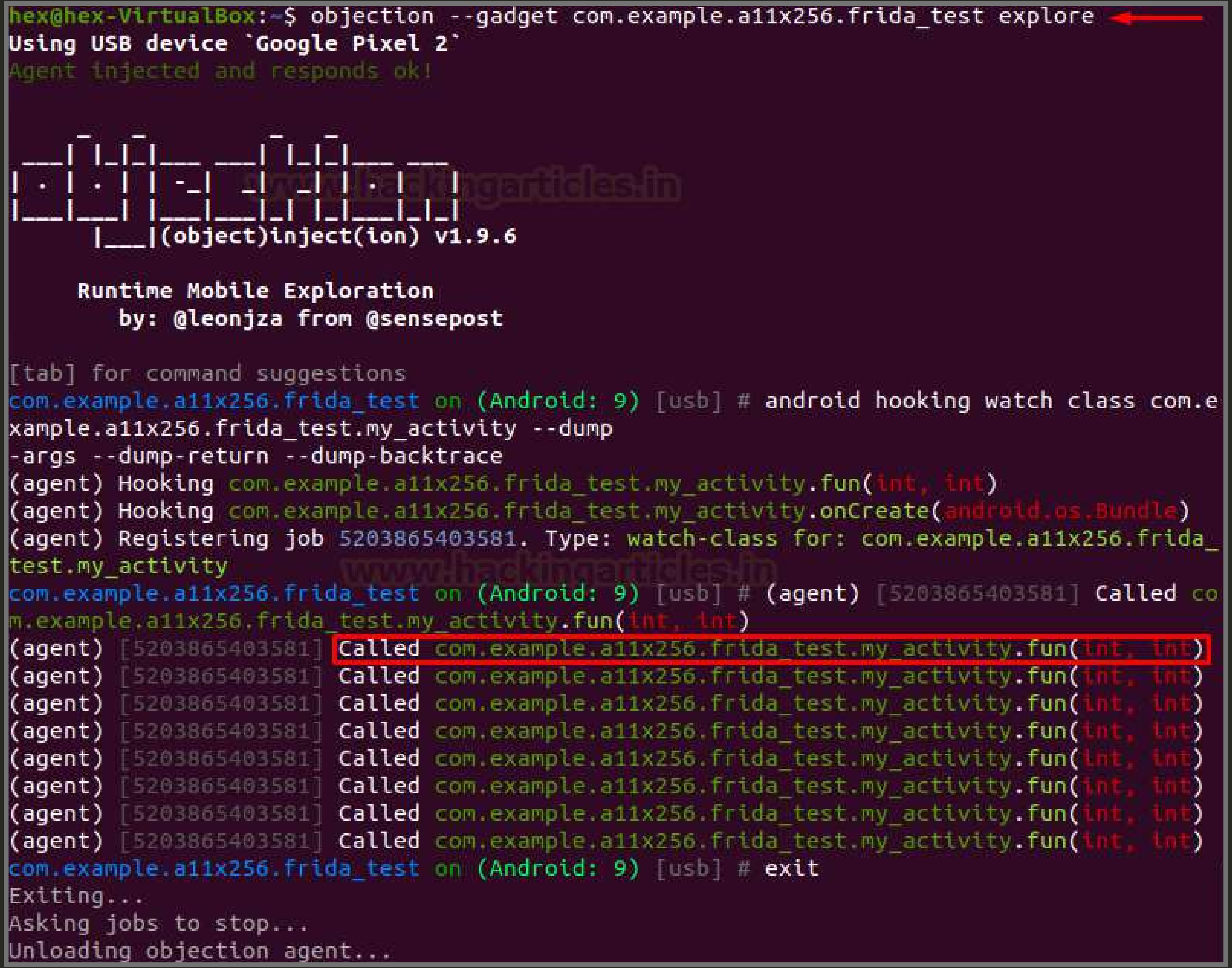
- setAllowFileAccessFromFileURLs, default value=false, disabled by default
- setAllowUniversalAccessFromFileURLs, default value=false, disabled by default

BugBase

Resilience

These include:

- SSL Pinning Bypass
- Root Detection Bypass
- Play Integrity Bypass
- Application Tampering
- Obfuscation



```
hex@hex-VirtualBox:~$ objection --gadget com.example.a11x256.frida_test explore
Using USB device `Google Pixel 2`
Agent injected and responds ok!

     _   _         _   _
 ___| |_|_|___ ___| |_|_|___ ___
| . | . | | -_|  _|  _| | . |   |
|___|___| |___|___|_| |_|___|_|_|
      |___|(object)inject(ion) v1.9.6

     Runtime Mobile Exploration
        by: @leonjza from @sensepost

[tab] for command suggestions
com.example.a11x256.frida_test on (Android: 9) [usb] # android hooking watch class com.e
xample.a11x256.frida_test.my_activity --dump
-args --dump-return --dump-backtrace
(agent) Hooking com.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) Hooking com.example.a11x256.frida_test.my_activity.onCreate(android.os.Bundle)
(agent) Registering job 5203865403581. Type: watch-class for: com.example.a11x256.frida_
test.my_activity
com.example.a11x256.frida_test on (Android: 9) [usb] # (agent) [5203865403581] Called co
m.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) [5203865403581] Called com.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) [5203865403581] Called com.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) [5203865403581] Called com.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) [5203865403581] Called com.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) [5203865403581] Called com.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) [5203865403581] Called com.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) [5203865403581] Called com.example.a11x256.frida_test.my_activity.fun(int, int)
(agent) [5203865403581] Called com.example.a11x256.frida_test.my_activity.fun(int, int)
com.example.a11x256.frida_test on (Android: 9) [usb] # exit
Exiting...
Asking jobs to stop...
Unloading objection agent...
```

Resilience

- Root Detection Bypass: frida -l <location>root-detection-byass.js -U -f <package name>- - no-pause

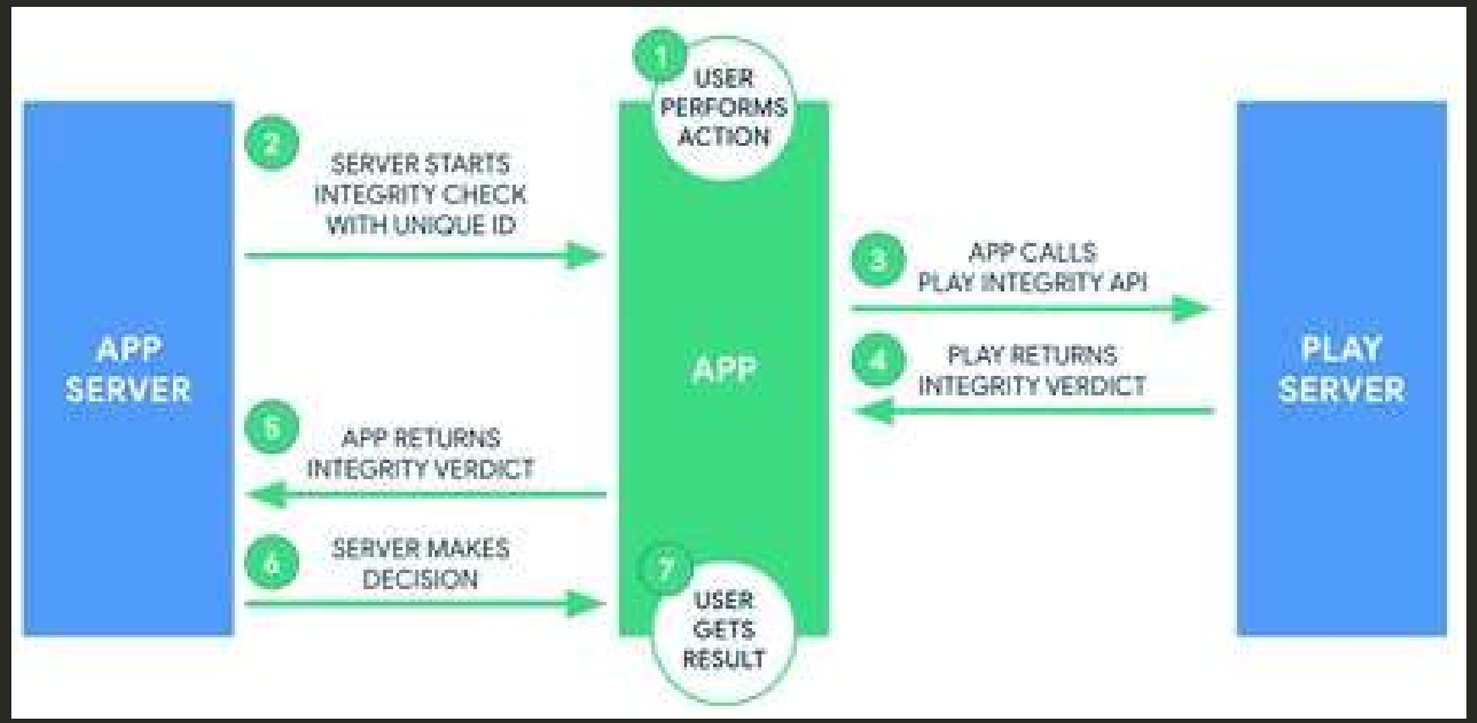
root-detection-bypass.js

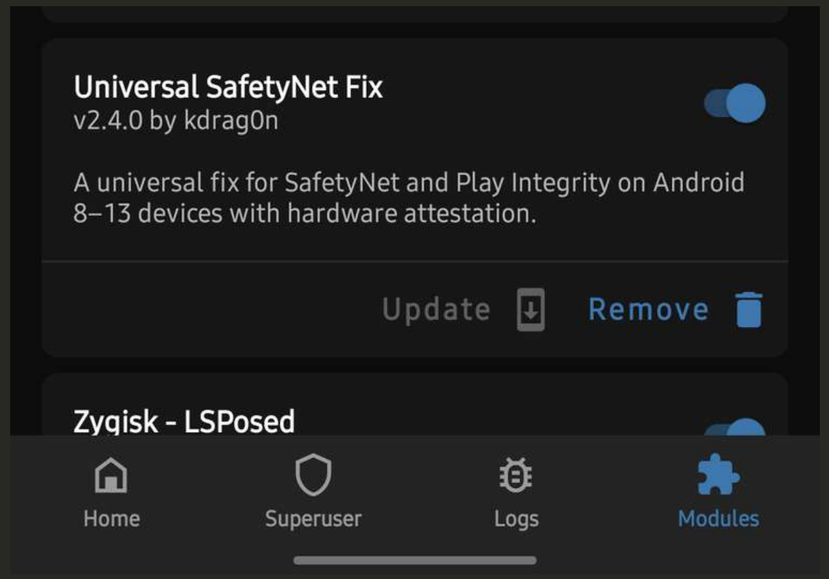
```
Java.perform(function(){
    console.log("\nRoot detection bypass with Frida");
    var DeviceUtils = Java.use("DeviceUtils.isDeviceRooted");
    console.log("\nHijacking isDeviceRooted function in DeviceUtils class");
    DeviceUtils.isDeviceRooted.implementation = function(){
        console.log("\nInside the isDeviceRooted function");
        return false;
    };
    console.log("\nRoot detection bypassed");
});
```

BugBase

Resilience

Play integrity Bypass:

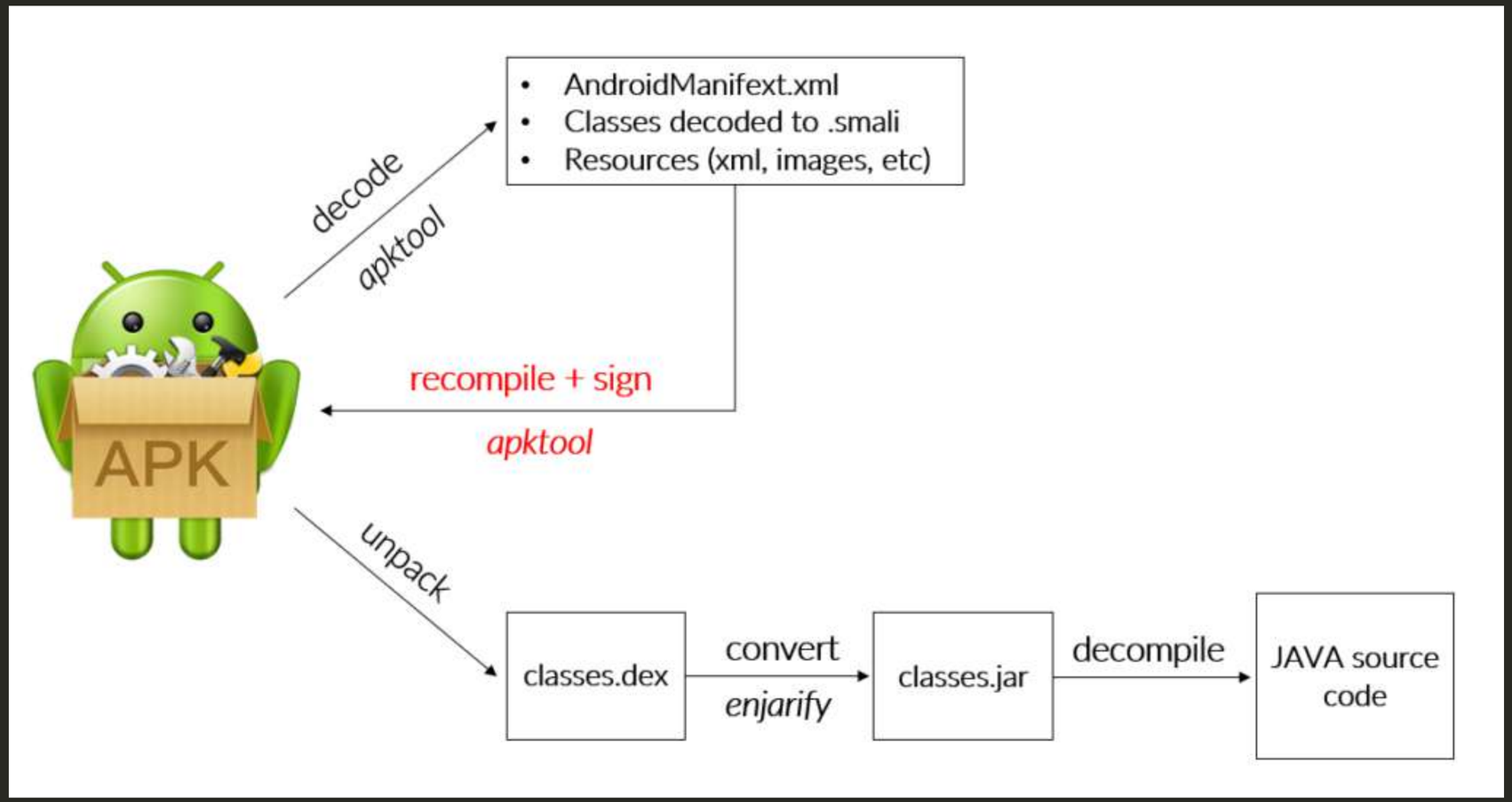




BugBase

Resilience

Application Tampering:



BugBase

ASK ANYTHING?

BugBase