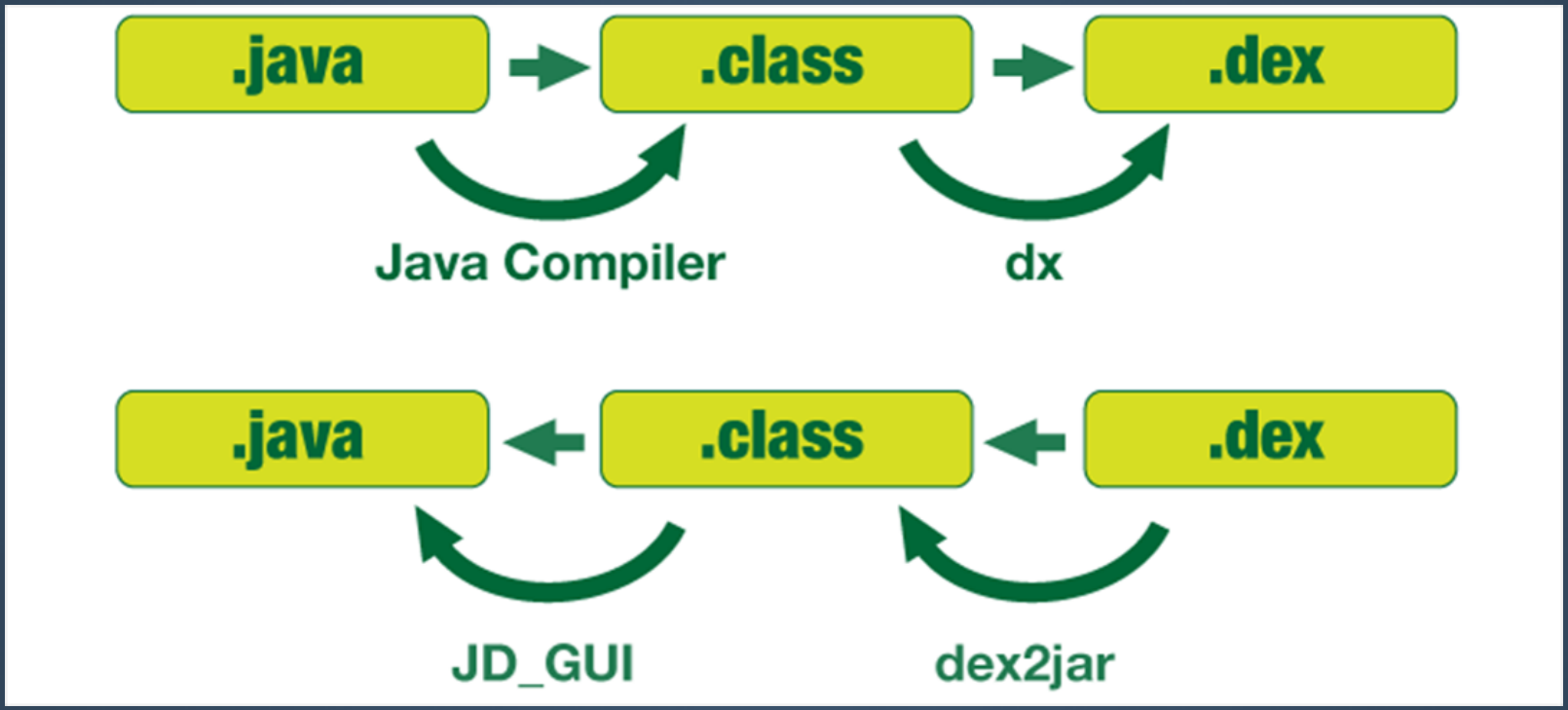
Bulletproof Android

Godfrey Nolan

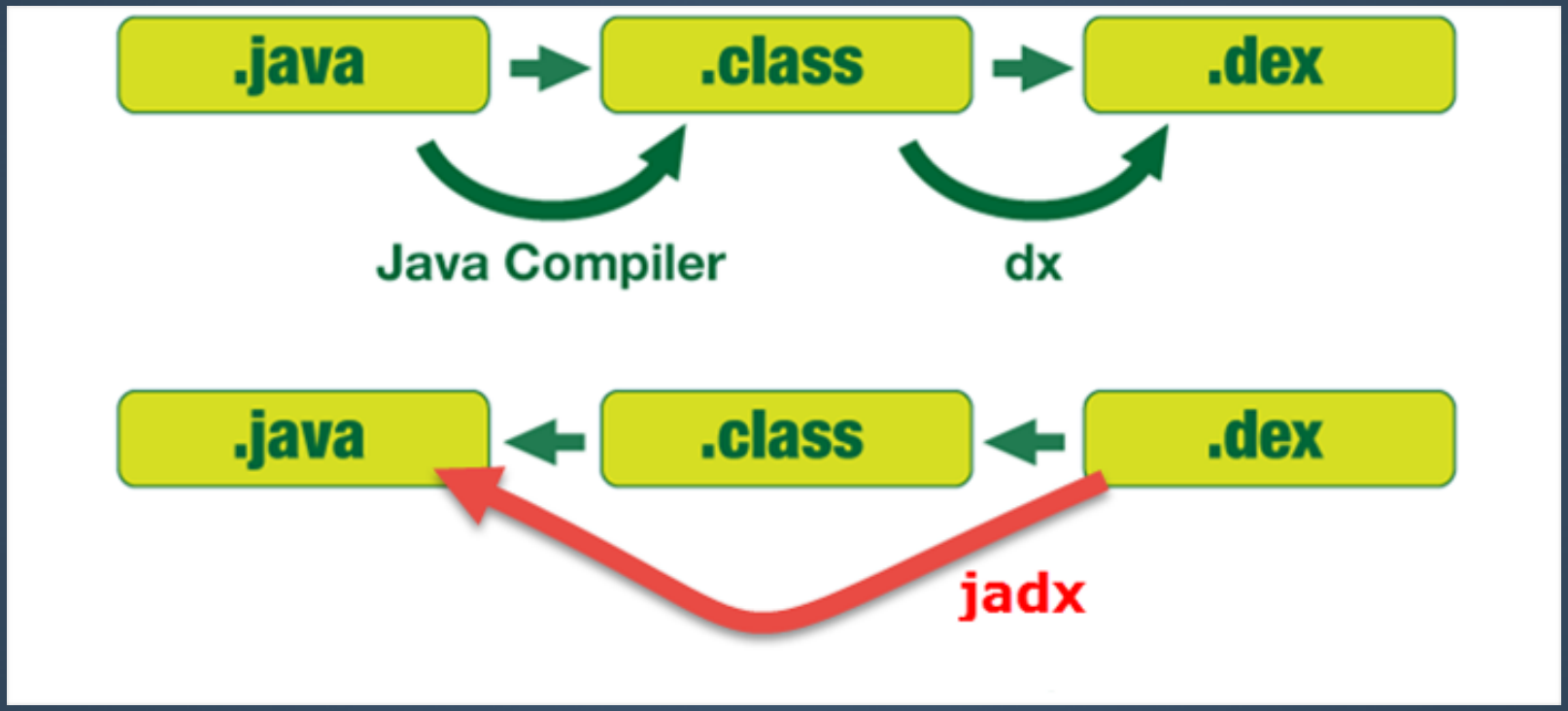
How did we get here

- Virtual Machines
- Static information
- Dynamic information

Decompilation 101



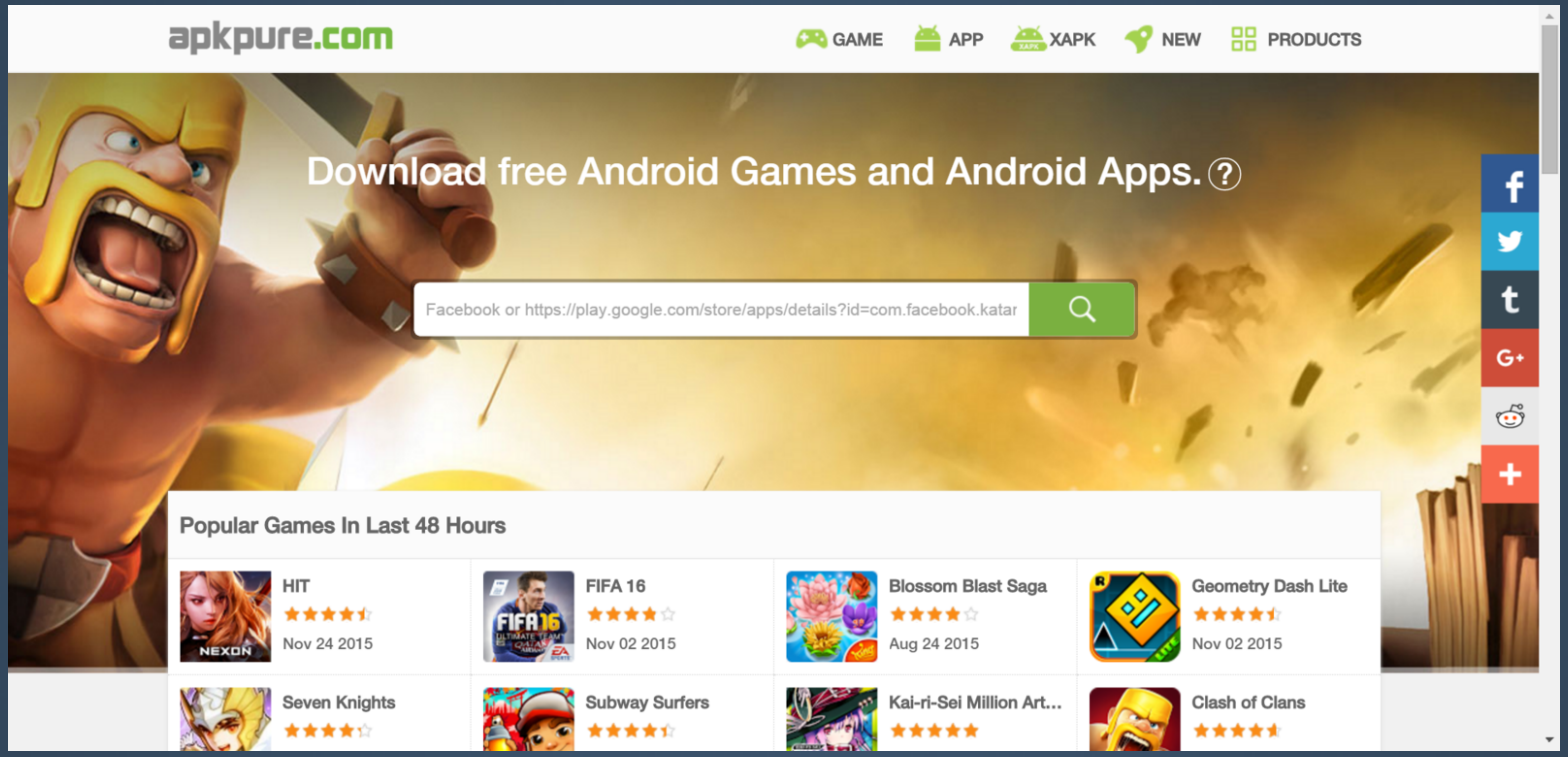
Decompilation 101



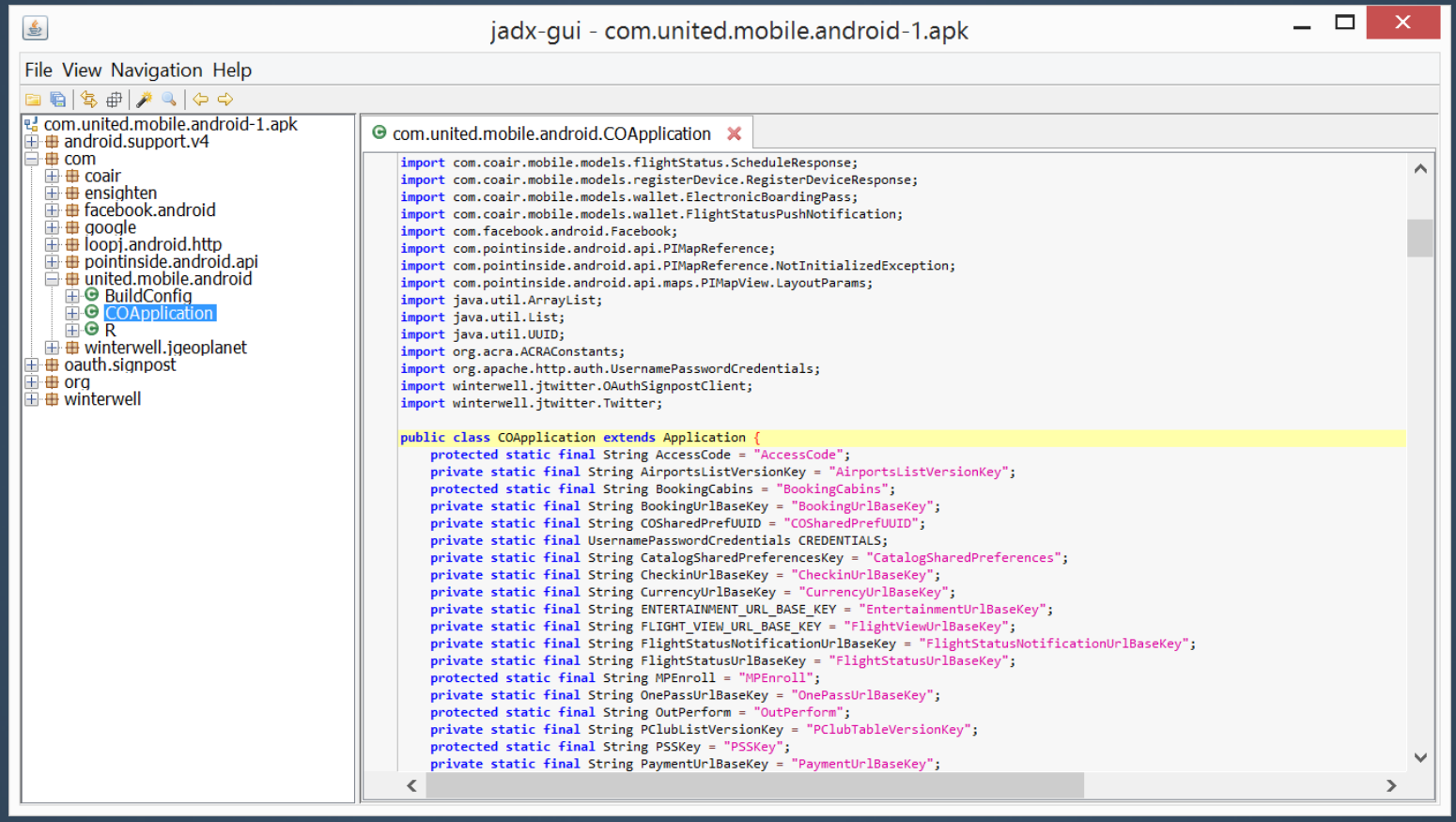
Decompilation 101

```
$ adb shell pm path com.united.mobile.android
package:/data/app/com.united.mobile.android-1/base.apk

$ adb pull /data/app/com.united.mobile.android-1/base.apk
4349 KB/s (51855610 bytes in 11.642s)

$ jadx-gui base.apk

$ adb backup com.united.mobile.android
Now unlock your device and confirm the backup operation.

$ java -jar abe.jar unpack backup.ab backup.tar

$ tar -xvf backup.tar

$ sqlite3 apps/com.united.mobile.android/db/united.db
```

Decompilation 101



Decompilation 101



Audit Reports

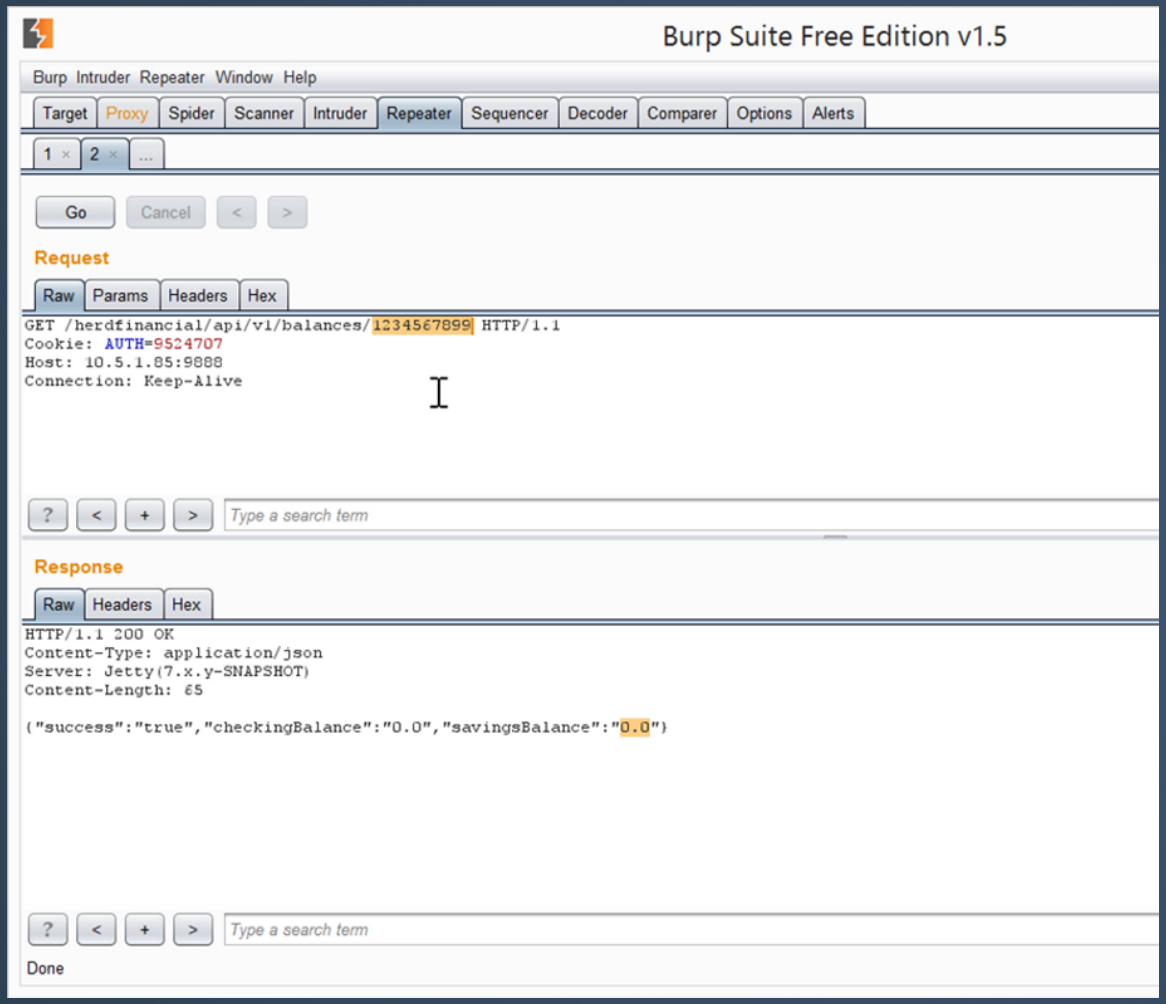
OWASP Top 10

- Weak Server Side Controls
- Insecure Data Storage
- Insufficient Transport Layer Protection
- Unintended Data Leakage
- Poor Authorization and Authentication
- Broken Cryptography
- Client Side Injection
- Security Decision via Untrusted Input
- Improper Session Handling
- Lack of Binary Protections

OWASP Top 10

- Identify Problem
- Show real world example
- Fix it!

Problem



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

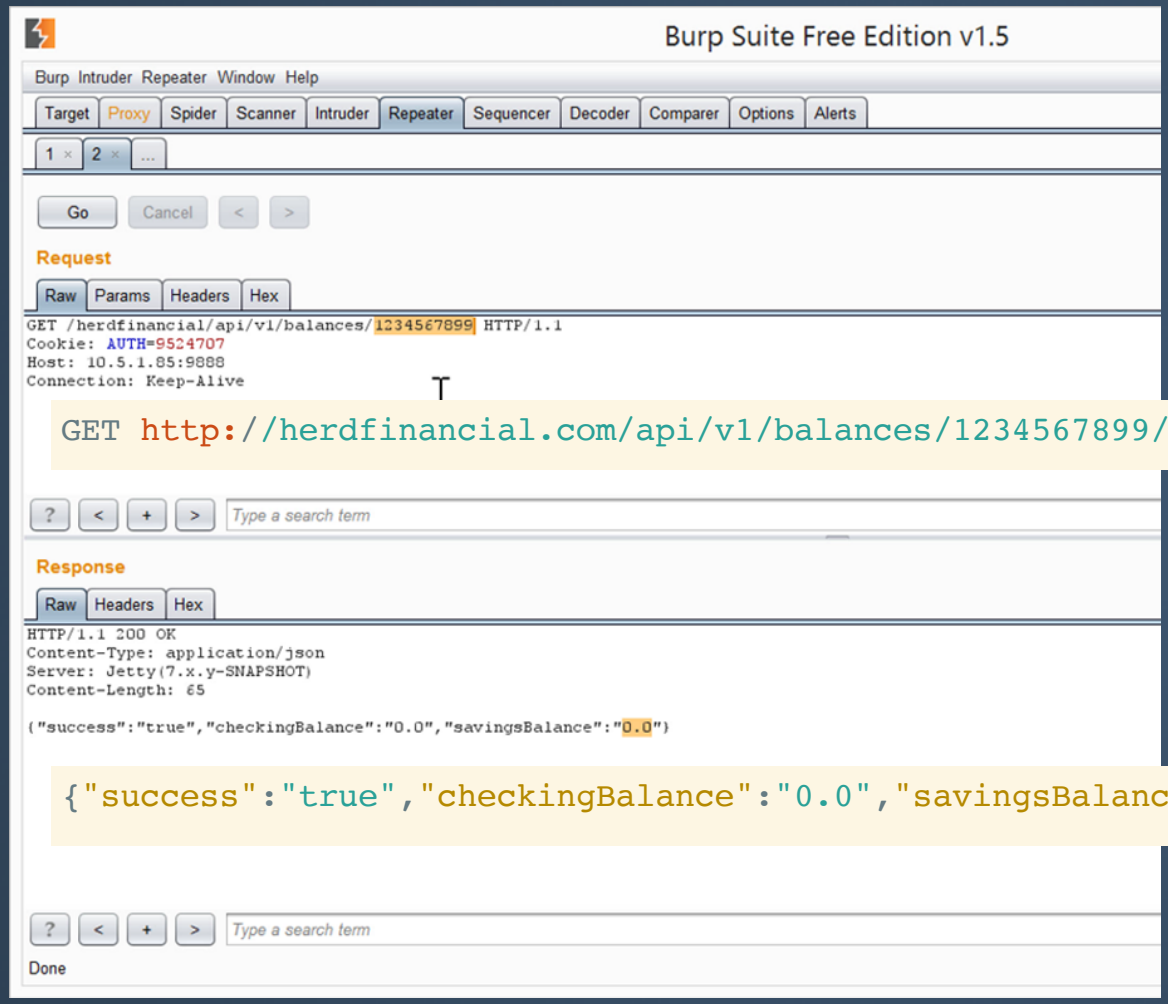
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

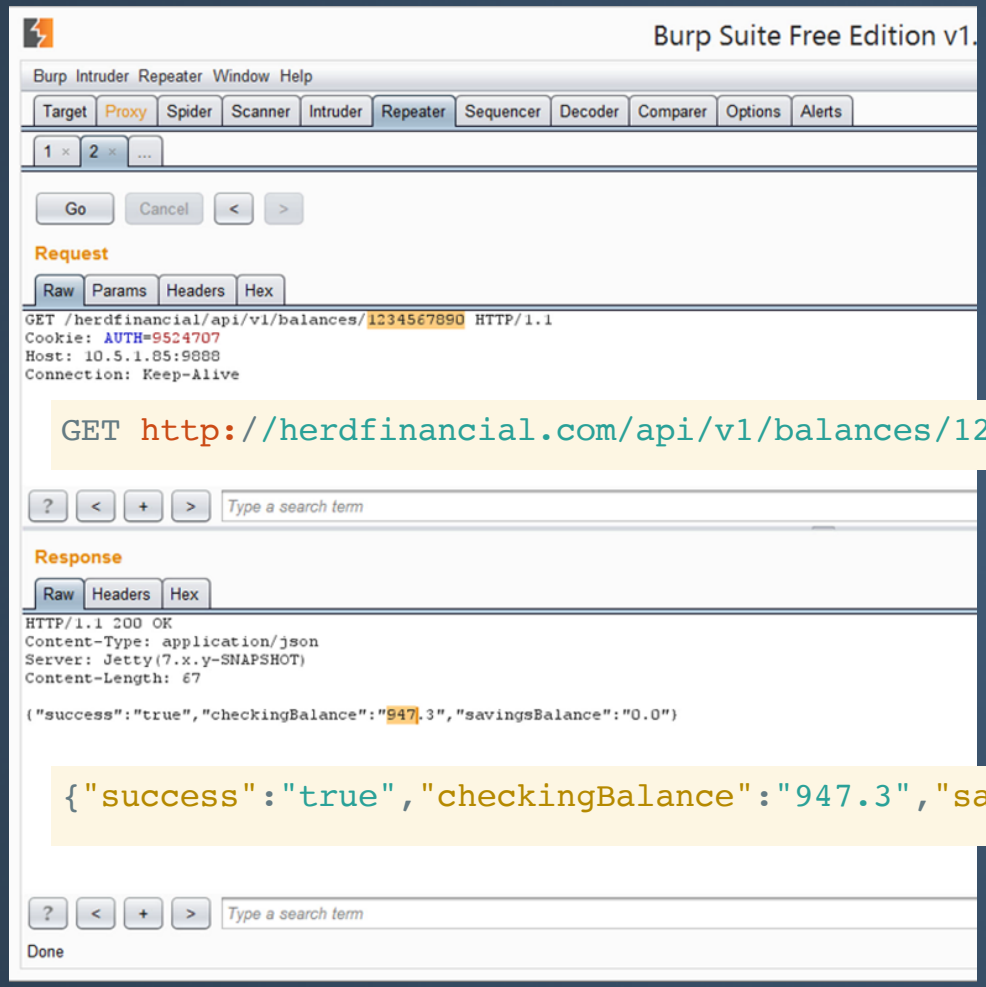
Client Side Injection

Security Decision via Untrusted Input

Session Handling

Lack of Binary Protections

Problem




```
GET http://herdfinancial.com/api/v1/balances/1234567890/
```

```
{"success":"true","checkingBalance":"947.3","savingsBalance":"0.0"}
```

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

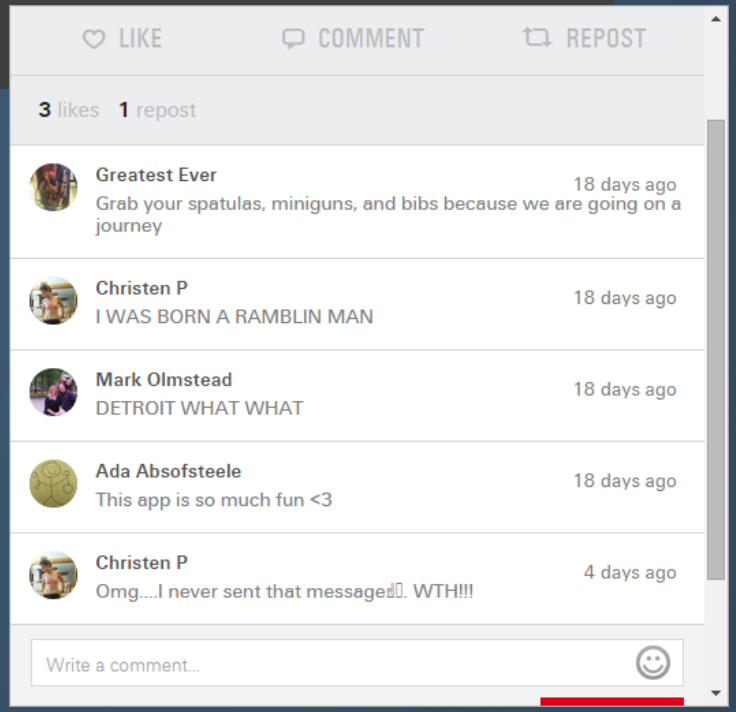
Security Decision via Untrusted Input

ssion Handling

Lack of Binary Protections

Example


```
"actor": {“first_name": "Rita","last_name": "D.","title": "Rita D.","gender": "F",
          "is_mvp": false,
          "preferred_brand": 32,
          "_links": {"self": [{"href": "\/v7.0\/user\/3273986\/","id": "3273986"}]},
          "type": "user",
          "friendship": null,
          "id": 3273986
},"id": "1-3273986-9-1440092847",
```

Fix

- Use GUID that maps to ID
- REST verbs are easy to guess
- OWASP Web/Cloud top 10
- Don’t trust the client, verify

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

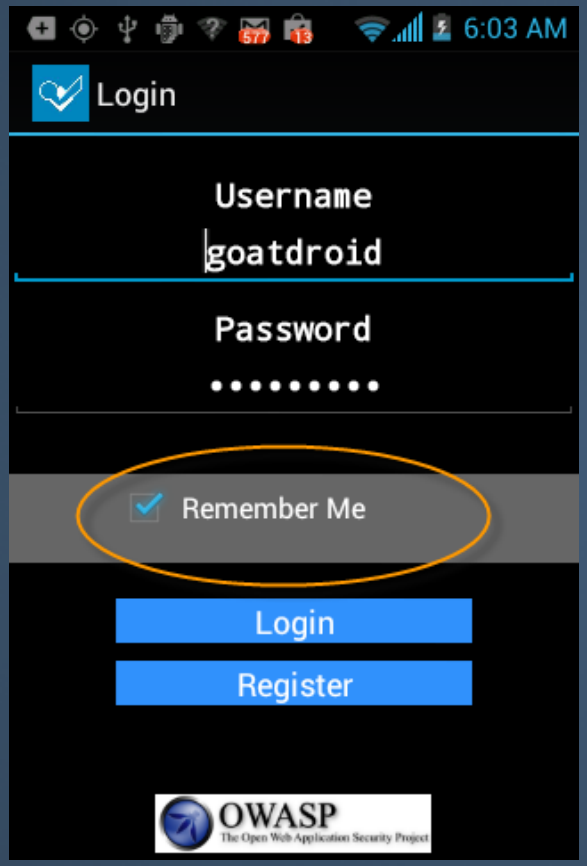
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<boolean name="remember" value="true" />
<string name="password">goatdroid</string>
<string name="username">goatdroid</string>
</map>
```

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

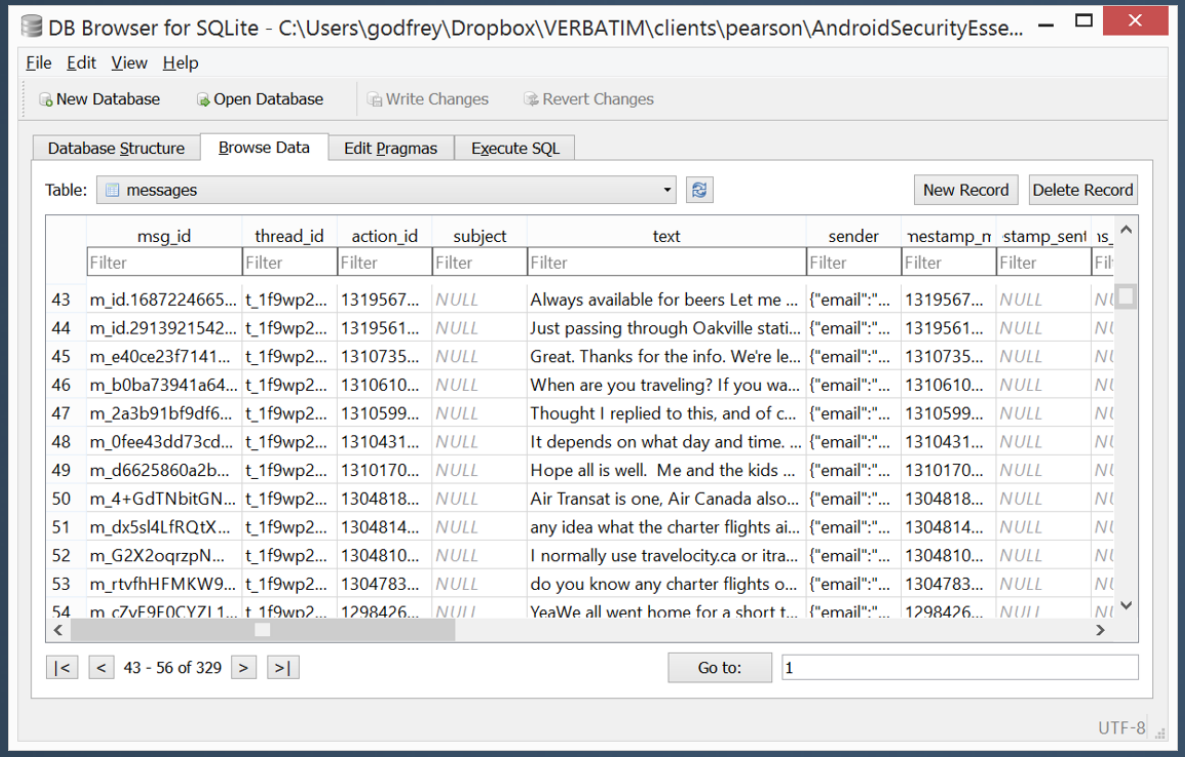
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix

- No caching of passwords, SSNs etc.
- Multi-factor authentication
- Client / Server side access control
- "Sensitive data should be encrypted and very sensitive data should be stored on server" - Zapata

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

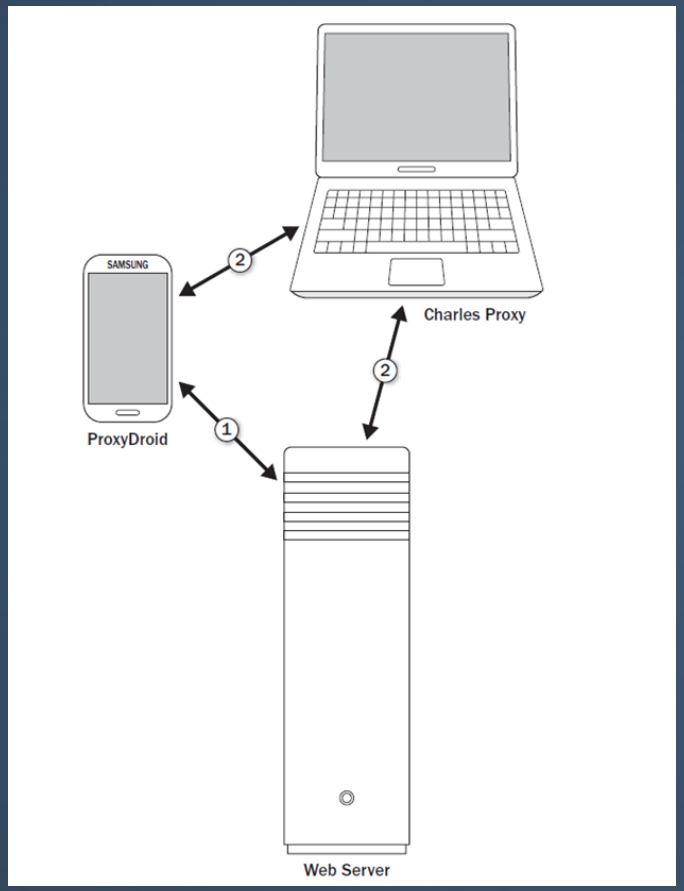
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

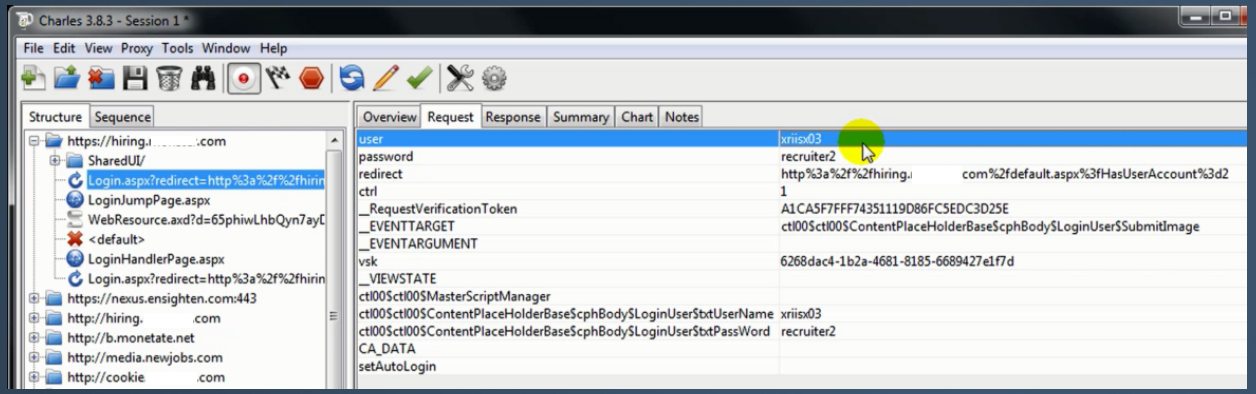
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

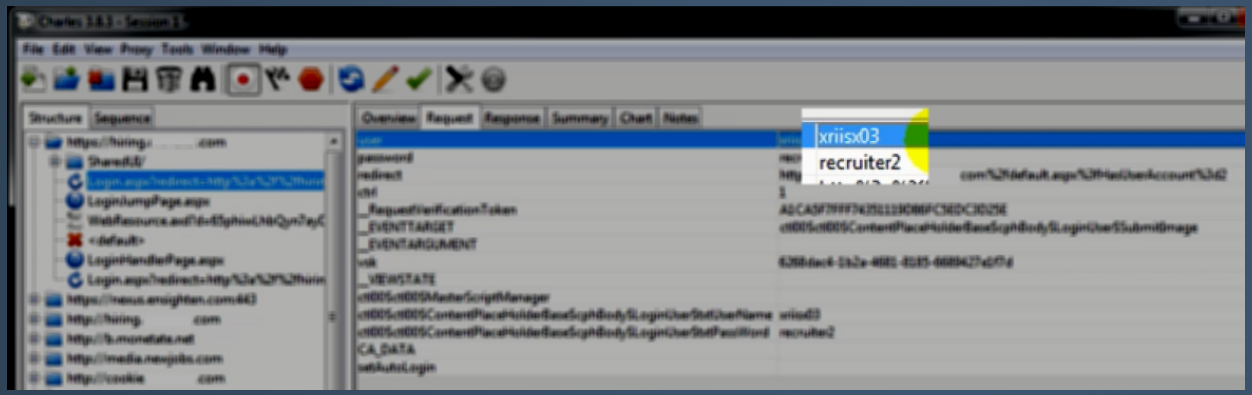
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

More Problems

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix

```
 private static String PUB_KEY = "30820122300d06092a864886f70d0101" +
   "0105000382010f003082010a0282010100b35ea8adaf4cb6db86068a836f3c85" +
   "5a545b1f0cc8afb19e38213bac4d55c3f2f19df6dee82ead67f70a990131b6bc" +
   "ac1a9116acc883862f00593199df19ce027c8eaaae8e3121f7f329219464e657" +
   "2cbf66e8e229eac2992dd795c4f23df0fe72b6ceef457eba0b9029619e0395b8" +
   "609851849dd6214589a2ceba4f7a7dcceb7ab2a6b60c27c69317bd7ab2135f50" +
   "c6317e5dbfb9d1e55936e4109b7b911450c746fe0d5d07165b6b23ada7700b00" +
   "33238c858ad179a82459c4718019c111b4ef7be53e5972e06ca68a112406da38" +
   "cf60d2f4fda4d1cd52f1da9fd6104d91a34455cd7b328b02525320a35253147b" +
   "e0b7a5bc860966dc84f10d723ce7eed5430203010001";

// Pin it!
final boolean expected = PUB_KEY.equalsIgnoreCase(encoded);
if (!expected) {
     throw new CertificateException("checkServerTrusted: Expected public key: "
            + PUB_KEY + ", got public key:" + encoded);
   }
}
```

Fix

- Error out on SSLHandshakeException
- Assume SSL is broken, root level CA's
- SSL pinning but use SafetyNet API
- Do more on the server
- Scan server with nogotofail

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

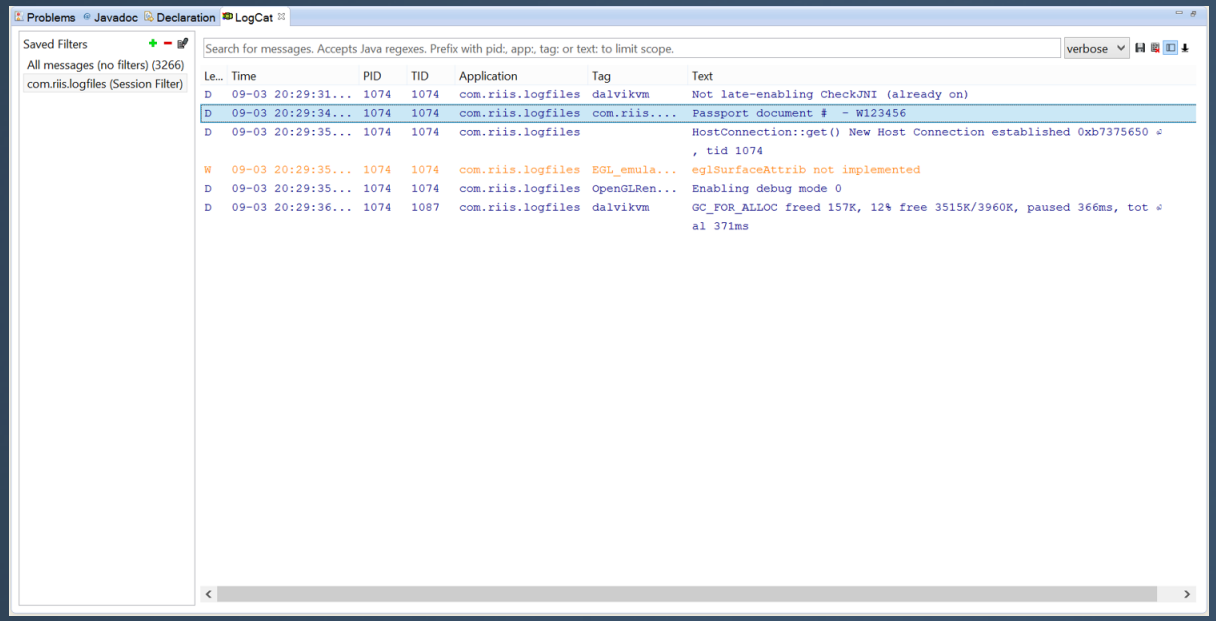
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

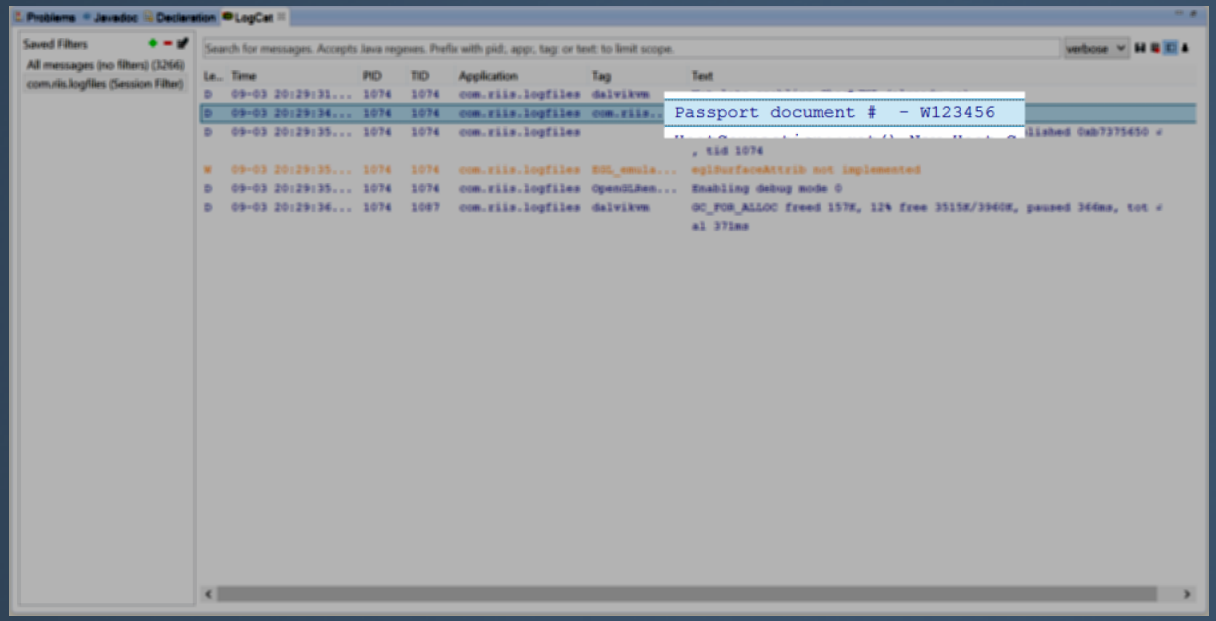
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

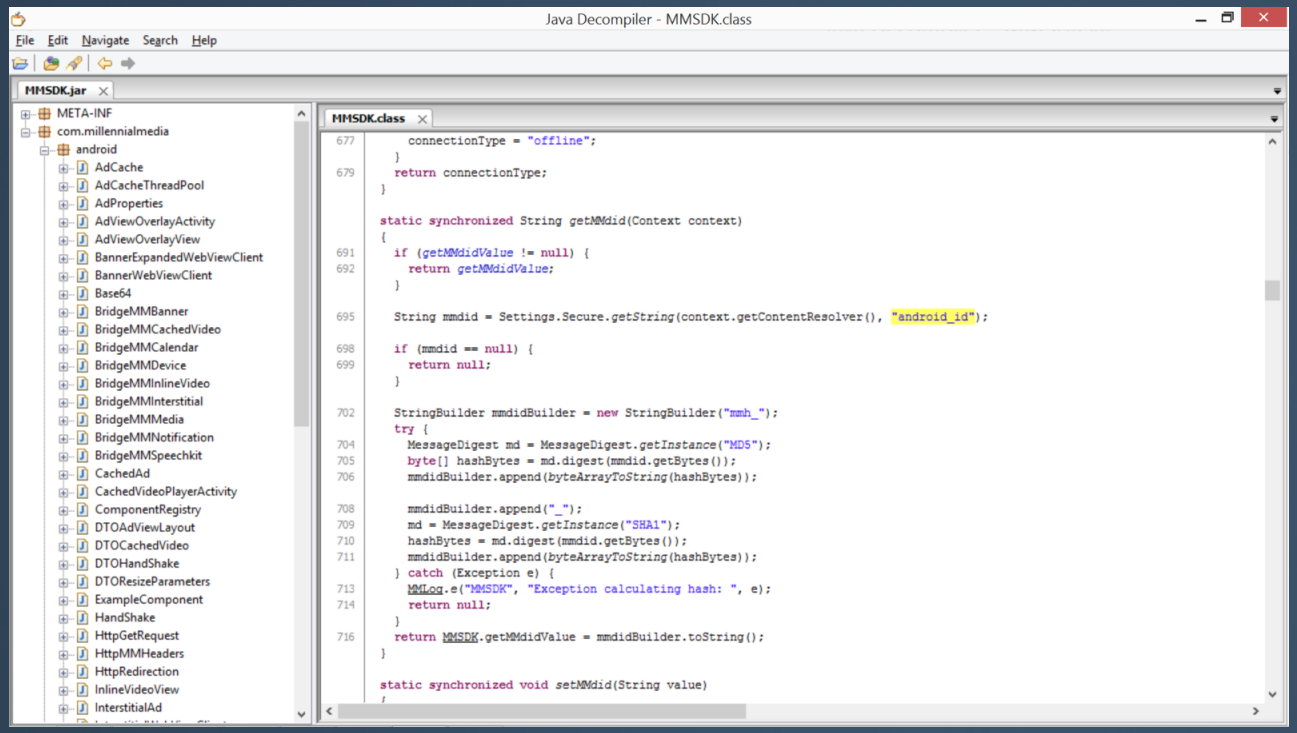
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example

```
public ActivityLaunchAppLoad() {
    this.WAY_WAY_TOO_LOW = 49;
    this.A_LITTLE_LESS_WAY_TOO_LOW = 50;
    this.LESSER_WAY_TOO_LOW = 51;
    this.BIT_TOO_LOW = 52;
    this.TOO_LOW = 53;
    this.MORE = 54;
    this.A_LITTLE_MORE = 55;
    this.WAY_TOO_MORE = 97;
    this.BIG_DADDY = 102;
    this.orderOfTheThronesTrois = new int[]{this.BIG_DADDY, this.MORE, this.WAY_TOO_MORE, this.MORE};
    this.orderOfTheThronesQuatre = new int[]{this.LESSER_WAY_TOO_LOW, this.MORE, this.LESSER_WAY_TOO_LOW, this.TOO_LOW};
    this.orderOfTheThronesUn = new int[]{this.BIT_TOO_LOW, this.BIT_TOO_LOW, this.WAY_WAY_TOO_LOW, this.BIT_TOO_LOW};
    this.orderOfTheThronesDeux = new int[]{this.MORE, this.A_LITTLE_MORE, this.A_LITTLE_LESS_WAY_TOO_LOW, this.BIT_TOO_LOW};
}

String createTheHalfBloodPrince() {
    String strTemp = StringUtils.EMPTY;
    int x = 0;
    while (x < 4) {
        int[] xyz = null;
        if (x == 0) {
            xyz = this.orderOfTheThronesTrois;
        } else if (x == 1) {
            xyz = this.orderOfTheThronesQuatre;
        } else if (x == 2) {
            xyz = this.orderOfTheThronesUn;
        } else if (x == 3) {
            xyz = this.orderOfTheThronesDeux;
        }
        int y = 3;
        while (y >= 0) {
            strTemp = new StringBuilder(String.valueOf(strTemp)).append(Character.toString((char) xyz[y])).toString();
            y--;
        }
        x++;
    }
    return strTemp;
}
```

Fix

- Strip out unneccesary logging code
- Obfuscate method names
- Check any third party libraries
- Double check your webview caches
- Download and unzip your APK

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

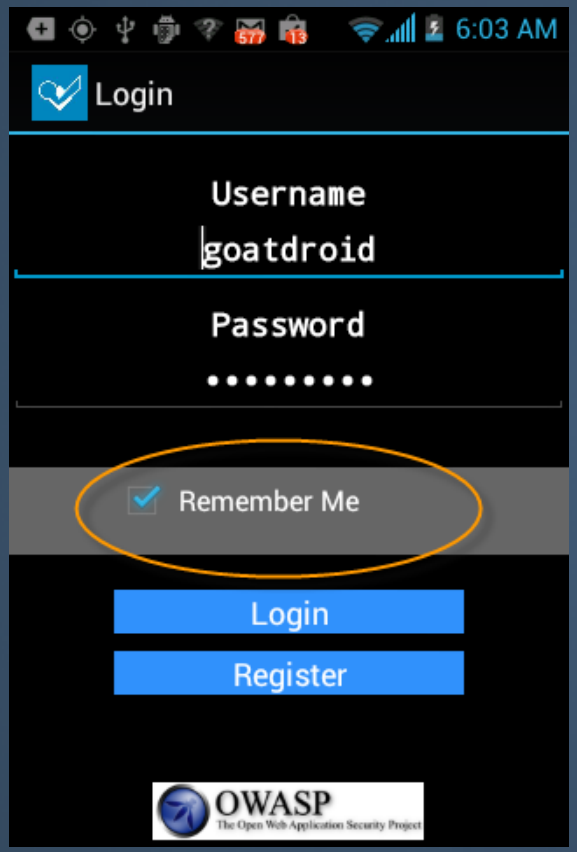
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="TM_MEMBER_EMAIL">godfrey@riis.com</string>
<int name="TM_MEMBER_MARKET_ID" value="7" />
<string name="TM_MEMBER_TAP_ID">77ef62159ad9c32913dfdbee0e58aea3</string>
<string name="TM_MEMBER_LNAME"></string>
<string name="TM_MEMBER_LANGUAGE">en-us</string>
<int name="TM_BILLING_COUNTRY_CODE" value="-1" />
<string name="TM_MEMBER_POSTCODE">48070</string>
<string name="TM_LAST_BILLING_ID"></string>
<int name="TM_MEMBER_COUNTRY" value="840" />
<string name="TM_MEMBER_PASSWORD">2secret4me</string>
<string name="TM_MEMBER_FNAME">Godfrey</string>
</map>
```

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix

- No password caching
- Multi Factor Authentication
- Encryption
  - Public-Private Key exchange
- Tokens, tokens, tokens
  - OAuth
  - Use Server side nonce’s

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

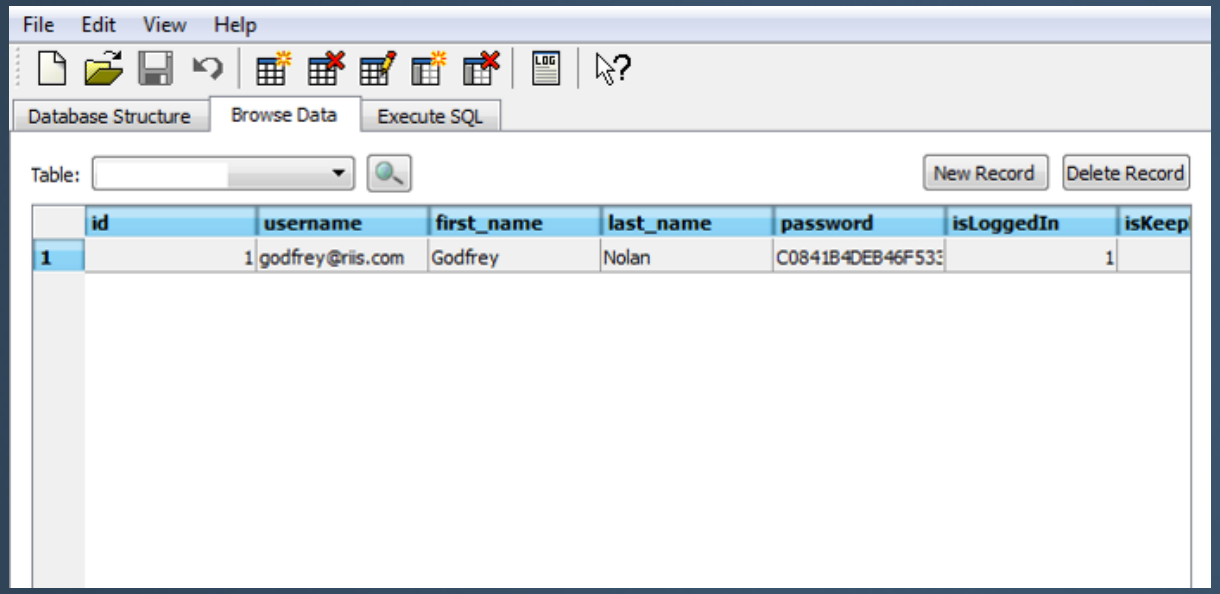
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

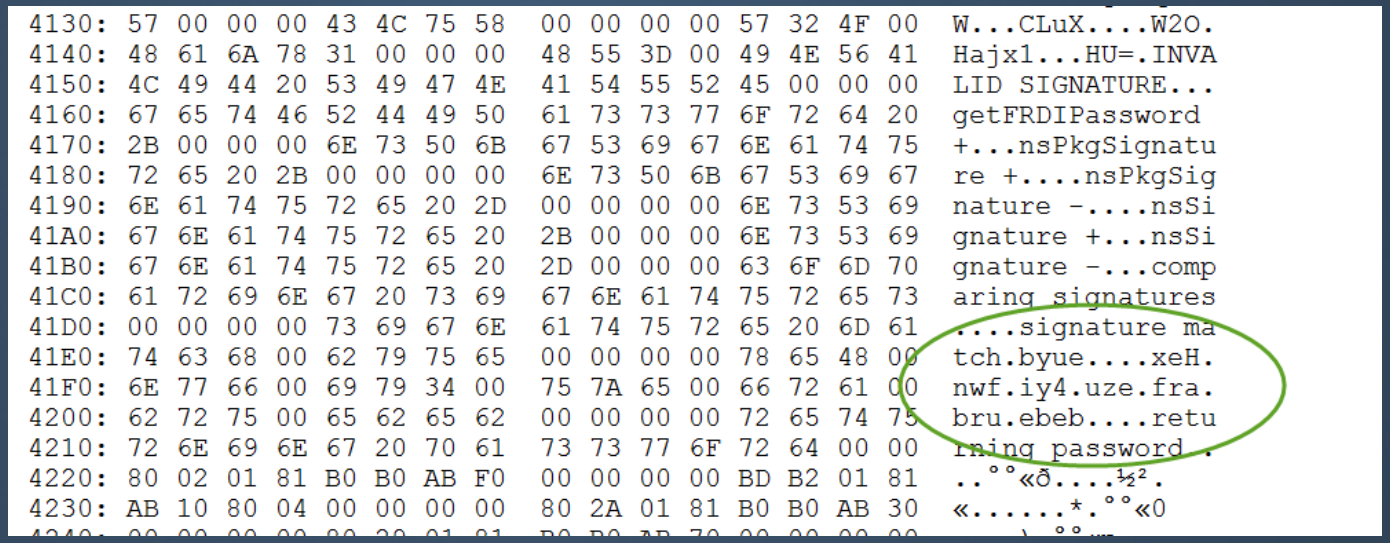
Lack of Binary Protections

Example

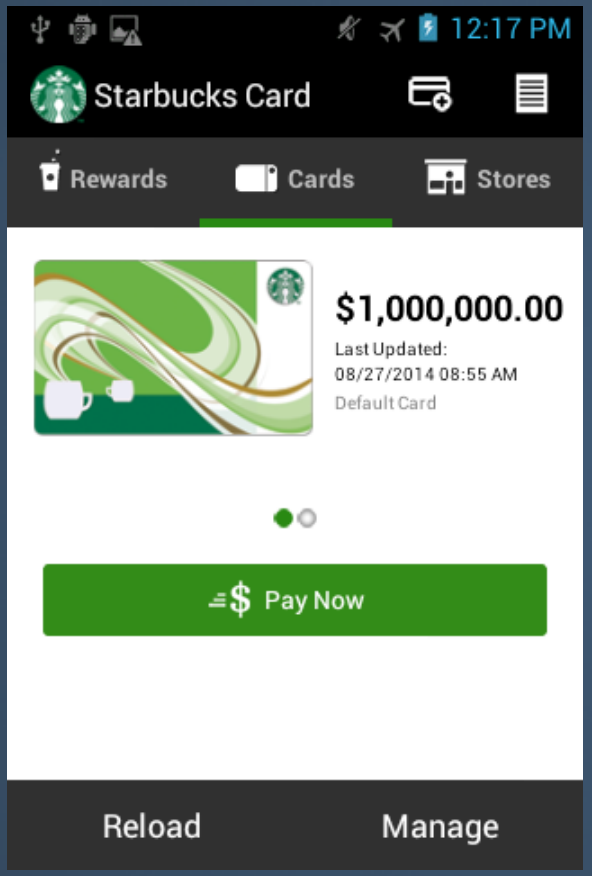
```
public static String decrypt(String paramString)
  throws Exception
{
  if (paramString != null)
    return new String(decrypt(getRawKey("3lIoM_d0idrn4|4TleD".getBytes()), toByte(paramString)));
  return null;
}

private static byte[] decrypt(byte[] paramArrayOfByte1, byte[] paramArrayOfByte2)
  throws Exception
{
  SecretKeySpec localSecretKeySpec = new SecretKeySpec(paramArrayOfByte1, "AES");
  Cipher localCipher = Cipher.getInstance("AES");
  localCipher.init(2, localSecretKeySpec);
  return localCipher.doFinal(paramArrayOfByte2);
}
```

Example

```
// NDK code - still see the code in disassembler

jstring Java_com_riis_decompilingandroid_getPassword(JNIEnv* env, jobject thiz)
{
    return  (*env)->NewStringUTF(env, "xeHnwfiy4uzefrabruebeb");
}
```



Example



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

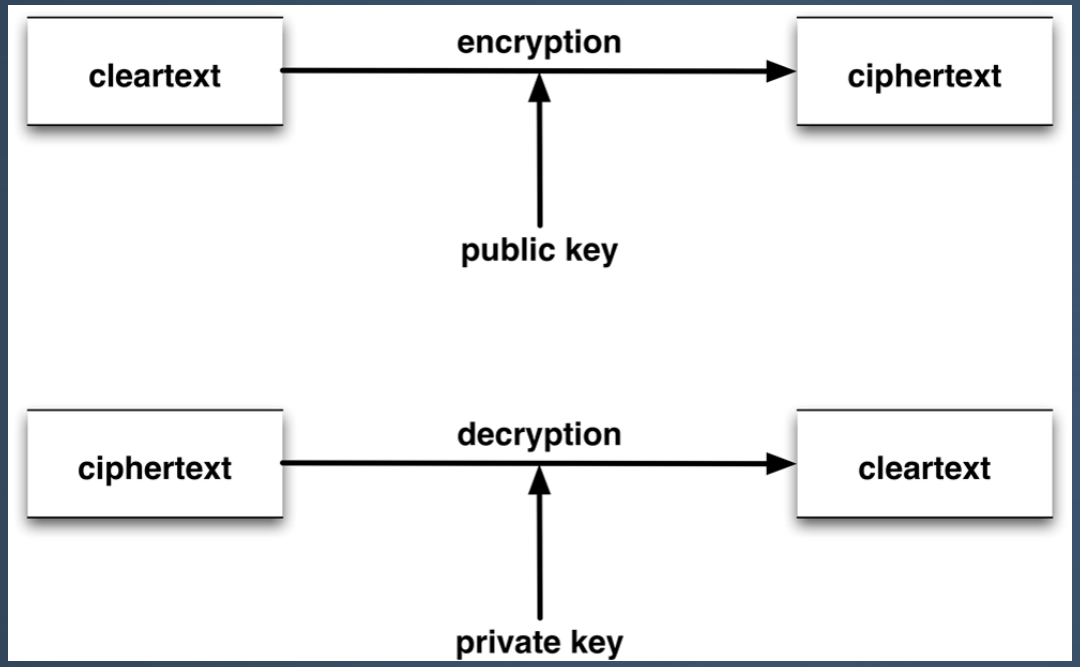
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

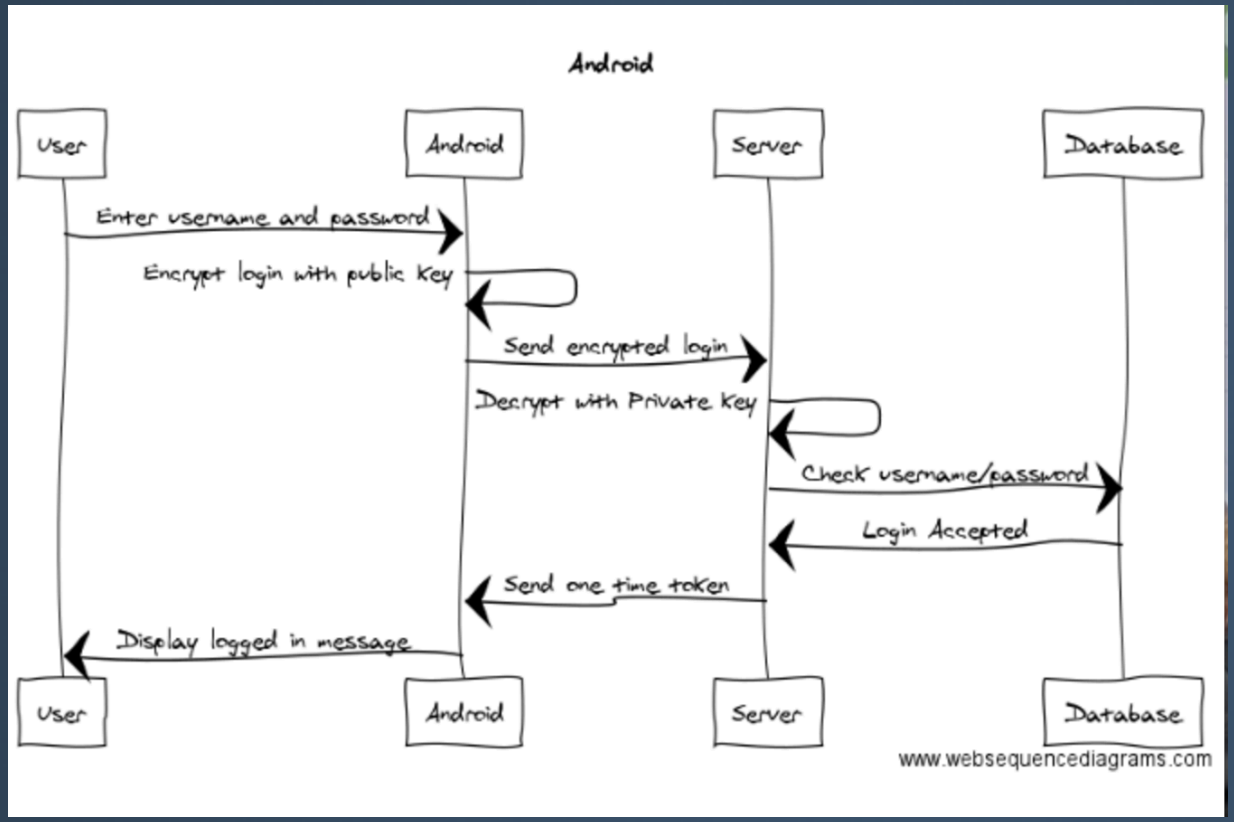
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix

- Use asymmetric encryption
- Encrypt databases
- Android Keystore is broken

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem

```
public boolean checkLogin(String param1, String param2)
{
    boolean bool = false;

    Cursor cursor = db.rawQuery("select * from login where USERNAME = '" +
                        param1 + "' and PASSWORD = '" + param2 + "';", null);

    if (cursor != null) {
        if (cursor.moveToFirst())
            bool = true;
            cursor.close();
    }
    return bool;
}
```

```
select * from login where USERNAME = '' OR 1=1 --' and PASSWORD = 'test'
```

Fix

```
public boolean checkLogin(String param1, String param2)
{
    boolean bool = false;

    Cursor cursor = db.rawQuery("select * from login where " +
                    "USERNAME = ? and PASSWORD = ?", new String[]{param1, param2});

    if (cursor != null) {
        if (cursor.moveToFirst())
            bool = true;
            cursor.close();
    }
    return bool;
}
```

Problem

```
WebView myWebView = (WebView) findViewById(R.id.webview);
WebSettings webSettings = myWebView.getSettings();
webSettings.setJavaScriptEnabled(true);
```

```
<script>alert("xss");</script>
```

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix

- Use parameterized queries
- setJavaScriptEnabled(false)

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

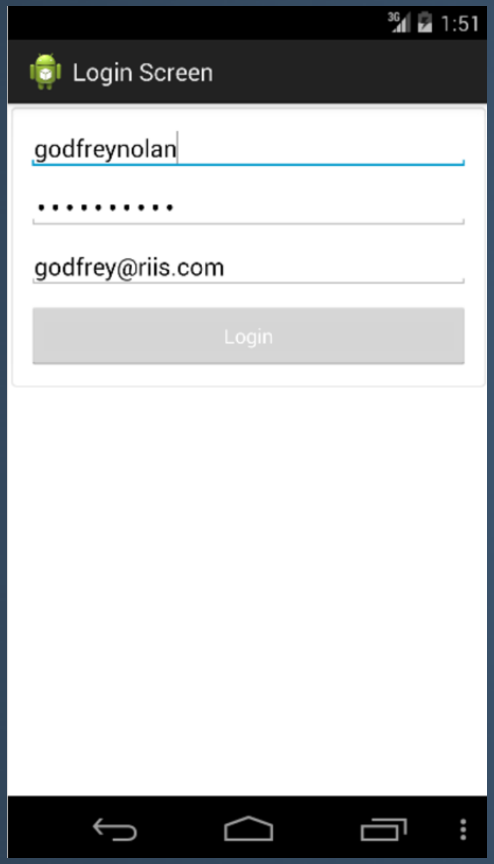
Client Side Injection

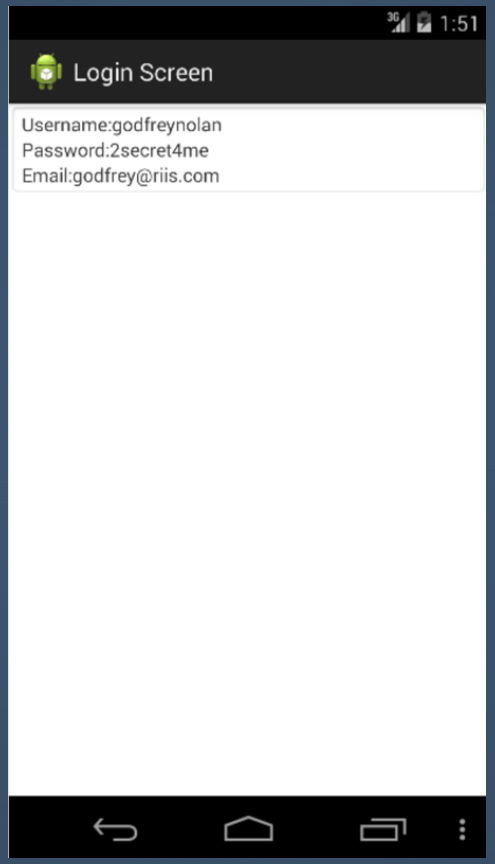
Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem





Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

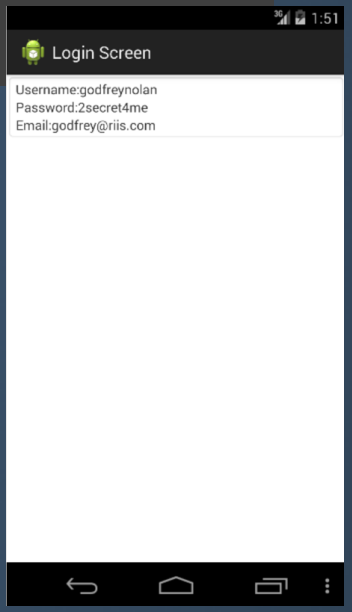
Lack of Binary Protections

Problem

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.riis.login"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.riis.login.LoginActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.riis.login.IntentReceiverActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="com.riis.login.IntentReceiverActivity" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Problem

```
<activity
     android:name="com.riis.login.IntentReceiverActivity"
     android:label="@string/app_name" >
     <intent-filter>
         <action android:name="com.riis.login.IntentReceiverActivity" />
         <category android:name="android.intent.category.DEFAULT" />
     </intent-filter>
 </activity>
```

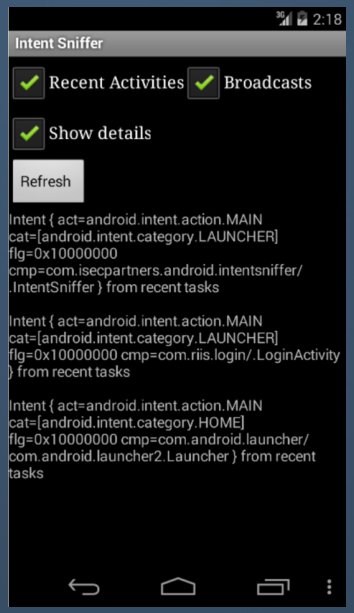


Problem

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.riis.hellointent"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.riis.hellointent.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action android:name="com.riis.login.IntentReceiverActivity" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Fix

```
// implicit
Intent intent = new Intent();

// explicit
Intent intent = new Intent(this, IntentReceiverActivity.class);
```



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix

- Use explicit intents
- Scan using Intent Sniffer / Drozer

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem

```
if (dao.isDevicePermanentlyAuthorized(deviceID)) {

        String newAuthToken = Utils.generateAutToken();
        doa.updateAuthrizedDeviceAuth(deviceID, newAuthToken);
        login.setAuthToken(newAuthToken);
        login.setUserName(dao.getUserName(newAuthToken));
        login.setAccountNumber(dao.getAccountNumber(newAuthToken));
        login.setSuccess(true);

}
```

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

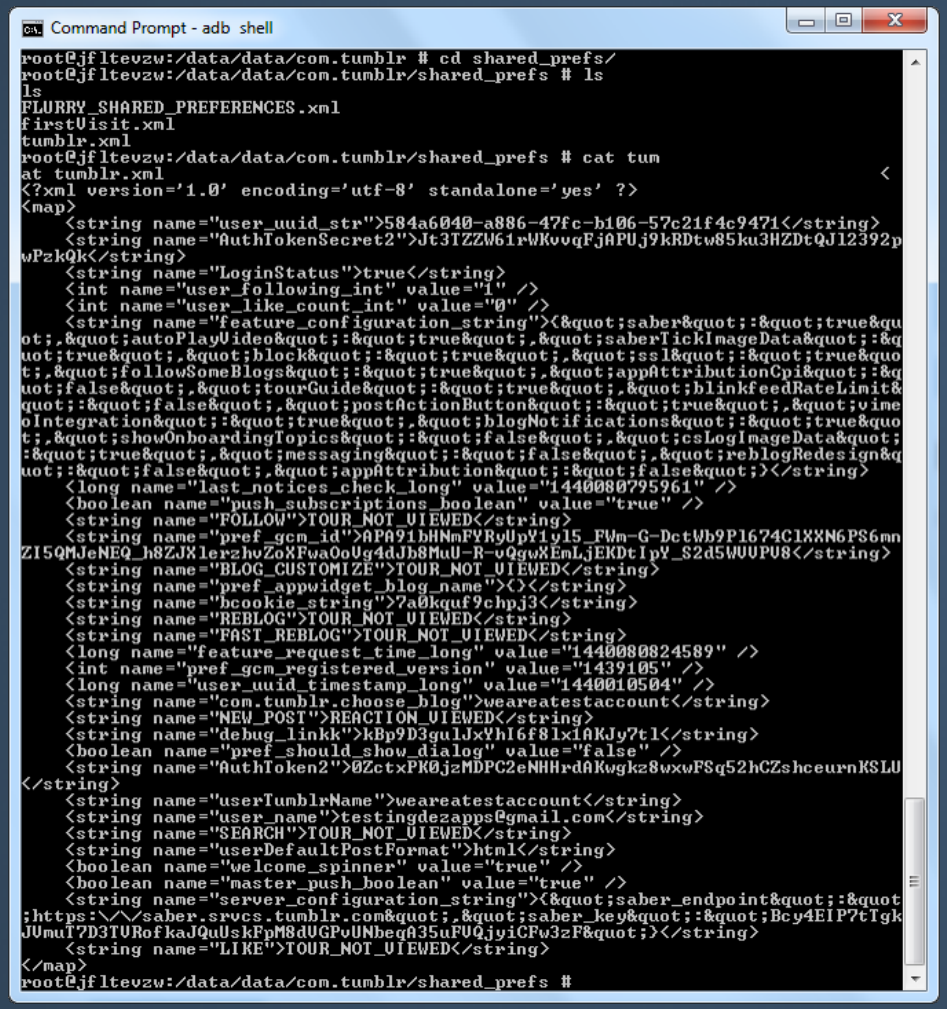
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

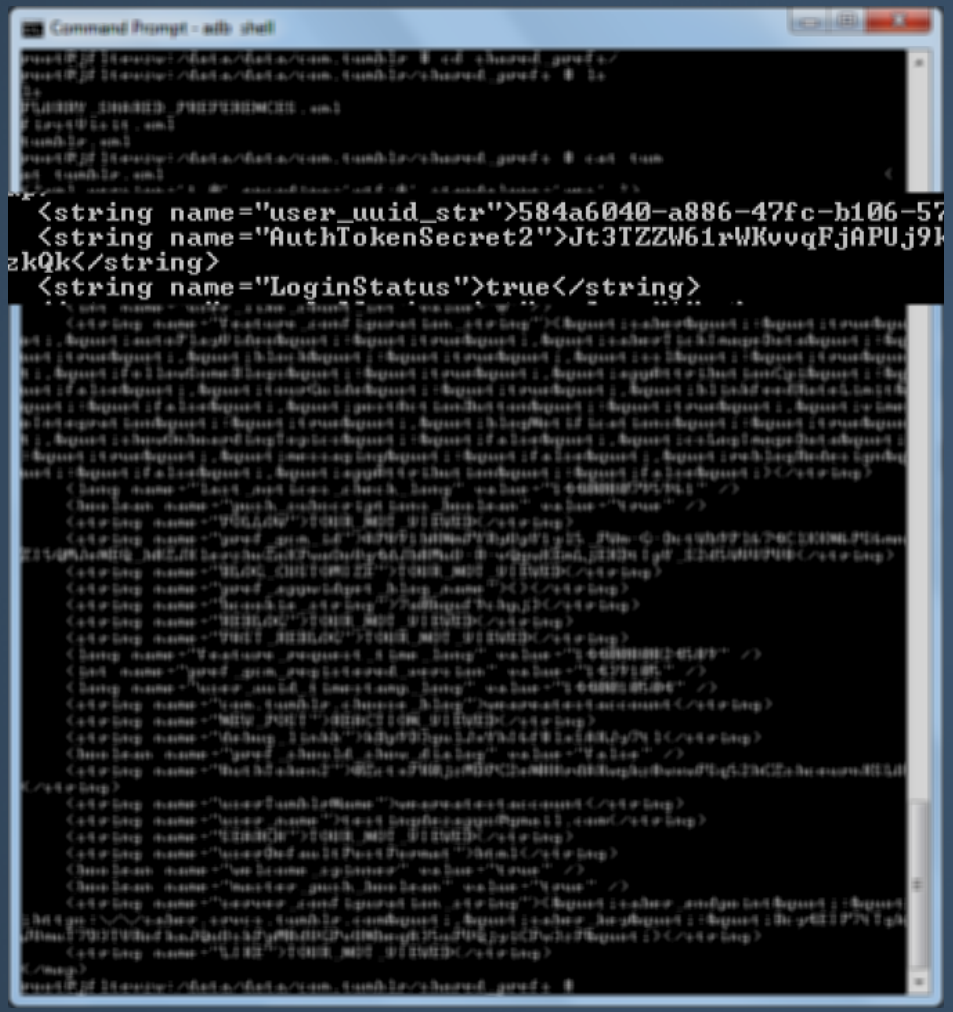
Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Fix

- Expire sessions
- Try backup to another phone
- Careful using OAuth logins to FB etc.

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Problem

```
/**
* Logs you into your SIP provider, registering this device as the location to
* send SIP calls to for your SIP address.
*/
public void initializeLocalProfile() {
   if (manager == null) {
        return;
   }

   if (me != null) {
        closeLocalProfile();
   }

   SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(getBaseContext());
   String username = prefs.getString("namePref", "");
   String domain = prefs.getString("domainPref", "");
   String password = prefs.getString("passPref", "");
```

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections

Example

```
public void initializeLocalProfile()
{
  if (this.manager == null)
    return;
  if (this.me != null)
    closeLocalProfile();
  SharedPreferences localSharedPreferences = PreferenceManager.getDefaultSharedPrefe
  String str1 = localSharedPreferences.getString("namePref", "");
  String str2 = localSharedPreferences.getString("domainPref", "");
  String str3 = localSharedPreferences.getString("passPref", "");
  if ((str1.length() == 0) || (str2.length() == 0) || (str3.length() == 0))
  {
    showDialog(3);
    return;
  }
```

```
public void b()
{
  if (this.b == null)
    return;
  if (this.c != null)
    c();
  SharedPreferences localSharedPreferences = PreferenceManager.getDefaultSharedPrefers
  String str1 = localSharedPreferences.getString("namePref", "");
  String str2 = localSharedPreferences.getString("domainPref", "");
  String str3 = localSharedPreferences.getString("passPref", "");
  if ((str1.length() == 0) || (str2.length() == 0) || (str3.length() == 0))
  {
    showDialog(3);
    return;
  }
  try
```

```
public class WalkieTalkieActivity extends Activity
  implements View.OnTouchListener
{
  public String ´ = null;
  public SipManager ` = null;
  public SipProfile ˛ = null;
  public SipAudioCall ¸ = null;
  private if ˚;

  // ERROR //
  private void ´()
  {
    // Byte code:
    //   0: aload_0
    //   1: getfield 24 com/example/android/sip/WalkieTalkieActivity:` Landroid/net/sip/SipManager;
    //   4: ifnonnull +4 -> 8
    //   7: return
    //   8: aload_0
    //   9: getfield 26 com/example/android/sip/WalkieTalkieActivity:˛ Landroid/net/sip/SipProfile;
    //   12: ifnull +7 -> 19
    //   15: aload_0
    //   16: invokespecial 34 com/example/android/sip/WalkieTalkieActivity:` ()V
    //   19: aload_0
    //   20: invokevirtual 38 com/example/android/sip/WalkieTalkieActivity:getBaseContext ()Landroid/
    //   23: invokestatic 44 android/preference/PreferenceManager:getDefaultSharedPreferences (Landro
```

Fix

- Obfuscation helps remove useful info
  - Set minifyEnabled = true
- Not a silver bullet
  - Anti ProGuard apps out there
  - Hackers just move to Smali
- Code in C++ using NDK
  - Much harder to read
  - Can still disassemble C++

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

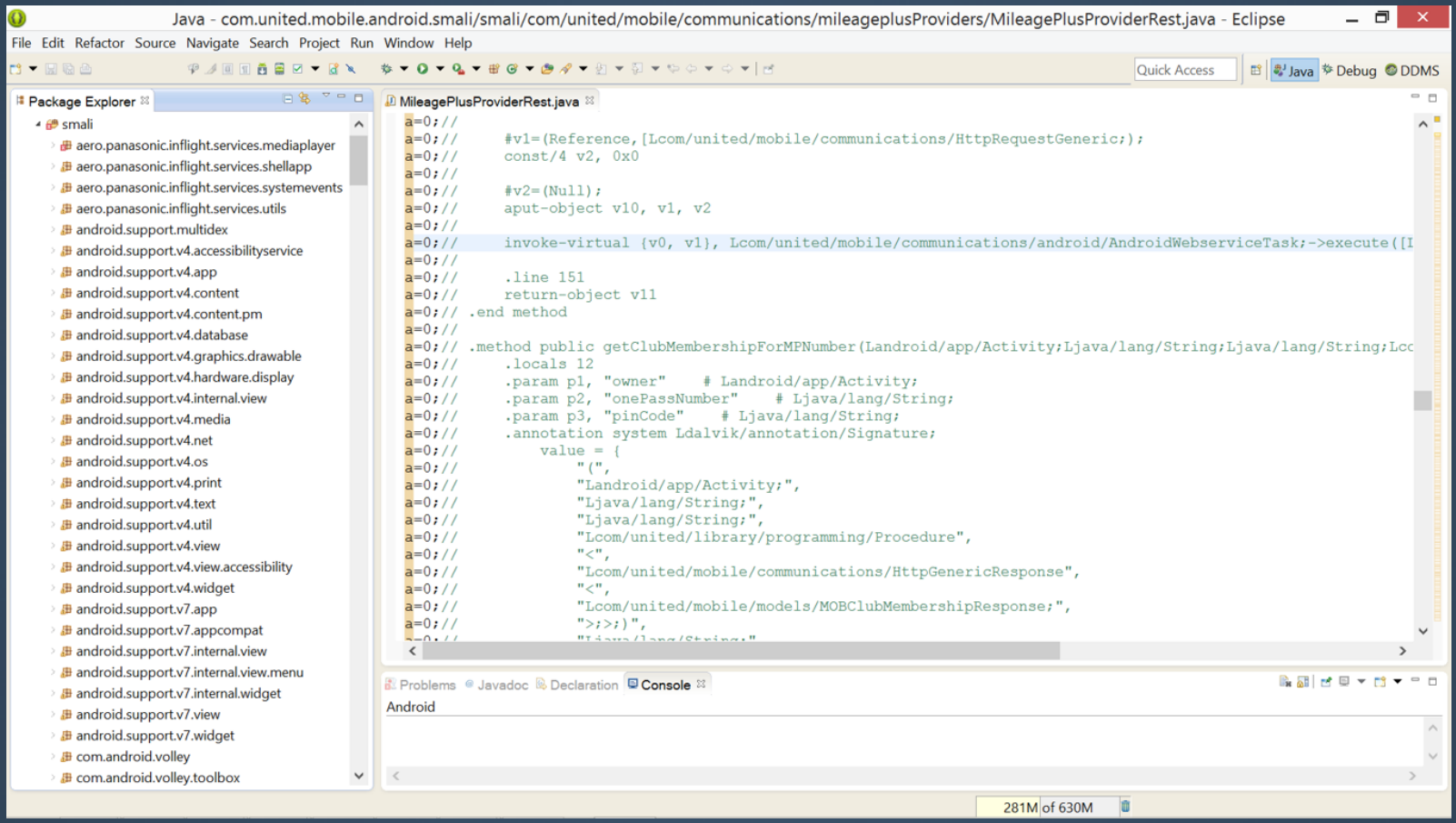
Improper Session Handling

Lack of Binary Protections

The Leftovers

- android:debuggable(true)
  - some Smali required
- SSL Pinning
- Bug Bounties
- SafetyNet API
- Frida

The Leftovers



The Leftovers

- Disassemble using apktool

```
java -jar apktool.jar d -d test.apk -o out
```

- Find main class in AdroidManifest.xml

```
<activity android:label="@string/app_name" android:name="com.riis.helloworld.MainActivity">
```

- Add debug wait to onCreate method

```
a=0;// # virtual methods
a=0;// .method protected onCreate(Landroid/os/Bundle;)V
a=0;//     invoke-static {}, Landroid/os/Debug;->waitForDebugger()V
a=0;//
a=0;//     .locals 1
a=0;//     .param p1, "savedInstanceState"    # Landroid/os/Bundle;
```

- Recompile using apktool

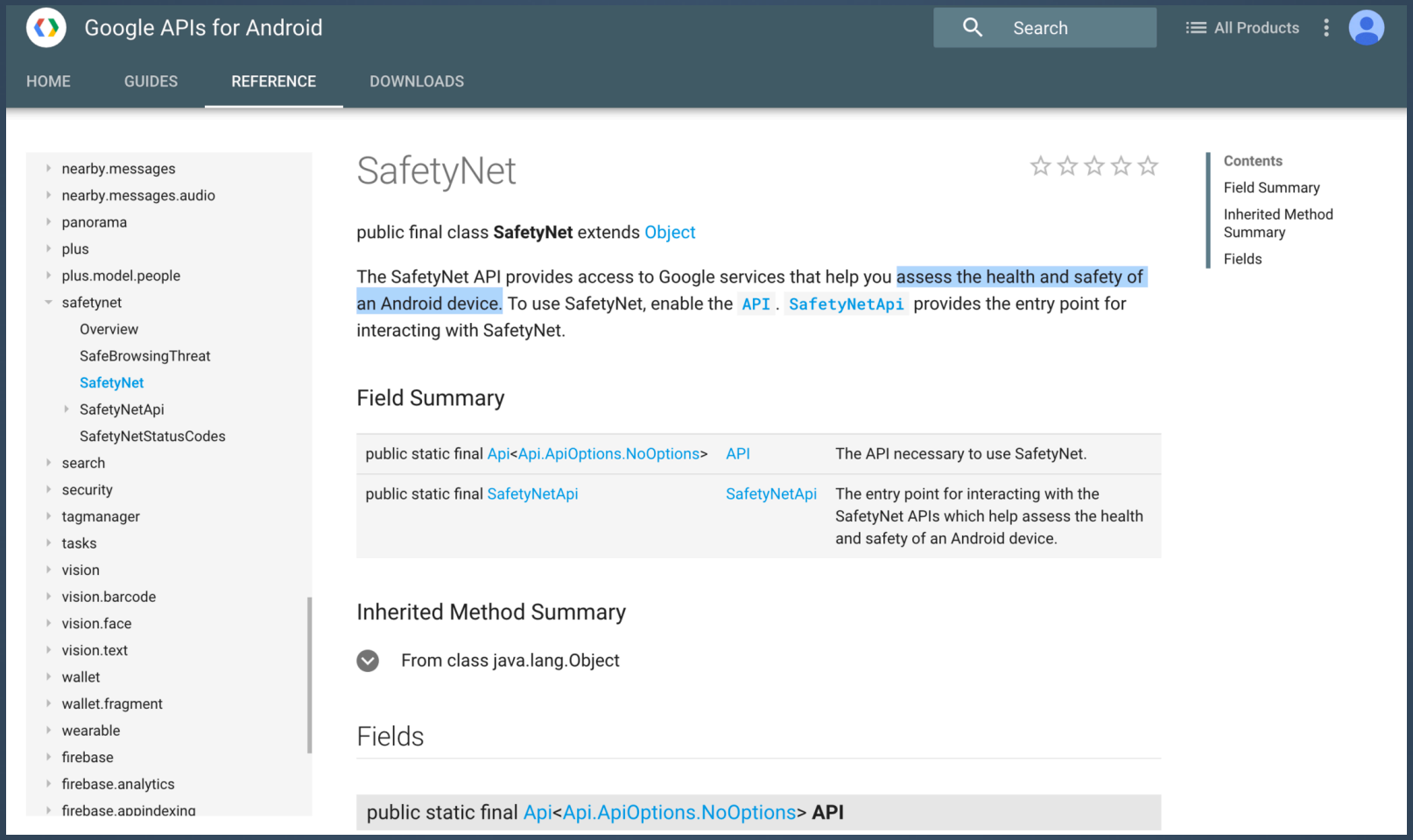
```
java -jar apktool.jar b -d out -o debug.apk
```

- Sign and install

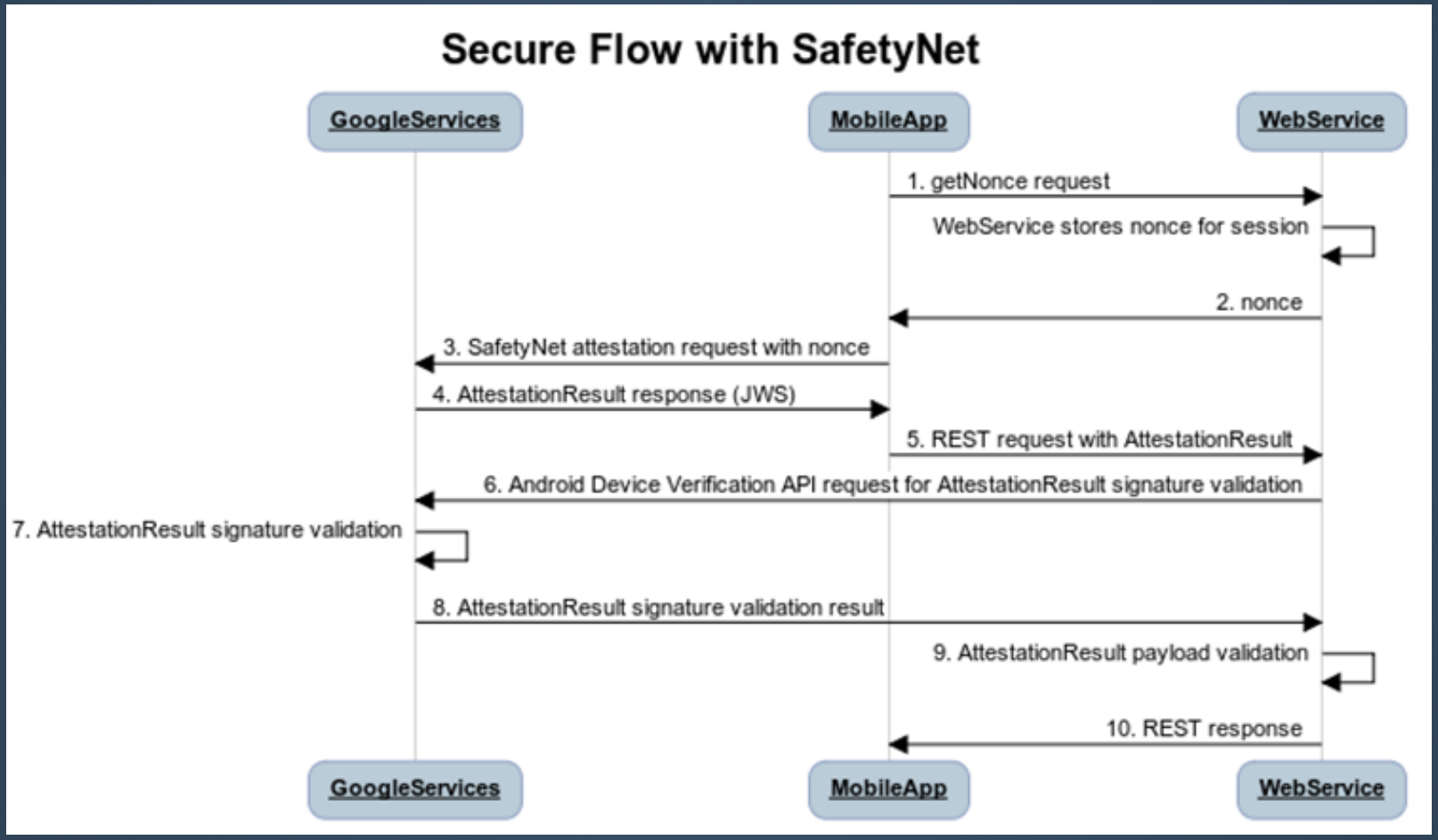
The Leftovers

- Security is too difficult to keep up with??
  - Crowdsource it with Bug Bounties
  - United Airlines offering substantial airmiles
- Lessons Learned
  - Requires effort to keep up with submissions
  - Update your app often to keep interest alive
  - Not a tool for shutting down researchers

The Leftovers



The Leftovers

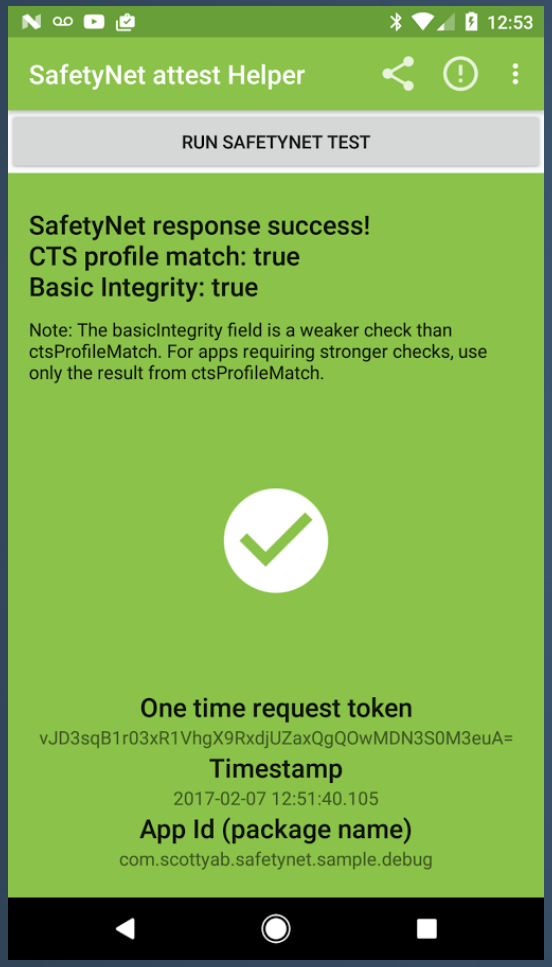


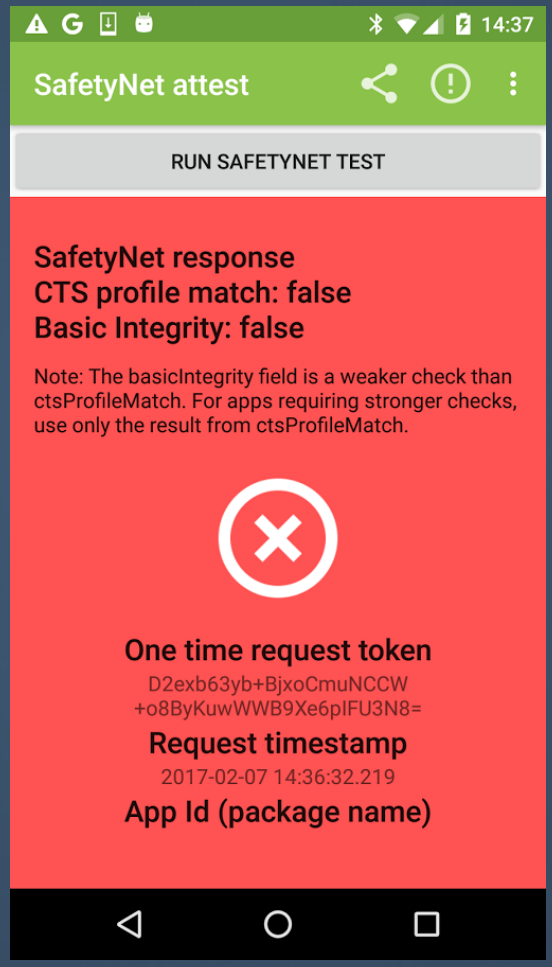
The Leftovers

```
byte[] nonce = getRequestNonce(); // Should be at least 16 bytes in length.
SafetyNet.SafetyNetApi.attest(mGoogleApiClient, nonce)
        .setResultCallback(new ResultCallback<SafetyNetApi.AttestationResult>() {

    @Override
    public void onResult(SafetyNetApi.AttestationResult result) {
        Status status = result.getStatus();
        if (status.isSuccess()) {
            // Indicates communication with the service was successful.
            // Use result.getJwsResult() to get the result data.
        } else {
            // An error occurred while communicating with the service.
        }
    }
});
```

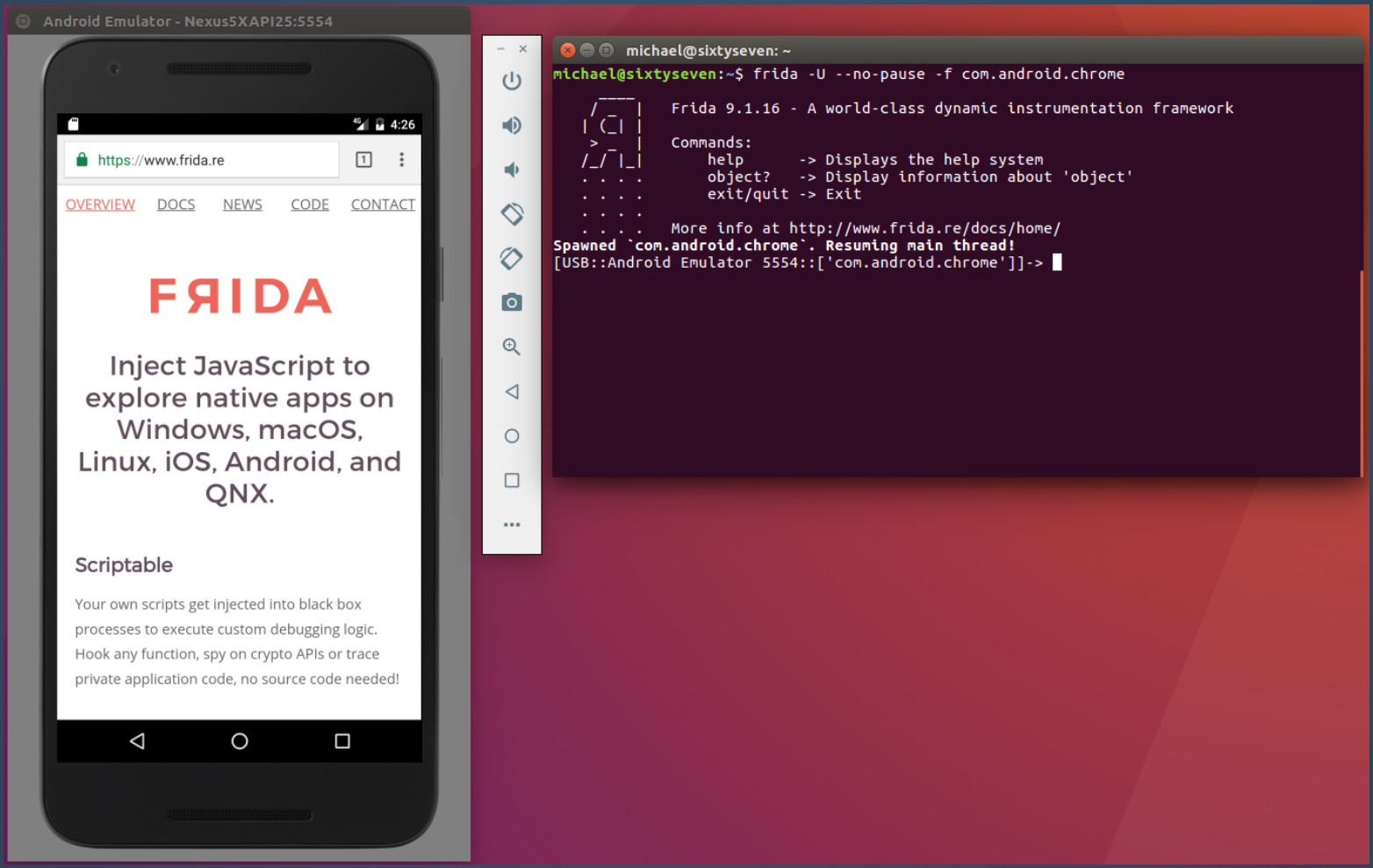
```
{
  "nonce": "R2Rra24fVm5xa2Mg",
  "timestampMs": 9860437986543,
  "apkPackageName": "com.package.name.of.requesting.app",
  "apkCertificateDigestSha256": ["base64 encoded, SHA-256 hash of the
                                  certificate used to sign requesting app"],
  "apkDigestSha256": "base64 encoded, SHA-256 hash of the app's APK",
  "ctsProfileMatch": true, // Compatibility Testing Suite
  "basicIntegrity": true,
}
```

The Leftovers





The Leftovers



Reasons to Ignore Security

- Security is too difficult to keep up with
- Requires physical access
  - Avast report - 80k old phones on eBay
- allowBackup=false
- Proguard / DexGuard is too hard to use
- The code is already obfuscated
- You need to talk to the API team
- Fragmentation
- We don't have time

Recommendations

- Understand debuggable=true, allowbackup=true
- Don’t trust, verify
- Rewrite SSL code, use asymmetric encryption
- Provide an email or security page for white hats
- Attacks are going to get more complex
- Start a Bug Bounty
- Store nothing important on the device
- Don't ignore Smali attacks
- Secure your server
- Use SafetyNet API

Resources

http://www.decompilingandroid.com

http://www.owasp.org

https://github.com/nelenkov/android-backup-extractor

http://www.charlesproxy.com

http://www.programering.com/a/MjM5UTMwATg.html

http://www.cs.ru.nl/~joeri/papers/spsm14.pdf

https://www.mwrinfosecurity.com/products/drozer

https://github.com/skylot/jadx

http://keyczar.org

https://www.nccgroup.trust/us/about-us/resources/intent-sniffer/

http://www.guardsquare.com

http://sqlitebrowser.org

http://bit.ly/1JlPoiY - How to hide your android API key

http://bit.ly/1hIeNNi - Where to store your password

https://github.com/google/nogotofail

https://github.com/godfreynolan/bulletproof

http://riis.com/blog/android-obfuscation

http://riis.com/blog/android-safetynet

http://frida.re

Gist List of Old* Hacks

Delta: https://gist.github.com/cbeyer-riis/32e3d028c0deebca4057
Groupon: https://gist.github.com/cbeyer-riis/151a3eeed66a0516d50f
Walgreens: https://gist.github.com/cbeyer-riis/4f3758f9a58f554d40a4
Target: https://gist.github.com/cbeyer-riis/a55d90e38554c7122c89
Match: https://gist.github.com/cbeyer-riis/73318ee997132024b17d
Walgreens: https://gist.github.com/cbeyer-riis/372212c1fb5128841dcf
eHarmony: https://gist.github.com/cbeyer-riis/9e21e9b9996ea536cc5c
Hilton Honors: https://gist.github.com/cbeyer-riis/0834606d33c581b2a045
Hyatt: https://gist.github.com/cbeyer-riis/bfcab3d7673fba868624
Holiday Inn: https://gist.github.com/godfreynolan/e01f6ae1fab31ab66c39

*Find older apk's on apkpure.com

Contact Details

godfrey@riis.com

@godfreynolan

slides.com/godfreynolan/bulletproofandroidmeetup