

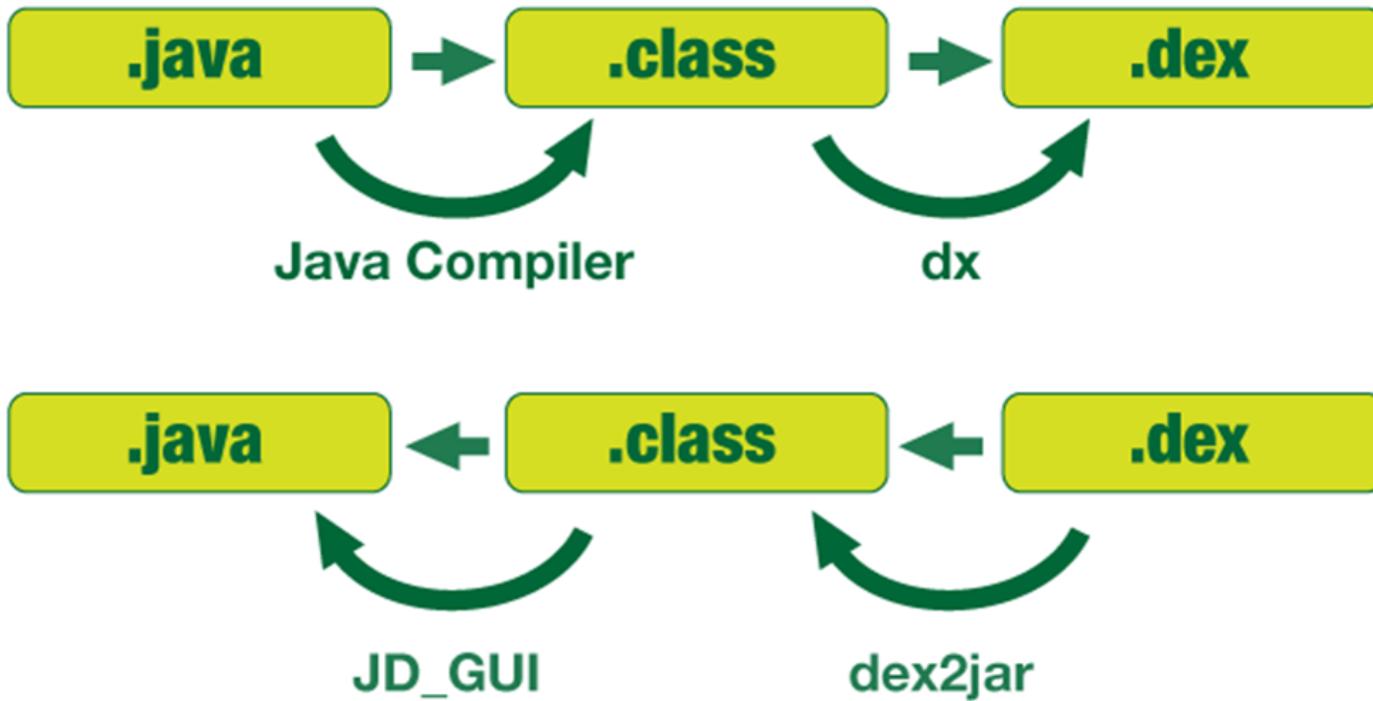
# Bulletproof Android

Godfrey Nolan

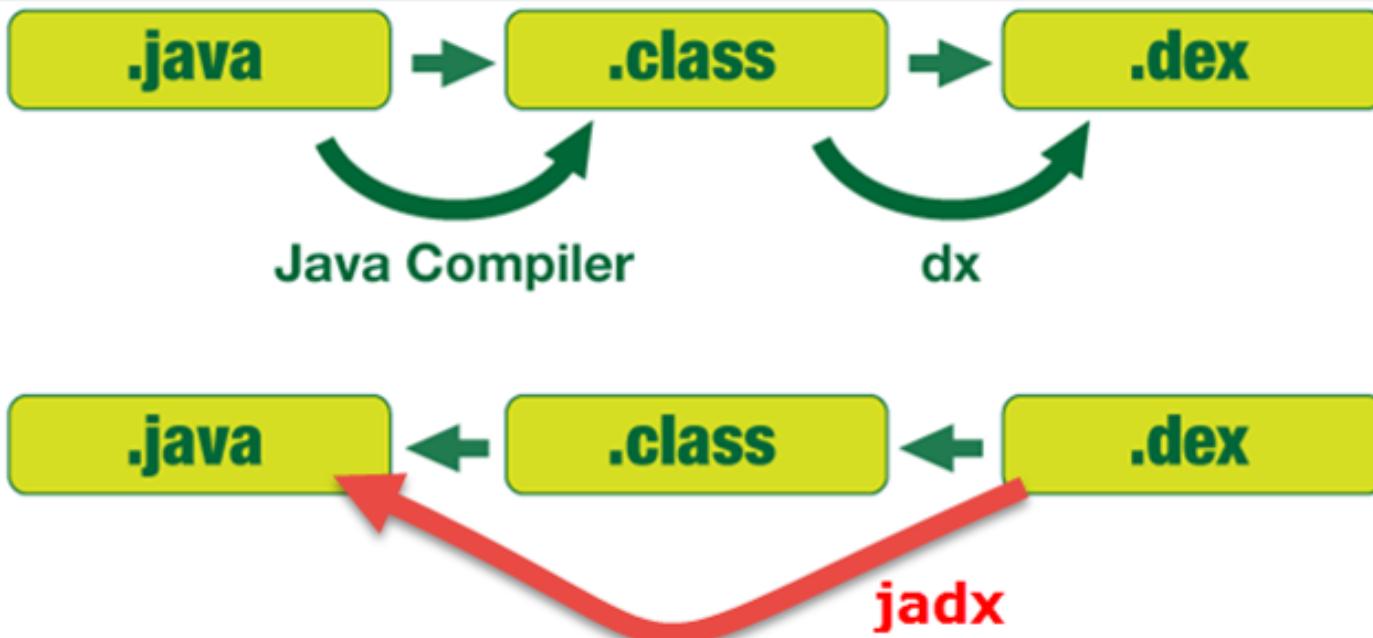
# How did we get here

- 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

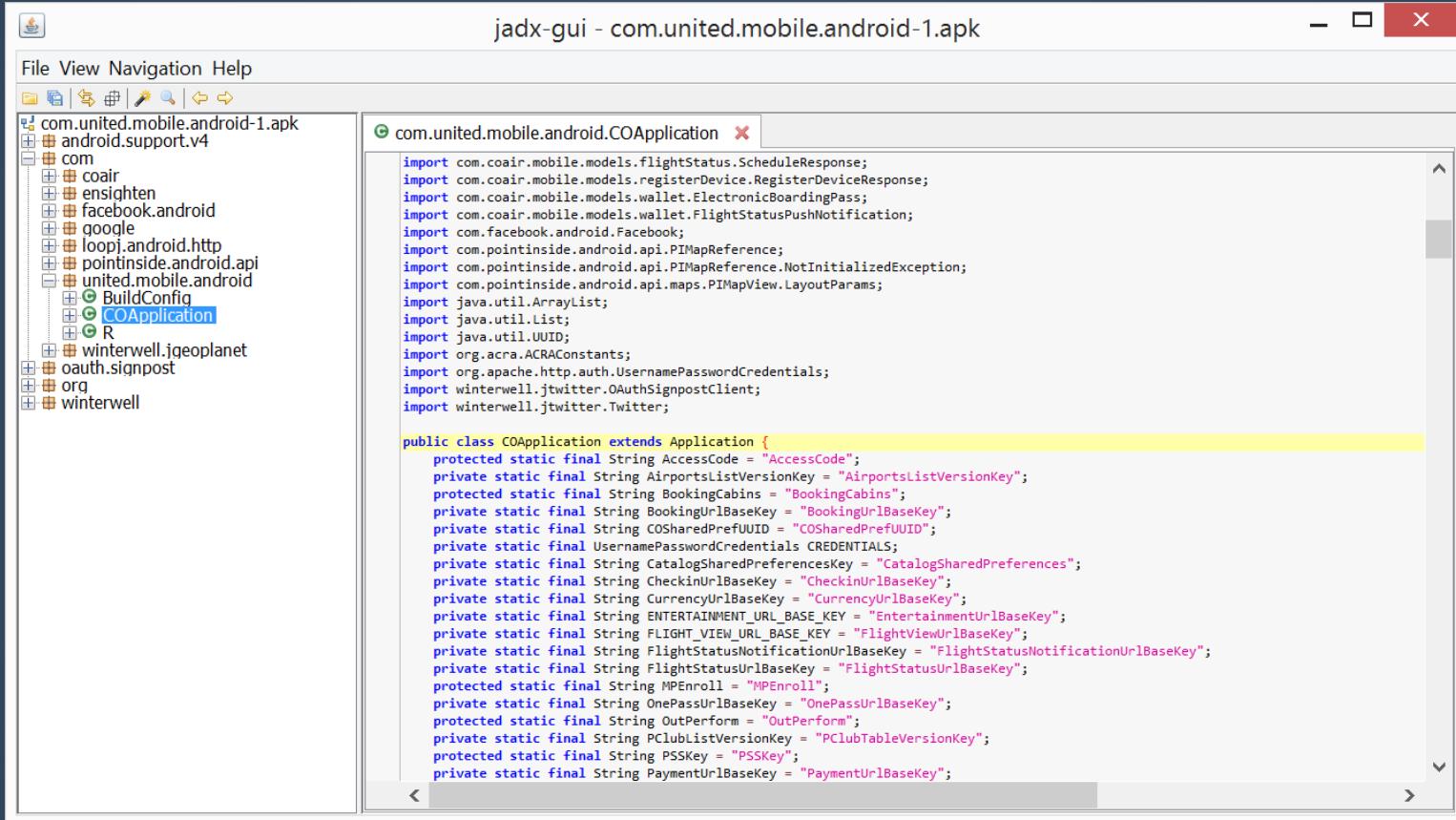
$ sqlite3 apps/com.united.mobile.android/db/united.db
```

# Decompilation 101

The screenshot shows the homepage of apkpure.com. At the top, there is a large banner featuring a character from Clash Royale. Below the banner, the text "Download free Android Games and Android Apps. ?" is displayed. A search bar contains the text "Facebook or https://play.google.com/store/apps/details?id=com.facebook.katana". To the right of the search bar is a green search button with a magnifying glass icon. On the far right, there is a vertical column of social media sharing icons for Facebook, Twitter, Tumblr, Google+, Reddit, and a plus sign for more options. The main content area features a section titled "Popular Games In Last 48 Hours" with a grid of game cards. The games listed are HIT (NEXON), FIFA 16 (EA Sports), Blossom Blast Saga, Geometry Dash Lite, Seven Knights, Subway Surfers, Kai-ri-Sei Million Art..., and Clash of Clans. Each card includes a small thumbnail image, the game name, developer, rating (from 5 stars), and the release date.

Popular Games In Last 48 Hours	
HIT NEXON Nov 24 2015	FIFA 16 EA Sports Nov 02 2015
Blossom Blast Saga Aug 24 2015	Geometry Dash Lite Nov 02 2015
Seven Knights Nov 24 2015	Subway Surfers Nov 02 2015
Kai-ri-Sei Million Art... Nov 24 2015	Clash of Clans Nov 02 2015

# Decompilation 101



The screenshot shows the jadx-gui interface with the title "jadx-gui - com.united.mobile.android-1.apk". The left pane displays the file tree of the APK, showing packages like com.united.mobile.android-1.apk, android.support.v4, com, coair, ensighten, facebook.android, google, loop.android.http, pointinside.android.api, united.mobile.android, oauth.signpost, org, and winterwell. The right pane shows the decompiled Java code for the COApplication class.

```
import com.coair.mobile.models.flightStatus.ScheduleResponse;
import com.coair.mobile.models.registerDevice.RegisterDeviceResponse;
import com.coair.mobile.models.wallet.ElectronicBoardingPass;
import com.coair.mobile.models.wallet.FlightStatusPushNotification;
import com.facebook.android.Facebook;
import com.pointinside.android.api.PIMapReference;
import com.pointinside.android.api.PIMapReference.NotInitializedException;
import com.pointinside.android.api.maps.PIMapView.LayoutParams;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
import org.acra.ACRAConstants;
import org.apache.http.auth.UsernamePasswordCredentials;
import winterwell.jtwitter.OAuthSignpostClient;
import winterwell.jtwitter.Twitter;

public class COApplication extends Application {
    protected static final String AccessCode = "AccessCode";
    private static final String AirportsListVersionKey = "AirportsListVersionKey";
    protected static final String BookingCabins = "BookingCabins";
    private static final String BookingUrlBaseKey = "BookingUrlBaseKey";
    private static final String CSharedPrefUUID = "CSharedPrefUUID";
    private static final UsernamePasswordCredentials CREDENTIALS;
    private static final String CatalogSharedPreferencesKey = "CatalogSharedPreferences";
    private static final String CheckinUrlBaseKey = "CheckinUrlBaseKey";
    private static final String CurrencyUrlBaseKey = "CurrencyUrlBaseKey";
    private static final String ENTERTAINMENT_URL_BASE_KEY = "EntertainmentUrlBaseKey";
    private static final String FLIGHT_VIEW_URL_BASE_KEY = "FlightViewUrlBaseKey";
    private static final String FlightStatusNotificationUrlBaseKey = "FlightStatusNotificationUrlBaseKey";
    private static final String FlightStatusUrlBaseKey = "FlightStatusUrlBaseKey";
    protected static final String MPEnroll = "MPEnroll";
    private static final String OnePassUrlBaseKey = "OnePassUrlBaseKey";
    protected static final String OutPerform = "OutPerform";
    private static final String PClubListVersionKey = "PClubTableVersionKey";
    protected static final String PSSKey = "PSSKey";
    private static final String PaymentUrlBaseKey = "PaymentUrlBaseKey";
```

# 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

Burp Suite Free Edition v1.5

Request

Raw Params Headers Hex

```
GET /herdfinancial/api/v1/balances/1234567899 HTTP/1.1
Cookie: AUTH=9524707
Host: 10.5.1.85:9888
Connection: Keep-Alive
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Content-Type: application/json
Server: Jetty(7.x.y-SNAPSHOT)
Content-Length: 65

{"success": "true", "checkingBalance": "0.0", "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

Improper Session Handling

Lack of Binary Protections

# Problem

Burp Suite Free Edition v1.5

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Options Alerts

1 × 2 × ...

Go Cancel < >

**Request**

Raw Params Headers Hex

GET /herdfinancial/api/v1/balances/**1234567899** HTTP/1.1  
Cookie: AUTH=9524707  
Host: 10.5.1.85:9888  
Connection: Keep-Alive

GET <http://herdfinancial.com/api/v1/balances/1234567899/>

? < + > Type a search term

**Response**

Raw Headers Hex

HTTP/1.1 200 OK  
Content-Type: application/json  
Server: Jetty(7.x.y-SNAPSHOT)  
Content-Length: 65

{"success": "true", "checkingBalance": "0.0", "savingsBalance": "**0.0**"}

{ "success": "true", "checkingBalance": "0.0", "savingsBalance": "**0.0**" }

? < + > Type a search term

Done

## 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

Burp Suite Free Edition v1.1

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Options Alerts

1 × 2 × ...

Go Cancel < >

Request

Raw Params Headers Hex

GET /herdfinancial/api/v1/balances/**1234567890** HTTP/1.1  
Cookie: AUTH=9524707  
Host: 10.5.1.85:9088  
Connection: Keep-Alive

GET <http://herdfinancial.com/api/v1/balances/1234567890/>

Type a search term

Response

Raw Headers Hex

HTTP/1.1 200 OK  
Content-Type: application/json  
Server: Jetty(7.x.y-SNAPSHOT)  
Content-Length: 67

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

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

Type a search term

Done

## 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

# 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",
```

The screenshot shows a social media post interface. At the top, there are three buttons: 'LIKE' with a heart icon, 'COMMENT' with a speech bubble icon, and 'REPOST' with a retweet icon. Below these buttons, it says '3 likes 1 repost'. The main area displays five comments from different users:

- Greatest Ever** (18 days ago) - Grab your spatulas, miniguns, and bibs because we are going on a journey
- Christen P** (18 days ago) - I WAS BORN A RAMBLIN MAN
- Mark Olmstead** (18 days ago) - DETROIT WHAT WHAT
- Ada Absofsteel** (18 days ago) - This app is so much fun <3
- Christen P** (4 days ago) - Omg....I never sent that message!! WTH!!!

At the bottom, there is a text input field with the placeholder 'Write a comment...' and a smiley face emoji icon.

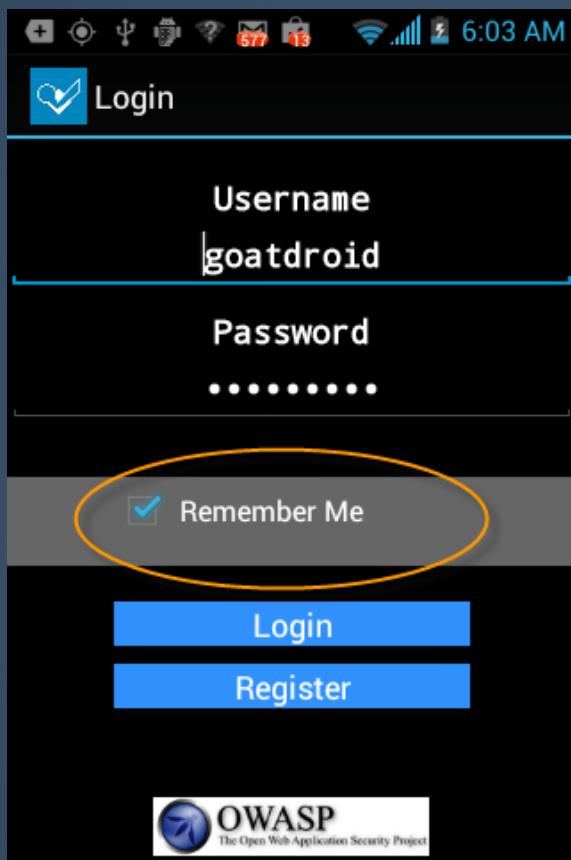
# Fix

- 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

msg_id	thread_id	action_id	subject	text	sender	nestamp_n	stamp_sent	is
43	m_id.1687224665...	t_1f9wp2...	1319567...	NULL	Always available for beers Let me ...	{"email": "..."} 1319567...	NULL	NULL
44	m_id.2913921542...	t_1f9wp2...	1319561...	NULL	Just passing through Oakville stati...	{"email": "..."} 1319561...	NULL	NULL
45	m_e40ce23f7141...	t_1f9wp2...	1310735...	NULL	Great. Thanks for the info. We're le...	{"email": "..."} 1310735...	NULL	NULL
46	m_b0ba73941a64...	t_1f9wp2...	1310610...	NULL	When are you traveling? If you wa...	{"email": "..."} 1310610...	NULL	NULL
47	m_2a3b91bf9df6...	t_1f9wp2...	1310599...	NULL	Thought I replied to this, and of c...	{"email": "..."} 1310599...	NULL	NULL
48	m_0fee43dd73cd...	t_1f9wp2...	1310431...	NULL	It depends on what day and time. ...	{"email": "..."} 1310431...	NULL	NULL
49	m_d6625860a2b...	t_1f9wp2...	1310170...	NULL	Hope all is well. Me and the kids ...	{"email": "..."} 1310170...	NULL	NULL
50	m_4+GdTNbitGN...	t_1f9wp2...	1304818...	NULL	Air Transat is one, Air Canada also...	{"email": "..."} 1304818...	NULL	NULL
51	m_dx5sl4LfRQtX...	t_1f9wp2...	1304814...	NULL	any idea what the charter flights ai...	{"email": "..."} 1304814...	NULL	NULL
52	m_G2X2oqrzpN...	t_1f9wp2...	1304810...	NULL	I normally use travelocity.ca or itra...	{"email": "..."} 1304810...	NULL	NULL
53	m_rtvhHFMKW9...	t_1f9wp2...	1304783...	NULL	do you know any charter flights o...	{"email": "..."} 1304783...	NULL	NULL
54	m_c7vF9F0CY7I1...	t_1f9wn2...	1298426...	NULL	YeaWe all went home for a short t...	{"email": "..."} 1298426...	NULL	NULL

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

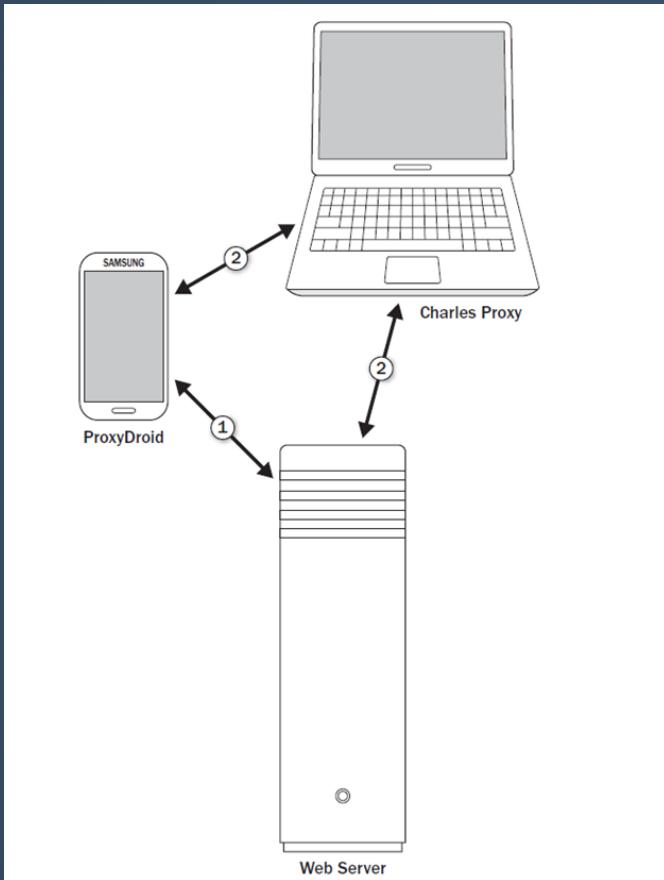
Client Side Injection

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

Name	Value
user	xriix03
password	recruiter2
redirect	http%3a%2f%2fhiring..com%2fdefault.aspx%3fHasUserAccount%3d2
ctrl	1
_RequestVerificationToken	A1CAF7FFF74351119D86FC5EDC3D25E
_EVENTTARGET	ctl00\$ctl00\$ContentPlaceHolderBase\$cphBody\$LoginUser\$SubmitImage
_EVENTARGUMENT	
vsk	
_VIEWSTATE	6268dac4-1b2a-4681-8185-6689427e1f7d
ct00\$ctl00\$MasterScriptManager	
ct00\$ctl00\$ContentPlaceHolderBase\$cphBody\$LoginUser\$txtUserName	xriix03
ct00\$ctl00\$ContentPlaceHolderBase\$cphBody\$LoginUser\$txtPassWord	recruiter2
CA_DATA	
setAutoLogin	

# Example

Weak Server Side Controls

Insecure Data Storage

## Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

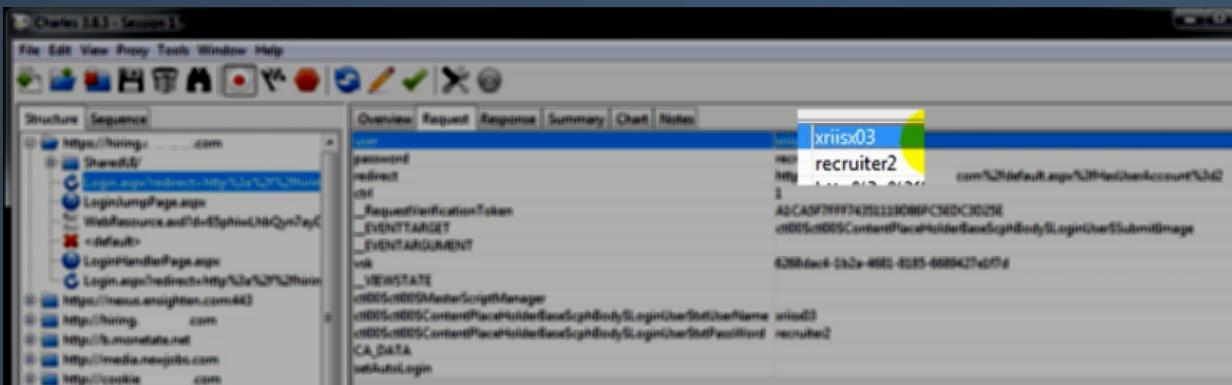
Broken Cryptography

Client Side Injection

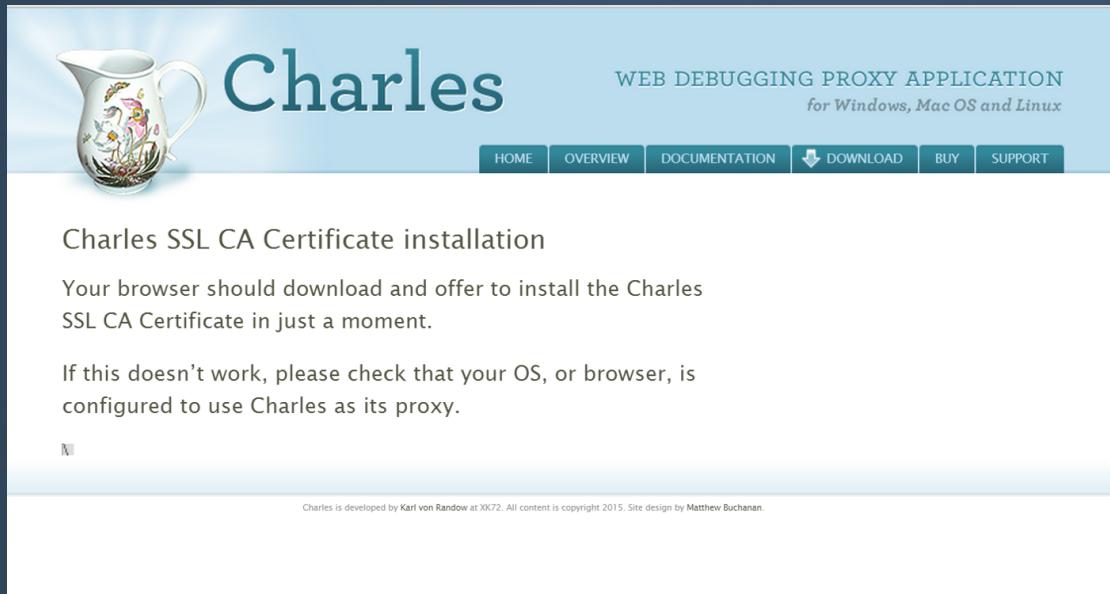
Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections



# More Problems



The screenshot shows the Charles SSL CA Certificate installation page. At the top left is a decorative illustration of a white pitcher with a colorful floral pattern. To its right, the word "Charles" is written in a large, teal, sans-serif font. Above the main content area, there's a header bar with the text "WEB DEBUGGING PROXY APPLICATION" and "for Windows, Mac OS and Linux". Below this are navigation links: HOME, OVERVIEW, DOCUMENTATION, DOWNLOAD (with a download icon), BUY, and SUPPORT. The main content area contains the following text:

Charles SSL CA Certificate installation

Your browser should download and offer to install the Charles SSL CA Certificate in just a moment.

If this doesn't work, please check that your OS, or browser, is configured to use Charles as its proxy.

Charles is developed by Karl von Rando at XK72. All content is copyright 2015. Site design by Matthew Buchanan.

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" +
    "ac1a9116acc883862f00593199df19ce027c8eaaaee8e3121f7f329219464e657" +
    "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

The screenshot shows the Android LogCat interface. The title bar includes tabs for Problems, Javadoc, Declaration, and LogCat. The main area displays a table of log messages with columns for Level, Time, PID, TID, Application, Tag, and Text. A search bar at the top allows filtering by Java regexes. The application listed in the logs is com.riis.logfiles. The log entries include:

Level	Time	PID	TID	Application	Tag	Text
D	09-03 20:29:31...	1074	1074	com.riis.logfiles	dalvikvm	Not late-enabling CheckJNI (already on)
D	09-03 20:29:34...	1074	1074	com.riis.logfiles	com.riis....	Passport document # - W123456
D	09-03 20:29:35...	1074	1074	com.riis.logfiles		HostConnection::get() New Host Connection established 0xb7375650 e , tid 1074
W	09-03 20:29:35...	1074	1074	com.riis.logfiles	EGL_emula...	eglSurfaceAttrib not implemented
D	09-03 20:29:35...	1074	1074	com.riis.logfiles	OpenGLRen...	Enabling debug mode 0
D	09-03 20:29:36...	1074	1087	com.riis.logfiles	dalvikvm	GC_FOR_ALLOC freed 157K, 12% free 3515K/3960K, paused 366ms, tot al 371ms

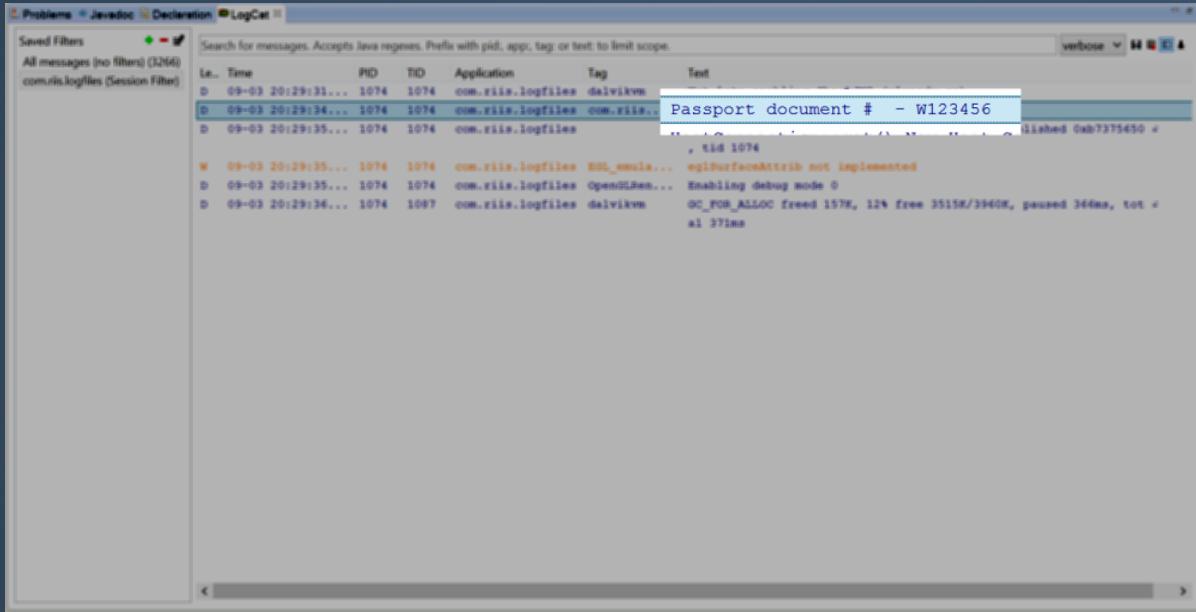
# Problem

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

## Unintended Data Leakage



The screenshot shows the Android LogCat interface. A specific log entry is highlighted, revealing sensitive information:

Time	PID	TID	Application	Tag	Text
09-03 20:29:31...	1074	1074	com.riis.logfiles	dalvikvm	Passport document # - W123456
09-03 20:29:34...	1074	1074	com.riis.logfiles	com.riis.	
09-03 20:29:35...	1074	1074	com.riis.logfiles		
09-03 20:29:35...	1074	1074	com.riis.logfiles	EGL_emu...	eglSurfaceAttrib not implemented
09-03 20:29:35...	1074	1074	com.riis.logfiles	OpenGL ES...	Enabling debug mode 0
09-03 20:29:36...	1074	1087	com.riis.logfiles	dalvikvm	GC_FOR_ALLOC freed 157K, 12% free 3515K/3940K, paused 36ms, tot al 37ms

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

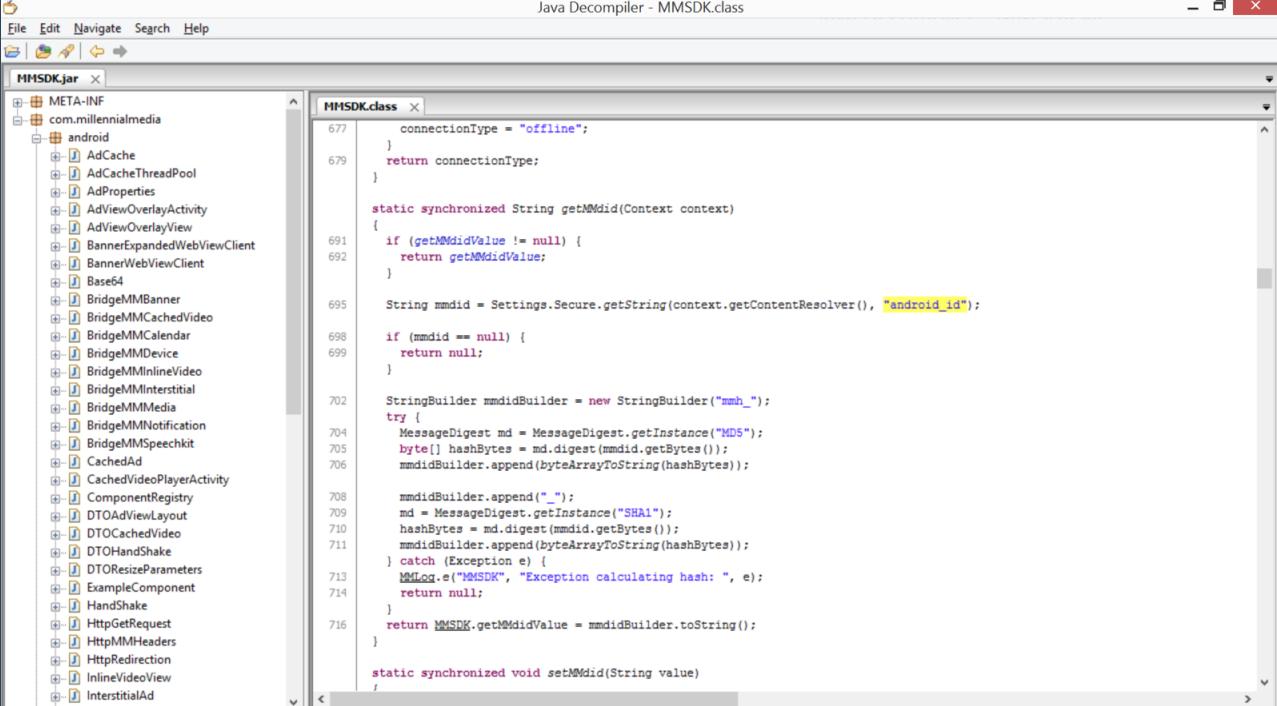
Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections



The screenshot shows a Java decompiler interface with the title "Java Decomiler - MMSDK.class". On the left is a tree view of the class hierarchy under "MMSDK.jar", including packages like META-INF, com.millennialmedia, and android, and classes such as AdCache, AdProperties, AdViewOverlayActivity, AdViewOverlayView, BannerExpandedWebViewClient, BannerWebViewClient, Base64, BridgeMMBanner, BridgeMMCachedVideo, BridgeMMCalendar, BridgeMMDevice, BridgeMMInlineVideo, BridgeMMInterstitial, BridgeMMMedia, BridgeMMNotification, BridgeMMSpeechkit, CachedAd, CachedVideoPlayerActivity, ComponentRegistry, DTOADViewLayout, DTOCachedVideo, DTOHandShake, DTOResizeParameters, ExampleComponent, HandShake, HttpRequest, HttpMMHeaders, HttpRedirection, InlineVideoView, and InterstitialAd. The main window displays the decompiled code for the MMSDK class. The code includes methods for getting and setting the mmid (mobile marketing identifier) using MD5 and SHA1 hashing. It also handles exceptions and logs errors.

```
    connectionType = "offline";
}
return connectionType;
}

static synchronized String getMmid(Context context)
{
    if (getMmidValue != null) {
        return getMmidValue;
    }

    String mmid = Settings.Secure.getString(context.getContentResolver(), "android_id");

    if (mmid == null) {
        return null;
    }

    StringBuilder mmidBuilder = new StringBuilder("mmh_");
    try {
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] hashBytes = md.digest(mmid.getBytes());
        mmidBuilder.append(bytetoString(hashBytes));

        mmidBuilder.append("_");
        md = MessageDigest.getInstance("SHA1");
        hashBytes = md.digest(mmid.getBytes());
        mmidBuilder.append(bytetoString(hashBytes));
    } catch (Exception e) {
        MLog.e("MMSDK", "Exception calculating hash: ", e);
        return null;
    }
    return MMSDK.getMmidValue = mmidBuilder.toString();
}

static synchronized void setMmid(String value)
;
```

# Example

```
public ActivityLaunchAppLoad() {
    this.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_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 unnecessary 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

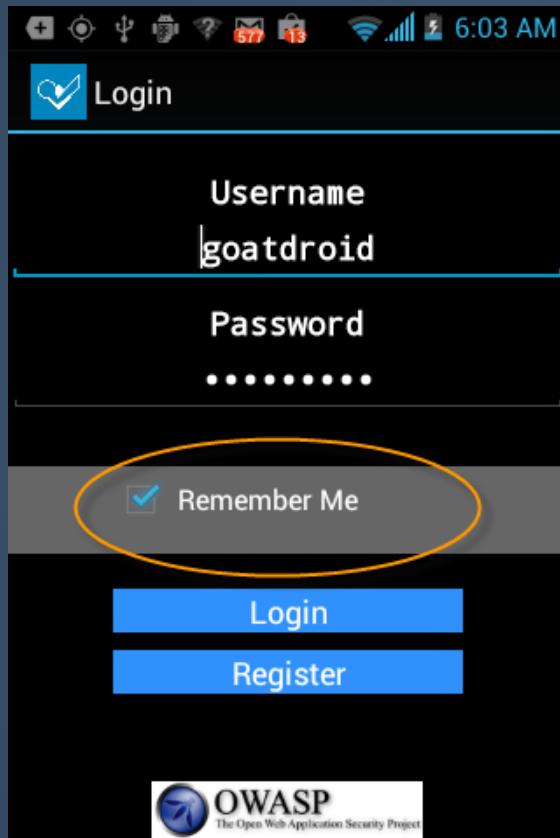
Client Side Injection

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">77ef62159ad9c32913dfdbbee0e58aea3</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

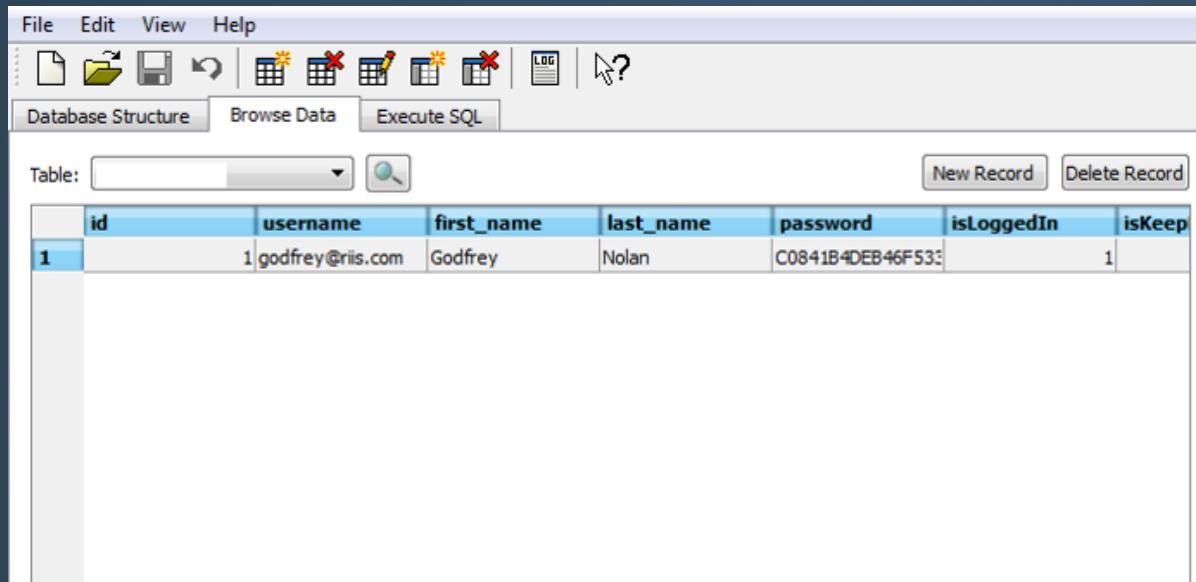
## Broken Cryptography

Client Side Injection

Security Decision via Untrusted Input

Improper Session Handling

Lack of Binary Protections



A screenshot of a database management software interface. The menu bar includes File, Edit, View, Help, and a toolbar with various icons. Below the toolbar are buttons for Database Structure, Browse Data, and Execute SQL. A search bar labeled 'Table:' is present. The main area displays a table with columns: id, username, first\_name, last\_name, password, isLoggedIn, and isKeep. One record is shown with values: id=1, username='godfrey@riis.com', first\_name='Godfrey', last\_name='Nolan', password='C0841B4DEB46F533', isLoggedIn=1, and isKeep=1.

	<b>id</b>	<b>username</b>	<b>first_name</b>	<b>last_name</b>	<b>password</b>	<b>isLoggedIn</b>	<b>isKeep</b>
	1	godfrey@riis.com	Godfrey	Nolan	C0841B4DEB46F533	1	1

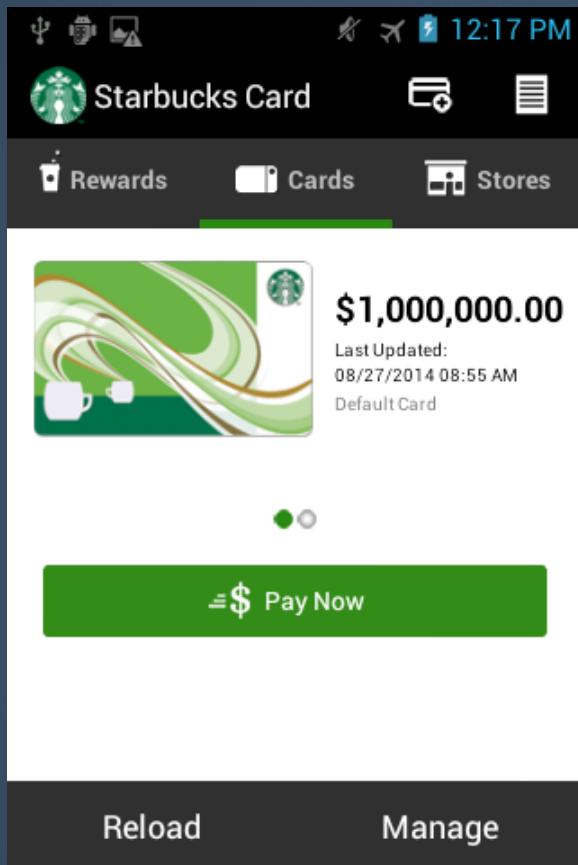
# Example

```
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

# Example



Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

## Broken Cryptography

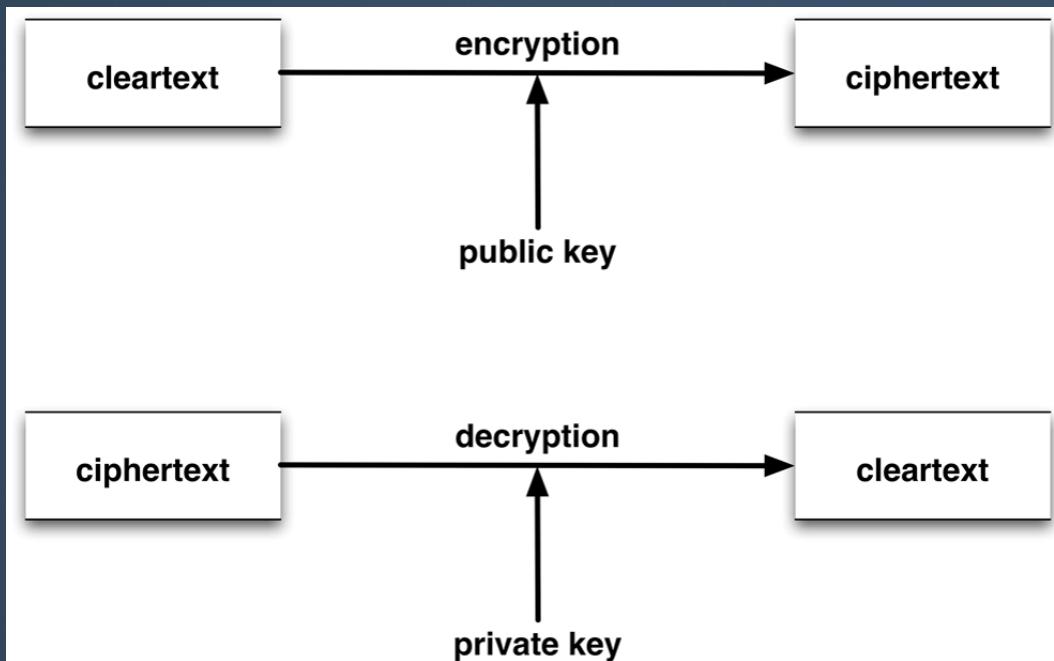
Client Side Injection

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

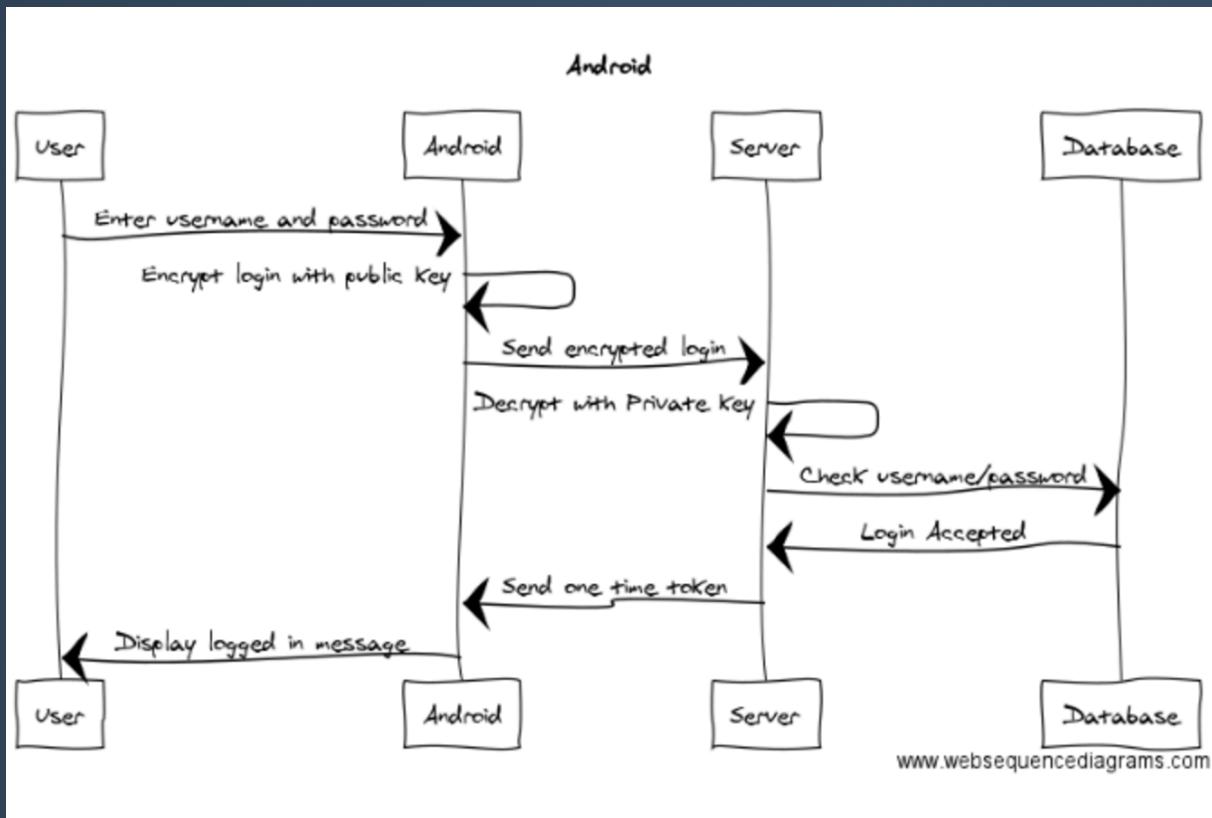
Client Side Injection

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

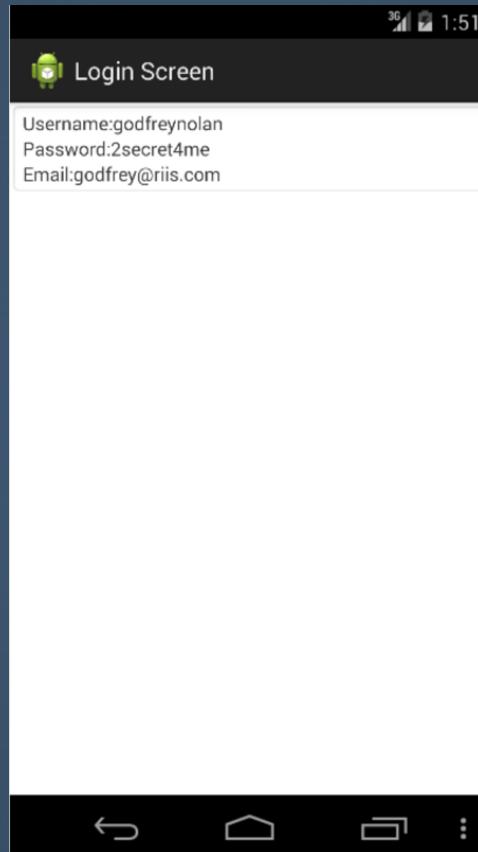
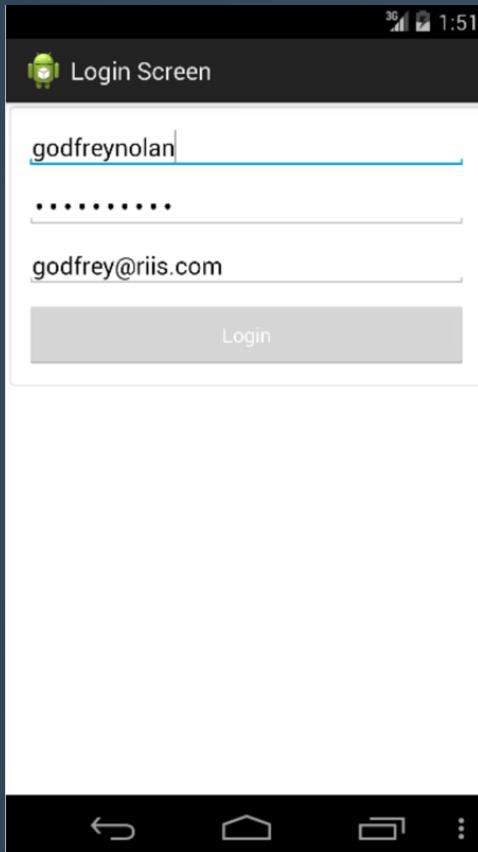
## **Client Side Injection**

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

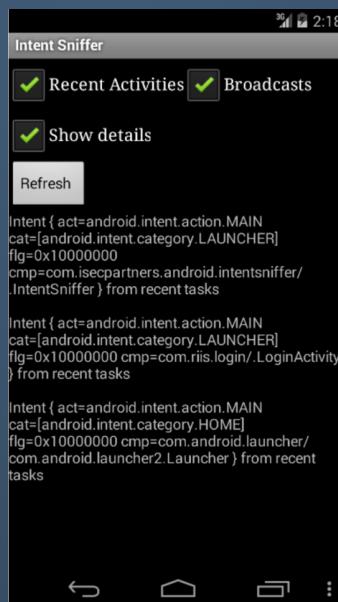
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.riis.hellointent"
    android:versionCode="1"
    android:versionName="1.0" >

    <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.generateAuthToken();  
    doa.updateAuthorizedDeviceAuth(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

```
root@jf1tevzu:/data/data/com.tumblrr # cd shared_prefs/
root@jf1tevzu:/data/data/com.tumblrr/shared_prefs # ls
ls
FLURRY_SHARED_PREFERENCES.xml
firstVisit.xml
tumblrr.xml
root@jf1tevzu:/data/data/com.tumblrr/shared_prefs # cat tum
at tumblrr.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="user_uuid_str">584a6040-a88e-47fc-b106-57c21f4c9471</string>
    <string name="AuthTokenSecret2">Jt3TZZW61rWkvvqFjAPUj9kRDt85ku3HZDtQJ12392p
wPzKQk</string>
        <string name="LoginStatus">true</string>
        <int name="user_following_int" value="1" />
        <int name="user_like_count_int" value="0" />
        <string name="feature_configuration_string">&quot;saber&quot;:&quot;true&qu
ot;,&quot;autoPlayVideo&quot;:&quot;true&quot;,&quot;saberTickImageData&quot;:&q
uot;true&quot;,&quot;blockQuot:&quot;true&quot;,&quot;sslQuot:&quot;true&quo
t;,&quot;followSomeBlogs&quot;:&quot;true&quot;,&quot;appAttributionCpi&quot;:&q
uot;false&quot;,&quot;tourGuideQuot:&quot;true&quot;,&quot;blinkfeedRateLimit&
quot;:&quot;false&quot;,&quot;postActionButtonQuot:&quot;true&quot;,&quot;vime
oIntegrationQuot:&quot;true&quot;,&quot;blogNotificationsQuot:&quot;true&quo
t;,&quot;showOnboardingTopicsQuot:&quot;false&quot;,&quot;csLogImageData&quot;
:&quot;true&quot;,&quot;messaging&quot;:&quot;false&quot;,&quot;reblogRedesign&q
uot;:&quot;false&quot;,&quot;appAttributionQuot:&quot;false&quot;></string>
        <long name="last_notices_check_long" value="1440080795961" />
        <boolean name="push_subscriptions_boolean" value="true" />
        <string name="FOLLOW">TOUR_NOT_VIEWED</string>
        <string name="pref_gcm_id">PA91bNmFRyUpY15_FWm-G-DctWb9P1674C1XXN6PS6mn
ZI50MJeMQ_h8ZJXlernhuZoXFwaOu94dJbMuU-R-vQgwXEmljEKDtIpY_S2d5WUUPU8</string>
        <string name="BLOG_CUSTOMIZE">TOUR_NOT_VIEWED</string>
        <string name="pref_appwidget_blog_name"></string>
        <string name="bcookie_string">?a0kqut9chpj3</string>
        <string name="REBLOG">TOUR_NOT_VIEWED</string>
        <string name="FAST_REBLOG">TOUR_NOT_VIEWED</string>
        <long name="feature_request_time_long" value="1440080824589" />
        <int name="pref_gcm_registered_version" value="1439105" />
        <long name="user_uuid_timestamp_long" value="1440010504" />
        <string name="com.tumblrr.choose_blog">weareatestaccount</string>
        <string name="NEW_POST">REACTION_VIEWED</string>
        <string name="debug_linkk">kBp9D3gulJxWhI6f8lxIAKJyvt1</string>
        <boolean name="pref_should_show_dialog" value="false" />
        <string name="AuthToken2">0ZctxPK0jzMDPC2eNHrdaKwgkz8wxwFsq52hCZshceurnKSLU
</string>
        <string name="userTumblrName">weareatestaccount</string>
        <string name="user_name">testingdezapps@gmail.com</string>
        <string name="SEARCH">TOUR_NOT_VIEWED</string>
        <string name="userDefaultPostFormat">html</string>
        <boolean name="welcome_spinner" value="true" />
        <boolean name="master_push_boolean" value="true" />
        <string name="server_configuration_string">&quot;saber_endpoint&quot;:&quot
;https://&quot;saber.srvcs.tumblrr.com&quot;,&quot;saber_key&quot;:&quot;Bcy4EIP7tTgk
JUmuT7D3TRofkaJQuUskFpM8dUGPvUNbegqA35uFUQjyiCFw3zF&quot;></string>
        <string name="LIKE">TOUR_NOT_VIEWED</string>
</map>
root@jf1tevzu:/data/data/com.tumblrr/shared_prefs #
```

Weak Server Side Controls

Insecure Data Storage

Insufficient Transport Layer Protection

Unintended Data Leakage

Poor Authorization and Authentication

Broken Cryptography

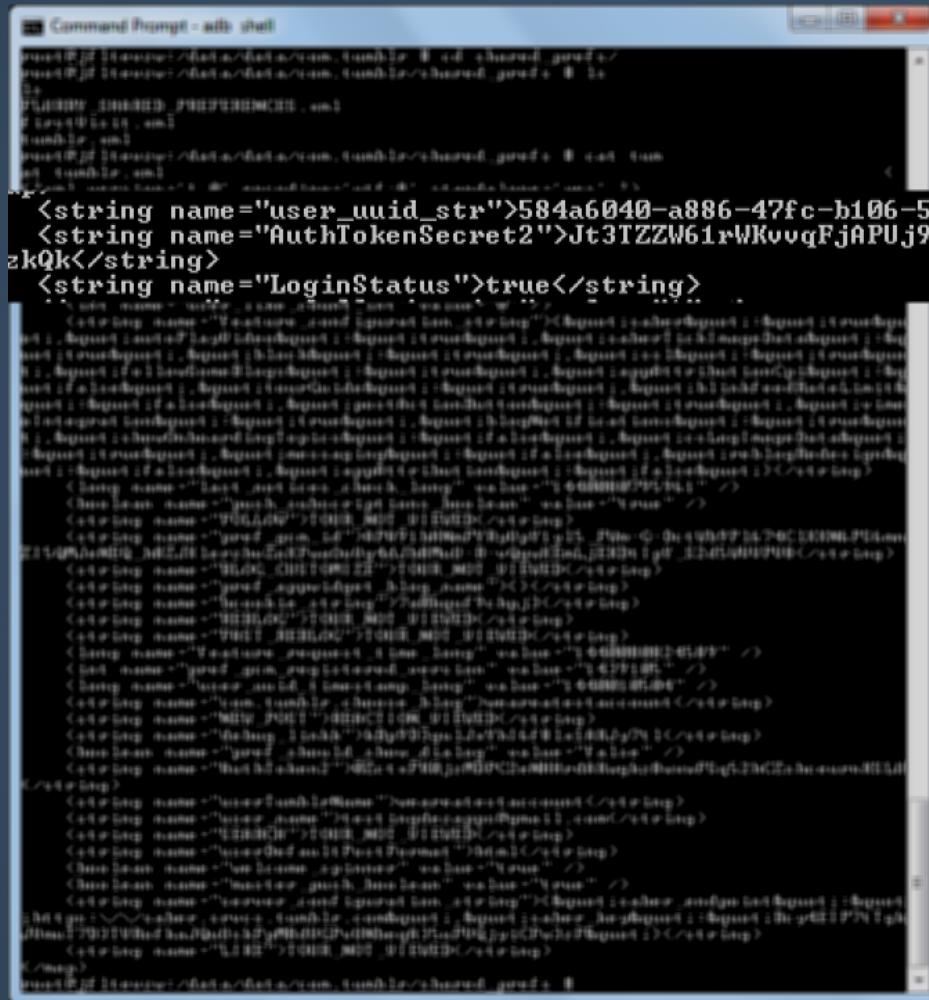
Client Side Injection

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

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**

```
/**  
 * 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(getApplicationContext());  
    String username = prefs.getString("namePref", "");  
    String domain = prefs.getString("domainPref", "");  
    String password = prefs.getString("passPref", "");
```

# Example

```
public void initializeLocalProfile()
{
    if (this.manager == null)
        return;
    if (this.me != null)
        closeLocalProfile();
    SharedPreferences localSharedPreferences = PreferenceManager.getDefaultSharedPreferences();
    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.getDefaultSharedPreferences();
    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 screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java - com.united.mobile.android.smali/smali/com/united/mobile/communications/mileageplusProviders/MileagePlusProviderRest.java - Eclipse
- Menu Bar:** File Edit Refactor Source Navigate Search Project Run Window Help
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, and others.
- Quick Access:** Buttons for Java, Debug, and DDMS.
- Package Explorer:** Shows the project structure under the package smali, which contains numerous Android support libraries and aero.panasonic.inflight.services.\*.
- MileagePlusProviderRest.java:** The active editor window displays Java code for a class named MileagePlusProviderRest. The code is heavily annotated with comments and includes several invokevirtual and invokeinterface instructions, indicating it's a generated or heavily modified Java file.
- Problems View:** Shows no errors or warnings.
- Console View:** Displays the text "Android".
- Bottom Status Bar:** Shows memory usage: 281M of 630M.

# 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

Google APIs for Android

HOME GUIDES REFERENCE DOWNLOADS

Search All Products :

SafetyNet

public final class **SafetyNet** extends Object

The SafetyNet API provides access to Google services that help you assess the health and safety of an Android device. To use SafetyNet, enable the [API](#). **SafetyNetApi** provides the entry point for interacting with SafetyNet.

Field Summary

public static final <a href="#">Api&lt;Api.ApiOptions.NoOptions&gt;</a> <b>API</b>	The API necessary to use SafetyNet.
public static final <b>SafetyNetApi</b>	<b>SafetyNetApi</b> The entry point for interacting with the SafetyNet APIs which help assess the health and safety of an Android device.

Inherited Method Summary

From class java.lang.Object

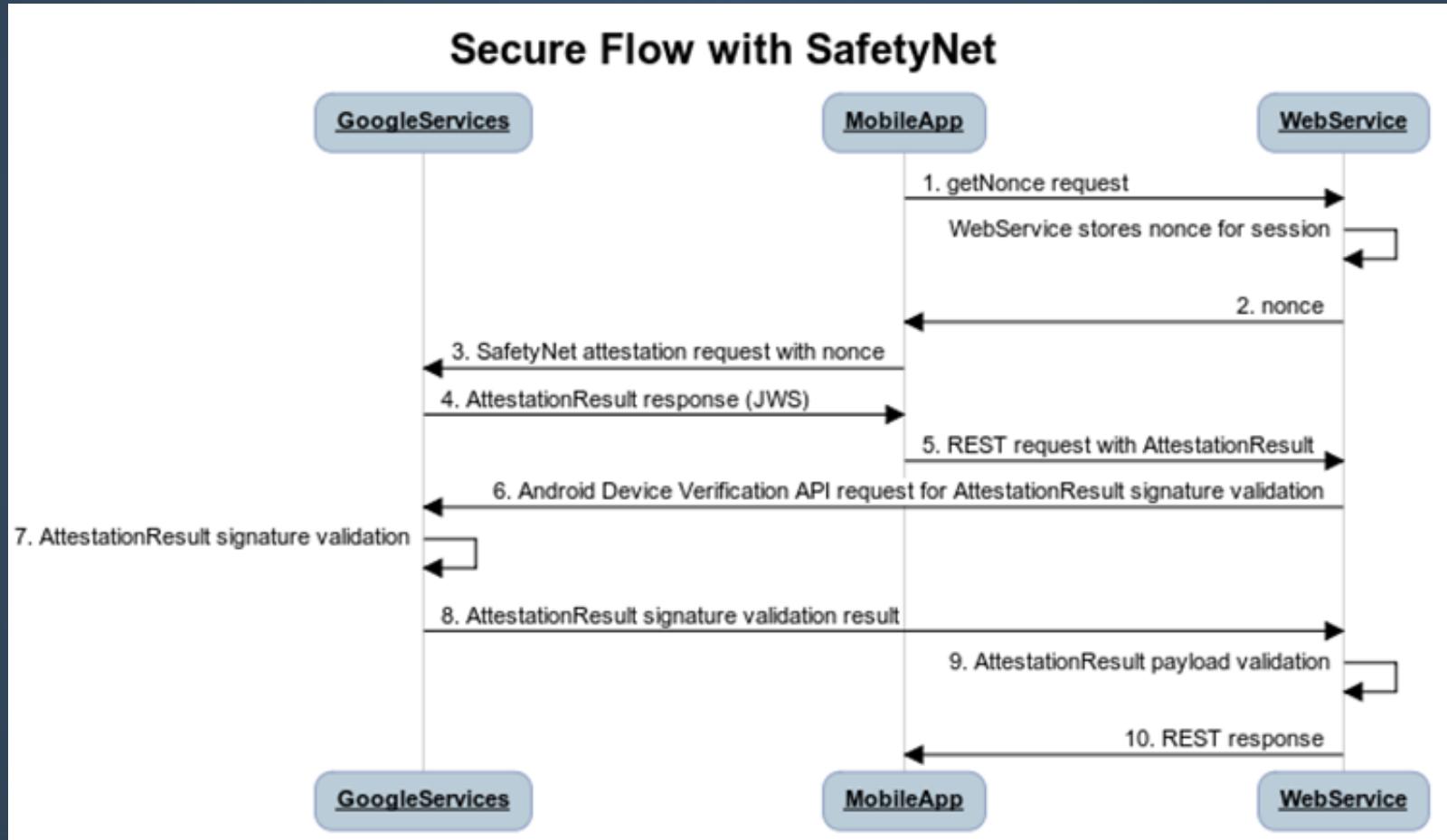
Fields

public static final <a href="#">Api&lt;Api.ApiOptions.NoOptions&gt;</a> <b>API</b>
--

- ▶ nearby.messages
- ▶ nearby.messages.audio
- ▶ panorama
- ▶ plus
- ▶ plus.model.people
- ▼ safetynet
  - Overview
  - SafeBrowsingThreat
  - SafetyNet**
    - ▶ SafetyNetApi
    - SafetyNetStatusCodes
  - search
  - security
  - tagmanager
  - tasks
  - vision
  - vision.barcode
  - vision.face
  - vision.text
  - wallet
  - wallet.fragment
  - wearable
  - firebase
  - firebase.analytics
  - firebase.appindexing

Contents  
Field Summary  
Inherited Method Summary  
Fields

# The Leftovers

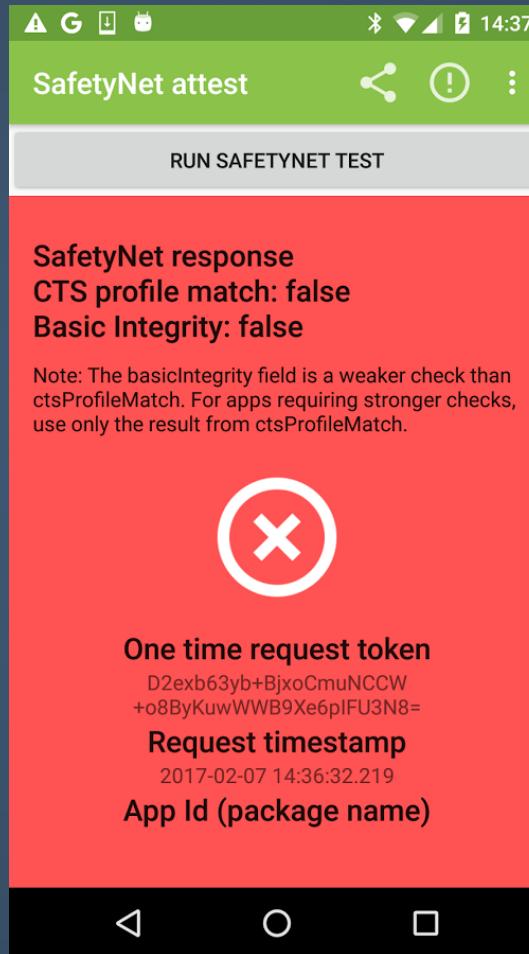
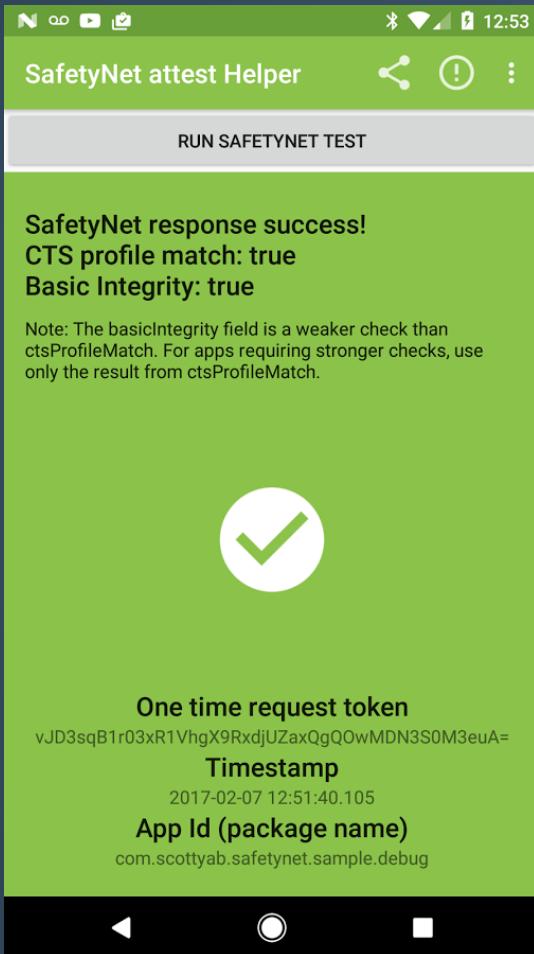


# The Leftovers

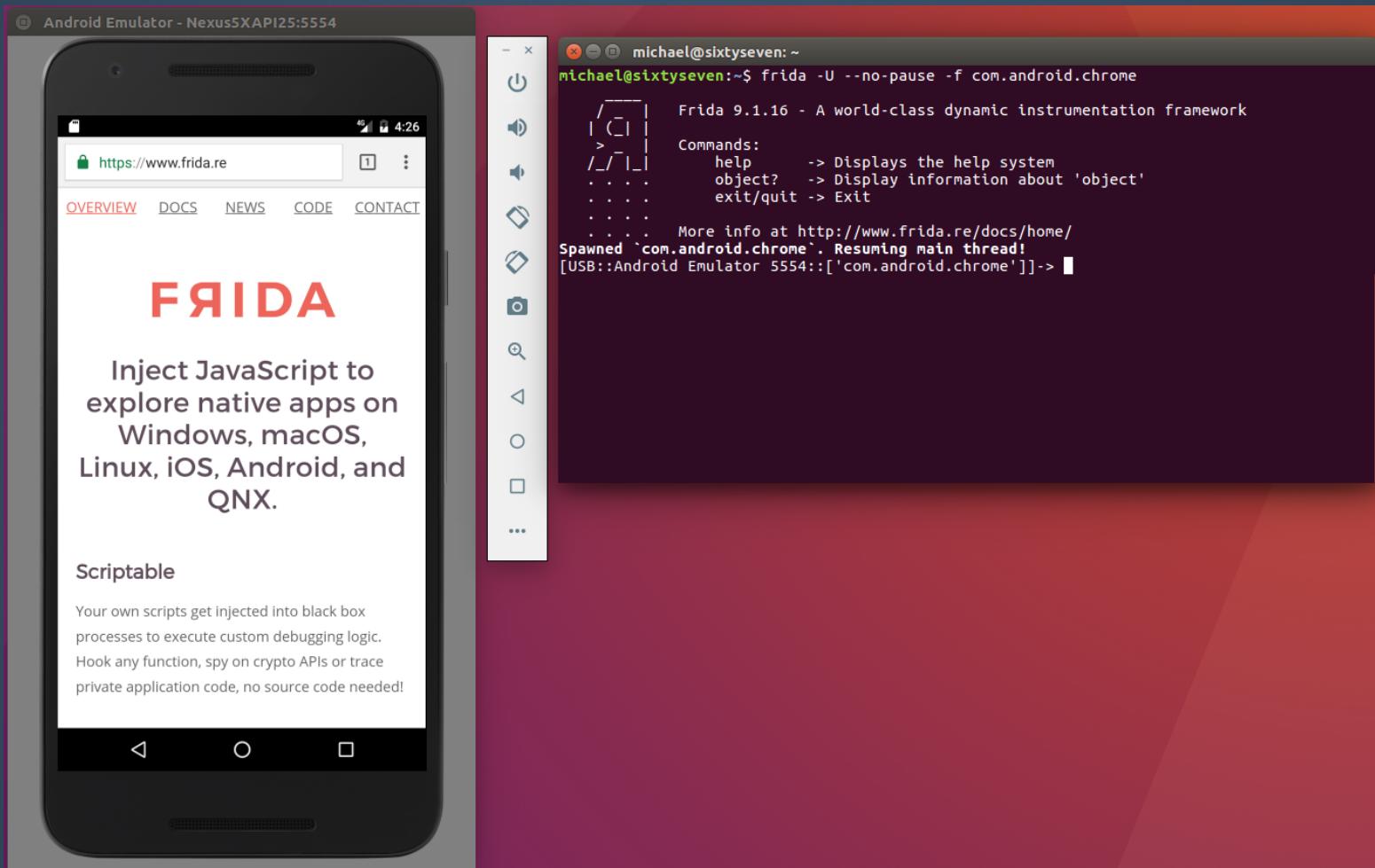
```
byte[] nonce = getRequestNonce(); // Should be at least 16 bytes in length.  
SafetyNet.SafetyNetApi.attest(mGoogleApiClient, nonce)  
    .setResultCallback(new ResultCallback<SafetyNetApi.AttestationResult>() {  
  
    @Override  
    public void onResult(SafetyNetApi.AttestationResult result) {  
        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/1hleNNi> - 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](http://slides.com/godfreynolan/bulletproofandroidmeetup)

