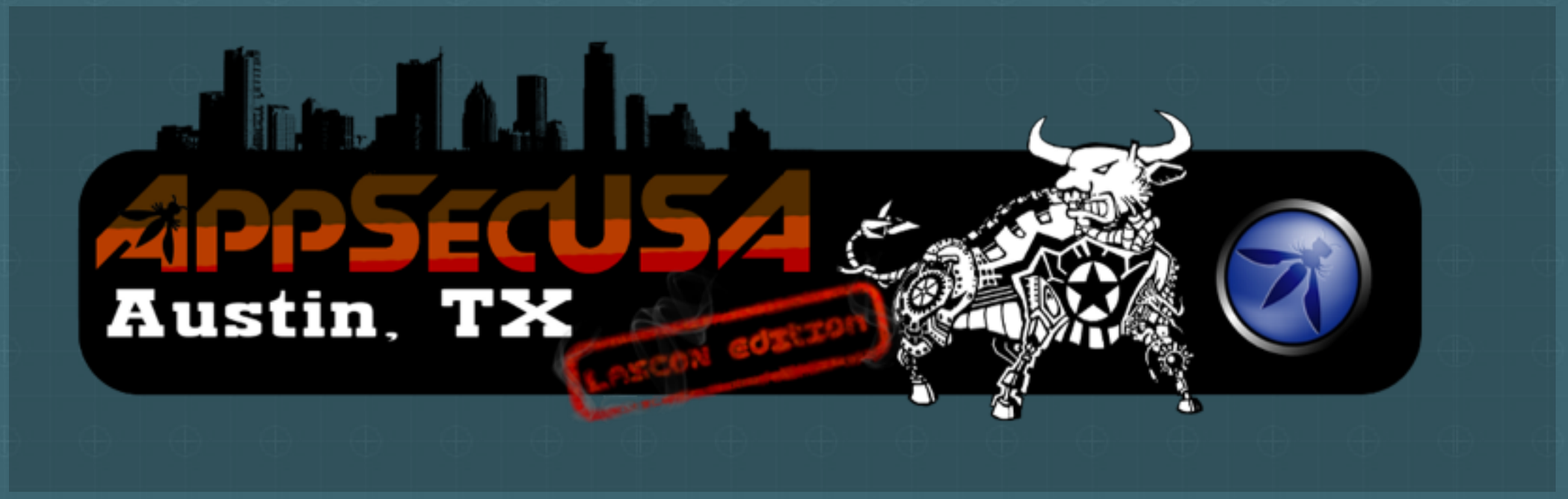


How we tear into that little green man

matasano SECURITY

Who are you?!

Mathew Rowley (@wuntee)

Senior security consultant at Matasano

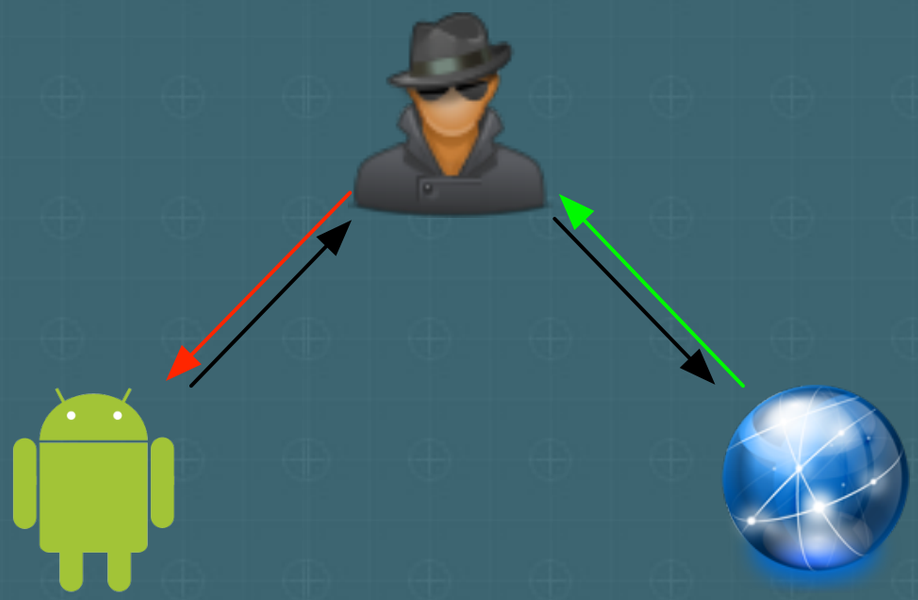
Agenda

- Techniques
  - MITM - SSL
  - Static analysis -> Skype secret menu
  - Modifying an app -> Injecting Java to APK
  - Manual debugging without source
  - Automated debugging (NEW TOOLS: JavaTap, CryptoTap)

matasano SECURITY

technique one

MITM - SSL



We need to trick the android device to think that the malicious user is “http://somesite.com"

That is not good enough though

Android HTTPS communication will fail

Tools to help with SSL MITM:

- Mallory
- Burp

matasano SECURITY

SSL

- By default SSL certificate checking is performed by the Android OS
- Optionally, developers can add or override those checks with their own
- This presents two problems when attempting to MITM an SSL communication stream
  - How do we trick the OS to trust endpoint
  - How can we get around custom developer checks

Circumventing OS checks

- Install the CA.
- There is no interface for doing this.
- Demos...

Before ICS

```
$ cd $JAVA_HOME/lib/ext
$ sudo wget 'http://www.bouncycastle.org/download/bcprov-jdk15on-147.jar'
$ cd -
$ adb pull /system/etc/security/cacerts.bks
$ keytool -keystore cacerts.bks -storetype BKS -provider
org.bouncycastle.jce.provider.BouncyCastleProvider -storepass changeit -importcert -
trustcacerts -alias idontcare -file ca.cer
$ adb shell mount | grep system
$ adb shell mount -o remount,rw /dev/block/mtdblock0 /system
$ adb shell chmod 777 /system/etc/security/cacerts.bks
$ adb push carerts.bks /system/etc/security/cacerts.bks
$ adb shell chmod 644 /system/etc/sercurity/cacerts.bks
```

ICS+

```
$ adb shell mount -o remount,rw /dev/block/
mtdblock0 /system
```

```
$ openssl x509 -in [cert] -text -noout >
cert.txt
```

```
$ adb push cert.txt /system/etc/security/
cacerts/
```

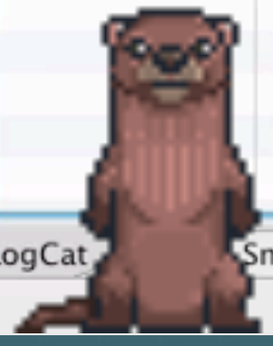
Otertool File LogCat FsDiff Smali Java to Smali Tools

Otertool

Message Filter:

Debug Info Warn Error Verbose Auto-scroll Clear Start Stop

Timestamp	Level	Class	PID	Message



LogCat Smali Java to Smali Package Manager

matasano SECURITY

Caveats

- Emulator
  - Does not persist changes to the .img file. Bug reported, was supposed to be fixed in r17 (still not fixed - android 4.1.2)
    - Bug or feature?
    - Can be done with unyaffs/mkyaffs2image
- Ice Cream Sandwitch
  - Switched from BouncyCastle to a directory of x509 text certificates

Persistent /system workaround

- The current /system image is:
  - /tmp/android-[user]/emulator-[rand]
- Make changes
- Before powering off, copy that somewhere
- emulator -system [copy] @[avd]

matasano SECURITY

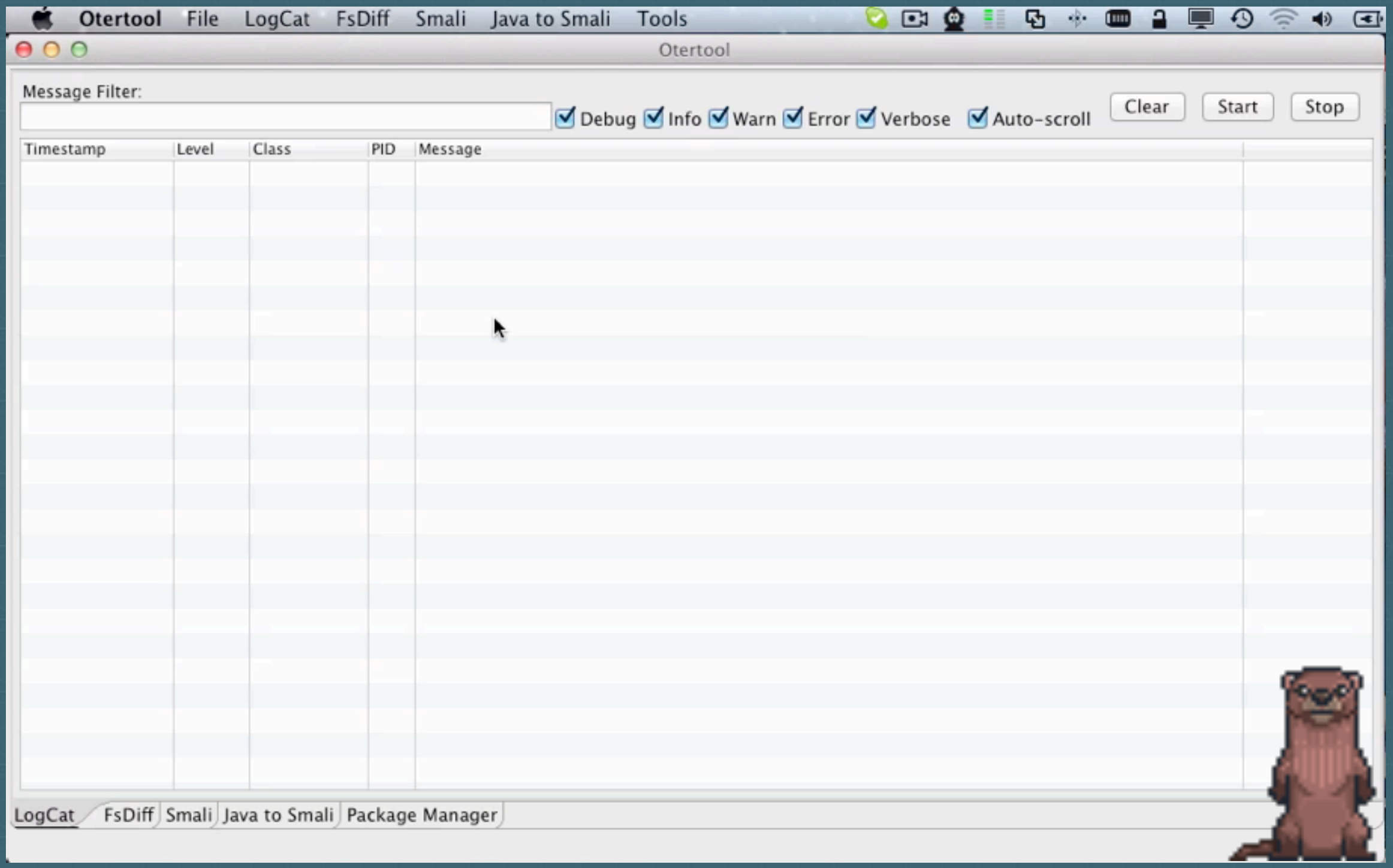
Preventative Measures

- Custom certificate checks
  - This is done by extending X509TrustManager
    - (void) checkServerTrusted(...)
- Can we get around this? Yes...
  - Baksmali
  - Modify smal source
  - re-smali
  - re-package
  - re-sign

For Reference.
But it must be easier...

```
$ java -jar apktool.jar d X509ModfiedSSL.apk
$ grep -ri x509 * | grep -i implements
-- Edit file to have 'checkServerTrusted' return void
$ java -jar apktool.jar b X509ModifiedSSL X509ModifiedSSL-mod.apk
$ keytool -genkey -v -keystore keystore.ks -alias idontcare -keyalg RSA -
keysize 2048 -validity 10000
$ jarsigner -keystore keystore.ks X509ModifiedSSL-mod.apk idontcare
$ adb install X509ModifiedSSL-mod.apk
```

matasano SECURITY



matasano SECURITY

I have SSL MITM, now what?

- This will typically be used to attack backend infrastructure, not the device
- Backend servers typically expect sanitized input, and dont expect non-device interaction.
- What I have seen:
  - 101 read/write of arbitrary files through SOAP interface
  - XML entity inclusion to gain access to private ssh key
  - Ability to arbitrarily lock/wipe another persons device

How to prevent MITM?

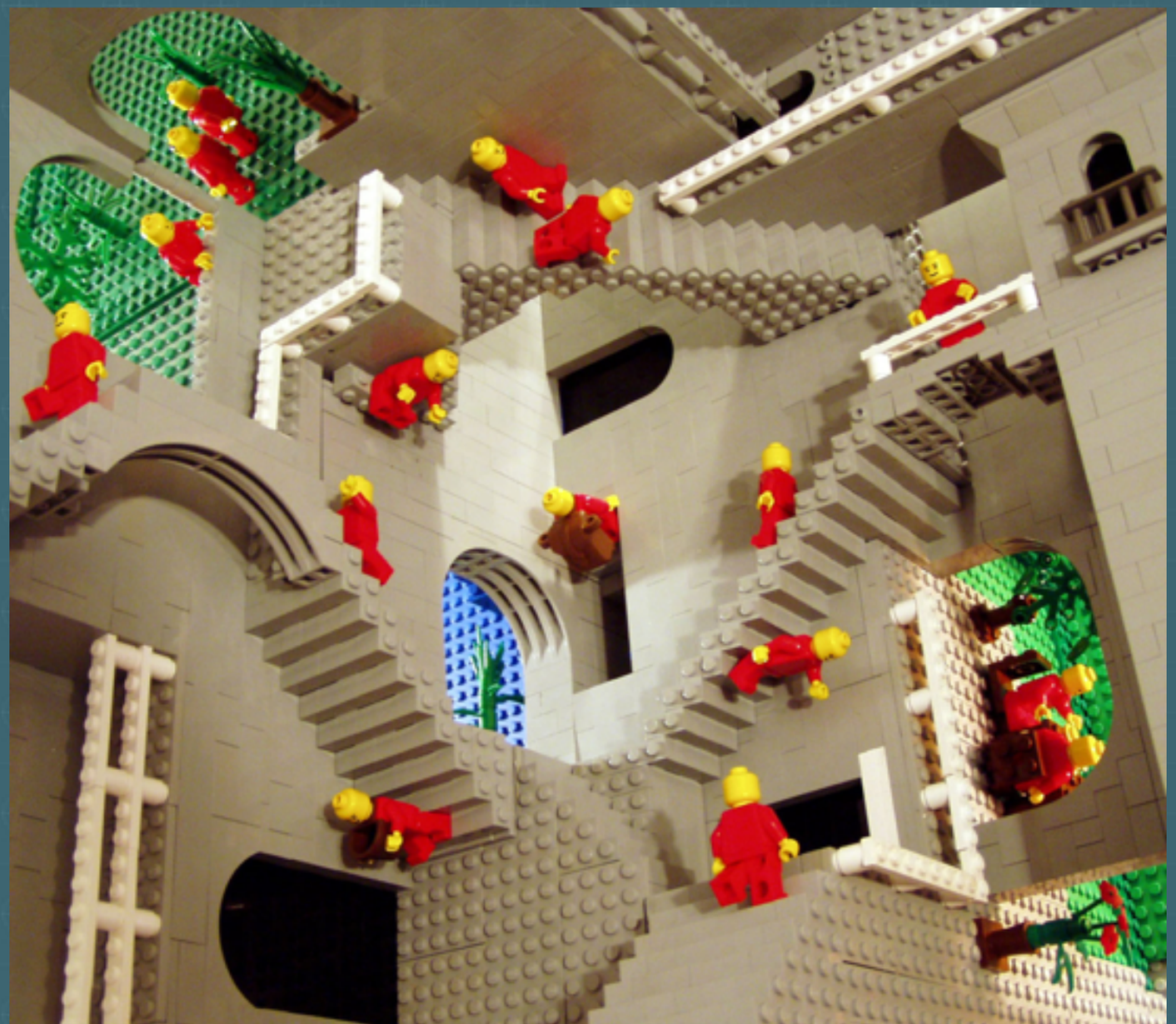
- You can't... Remember the device is out of your trust zone. ALWAYS treat user input as untrusted
- But, we can make it really hard
  - Real obfuscation - means obfuscating strings
  - Custom certificate checks
  - Authenticate client app - The application has access to itself (APK), send a custom hash of it for each request

matasano SECURITY

technique two

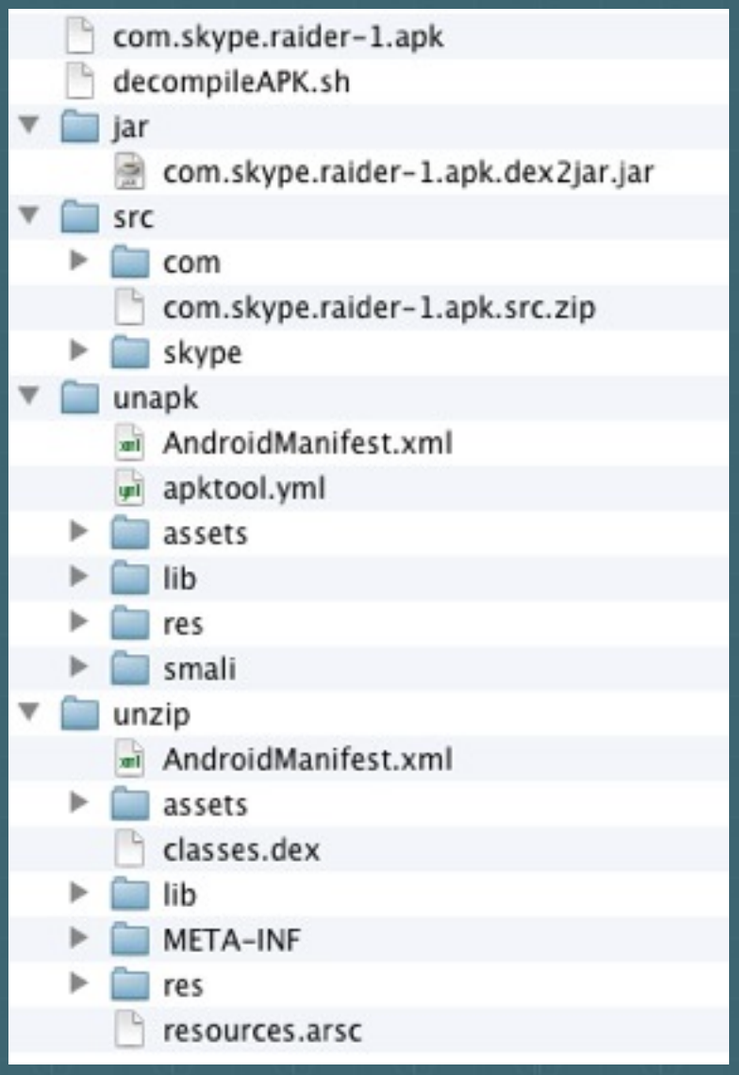
Static analysis

It’s a skill



Project Setup

- jar - dex2jar(*.apk)
- src - jdgui(*.jar)
- unapk - baksmali(*.apk)
- unzip - unzip(*.apk)



Reference Script

```
#!/bin/bash
# Scripts
apktool="/Applications/hacking/apktool-install-macosx-r04-brut1/apktool"
dex2jar="/Applications/hacking/dex2jar-0.0.9.7/dex2jar.sh"
jdgui="/Applications/JD-GUI.app/Contents/MacOS/jd-gui"
apk=$1
mkdir jar src unapk unzip
# Unzip
pushd unzip
unzip  ../$apk
popd
# Unapk
pushd unapk
$apktool d -f ../$apk .
popd
# Get jar
$dex2jar $apk
mv *.jar ./jar
# Decompile
$jdgui jar/*.jar
```

Example
Skypes hidden menu

- Logcat... What is skype.properties?

12:01:16:807	info	ActivityManager	61	Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.c
12:01:17:0	info	ActivityManager	61	Start proc com.skype.raider for activity com.skype.raider/.Main: pid=3
12:01:18:609	info	ActivityThread	347	Pub com.skype.SearchProvider: com.skype.SearchProvider
12:01:18:621	info	com.skype.raider.MainApp	347	onCreate +
12:01:18:621	info	com.skype.raider.MainApp	347	DEFAULTING TO PRODUCTION CONFIGUATION SETTINGS
12:01:18:664	info	com.skype.raider.MainApp	347	onCreate -
12:01:18:677	verbose	com.skype.ni	347	lib:/data/data/com.skype.raider/lib/libvideohost_skype.so
12:01:18:704	verbose	com.skype.ni	347	lib:/data/data/com.skype.raider/lib/libpcmhost_skype.so
12:01:18:730	warn	com.skype.mt	347	No config file name:/mnt/sdcard/skype.properties
12:01:18:762	warn	com.skype.mt	347	No config file name:/sdcard/skype.properties
12:01:18:788	warn	com.skype.mt	347	No config file name:/mnt/sdcard/skype.properties
12:01:18:833	warn	com.skype.mt	347	No config file name:skype.properties
12:01:18:861	warn	com.skype.qp	347	data class:com.skype.LiveData
12:01:18:953	info	com.skype.raider.MainApp	347	cert: raider-2.0-market-live.cert
12:01:19:119	debug	dalvikvm	347	GC_EXTERNAL_ALLOC freed 331K, 48% free 3255K/6215K, external 1
12:01:19:281	debug	dalvikvm	347	GC_EXTERNAL_ALLOC freed 20K, 48% free 3278K/6215K, external 26
12:01:19:395	debug	FlurryAgent	347	Starting new session
12:01:19:412	warn	Settings	347	Setting android_id has moved from android.provider.Settings.System t
12:01:19:453	warn	com.skype.DefaultUiNavig...	347	foreground() not calling show() because view stack is empty
12:01:19:482	verbose	com.skype.kit.dy	347	CONNECTIVITY_ACTION connected:true

Locate string "skype.properties"

- grep is your friend
- Notice how it is only located in .smali files
- What about .java?

```
[16:06:29]wuntee:~/matasano/tmp$ ls -l
total 20128
-rw-r--r--  1 wuntee  staff  10297777 Apr 18 16:05 com.skype.raider-1.apk
-rwxr-xr-x  1 wuntee  staff       424 Apr 18 16:05 decompileAPK.sh
drwxr-xr-x  3 wuntee  staff       102 Apr 18 16:05 jar
drwxr-xr-x  5 wuntee  staff       170 Apr 18 16:06 src
drwxr-xr-x  8 wuntee  staff       272 Apr 18 16:05 unapk
drwxr-xr-x  9 wuntee  staff       306 Apr 18 16:05 unzip
[16:06:30]wuntee:~/matasano/tmp$ grep -ri skype.properties *
unapk/smali/com/skype/mt.smali:    const-string v2, "skype.properties"
unapk/smali/com/skype/mt.smali:    const-string v1, "/sdcard/skype.properties"
unapk/smali/com/skype/mt.smali:    const-string v2, "/mnt/sdcard/skype.properties"
unapk/smali/com/skype/mt.smali:    const-string v2, "skype.properties"
Binary file unzip/classes.dex matches
```

matasano SECURITY

```
.end method

.method public static final a(Landroid/app/Application;)V
    .locals 11

    move-result-object v1

    const-string v2, "skype.properties"

    invoke-virtual {v1, v2}, Ljava/lang/StringBuilder;->append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    move-result-object v1

    invoke-virtual {v1}, Ljava/lang/StringBuilder;->toString()Ljava/lang/String;

    move-result-object v1

    aput-object v1, v0, v7

    const-string v1, "/sdcard/skype.properties"

    aput-object v1, v0, v8

    const/4 v1, 0x2

    const-string v2, "/mnt/sdcard/skype.properties"

    aput-object v2, v0, v1
```

```
    String str4 = I;
    StringBuilder localStringBuilder33 = localStringBuilder32.append(str4);
    StringBuilder localStringBuilder34 = localStringBuilder1.append(" kitLogging:");
    boolean bool11 = o;
    StringBuilder localStringBuilder35 = localStringBuilder34.append(bool11);
    return localStringBuilder1.toString();
  }

  // ERROR //
  public static final void a(android.app.Application paramApplication)
  {
    // Byte code:
    //   0: ldc 2
    //   2: invokevirtual 224   java/lang/Class:getName ()Ljava/lang/String;
    //   5: invokestatic 229    com/skype/tj:a  (Ljava/lang/String;)Z
    //   8: ifeq +44 -> 52
    //   11: ldc 2
    //   13: invokevirtual 224  java/lang/Class:getName ()Ljava/lang/String;
    //   16: astore_1
    //   17: new 153    java/lang/StringBuilder
    //   20: dup
    //   21: invokespecial 230  java/lang/StringBuilder:<init>  ()V
    //   24: ldc 232
    //   26: invokevirtual 164  java/lang/StringBuilder:append  (Ljava/lang/String;)Ljava/lang/StringBuilder;
    //   29: astore_2
    //   30: invokestatic 238   android/os/Environment:getExternalStorageDirectory      ()Ljava/io/File;
    //   33: astore_3
    //   34: aload_2
    //   35: aload_3
```

matasano SECURITY

skype.properties processing

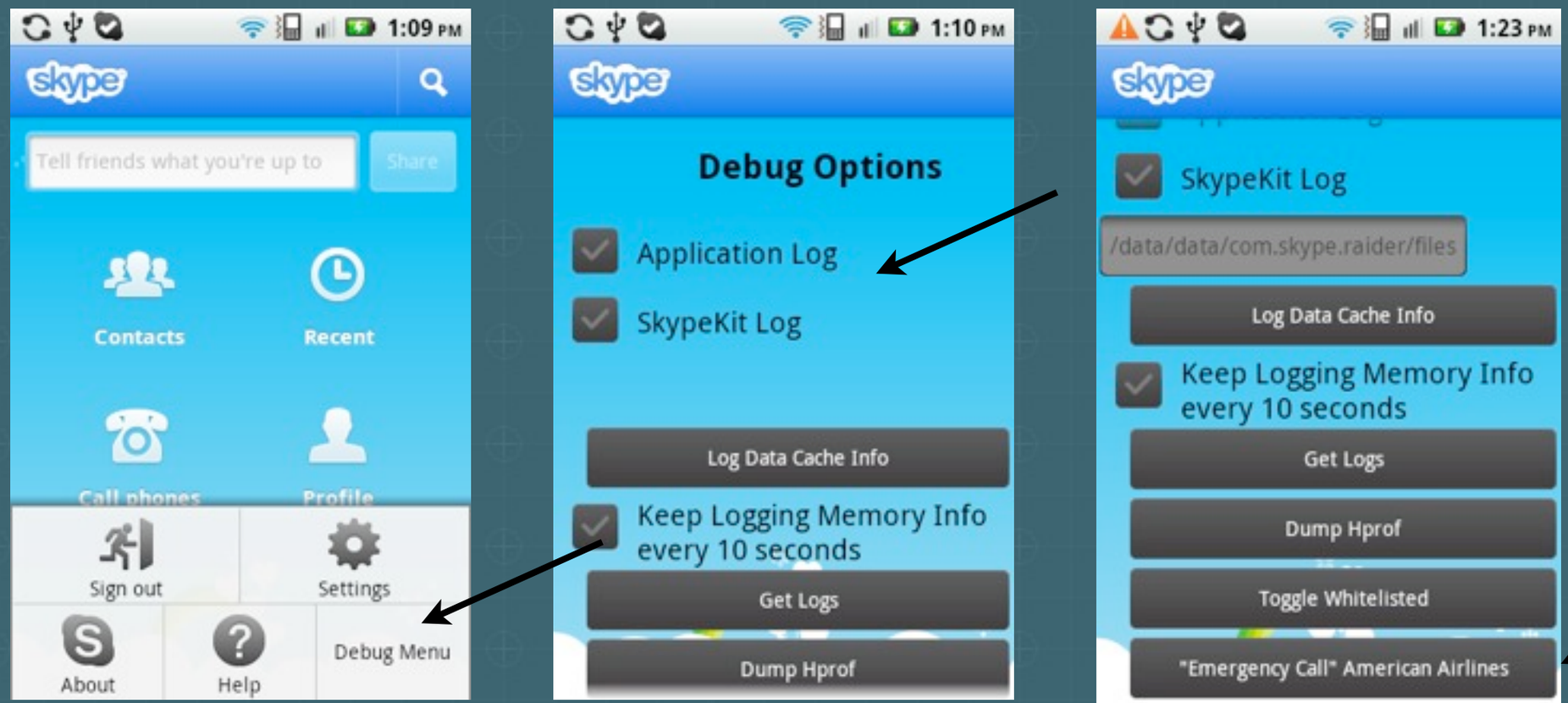
```
:try_start_ce
const-string v4, "login"
invoke-virtual {v5, v4}, Ljava/util/Properties;->containsKey(Ljava/lang/Object;)Z
move-result v4
if-eqz v4, :cond_e7
const-string v4, "login"
invoke-virtual {v5, v4}, Ljava/util/Properties;->getProperty(Ljava/lang/String;)Ljava/lang/String;
move-result-object v4
const-string v6, "0"
invoke-virtual {v4, v6}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z
move-result v4
if-nez v4, :cond_287
move v4, v8
:goto_e5
sput-boolean v4, Lcom/skype/mt;->z:Z
:cond_e7
const-string v4, "daemon"
invoke-virtual {v5, v4}, Ljava/util/Properties;->containsKey(Ljava/lang/Object;)Z
move-result v4
if-eqz v4, :cond_100
const-string v4, "daemon"
invoke-virtual {v5, v4}, Ljava/util/Properties;->getProperty(Ljava/lang/String;)Ljava/lang/String;
move-result-object v4
const-string v6, "0"
invoke-virtual {v4, v6}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z
```

```
login=Boolean
daemon=Boolean
update=Boolean
answer=Boolean
callVoiceMailDelay=Integer
videoQualityLow=Integer
userWantsVideo=Boolean
checkSharedXML=Boolean
kit.logging=Boolean
debugMenu=Boolean
test.monkey.enabled=Boolean
test.acs.enabled=Boolean
test.video=Boolean
test.skypename=String
test.password=String
videoInfo=String
```

matasano SECURITY

The test...

- Create skype.properties with "debugMenu=1" and place it on SD card



matasano SECURITY

Logging...

```
09-21 13:13:02.872  5166  5172 V com.skype.live.aa: skype account getAvatarImage +
09-21 13:13:02.880  5166  5172 V com.skype.live.aa: skype account getAvatarImage 2ms -
09-21 13:13:02.880  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable 3ms -
09-21 13:13:02.888  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable +
09-21 13:13:02.888  5166  5172 V com.skype.live.aa: skype account getAvatarImage +
09-21 13:13:02.888  5166  5172 V com.skype.live.aa: skype account getAvatarImage 2ms -
09-21 13:13:02.888  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable 3ms -
09-21 13:13:02.903  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable +
09-21 13:13:02.903  5166  5172 V com.skype.live.aa: skype account getAvatarImage +
09-21 13:13:02.903  5166  5172 V com.skype.live.aa: skype account getAvatarImage 2ms -
09-21 13:13:02.903  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable 2ms -
09-21 13:13:02.950  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable +
09-21 13:13:02.950  5166  5172 V com.skype.live.aa: skype account getAvatarImage +
09-21 13:13:02.950  5166  5172 V com.skype.live.aa: skype account getAvatarImage 2ms -
09-21 13:13:02.950  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable 2ms -
09-21 13:13:02.958  5166  5166 V com.skype.ui.widget.x: refreshing active views
09-21 13:13:02.958  5166  5166 V com.skype.ui.widget.y: UI THREAD:total content height finished 92ms -
09-21 13:13:02.966  5166  5166 V com.skype.ui.aa: UI THREAD:update chat content +
09-21 13:13:02.966  5166  5172 V com.skype.ui.widget.EnhancedListView: NON-UI THREAD:compute total content height +
09-21 13:13:02.966  5166  5172 V com.skype.ui.widget.y: content list height for 6 elements (481) computed in 0ms
09-21 13:13:02.966  5166  5172 V com.skype.ui.widget.EnhancedListView: NON-UI THREAD:compute total content height 0ms -
09-21 13:13:02.966  5166  5166 V com.skype.ui.aa: UI THREAD:update chat content 2ms -
09-21 13:13:03.013  5166  5166 V com.skype.ui.widget.y: UI THREAD:total content height finished +
09-21 13:13:03.020  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable +
09-21 13:13:03.020  5166  5172 V com.skype.live.aa: skype account getAvatarImage +
09-21 13:13:03.020  5166  5172 V com.skype.live.aa: skype account getAvatarImage 3ms -
09-21 13:13:03.020  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable 3ms -
09-21 13:13:03.028  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable +
09-21 13:13:03.028  5166  5172 V com.skype.live.aa: skype account getAvatarImage +
09-21 13:13:03.028  5166  5172 V com.skype.live.aa: skype account getAvatarImage 2ms -
09-21 13:13:03.028  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable 2ms -
09-21 13:13:03.036  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable +
09-21 13:13:03.036  5166  5172 V com.skype.live.aa: skype account getAvatarImage +
09-21 13:13:03.044  5166  5172 V com.skype.live.aa: skype account getAvatarImage 2ms -
09-21 13:13:03.044  5166  5172 V com.skype.kit.cd: NON-UI THREAD:lazy load account avatar drawable 3ms -
09-21 13:13:03.052  5166  5221 V com.skype.live.at: IPC-THREAD Message onPropertyChange +
09-21 13:13:03.075  5166  5221 V com.skype.kit.er: notify str:WatchEvents.CHAT_INFO_CHANGED +
09-21 13:13:03.075  5166  5221 V com.skype.kit.ap: WatchEvents.CHAT_INFO_CHANGED
09-21 13:13:03.075  5166  5221 V com.skype.live.aj: skype message getBodyXml +
09-21 13:13:03.075  5166  5221 V com.skype.live.aj: skype message getBodyXml 3ms -
09-21 13:13:03.083  5166  5221 V com.skype.live.aj: skype message getConversationId +
09-21 13:13:03.083  5166  5221 V com.skype.live.aj: skype message getConversationId 4ms -
09-21 13:13:03.083  5166  5221 V com.skype.live.aj: skype message getGuid +
09-21 13:13:03.091  5166  5221 V com.skype.live.aj: skype message getGuid 7ms -
09-21 13:13:03.098  5166  5221 V com.skype.kit.DataCache: _getConversationMessage id:mathew_rowley msg_guid:b3c341430681788a1a48e027c16d53ccc6b551c793dfd35d675bae052df42
44e +
09-21 13:13:03.106  5166  5221 V com.skype.kit.DataCache: _getConversationMessage id:mathew_rowley msg_guid:b3c341430681788a1a48e027c16d53ccc6b551c793dfd35d675bae052df42
44e 4ms -
09-21 13:13:03.106  5166  5221 V com.skype.live.aj: skype message getSendingStatus +
09-21 13:13:03.106  5166  5221 V com.skype.live.aj: skype message getSendingStatus 4ms -
```

Preventing Static Analysis/ Decompilation

- You can't...
- But, we can make it really hard
- Real obfuscation - means obfuscating strings
- Disable logging
- Cause decompilers to crash
  - Blackhat 2012 - Practicing safe DEX [1]
  - APKfuscator [2]

[1]http://www.strazzere.com/papers/DexEducation-PracticingSafeDex.pdf
[2]https://github.com/strazzere/APKfuscator

HackersChallenge...

HackersChallenge...

- Challenge for Matasano coworkers - nobody has completed it
- Performs an HTTPS request to get a "secret" of a character from the movie Hackers, then displays it
- Goal is to figure out everyones secret
- Preventative measures
  - Tamper proof - custom hash of APK
  - Custom X509TrustManager

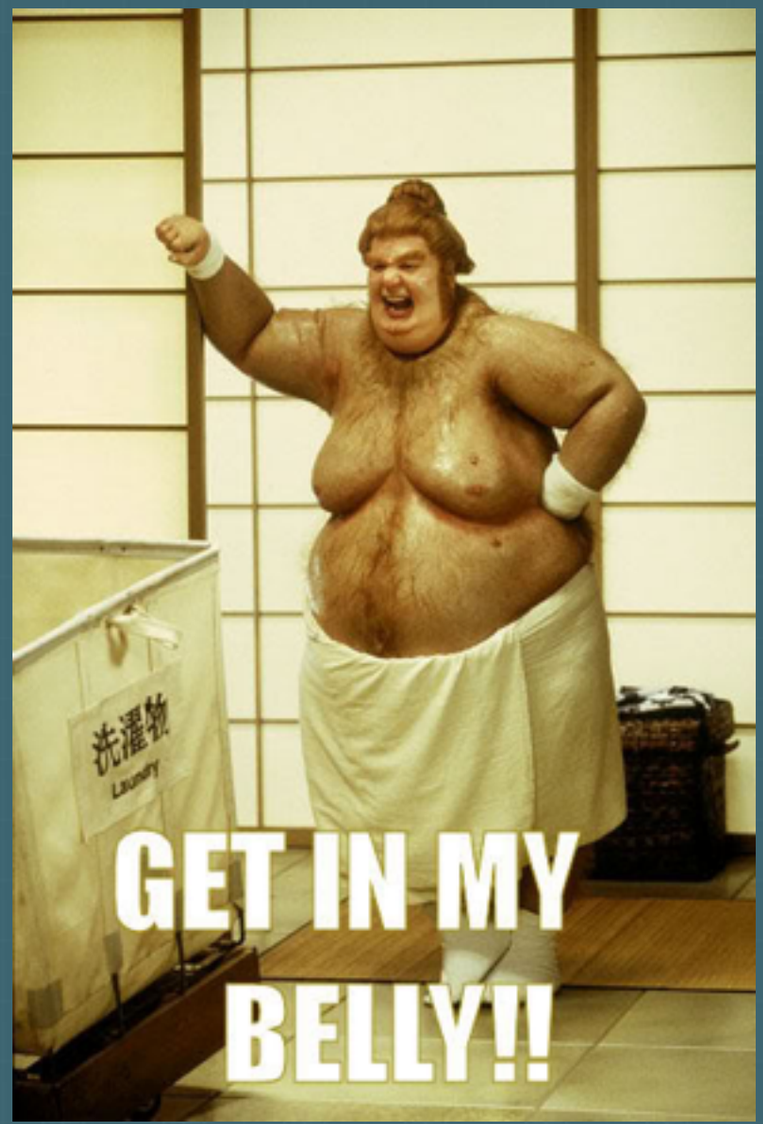
matasano SECURITY

technique three

Modifying an application

```
.field static staticString:Ljava/lang/String;
.method static constructor <clinit>()V
    .registers 1
    .prologue
    .line 8
    const-string v0, "static string"
    sput-object v0, Lcom/wuntee/Downloader;->staticString:Ljava/lang/String;
    .line 7
    return-void
.end method
.method public constructor <init>()V
    .registers 1
    .prologue
    .line 7
    invoke-direct {p0}, Landroid/app/Activity;-><init>()V
    return-void
.end method
# virtual methods
.method public onCreate(Landroid/os/Bundle;)V
    .registers 5
    .parameter "savedInstanceState"
    .prologue
    .line 11
    invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V
    .line 12
    const/high16 v1, 0x7f03
    invoke-virtual {p0, v1}, Lcom/wuntee/Downloader;->setContentView(I)V
    .line 13
    const-string v0, "testtest2"
    .line 14
    .local v0, anotherString:Ljava/lang/String;
    sget-object v1, Lcom/wuntee/Downloader;->staticString:Ljava/lang/String;
    invoke-virtual {v0, v1}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z
    move-result v1
    if-eqz v1, :cond_19
    .line 15
    sget-object v1, Lcom/wuntee/Downloader;->staticString:Ljava/lang/String;
    sget-object v2, Lcom/wuntee/Downloader;->staticString:Ljava/lang/String;
    invoke-static {v1, v2}, Landroid/util/Log;->d(Ljava/lang/String;Ljava/lang/String;)I
    .line 17
    :cond_19
    return-void
.end method
```

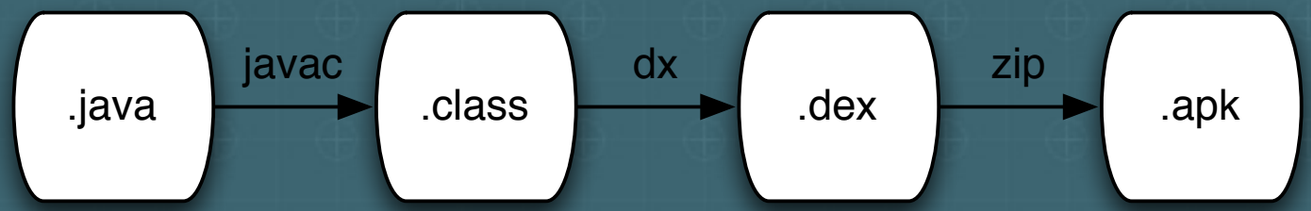
```
public static void logMap(Map map){
    for(Object key : map.keySet()){
        Object val = map.get(key);
        Log.e("WUNTEE", key.toString() + "=" + val.toString());
    }
}
```



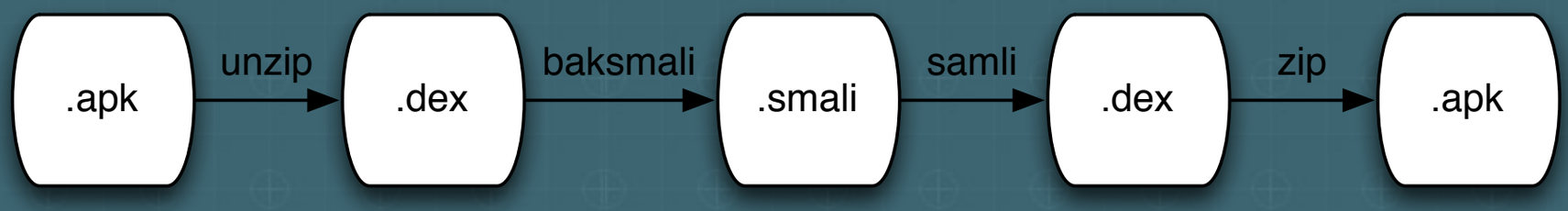
matasano SECURITY

Logic

Normal compilation path:

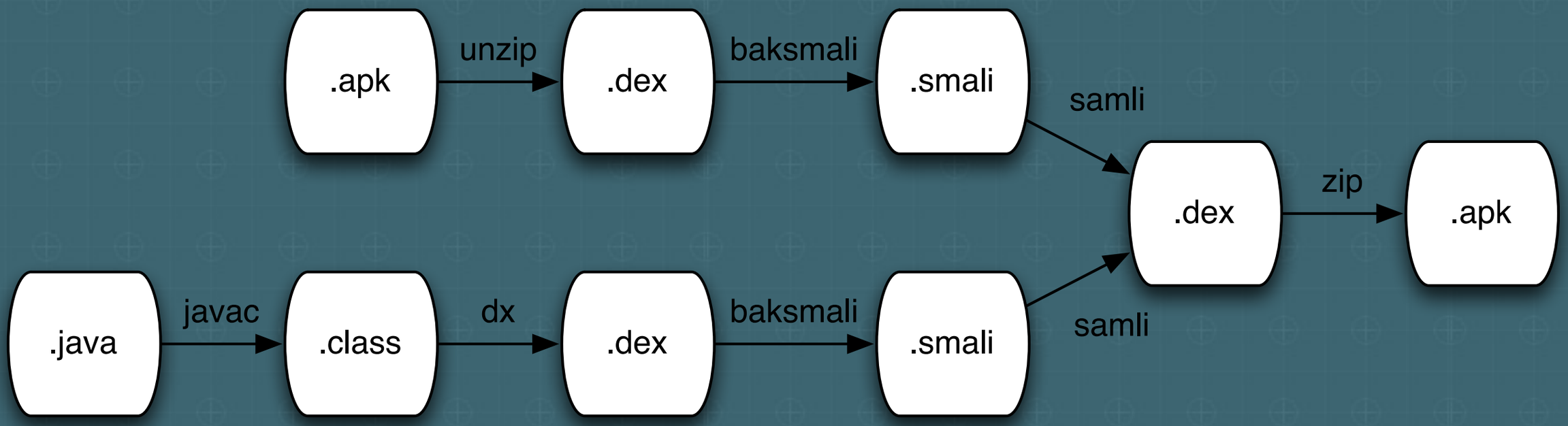


Simple modification path:



matasano SECURITY

Why cant I just...



You can. However...

- Nuances
  - Smali calling convention
  - Namespaces
- Demo (HackersChallenge - print hash algorithm output)...

emulator-arm Window

5554:HackersChallengeAvd

3G 1:22

HackersChallengeAndroid

Get zero cools secret

MENU

o-scroll Clear Start Stop

Thursday, October 25, 12

So what?

- Simple example of injecting arbitrary Java into an Android application
- Malware?
- Debugging obfuscation?
- Control circumvention aka “cracks”?
- You can inject any Java source into ANY application!

technique four

Debugging applications without source

Debugging

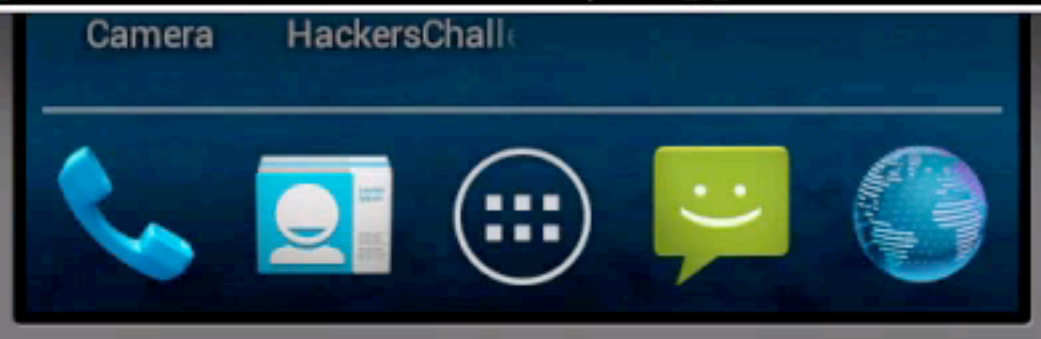
- System.out.println(debug_info) or attach to a debugger?
- Android applications are debugged via the typical Java Platform Debug Architecture (JPDA)
- Interacting with an Android application
  - Eclipse - GUI
  - JDB - command line
  - Java JPDA API
  - Ruby/JRuby with jdi-hook
- Caveats
  - You need to know “where” you must breakpoint
- Benefits
  - See method arguments and return values
  - HackersChallenge?
- Demo (HackersChallenge - obtain hash value without modification)...

iTerm Shell Edit View Profiles Toolbelt Window Help

2. screen

```
[11:19:42]wuntee:~/matasano/tmp$
```

```
[ wuntee-matasano-macbookpro ][          (0*bash)  1- bash          ][ 04-19 11:20 ]
```



Java

Type

Ok... And?

- Debugging is a key reversing concept
- Relate this to debugging using GDB or ImmunityDebugger/Ollydbg
- Not only can you break on local methods, but you can on core Android methods as well. Ever want to see all IPC?
  - Break: android.content.Intent.<init>

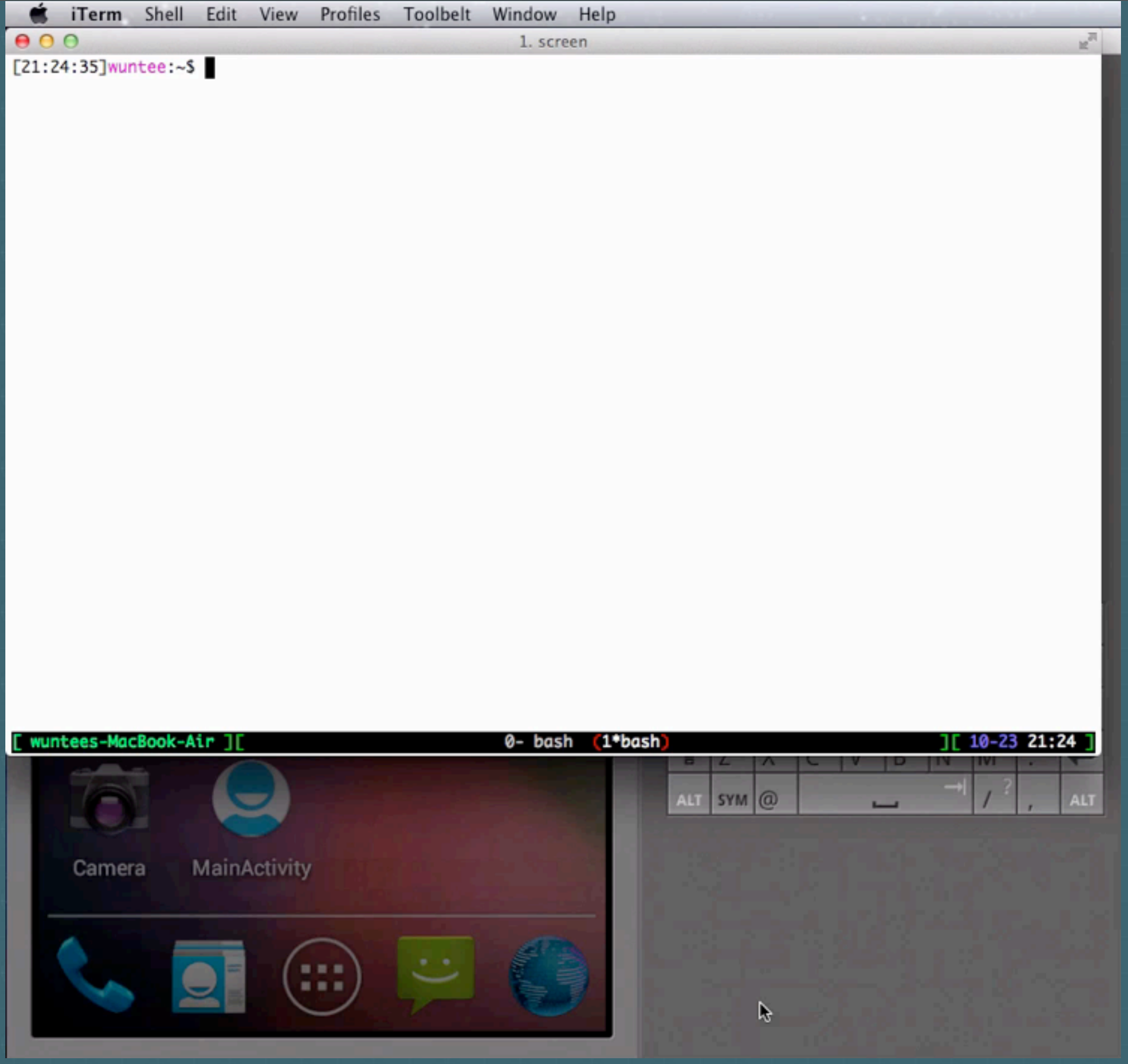
JavaTap/CryptoTap

- Originally part of OterTool
- Simplify debugging of an application
- CryptoTap
  - Original problem - pull AES keys from obfuscated application
  - Attach or launch a java process, and it will spit out all crypto info
- JavaTap
  - Config file defines entry/exit methods
  - Prints out passed in values and return values

matasano SECURITY

Live demo...

matasano SECURITY


iTerm Shell Edit View Profiles Toolbelt Window Help
1. screen
[21:24:35]wuntee:~$
[wuntees-MacBook-Air][0- bash (1*bash)][10-23 21:24]
Camera
MainActivity
ALT SYM @ ALT

Force Application to Wait for debugger

adb shell am set-debug-app -w [app]

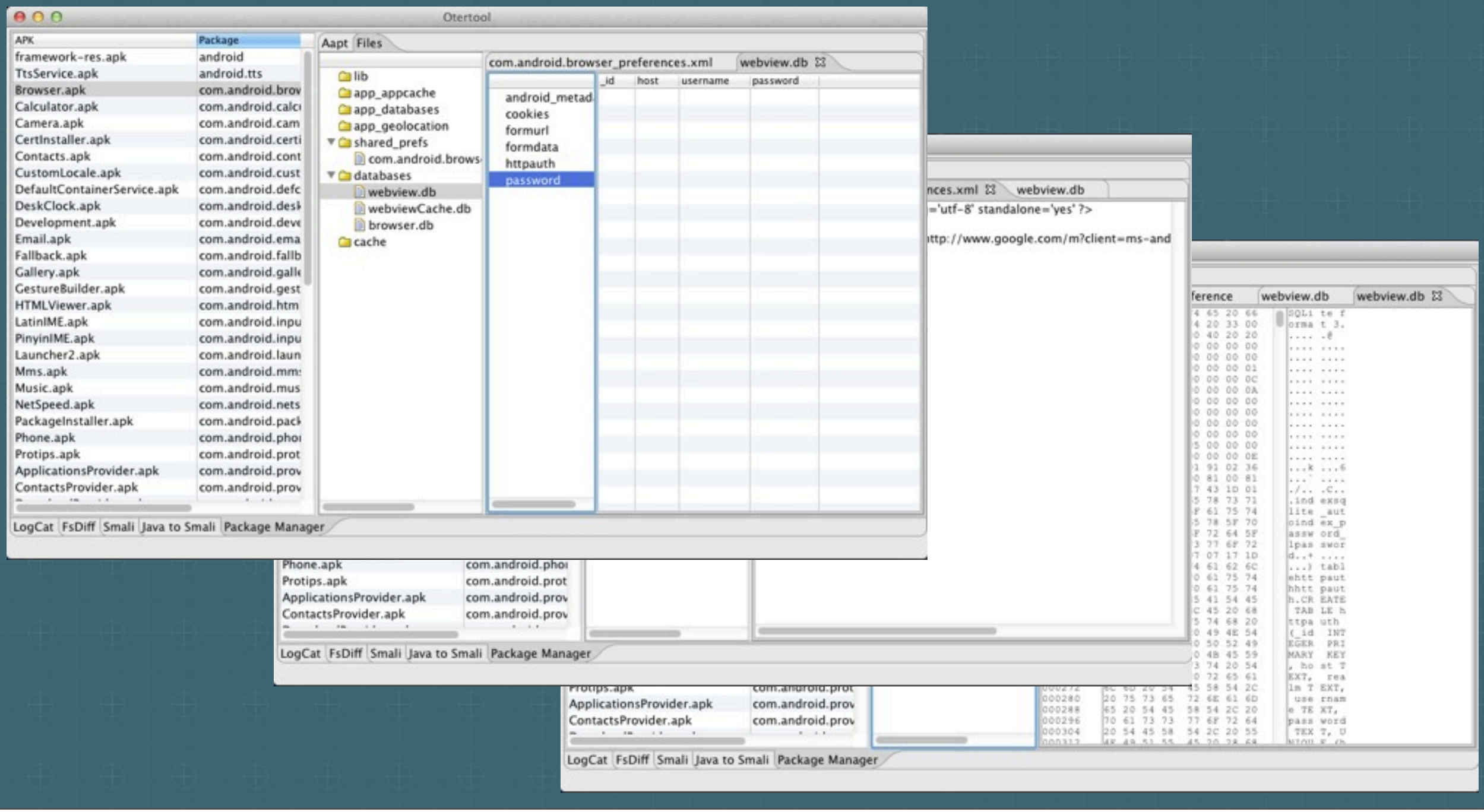
Prevention

- You can not
- AndroidManifest.xml - debuggable=false
  - What does this even do? I can still use ‘am’ to attach to it...
- If for some reason you can not attach, apktool can pull out the manifest, you can modify and re-packages

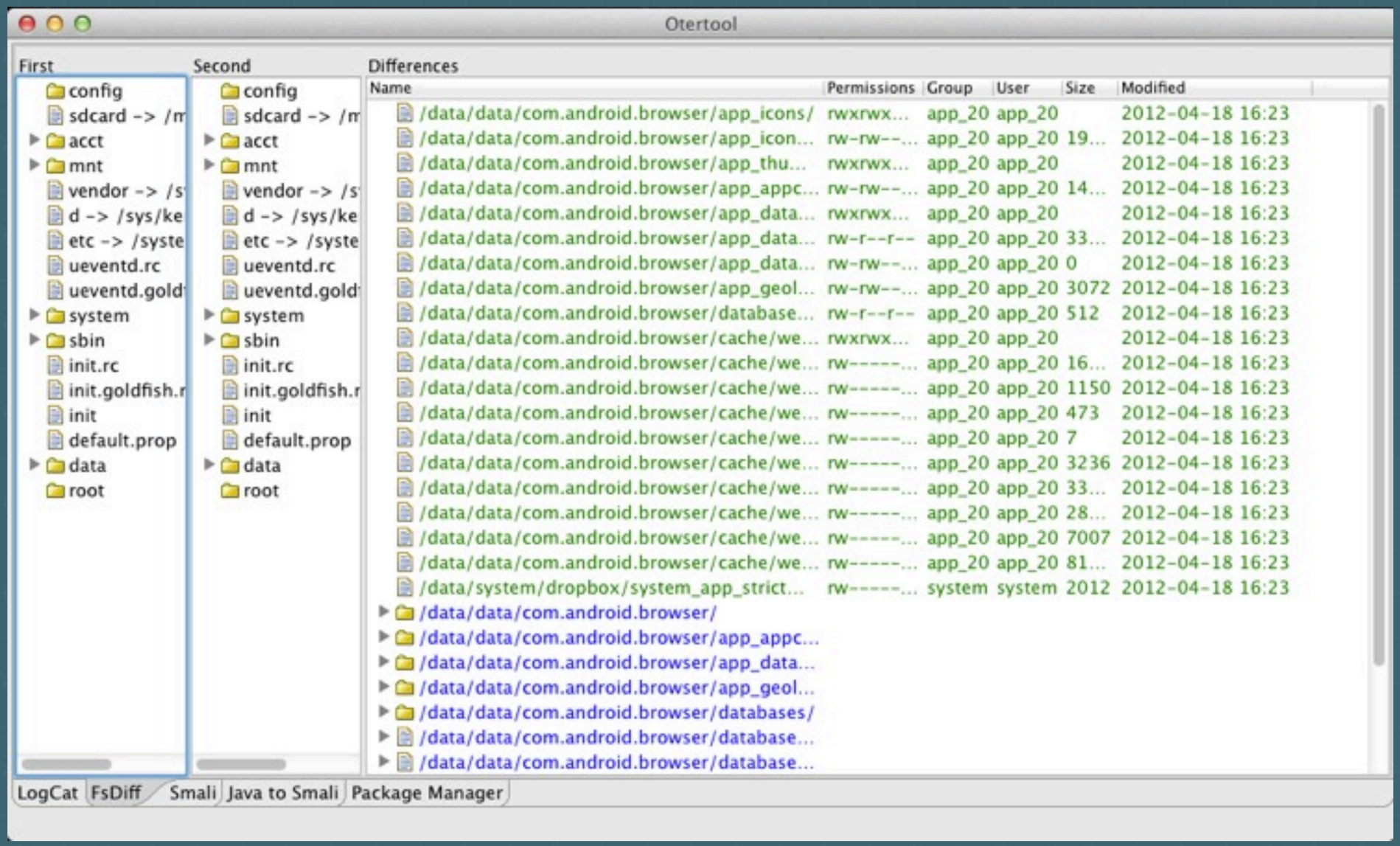
otertool

matasano SECURITY

Live File Browser
sqlite/text/hex

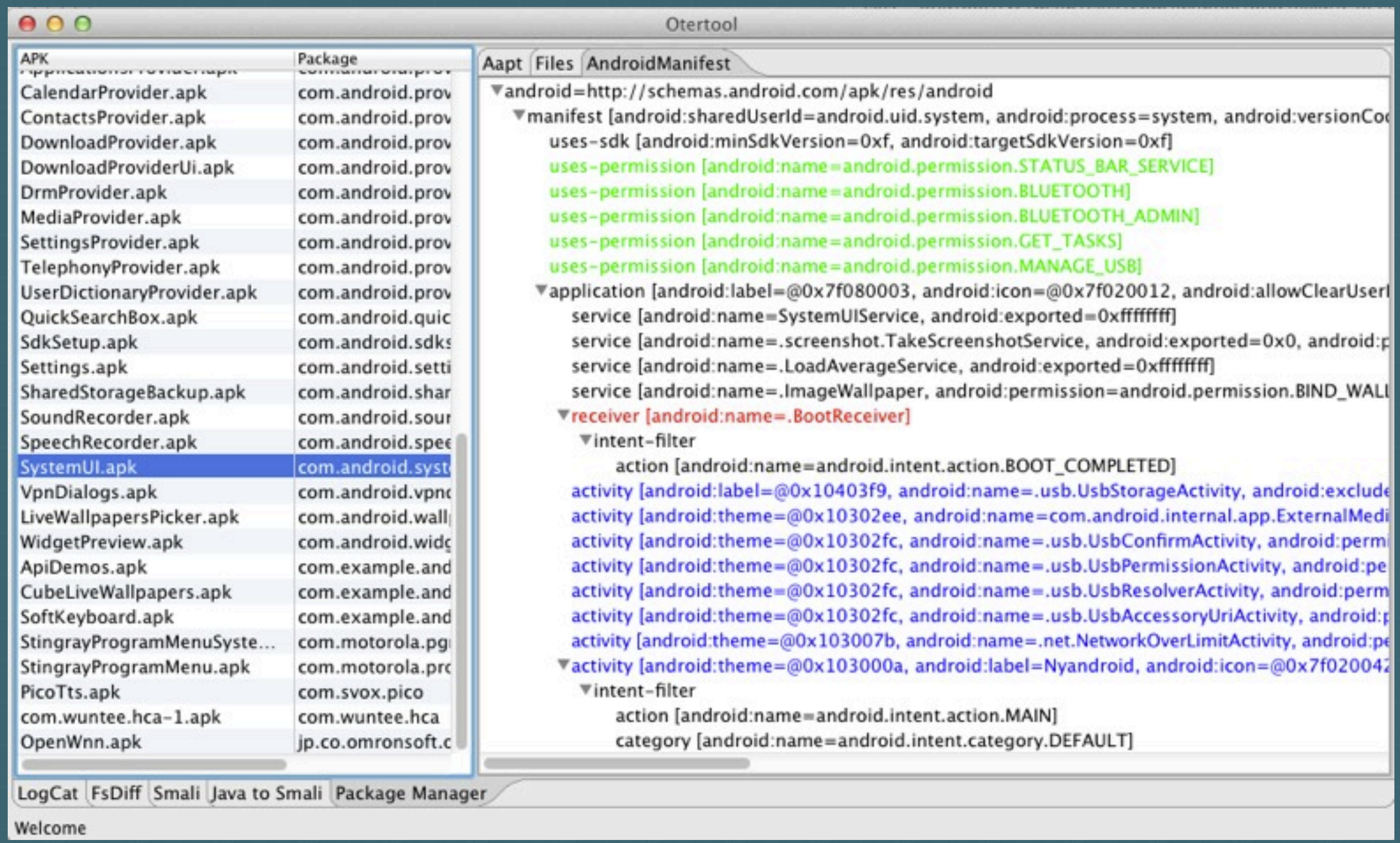


FSDiff



matasano SECURITY

Visual AndroidManifest.xml



matasano SECURITY

Questions?

mathew rowley (@wuntee)
mathew@matasano.com

otertool - https://github.com/wuntee/
CryptoTap - http://bit.ly/cryptotap

LOOKING FOR A JOB IN SECURITY CONSULTING?
(NYC, Chicago, Mountain View)
Send me your resume