

Leave The App Alone!

- Attack and Defense of Android App Hijack

SESSION ID: MBS-W05

Rongyu Zhou

Senior Research Engineer
Baidu Inc.



Outline

- ◆ Root or Not Root
- ◆ App Hijack
- ◆ Hook Insight
- ◆ Demo: App Hijack
- ◆ Detection & Fix for App Hijack
- ◆ Leave My App Alone – Create A Trusted App Runtime

RSAC CONFERENCE 2014
ASIA PACIFIC & JAPAN



Root or Not Root

Root or Not Root

- ◆ It's not the main topic I'd talk about today
- ◆ But it's indeed a simple question
 - ◆ Root: pre-installed, auto startup, customization, etc.
 - ◆ Not Root: unsafe!
- ◆ So, what we need is 'Safe Root'



Root Risks

- ◆ Android keeps each App's security by creating a different user for each App to distinguish permissions
- ◆ Each App could apply for Root permission
- ◆ After Root, your App could be accessed by others
 - ◆ Memory modifications
 - ◆ File access
 - ◆

```
u0_a5      936  181  889740 46828 ffffffff 00000000 S android.process.media
u0_a48     963  181  884940 47140 ffffffff 00000000 R com.google.android.inputmethod.pinyin
u0_a7      987  181  1018824 53324 ffffffff 00000000 S com.google.android.gms
radio     1002  181  886732 39532 ffffffff 00000000 S com.android.phone
radio     1014  181  868940 32092 ffffffff 00000000 S com.redbend.vdmc
nfc       1027  181  884840 36536 ffffffff 00000000 S com.android.nfc
u0_a20    1041  181  1045096 93236 ffffffff 00000000 S com.google.android.googlequicksearchbox
u0_a7     1153  181  902264 50052 ffffffff 00000000 S com.google.process.location
u0_a7     1175  181  911244 53672 ffffffff 00000000 S com.google.process.gapps
root      1259  1   7212   492   ffffffff 00000000 S /system/bin/mpdecision
dhcp      1391  1   1020   476   ffffffff 00000000 S /system/bin/dhccpd
u0_a101   1412  181  897728 56624 ffffffff 00000000 S com.lbe.security.service
u0_a101   1474  1412  1052   244   ffffffff 00000000 S lbsec.monitor
```

```

@hammerhead:/ $ su
hammerhead:/ # ps | grep chrome
02  1584  180  1156420 106048 ffffffff 4006773c S com.android.chrome
0  2370  180  1223960 71560 ffffffff 4006773c S com.android.chrome:sandboxed_process0
hammerhead:/ # cat /proc/1584/maps | grep /data/
0000-6f7dd000 r--s 00017000 b3:1c 82184 /data/data/de.robv.android.xposed.installer/bin/XposedBridge.jar
2000-71030000 r--p 00000000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
classes.dex
va-ADT 0000-71030000 r--p 0000e000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
lge.jar@classes.dex
0000-71041000 r--p 0000f000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
lge.jar@classes.dex
0000-71042000 r--p 0001f000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
lge.jar@classes.dex
2000-71070000 r--p 00020000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
lge.jar@classes.dex
0000-74d49000 r--s 01c7e000 b3:1c 162902 /data/app/com.android.chrome-1.apk
0000-7515f000 r--p 00000000 b3:1c 106021 /data/dalvik-cache/data@app@com.android.chrome-1.apk@classes.dex
2000-751d7000 r--s 01c7e000 b3:1c 162902 /data/app/com.android.chrome-1.apk
0000-753d5000 r--s 01a71000 b3:1c 162902 /data/app/com.android.chrome-1.apk
0000-77717000 r-xp 00000000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
0000-77867000 r--p 02238000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
0000-77868000 rwxp 02388000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
0000-7787c000 rw-p 02389000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
0000-7787d000 rwxp 0239d000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
0000-77883000 rw-p 0239e000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
0000-77975000 r--s 00000000 b3:1c 81805 /data/data/com.android.chrome/app_chrome/paks/zh-CN.pak
0000-7803b000 r--s 00000000 b3:1c 81801 /data/data/com.android.chrome/app_chrome/paks/chrome_100_percent.pak
0000-78402000 r--s 00000000 b3:1c 81804 /data/data/com.android.chrome/app_chrome/paks/resources.pak
0000-7d645000 r--s 00000000 b3:1c 82120 /data/data/com.android.chrome/files/tab6
0000-7d648000 r--s 00000000 b3:1c 82700 /data/data/com.android.chrome/files/tab20
0000-7f64c000 r--s 00019000 b3:1c 97945 /data/data/com.lbe.security/app_hips/client.jar
0000-7f65a000 r-xp 00000000 b3:1c 98005 /data/data/com.lbe.security/app_hips/liblbeclient.so
0000-7f65c000 r--p 0000e000 b3:1c 98005 /data/data/com.lbe.security/app_hips/liblbeclient.so
0000-7f65d000 rw-p 0000f000 b3:1c 98005 /data/data/com.lbe.security/app_hips/liblbeclient.so
0000-7fe78000 r--p 00000000 b3:1c 106036 /data/dalvik-cache/data@data@com.lbe.security@app_hips@client.jar@
classes.dex
0000-7fffa000 r--s 00000000 b3:1c 82101 /data/data/com.android.chrome/files/tab0
0000-80051000 r--s 00000000 b3:1c 82116 /data/data/com.android.chrome/files/tab1

```

Process of Chrome after hijacked

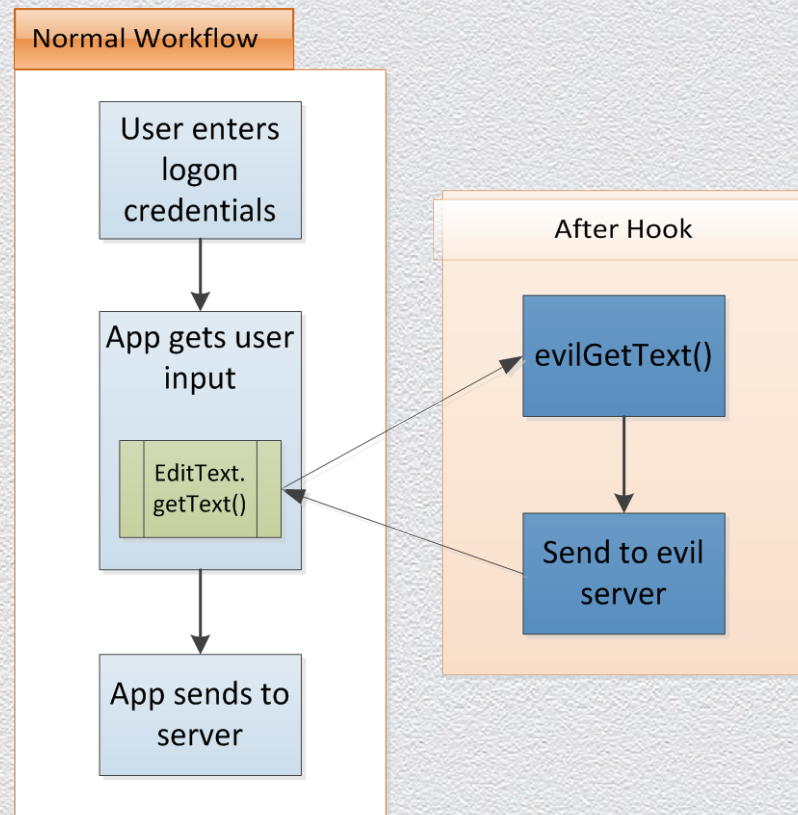
RSAC CONFERENCE 2014
ASIA PACIFIC & JAPAN



App Hijack

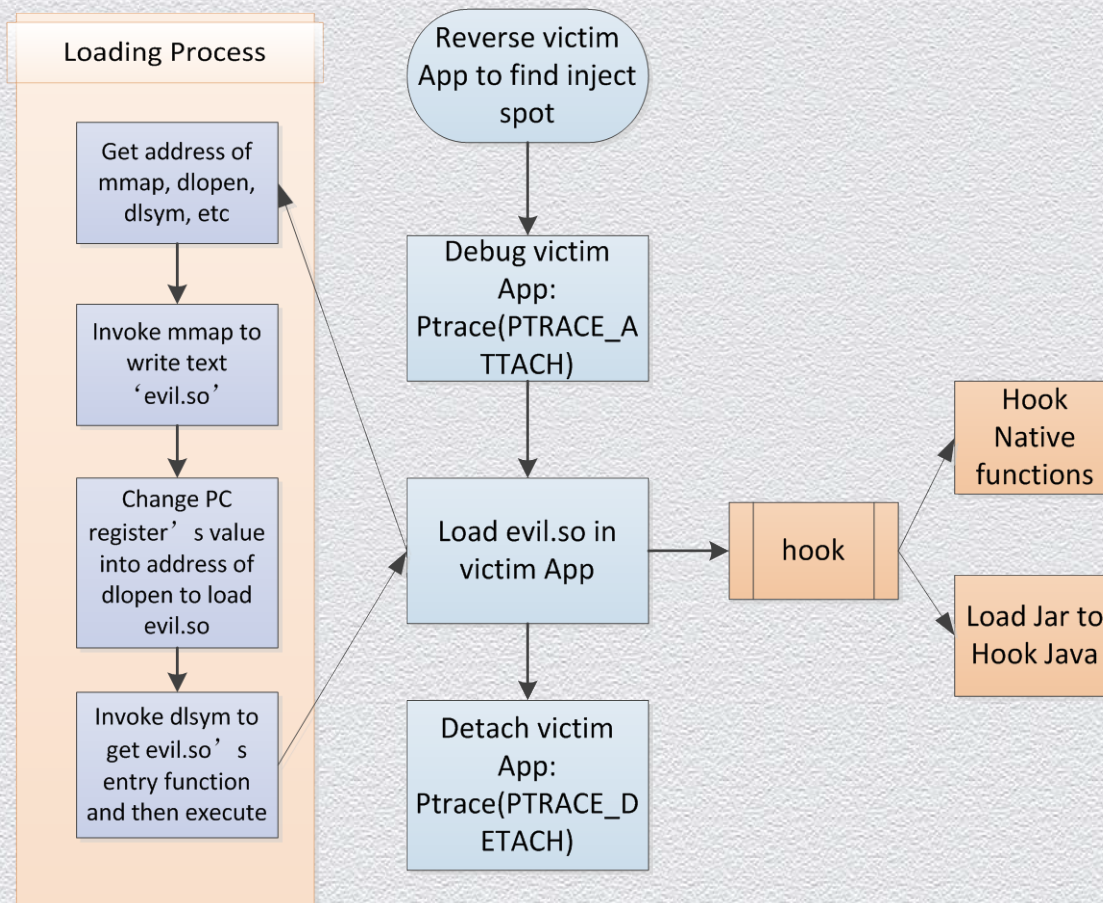
What's App Hijack?

- ◆ App Hijack: App's workflow is redirected by others
- ◆ Usually achieved by 'Inject' and 'Hook'



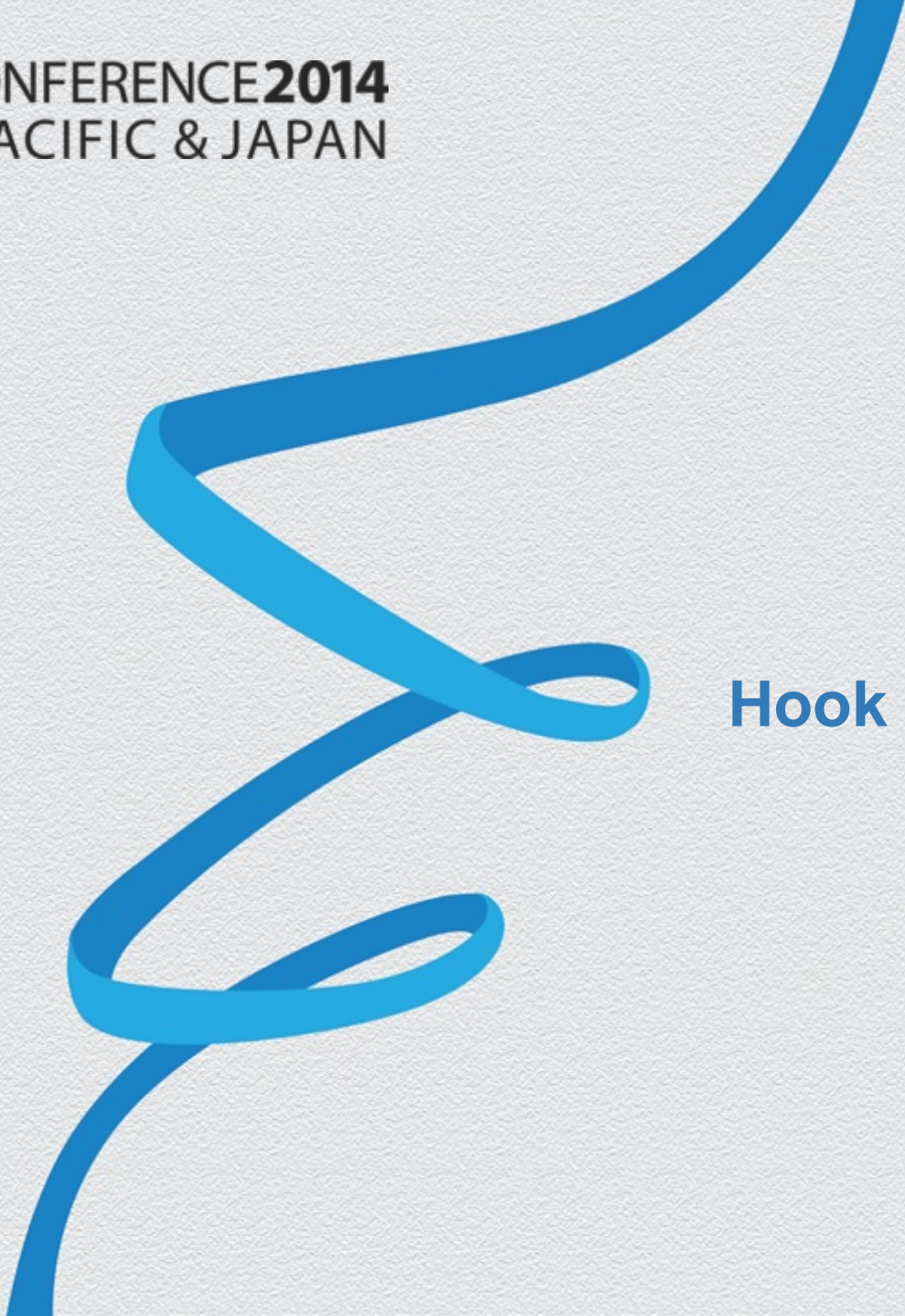
App Hijack's Process

- ◆ Reverse App to get its main logic
- ◆ Inject evil module into App's process
 - ◆ *Ptrace*
 - ◆ *Dlopen*
- ◆ Hook
 - ◆ Java Hook
 - ◆ Native(so) Hook



Process of Inject and Hook

RSAC CONFERENCE 2014
ASIA PACIFIC & JAPAN



Hook Insight

App which has Hook actions

- ◆ Virus: few
 - ◆ More costs to develop and maintain
 - ◆ Root is required
- ◆ Security App: some
 - ◆ Active Defense/Behavior Monitor

Hook: malware

◆ Wind Seeker



```
v12 = JNIEnv;  
v13 = (*(int (__fastcall **)(DWORD *, int, DWORD, DWORD))(*JNIEnv + 452))(  
    JNIEnv,  
    v11,  
    "getSystemClassLoader",  
    "()Ljava/lang/ClassLoader;");  
v14 = _JNIEnv::CallStaticObjectMethod(v12, v11, v13);  
v15 = _JNIEnv::NewObject(JNIEnv);  
_android_log_print(6, "FOR_AD", "DexClassLoader= %p", v15, v10, 0, v14);  
v16 = (*(int (**)(void))(*JNIEnv + 132))();  
_android_log_print(6, "FOR_AD", "loadClass= %p", v16);  
(*(void (**)(void))(*JNIEnv + 668))();  
v17 = _JNIEnv::CallObjectMethod(JNIEnv, v15, v16);  
_android_log_print(6, "FOR_AD", "Class= %p", v17);  
v18 = (*(int (**)(void))(*JNIEnv + 452))();  
_android_log_print(6, "FOR_AD", "setAppHook = %p", v18);  
_JNIEnv::CallStaticVoidMethod(JNIEnv, v17, v18);
```

```
long l3 = localCursor.getLong(localCursor.getColumnIndex("_id"));  
String str8 = localCursor.getString(localCursor.getColumnIndex("selfuin"));  
String str9 = localCursor.getString(localCursor.getColumnIndex("frienduin"));  
String str10 = localCursor.getString(localCursor.getColumnIndex("senderuin"));  
long l4 = localCursor.getLong(localCursor.getColumnIndex("time"));  
String str11 = localCursor.getString(localCursor.getColumnIndex("msg"));  
localCursor.getInt(localCursor.getColumnIndex("msgtype"));  
localCursor.getInt(localCursor.getColumnIndex("isread"));  
int n = localCursor.getInt(localCursor.getColumnIndex("issend"));  
localCursor.getLong(localCursor.getColumnIndex("msgseq"));  
localCursor.getLong(localCursor.getColumnIndex("shmsgseq"));  
localCursor.getInt(localCursor.getColumnIndex("istroop"));  
localCursor.getInt(localCursor.getColumnIndex("extraflag"));  
String str12 = localCursor.getString(localCursor.getColumnIndex("friendnick"));
```

Hook: Security App

Popular Security App:

Those which have hook actions:



Hook Types

- ◆ Java Hook
 - ◆ Static Field Hook
 - ◆ Method Hook
- ◆ Native So Hook
 - ◆ GOT Hook: Global Offset Table hook
 - ◆ SYM Hook: Dynamic Symbol hook
 - ◆ Inline Hook

Java Static Field Hook

- ◆ Change the value of Java Class's static field
- ◆ By reflecting
 - ◆ `Class<?> cls = Class.forName("")`
 - ◆ `Field fld = cls.getDeclaredField("")`
- ◆ Why static field?
 - ◆ Reflection only gets a Class DEFINITION, not a Class OBJECT


```

private static void a(Object arg3) {
    if(arg3 != null) {
        Object v0 = b.reflectField(KSCONST.decrypt("android.webkit.BrowserFrame$ConfigCallback")
            , KSCONST.decrypt("mHandlers"), arg3);
        if(v0 != null) {
            b.setField(KSCONST.decrypt("android.webkit.BrowserFrame$ConfigCallback"), KSCONST.decrypt(
                ("mHandlers"), arg3, new at(((ArrayList)v0)));
        }
    }
}

public static Object setField(String arg4, String arg5, Object arg6, Object arg7) {
    Exception v1_1;
    Object v0_1;
    Field v2;
    Object v1 = null;
    try {
        v2 = Class.forName(arg4).getDeclaredField(arg5);
        v2.setAccessible(true);
        v0_1 = v2.get(arg6);
    }
    catch(Exception v0) {
        Exception v3 = v0;
        v0_1 = v1;
        v1_1 = v3;
        goto label_12;
    }

    try {
        v2.set(arg6, arg7);
        goto label_7;
    }
    catch(Exception v1_1) {
    }

label_12:
    v1_1.printStackTrace();
label_7:
    return v0_1;
}

```

Java static field hook sample

Java Method Hook

- ◆ Change a Java class's method pointing to a custom method
 - ◆ Dalvik: Java Method -> Native Method
 - ◆ ART: Method Inline Hook

Java Method Hook

◆ Dalvik

- ◆ Java method is Dex byte code: *Method->insns*
- ◆ Q: Dex byte code inline hook? A: Too complicated!
- ◆ Change Java method into Native method
 - ◆ *Method->nativeFunc* points to a custom native method

```
// Replace method with our own code
SET_METHOD_FLAG(method, ACC_NATIVE);
method->nativeFunc = &xposedCallHandler;
method->insns = (const u2*) hookInfo;
method->registersSize = method->insSize;
method->outsSize = 0;
```

So GOT Hook

- ◆ Modify the a function's address stored in the GOT table of the so loaded by the App
- ◆ GOT
 - ◆ Global Offset Table
 - ◆ Store the addresses of functions imported by the library/executable
- ◆ PLT
 - ◆ Procedure Linkage Table
 - ◆ Will call GOT items to get the function's absolute address

So GOT Hook

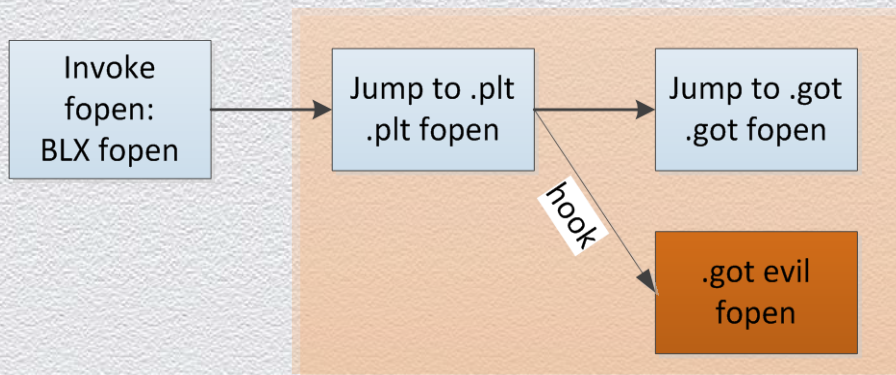
◆ Normal workflow

```
extern:0000C120                ; FILE *fopen(const char *filename, const char *modes)
extern:0000C120 00 00 00 00      IMPORT __imp_fopen      ; CODE XREF: fopen+8↑j
.text:00008C96 FF F7 E8 ED      BLX      fopen

; FILE *fopen(const char *filename, const char *modes)
.plt:00008868                fopen
.plt:00008868                ; CODE XREF: sub_8B90+12↓p
.plt:00008868                ; sub_8BE8+AE↓p
.plt:00008868 00 C6 8F E2      ADR      R12, 0x8870
.plt:0000886C 03 CA 8C E2      ADD      R12, R12, #0x3000
.plt:00008870 2C F7 BC E5      LDR      PC, [R12,#(fopen_ptr - 0xB870)]! ; __imp_fopen

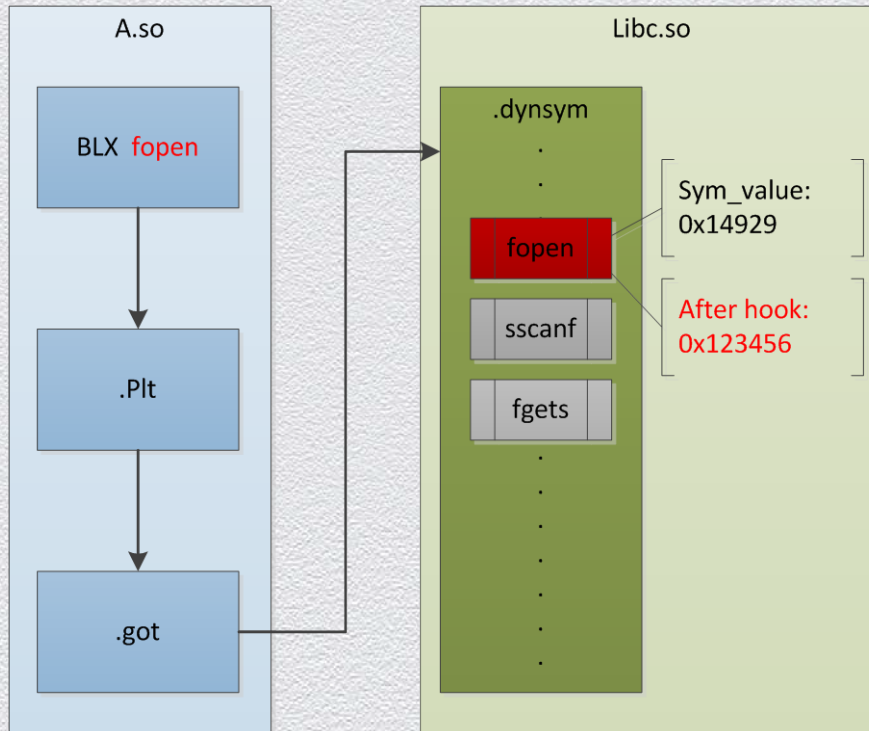
extern:0000C120                ; FILE *fopen(const char *filename, const char *modes)
extern:0000C120 00 00 00 00      IMPORT __imp_fopen      ; CODE XREF: fopen+8↑j
```

◆ Workflow after hook



So SYM Hook

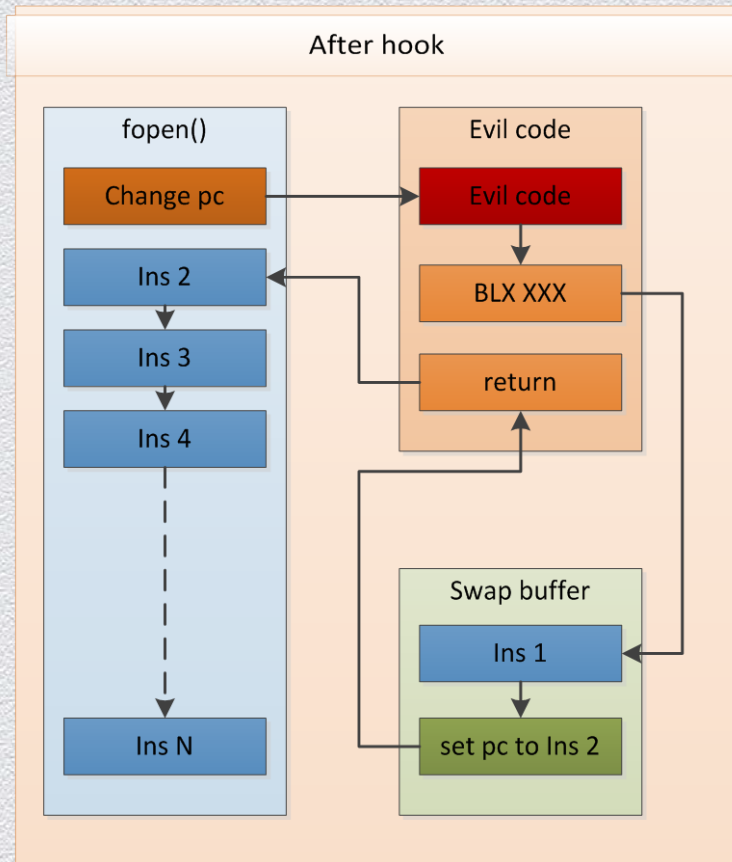
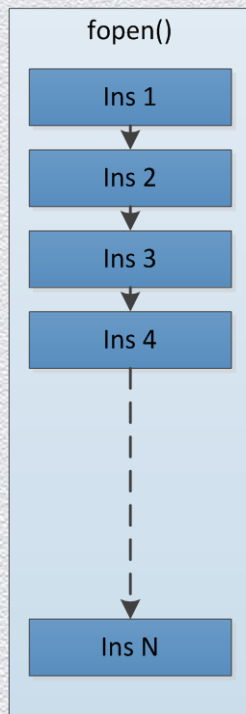
- ◆ Modify dependent library's exported symbol's value
- ◆ Weakness
 - ◆ Hook the needed library before loading current library/executable



So SYM Hook Process

So Inline Hook

- ◆ Modify a function's inline instructions
- ◆ Jumping has a distance restriction
- ◆ Compatibility for Arm and Thumb
- ◆ Advantage: global coverage
- ◆ Disadvantage: difficult and unstable



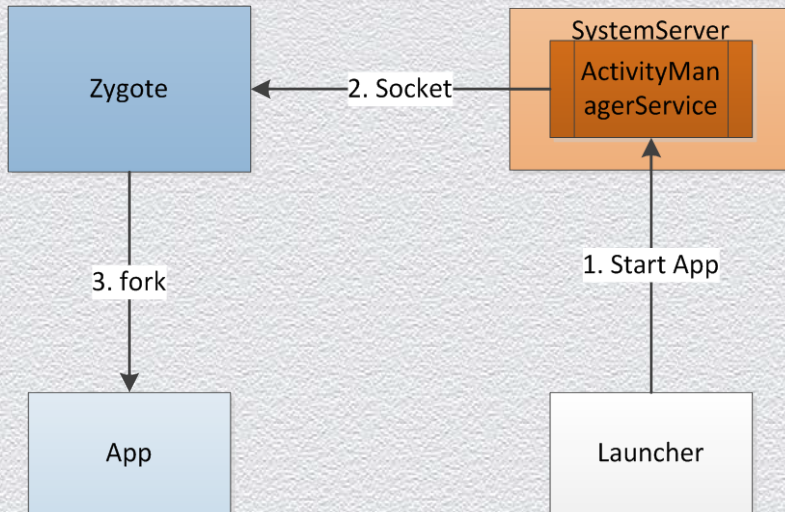
So Inline Hook Process

Hook Zygote

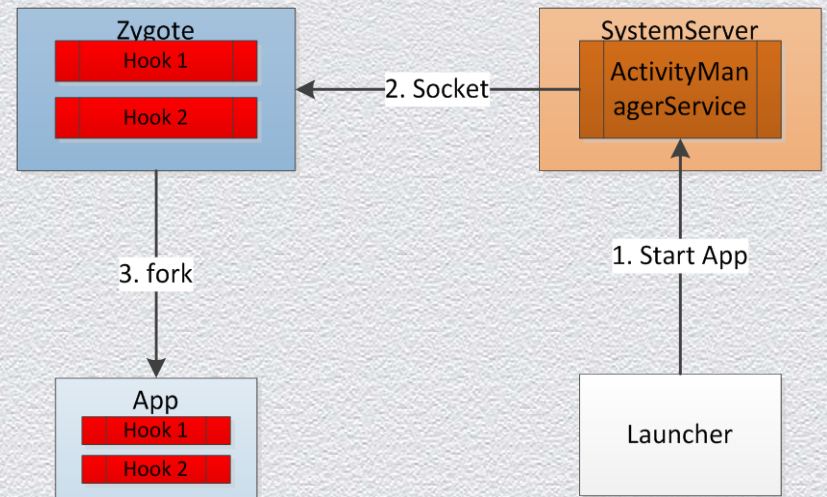
- ◆ Global Hook
 - ◆ 😞 Hook all processes already running, and monitor new processes
 - ◆ 😊 Hook Zygote
- ◆ All App processes are forked by Zygote
 - ◆ If Zygote is injected/hooked, all the forked App processes are injected/forked as well

Hook Zygote

Before Hook



After Hook



Popular Hook Frameworks

- ◆ Xposed
 - ◆ <http://repo.xposed.info/>
- ◆ Cydia Substrate
 - ◆ <http://www.cydiasubstrate.com/>
- ◆ ADBI/DDI
 - ◆ <https://github.com/crmulliner/adbi>
 - ◆ <https://github.com/crmulliner/ddi>

Pop Hook Frameworks: Xposed

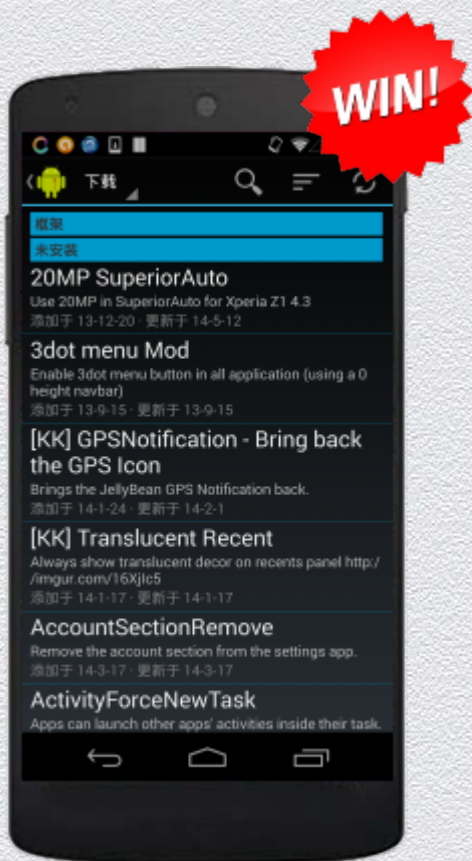
- ◆ The most popular open source hook framework in Android platform
- ◆ Replace *app_process*
- ◆ Java Method -> Native Method
- ◆ Before original method is executed, call *BeforeHook/AfterHook* interfaces
- ◆ Zygote restart is required to update hooks

Pop Hook Frameworks: Cydia Substrate

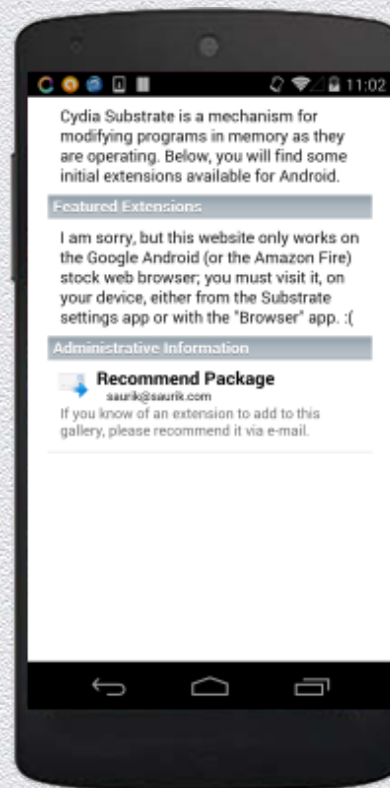
- ◆ Compatible with iOS and Android
- ◆ More popular in iOS
- ◆ Zygote restart is required

Pop Hook Frameworks: Comparison

XPosed



Cydia Substrate



RSAC CONFERENCE 2014
ASIA PACIFIC & JAPAN



App Hijack Demo



Demo: Facebook Logon Hijack

(Will replace with the correct snapshot in final version)



Demo: China Merchants Bank WebView Phishing Attack

(Will replace with the correct snapshot in final version)

RSAC CONFERENCE 2014
ASIA PACIFIC & JAPAN



Fix for App Hijack

Hook Fix: Java Hook

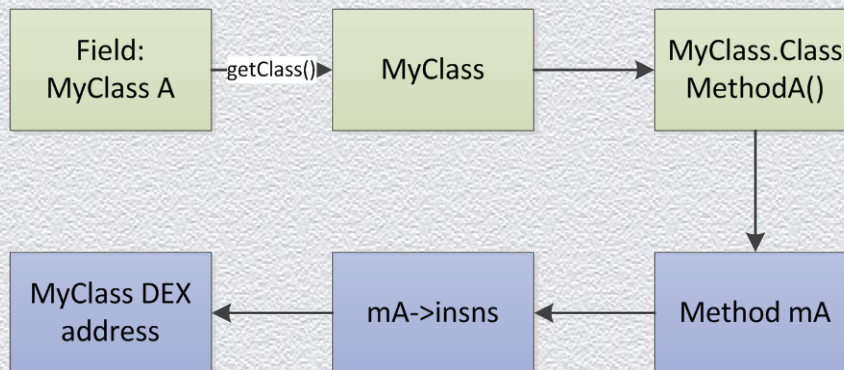
◆ Static Field Hook

◆ Detection

- ◆ Check whether the field's class is defined in current DEX or framework

◆ Fix

- ◆ Failed: original value is unreachable



Hook Fix: Java Hook

- ◆ Method hook

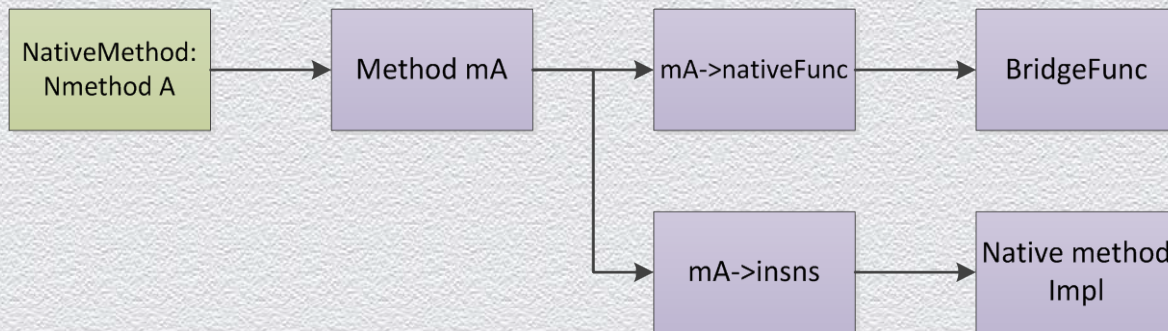
- ◆ Dalvik: Java->Native

- ◆ Detection

- ◆ Check whether the method's instructions lie in current App

- ◆ Fix

- ◆ Failed: Java->Native is not invertible



Hook Fix: So Hook

- ◆ GOT hook
 - ◆ Detection: check the library pointed by the GOT address
 - ◆ Fix: reload the library on the disk to get the correct address
- ◆ SYM hook
 - ◆ Detection: check the library pointed by the Symbol address
 - ◆ Fix: reload the library on the disk to get the correct address
- ◆ Inline hook
 - ◆ Detection: parse the instructions
 - ◆ Fix: reload the library and restore the modified instructions

Inject Fix

- ◆ Why fix injection?
 - ◆ Sometimes, inject only, no hook: remote monitor thread
- ◆ How to fix:
 - ◆ *munmap*
 - ◆ Hook Fix must be done first
 - ◆ Unstable, Crash

Fix Zygote


- ◆ Q: How to protect Zygote from hijacking?
 - ◆ Zygote is started by Init
 - ◆ Protect Zygote in a very early stage
 - ◆ Q: How to ensure the protector to start earlier than others?
 - ◆ Learn from Xposed to replace *app_process*?
 - ◆ System modifications are not first choice
 - ◆ Compatibility issues
 - ◆ Less configurability

Fix for App Hijack: Conclusion

- ◆ Three “Basic” Points:
 - ◆ Hooks could not be fixed basically
 - ◆ Injects could not be removed basically
 - ◆ Zygote could not be protected basically



RSAC CONFERENCE 2014
ASIA PACIFIC & JAPAN



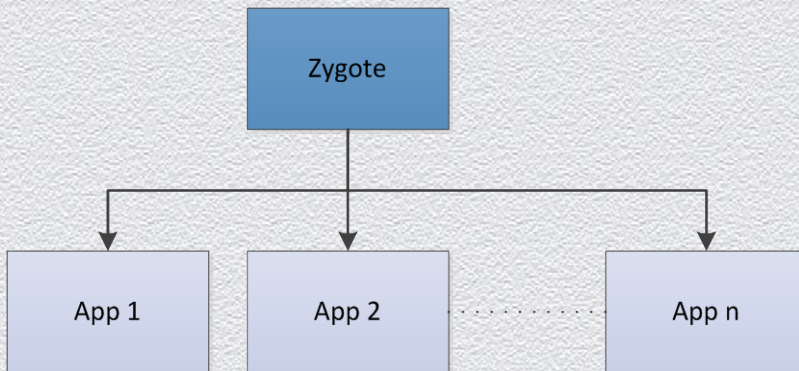
**Leave My App Alone –
Create A Trusted App
Runtime**

Create A Trusted App Runtime

- ◆ Create a trusted Zygote Process
- ◆ Protect the trusted Zygote
- ◆ Make App fork from the trusted Zygote

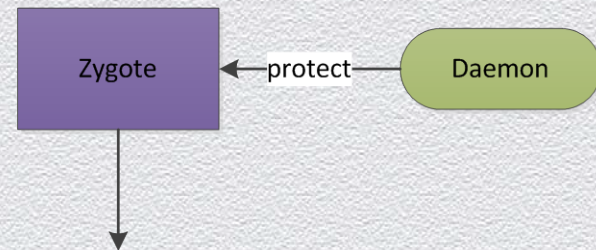
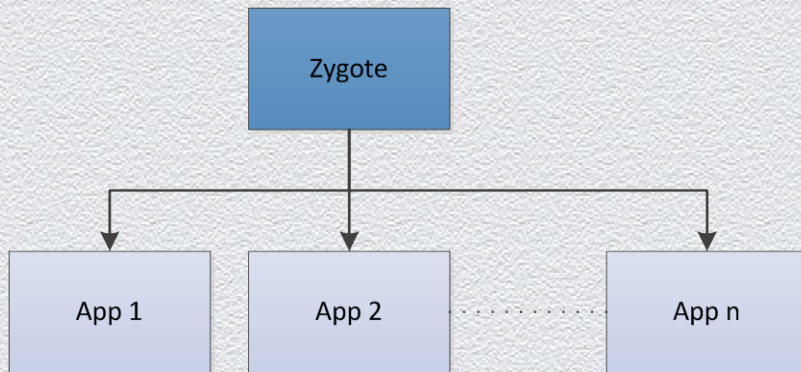
Create Trusted Zygote

- ◆ Step1: Start a new Zygote process
 - ◆ Compile a our own *app_process*
 - ◆ No restart required



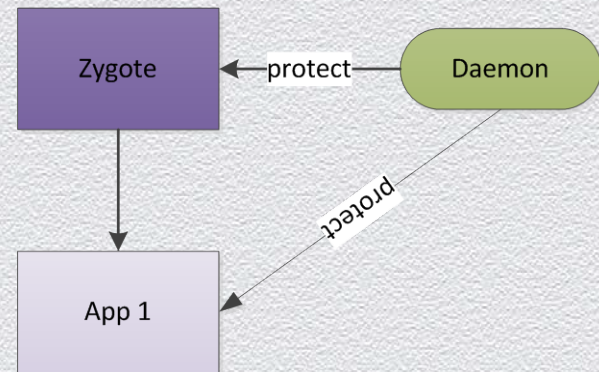
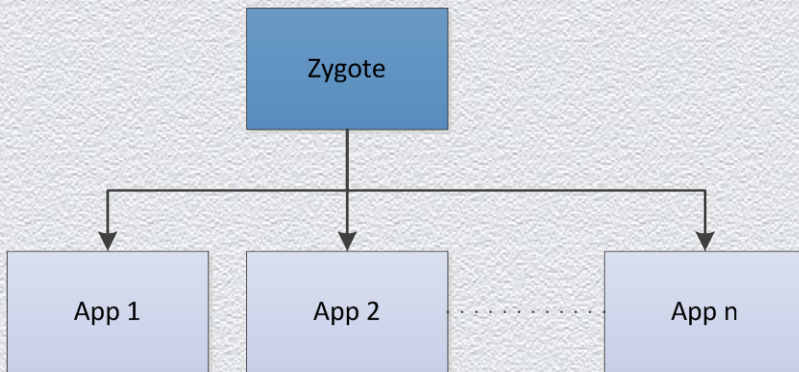
Protect Trusted Zygote

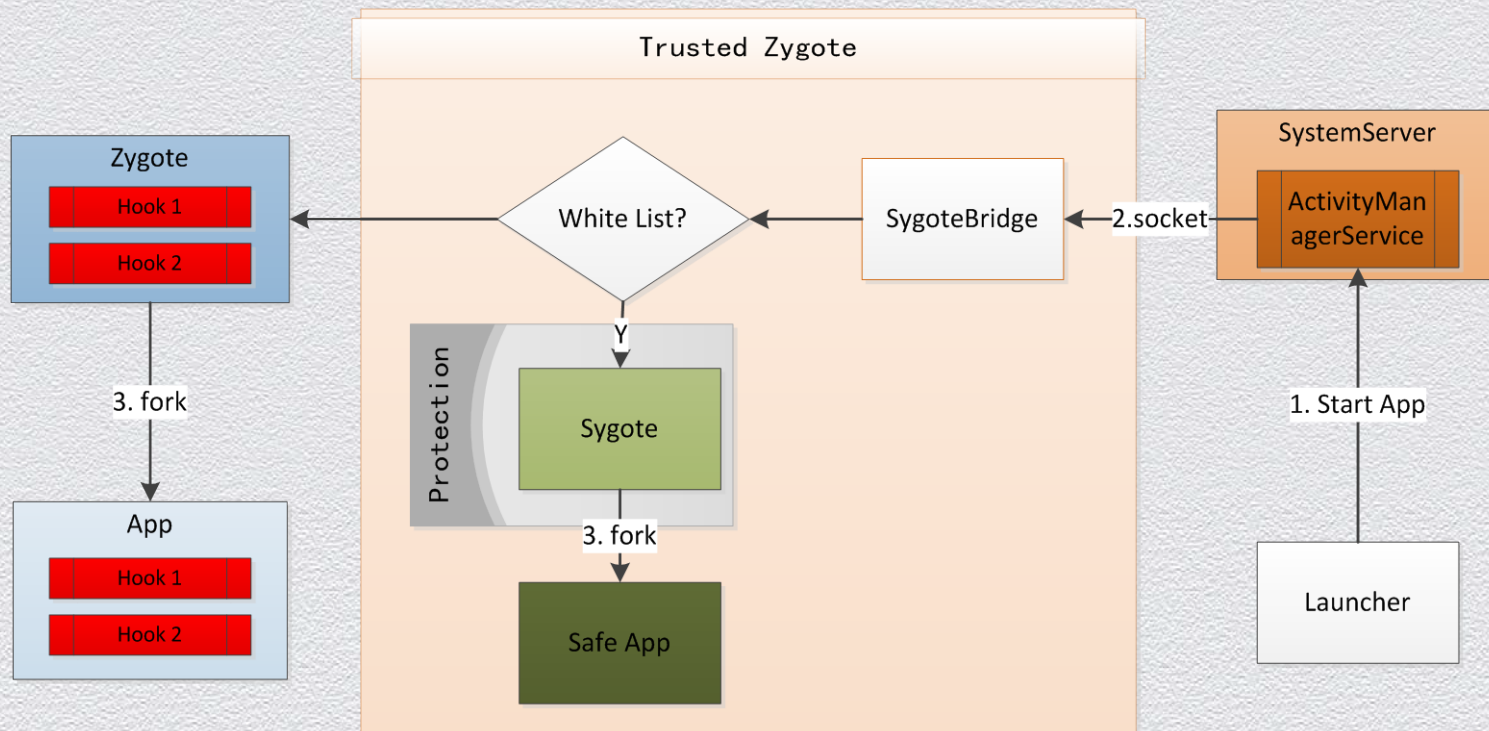
- ◆ Step2: Protect trusted Zygote process
 - ◆ Anti Debug/Anti Inject
 - ◆ *Ptrace_me*
 - ◆ Double-Process protection
 - ◆ Other protections



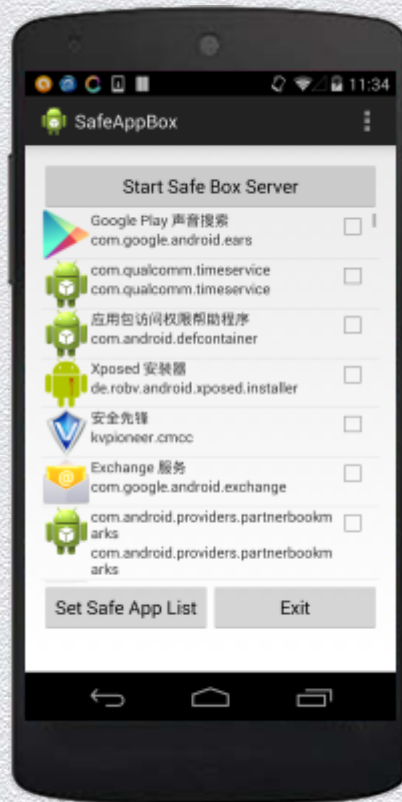
Control Trusted Zygote

- ◆ Fork App through trusted Zygote
 - ◆ Redirect the socket communication from *ActivityManagerService* to Zygote
 - ◆ Control
 - ◆ White List: redirect the socket to trusted zygote
 - ◆ Black List: redirect the socket to old zygote





Trusted App Runtime: the Whole Picture



Demo: Choose any App you want to run in a trusted runtime

RSAC CONFERENCE 2014
ASIA PACIFIC & JAPAN



Thanks / Q&A