Playing Hide and Seek with Dalvik Executables

Axelle Apvrille

Hack.Lu, October 2013
Who am i?

#!/usr/bin/perl -w
my $self = {
  realname => 'Axelle Apvrille',
  nickname => 'Crypto Girl',
  twitter => '@cryptax',
  job => 'Malware Analyst and Researcher',
  title => 'Senior', # white hair
  company => 'Fortinet, FortiGuard Labs',
  before => 'Security software eng.: protocols, crypto...',
  languages => 'French, English, Hexadecimal :)'  
};
Quick background

Android mobile phone
Quick background

Android mobile phone

Applications: APK

Inside the APK: DEX

Inside the DEX:
Classes, methods, fields, strings

'bytes', '** I am Mr Hyde **', '<init>'
Quick background

Android mobile phone

Applications: APK

Inside the APK: DEX

Dalvik Executable with Dalvik bytecode

dex.035.v...d..$g
Quick background

Android mobile phone

Applications: APK

Inside the APK: DEX
Dalvik Executable with Dalvik bytecode
dex.035.v..d..$g

Inside the DEX
Classes, methods, fields, strings
'bytes', '** I am Mr Hyde **', '<init>'...
Part 1: Hiding a method

Application source code

```java
public void thisishidden(boolean ismrhyde) {
    Log.i("HideAndSeek",
            "In thisishidden(): set mryde=
            +ismrhyde);
    try {
        File dir;
        if (context !=null) {
            ...
    }
```
Hiding / Revealing demo

Demo

https://github.com/cryptax/dextools
Hiding / Revealing demo

Demo

https://github.com/cryptax/dextools
Format of a DEX file

Header

Arrays

Data
Format of a DEX file

- Header
- Lists:
  - List of String Ids
  - List of Type Ids
  - List of Fields Ids
  - List of Method Ids
  - List of Class Defs
- Data
Inside the list of class definitions

- **access_flags:**
  - ACC_PUBLIC,
  - ACC_PRIVATE,
  - ACC_STATIC...

- **code_off:** offset to code from beginning of DEX file

- **method_idx_diff:** increment to method indexes

---

**encoded_method**

**Header**

**class_def_item**

**Arrays**

**Data**
Inside the list of class definitions

- **access_flags**: ACC_PUBLIC, ACC_PRIVATE, ACC_STATIC...
- **code_off**: offset to code from beginning of DEX file
- **method_idx_diff**: increment to method indexes

encoded_method

Header

Arrays

class_def_item

Type Lds

class_data_item
Inside the list of class definitions

- **access_flags**: ACC_PUBLIC, ACC_PRIVATE, ACC_STATIC...
- **code_off**: offset to code from beginning of DEX file
- **method_idx_diff**: increment to method indexes

### encoded_method
- **class_def_item**
  - **Type Lds**
  - **Arrays**
  - **List of fields**
  - **Direct methods**: encoded_method
  - **Virtual methods**: encoded_method
How to hide

**Trick**
Modify the chaining of methods and skip the hidden method
The info for the hidden method is still there, but won’t be read

**Implementation**

- **method_idx_diff:**
  - modify for hidden method
  - + modify for the 'other' method

- **code_off:** refer the other method

- **access_flags:** nothing to do

- **direct_methods_size** (or virtual_methods_size): nothing to do
Visual representation of chaining

Method Ids

encoded Method A

Encoded Method B

Encod Method C

Code of A

Code of B

Code of C

Code

[1] A

[2] B

[3] C
Visual representation of chaining

[1] A
[2] B
[3] C

Encoded Method A

Method IDs

Encoded Method B

Encoded Method C

Code of A

Code of B

Code of C

Code
Visual representation of chaining

Method IDs:
1. A
2. B
3. C

Encoded Method A
Encoded Method B
Encoded Method C

Code:
- Code of A
- Code of B
- Code of C
Visual representation of chaining

Method Ids

[1] A
[2] B
[3] C

Encoded Method A

Encoded Method B

Encoded Method C

Code

Code of A

Code of B

Code of C

Hack.Lu 2013 - A. Apvrille
Hiding - for advanced users

Some more tricks

- **Access flags**: you *may* modify but must choose a flag within *direct* methods or *virtual* methods

- **Single method?** Set `direct_methods_size` (or `virtual_methods_size`) and nullify encoded method
Re-build the APK

Build a valid DEX
- Compute the SHA-1 of the new DEX
- Write to header
- Compute the checksum of the new DEX
- Write to header
- https://github.com/cryptax/dextools

Re-build APK
- Unzip original APK: retrieve manifest, resources...
- Zip new APK with new DEX + same manifest and resources
- Sign package (jarsigner)
Part 2: calling the hidden method

calling thisishidden()

- The method is hidden to disassemblers
- ... but it can be run!

The strange case of Dr Jekyll and Mr Hyde – R. Stevenson

- Split personalities: Dr Jekyll or Mr Hyde
- Only one way to change into Mr Hyde: call thisishidden()
- Current personality displayed in main activity
DEMO :}
Implementation - Step 1/4

Load the current DEX file

openNonAsset() not directly accessible → use reflection

// get AssetManager class via reflection
Class localClass = Class.forName("....AssetManager");
Class[] arrayOfClass = new Class[1];
arrayOfClass[0] = String.class;

// get openNonAsset method
Method localMethod = localClass.getMethod("openNonAsset", ...)
AssetManager localAssetManager = this.context.getAssets();
Object[] arrayOfObject = new Object[1];
arrayOfObject[0] = paramString;

// invoke method
InputStream localInputStream = (InputStream)localMethod.invoke(...);
Patch the DEX

Undo what we did - re-chain the hidden method, re-hash and checksum the DEX

```java
int patch_index = 0x2c99;
dex[patch_index++] = 1; // method_idx_diff
dex[patch_index++] = 1; // access flag
dex[patch_index++] = (byte)0xcc; // code offset
dex[patch_index++] = (byte)0x28;
dex[patch_index++] = 1;
```
Open the modified DEX

- use reflection to call openDexFile()

    native private static int
    openDexFile(byte[] fileContents);

- returns a cookie = pointer to internal struct for DEX

- load modified class using defineClass()

    Class patchedHyde = null;
    Log.i("HideAndSeek", "retrieving patched MrHyde class");
    if (defineClassMethod != null) {
        patchedHyde = (Class) defineClassMethod.invoke(
            dexFileClass, params);
Invoke the hidden method

- Search for the hidden method (getDeclaredMethods())
- Instantiate an object
- Call thisishidden()

Object obj = patchedHyde.getDeclaredConstructor(Context.class)
    .newInstance(context);
Log.i("HideAndSeek", "after new Instance");
arg[0] = Boolean.valueOf(true);
Log.i("HideAndSeek", "invoking thisishidden()");
thisishiddenMethod.invoke(obj, arg);
It’s two different classes

Static field
Instance field
MrHyde
thisishidden

Static field
Instance field
Modified MrHyde
thisishidden
It's two different classes

Does not work!
It's two different classes

Does not work!
It's two different classes

Static field
Instance field
MrHyde
thisishidden

Static field
Instance field
Modified MrHyde
thisishidden

Does not work!

Use shared files
Hiding, so what?

Dangers
It can be used to hide some malicious feature

Detection
The strings are not hidden
The bytecode is there

Solutions
- Use my patch/unpatch tool: hidex.pl
- Disassemble bytecode at a given location: androdis.py
- Fix Android: verify consistency of encoded_method
- Google notified in June 2013
Thank You!

Thanks! to @pof ... and for your attention!

FortiGuard Labs
Follow us on twitter: @FortiGuardLabs
or on our blog http://blog.fortinet.com

Me
twitter: @cryptax
e-mail: aapvrille at fortinet dot com
source code: https://github.com/cryptax/dextools

Are those PowerPoint slides? No way! It’s \LaTeX+ TikZ + Beamer + Lobster