Dalvik Executable (DEX) Trick: Hidex

Axelle Apvrille

Insomni'Hack, March 2014
Who am i?

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

Android mobile phone

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>'
Quick Android 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>'
Quick Android background

Android mobile phone

Applications: APK

Inside the APK: DEX

Dalvik Executable with Dalvik bytecode
dex.035.v.d.$g
Quick Android 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
Part 1: goal and demo

Goal

1. Write an app
2. Hide a given *method* of the app to disassemblers
Part 1: goal and demo

**Goal**

1. Write an app
2. Hide a given *method* of the app to disassemblers

**Demo - source code:**

[https://github.com/cryptax/dextools](https://github.com/cryptax/dextools)

1. Example method `thisishidden()`:  
   ▶ Logs "In thisishidden(): set mrhyde=" etc  
   ▶ Accesses file "identity" in app dir  
   ▶ Exact prototype: `public void thisishidden(boolean ismrhyde)`

2. Hide `thisishidden()`: Baksmali, dex2jar, Androguard, JEB, IDA Pro do not see it!

3. Back: reveal `thisishidden()`
How it works / inside class_data items

- **method_ids**
  - method_idx

- Encoded Method A
  - code offset

- Code
  - Code of A
  - Code of B
  - Code of C

---

Insomni’Hack 2014 - A. Apvrille 6/18
How it works / inside class_data items

Encoded Method A

Encoded Method B

Code of A

Code of B

Code of C

method_ids

method_idx

method_idx_diff

code offset
How it works / inside class_data items

method_ids

Encoded Method A
Encoded Method B
Encoded Method C

method_idx
method_idx_diff
method_idx_diff

code offset

Code of A
Code of B
Code of C

Code
How it works / inside class_data_items

Encoded Method A
Encoded Method B
Encoded Method C

method_ids

method_idx

method_idx_diff

code offset

Code of A
Code of B
Code of C

Code
How it works / inside class_data items

encoded Method A
encoded Method B
encoded Method C

method_idx
method_idx_diff
code offset

Code of A
Code of B
Code of C
Hiding - Advanced

Access flags
You may modify it but must remain with the same category of methods:
- direct methods: static, private, constructors
- virtual methods: others ;)

Single method to hide?
In class_data_item, set direct_methods_size (or virtual_methods_size)
+ 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
- `hidex` or `dexrehash` in
  https://github.com/cryptax/dextools

Re-package: easy

- In original APK, replace DEX with new one
- Zip, sign package (`jarsigner`)

Did you know?

You can write a `.dex` ‘manually’ using Yasm - thanks @angealbertini
Hello World: 695 bytes
Part 2: calling the hidden method - PoC

Calling the hidden method
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

calling thisishidden()
- The method is hidden to disassemblers
- ... but it can be run!
I/HideAndSeek( 851): invoking thisishidden() with arg=true
I/HideAndSeek( 851): In thisishidden(): set mrhyde=true
I/HideAndSeek( 851): thisishidden(): context=android.app.Application@416ff148
I/HideAndSeek( 851): thisishidden(): dir=/data/data/com.fortiguard.hideandseek/files
I/HideAndSeek( 851): thisishidden(): file: /data/data/com.fortiguard.hideandseek/files/identity
I/HideAndSeek( 851): thisishidden(): done
Implementation… technical but illustrated ;)

MrHyde

**Reflection**

- `openNonAsset()`
- `classes.dex`
- `invoke`
- `Patch`
- `openDexFile()`
- `defineClass()`
- `Modified` `getDeclaredConstructor()`
- `Modified` `getDeclaredMethods()`
- `thisishidden()`

- `Instance field`
- `Instance field`
- `Static field`
- `Static field`

- Use shared files

Does not work!
Implementation... technical but illustrated ;)

MrHyde

Reflection

openNonAsset()
Implementation... technical but illustrated ;)

Reflection

openNonAsset()

invoke

classes.dex

MrHyde

Modified
getDeclaredConstructor()

Modified
getDeclaredMethods()

do Thing()

Instance field

Instance field

Static field

Static field

Use shared files

Does not work!
Implementation... technical but illustrated ;)

MrHyde

Reflection
openNonAsset()

invoke

classes.dex

Patch

Instance field

Instance field

Does not work!

Static field

Use shared files

MrHyde object

newInstance()
Implementation... technical but illustrated ;)

MrHyde

Reflection

openNonAsset()

openDexFile()

invoke

classes.dex

Patch

Modified

defineClass()

getDeclaredConstructor()

newInstance()

getDeclaredMethods()

thisishidden()

invoke

Instance field

Static field

Use shared files

Does not work!
Implementation... technical but illustrated ;)

MrHyde

Reflection

openNonAsset()

openDexFile()

invoke

classes.dex

Patch

MrHyde object

newInstance()

defineClass()

getDeclaredConstructor()

defineClass()

getDeclaredMethods()

thisishidden()
Implementation... technical but illustrated ;)

MrHyde

Reflection

openNonAsset()
openDexFile()
defineClass()

invoke

classes.dex

Patch

Modified
getDeclaredConstructor()

Modified
getDeclaredMethods()

thisishidden()

invoke

Instance field

Instance field

Static field

Static field

Use shared files

Does not work!
Implementation... technical but illustrated ;)

```
openNonAsset()
openDexFile()
defineClass()

Reflection

MrHyde

Modified
```

- openNonAsset()
- openDexFile()
- defineClass()

Does not work!
Implementation... technical but illustrated ;)

Reflection

- `openNonAsset()`
- `openDexFile()`
- `defineClass()`
- `getDeclaredConstructor()`

Modified

MrHyde

`newInstance()`

getDeclaredMethods()

this is hidden()
Implementation... technical but illustrated ;)
Implementation... technical but illustrated ;)
Implementation... technical but illustrated ;)

Reflection
- openNonAsset()
- openDexFile()
- defineClass()
- getDeclaredConstructor()
- getDeclaredMethods()
- thisishidden()

MrHyde object

Modified

MrHyde
Implementation... technical but illustrated ;)

Instance field

MrHyde

MrHyde object

Instance field

Modified

Does not work!
Implementation... technical but illustrated :)

Instance field

MrHyde

Does not work!

MrHyde object

Instance field

Modified
Implementation... technical but illustrated ;)

MrHyde object
- Instance field
- Static field
- Modified

MrHyde
- Instance field
- Static field

Does not work!
Implementation... technical but illustrated ;)

Instance field
Static field

MrHyde

Does not work!

Use shared files

MrHyde object
Instance field
Static field
Modified

Static field
Instance field

Insomni’Hack 2014 - A. Apvrille
It works :) 

- Dex manipulation: working on all versions
- Calling hidden method: < 4.4.2. 
  Prototypes for openDexFile and defineClass changed
  - minor modif for defineClass
  - openDexFile no longer works on byte[]

Work in progress: looks feasible.

- Android Security Team notified in June 2013
Detecting hidden methods

How? Use ‘--detect’ option in hidex.pl

$ ./hidex.pl --input classes.dex | grep -B 1 "WARNING"
$ ./nonreferenced-methods.sh classes.dex

- null code offset: just a hint
- null or negative method_idx_diff
- code offset or method_id already referenced
- method_id never referenced: beware, there are valid methods not implemented in the DEX.

Something is wrong with openNonAsset()

Class: Lcom../MrHyde; Method: openNonAsset Position: 0x2C99
WARNING: Code offset 0x13D8 ALREADY REFERENCED
Class: Lcom../MrHyde; Method: openNonAsset Position: 0x2C99
WARNING: method_idx_diff <= 0 detected
Class: Lcom../MrHyde; Method: openNonAsset Position: 0x2C99
WARNING: METHOD_IDX 22 ALREADY REFERENCED
Non Referenced Methods Detection results

Method Landroid/annotation/SuppressLint;->value is never used
Method Landroid/annotation/TargetApi;->value is never used

→ abstract methods indeed never used.

Method Lcom/fortiguard/hideandseek/MrHyde;->thisishidden is never used

→ hidden method!

Method Ljava/io/File;->delete is never used
Method Ljava/io/File;->exists is never used
Method Ljava/io/FileOutputStream;-><init> is never used
Method Ljava/io/FileOutputStream;->close is never used
Method Ljava/io/FileOutputStream;->write is never used
Method Ljava/lang/Object;->toString is never used

→ methods used only by the hidden method!
Reversing the hidden method

Fixing
- Implement method_idx_diff check?
- Implement code or method_id duplicate references check?

Working around
- Unpatch the DEX: hidex.pl
- Disassemble bytecode at a given location: androdis.py
How about hiding strings?

- **No idx_diff**. No chaining.
- **String as byte []** → not in strings list but visible in hex (e.g. strings Unix command)
- **Encode, encrypt, obfuscate string** → possible - nothing new.
Where’s the source code?
https://github.com/cryptax/dextools

FortiGuard Labs
Follow us on twitter: @FortiGuardLabs
or on our blog http://blog.fortinet.com
Me: @cryptax or aapvrille at fortinet dot com

Useless/shameless/stupid speaker challenge
I told @angealbertini I could keep it under 20 slides ;)

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