Playing Hide and Seek with Dalvik Executables

Axelle Apvrille

Hack.Lu, October 2013

Who am i?

whoami

```
#!/usr/bin/perl -w
my $self = {
    realname => 'Axelle Apvrille',
    nickname => 'Crypto Girl',
    twitter => '@cryptax',
    job => 'Malware Analyst and Researcher',
    # reverse engineering of incoming mobile malware
    # research and tools in related areas
    title => 'Senior', # white hair
    company => 'Fortinet, FortiGuard Labs',
    before => 'Security software eng.: protocols, crypto...',
    languages => 'French, English, Hexadecimal :)'
};
```

Quick background

Android mobile phone

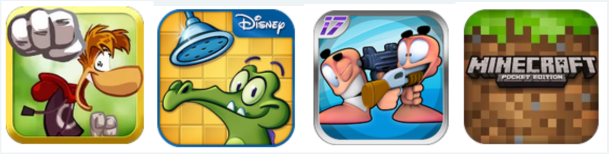
11:35
Natanya Ray
Checkin' out latest Hollywood action!
10:09
Barry Smyth
Want to meet up for coffee?
15 minutes ago
market
Androidify yourself by customising the little green Android as yourself, your family, your friends, anyone!
Maps
CardioTrai
Google Se
News & W

Quick background

Android mobile phone

11:35
Natanya Ray
Checkin' out latest Hollywood action!
10:09
Barry Smyth
Want to meet up for coffee?
15 minutes ago
market
Androidify yourself by customising the little green Android as yourself, your family, your friends, anyone!
Maps
CardioTrai
Google Se
News & W

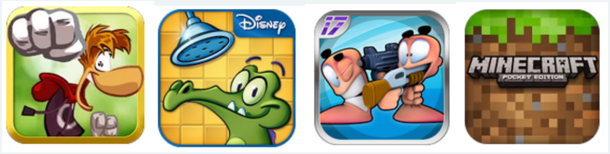
Applications: APK



Quick background

Android mobile phone

Applications: APK



Inside the APK: DEX

Dalvik Executable with Dalvik bytecode

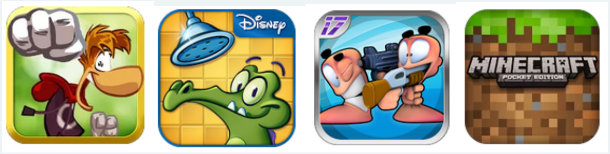
```
dex.035.V..d..$g
```

Quick background

Android mobile phone

11:35
Natanya Ray
Checkin' out latest Hollywood action!
10:09
Barry Smyth
Want to meet up for coffee?
15 minutes ago
market
Maps
CardioTrai
Google Se
News & W
Androidify yourself by customising the little green Android as yourself, your family, your friends, anyone!

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

```
public void thisishidden(boolean ismrhyde) {
  Log.i("HideAndSeek",
         "In thisishidden(): set mrhyde="
         +ismrhyde);
  try {
    File dir;
    if (context !=null) {
...
```

Method thisishidden(): hidden to dissassemblers

- Baksmali does not see it
- dex2jar does not see it
- IDA Pro does not see it
- Androguard does not see it

Hiding / Revealing demo

$\rightarrow$

Demo

https://github.com/cryptax/dextools

Hiding / Revealing demo

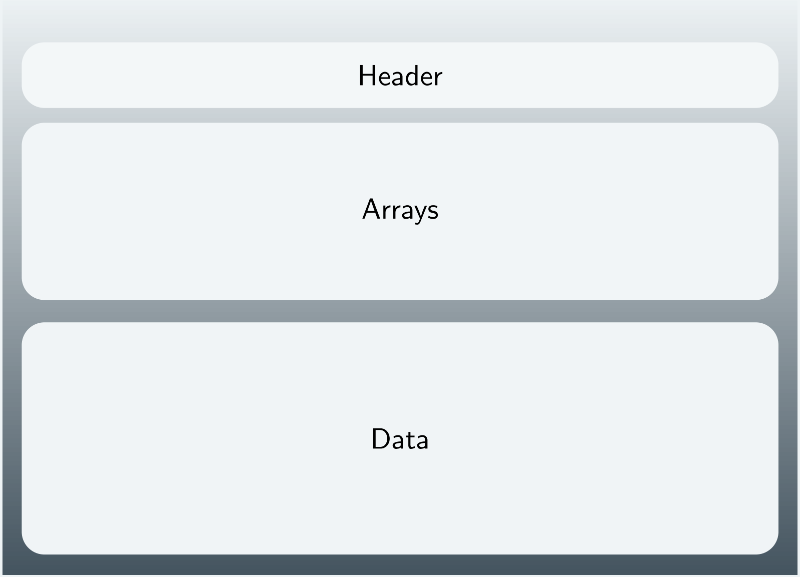
→

→

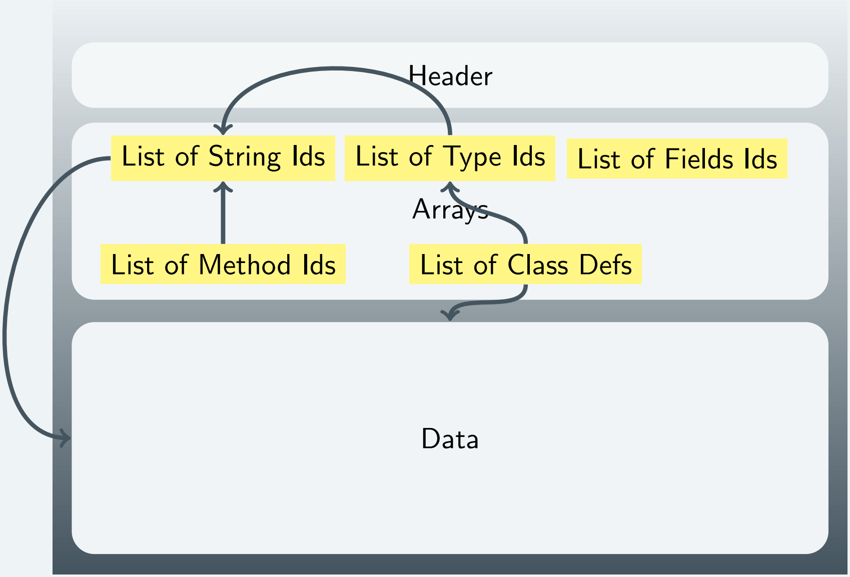
Demo

https://github.com/cryptax/dextools

Format of a DEX file



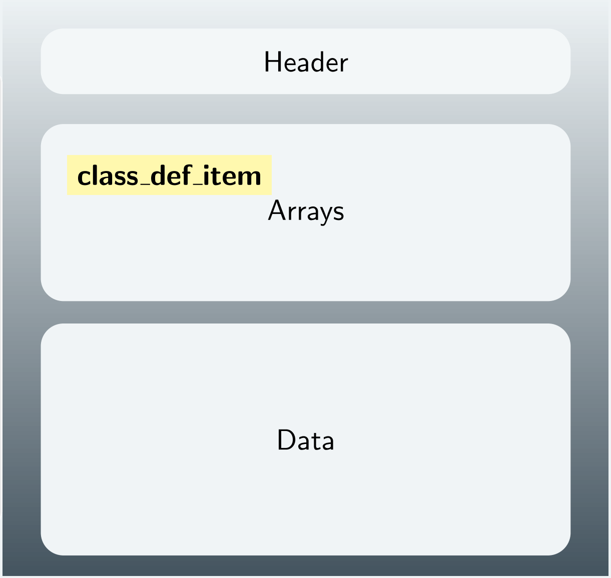
Format of a DEX file



Inside the list of class definitions

encoded_method

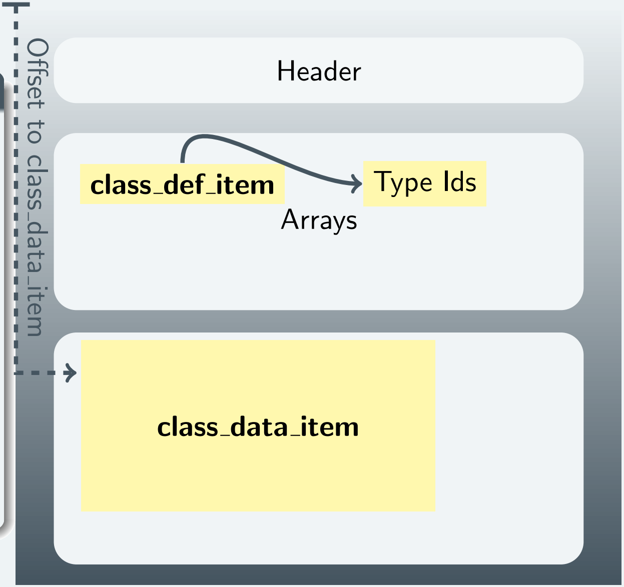
- **access_flags**: `ACC_PUBLIC`, `ACC_PRIVATE`, `ACC_STATIC`...
- **code_off**: offset to code from beginning of DEX file
- **method_idx_diff**: increment to method indexes



Inside the list of class definitions

encoded_method

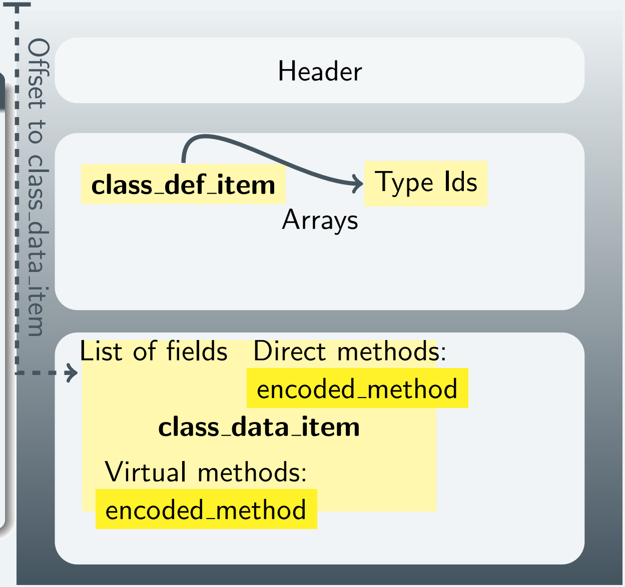
- **access_flags**: `ACC_PUBLIC`, `ACC_PRIVATE`, `ACC_STATIC`...
- **code_off**: offset to code from beginning of DEX file
- **method_idx_diff**: increment to method indexes



Inside the list of class definitions

encoded_method

- **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



How to hide

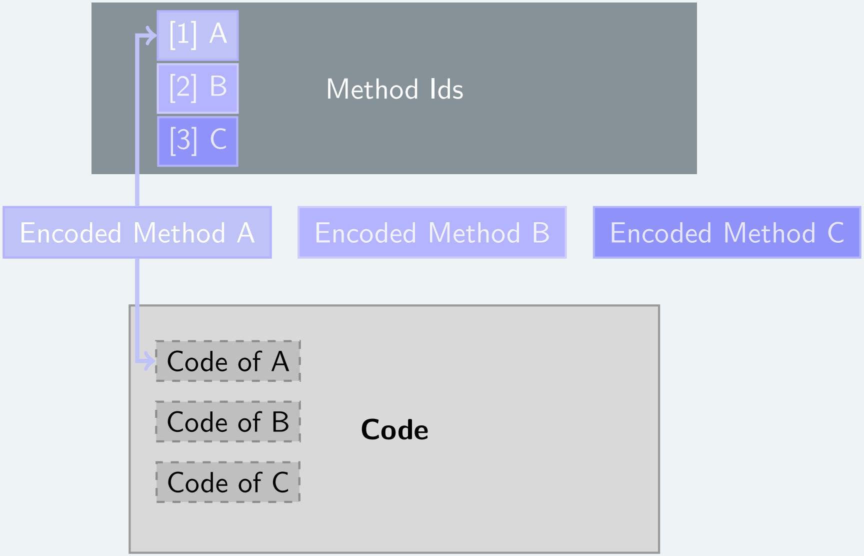
Trick

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

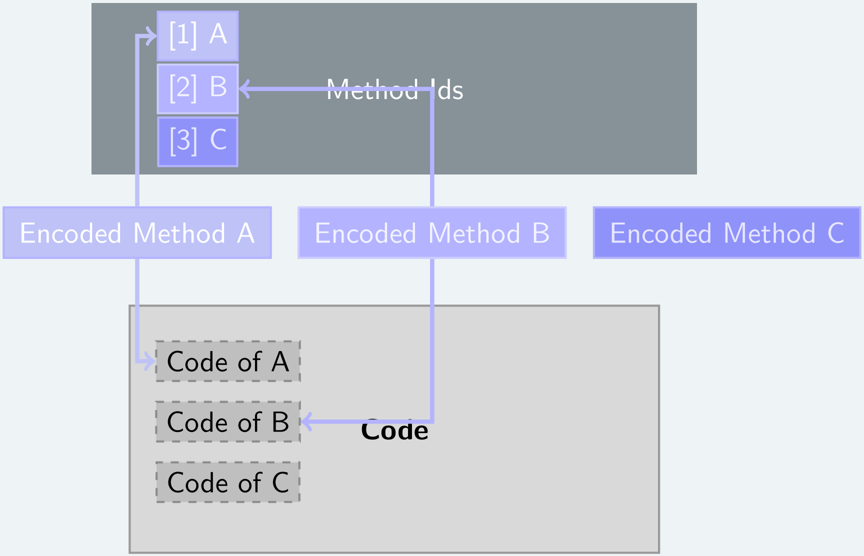
Implementation

- **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

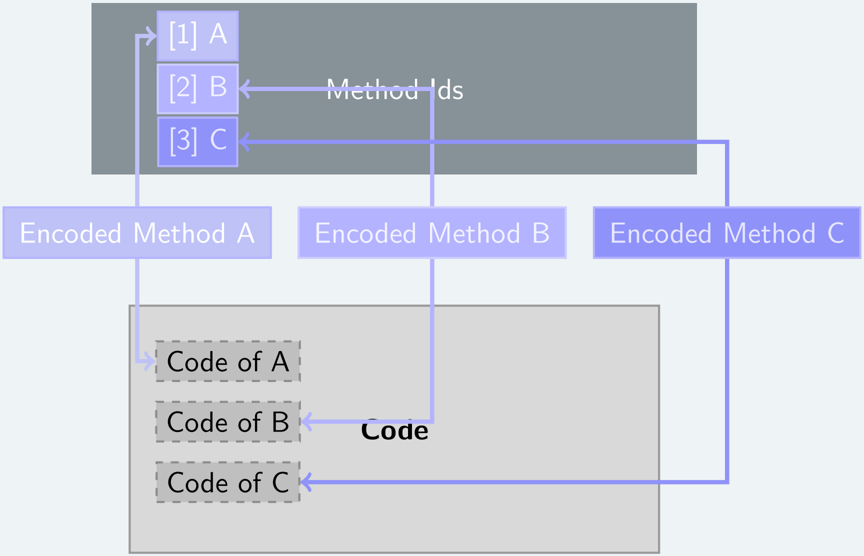
Visual representation of chaining



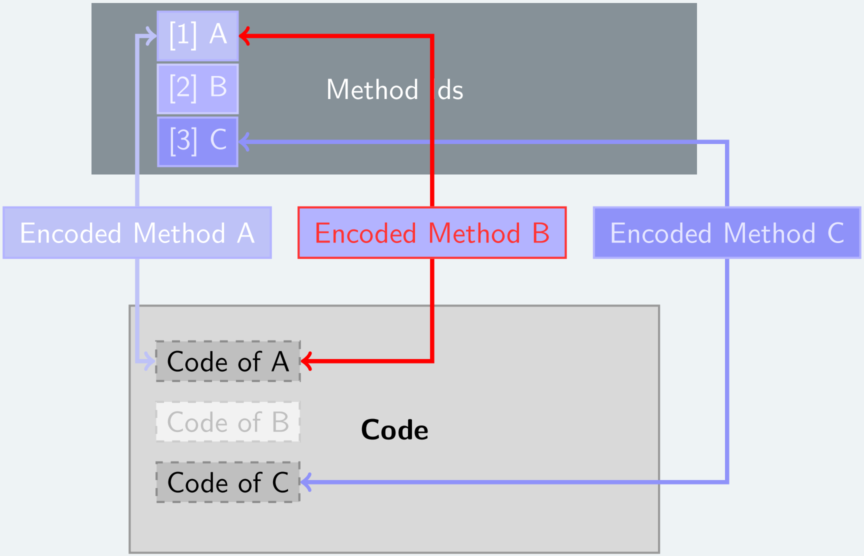
Visual representation of chaining



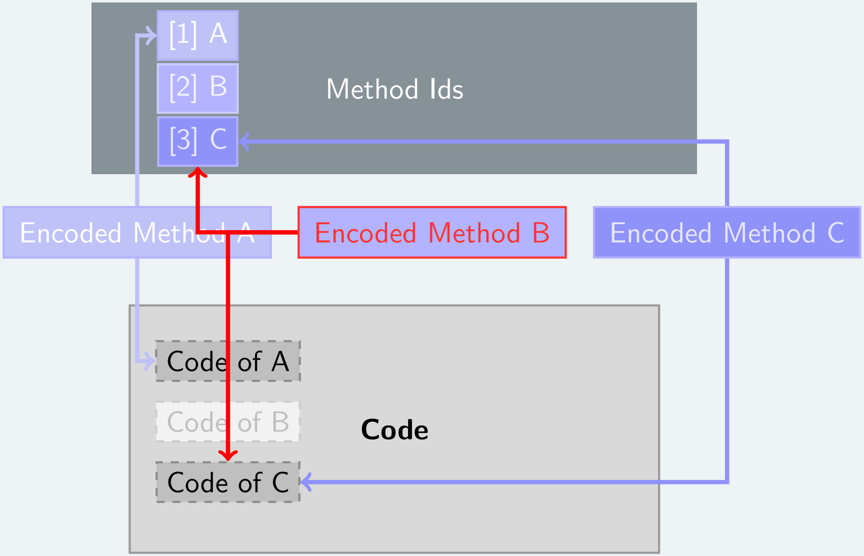
Visual representation of chaining



Visual representation of chaining



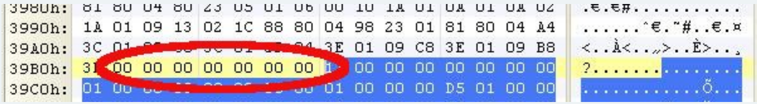
Visual representation of chaining



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



```
3980h: 81 80 04 80 23 05 01 06 00 10 1A 01 0A 01 0A 02  .€.€#...........
3990h: 1A 01 09 13 02 1C 88 80 04 98 23 01 81 80 04 A4  ......ˆ€.˜#..€.¤
39A0h: 3C 01 [illegible] 3E 01 09 C8 3E 01 09 B8  <..À<..„>..È>..¸
39B0h: 31 00 00 00 00 00 00 00 [illegible] 00 00 00 00 00 00 00  ?...............
39C0h: 01 00 [illegible] 01 00 00 00 D5 01 00 00  ............Õ...
```

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`

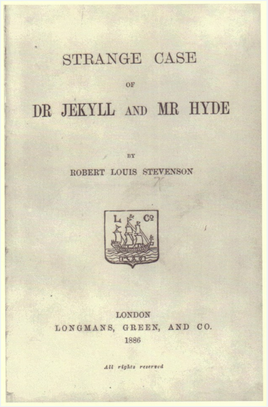
Re-build APK

- 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!


STRANGE CASE

OF

DR JEKYLL AND MR HYDE

BY

ROBERT LOUIS STEVENSON

LONDON

LONGMANS, GREEN, AND CO.

1886

All rights reserved

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

- Split personalities: Dr Jekyll or Mr Hyde
- Only one way to change into MrHyde: call thisishidden()
- Current personality displayed in main activity

Demo

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(...);
```

Implementation - Step 2/4

Patch the DEX

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

```
int patch_index = 0x2c99;
dex[patch_index++]= 1; // method_idx
dex[patch_index++]= 1; // access flag
dex[patch_index++]= (byte)0xcc; // code offset
dex[patch_index++]= (byte)0x28;
dex[patch_index++]= 1;
```

Implementation - Step 3/4

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);
```

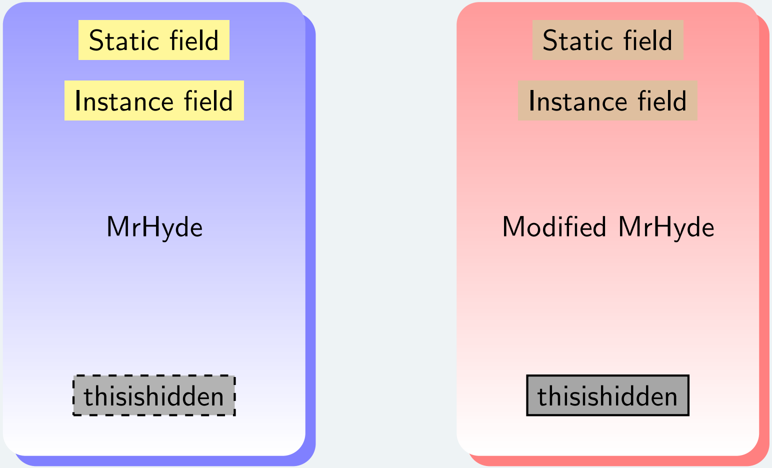
Implementation - Step 4/4

Invoke the hidden method

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

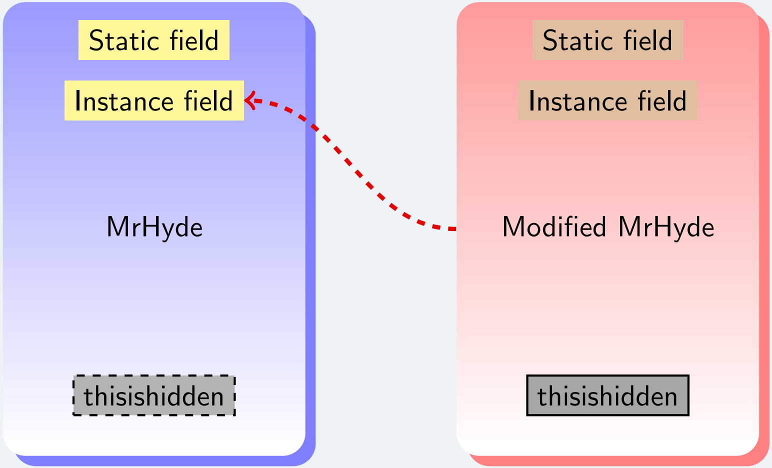
```
Object obj = patchedHyde.getDeclaredConstructor(Context.class)
               .newInstance(context);
Log.i("HideAndSeek", "after new Instance");
arg[0] = Boolean.valueOf(true);
Log.i("HideAndSeek", " invoking thisishidden()..
thisishiddenMethod.invoke(obj, arg);
```

It's two different classes

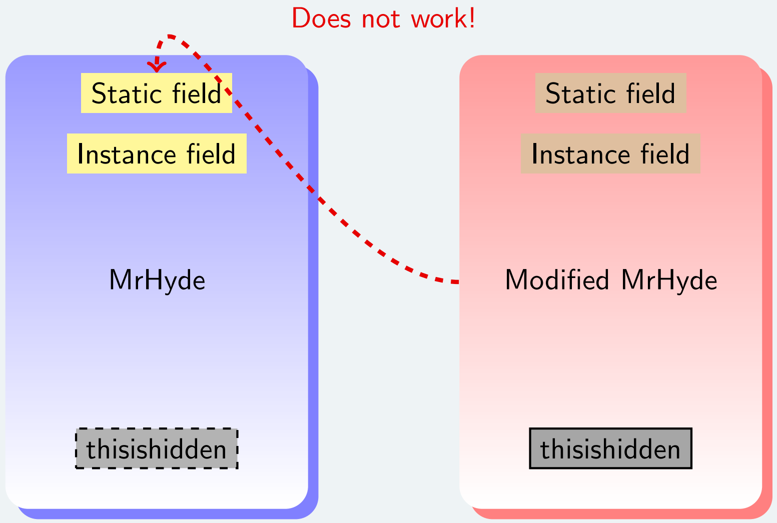


It's two different classes

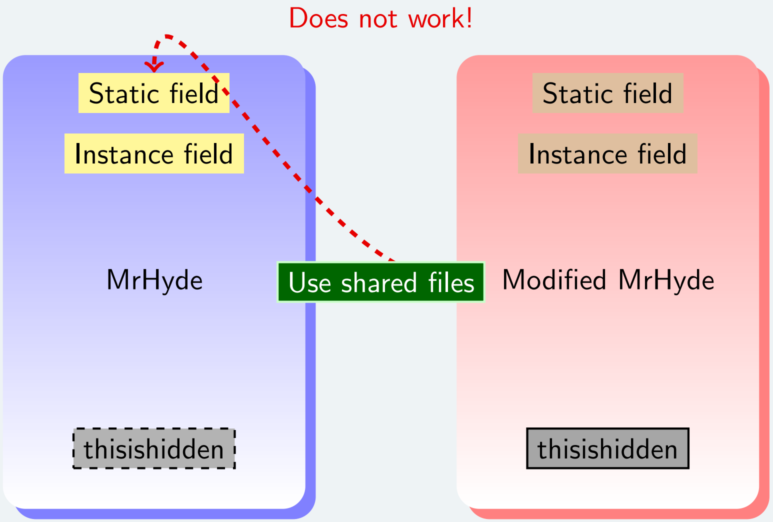
Does not work!



It's two different classes



It's two different classes



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 !

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

twitter: @cryptax
e-mail: aapvrille at fortinet dot com

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