Reverse Engineering in Android

Huỳnh Quang Thảo
Silicon Straits Saigon
Google Developer Expert on Android

What is reverse engineering

Why we need reverse engineering

- understand app behaviors.

Why we need reverse engineering

- understand app behaviors.
- get some necessary data.

Why we need reverse engineering

- understand app behaviors.
- get some necessary data.
- understand protocol between client and server.

Why we need reverse engineering

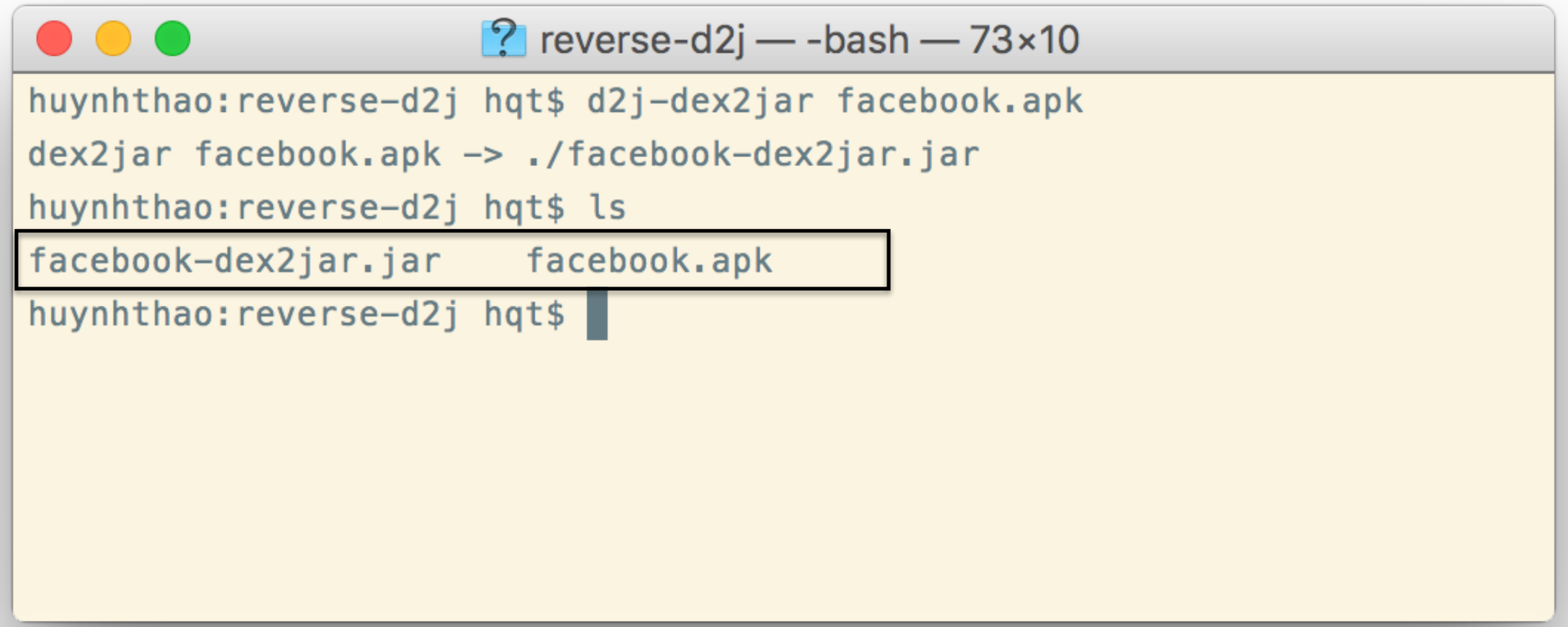
- understand app behaviors.
- get some necessary data.
- understand protocol between client and server.
- modify its behaviors.

Decompile app using Dex2Jar

d2j-dex2jar sample.apk

Decompile app using Dex2Jar

d2j-dex2jar sample.apk



SplashScreenActivity.class - Java Decompiler

facebook-dex2jar.jar

- X
- com
  - facebook
    - acra
    - analytics.appstatelogger
    - androidcompat
    - base
      - app
        - ApplicationLike.class
          - ApplicationLike
            - ApplicationLike()
            - a(int) : void
            - b() : void
            - c() : void
        - SplashScreenActivity.class
          - SplashScreenActivity
        - SplashScreenApplication$RedirectHack
      - init
    - breakpad
    - browser.lite
    - browserextensions.ipc
    - common
    - cpuprofiler
    - dalvikdistract
    - exoplayer.ipc
    - fbtrace
    - forker
    - intent.thirdparty
    - jni
    - katana.app
    - loom
    - nobreak
    - proxygen
    - quicklog
    - rti
    - secure.webkit
    - soloader

SplashScreenActivity.class

```
    localViewTreeObserver.addOnPreDrawListener(new 05t(this, localViewTreeObserver));
  }

  public final void onBackPressed()
  {
    setResult(1062849428);
    super.onBackPressed();
    ((000)getApplicationContext()).a(this, 7);
  }

  public void onCreate(Bundle paramBundle)
  {
    int i = Logger.a(2, 34, -1048628720);
    super.onCreate(null);
    paramBundle = getIntent();
    this.c = paramBundle.getLongExtra("com.facebook.base.app.splashId", 0L);
    this.d = paramBundle.getLongExtra("com.facebook.base.app.rhaId", 0L);
    Log.v("fb-SplashScreenActivity", String.format("SSA.onCreate ssaId:%x rhaId:%x", new Object[] { Long.valueOf(this.c)
    ((000)getApplicationContext()).a(this);
    if (!isFinishing()) {
      this.e = 1;
    }
    Logger.a(2, 35, 2009630515, i);
  }

  public void onDestroy()
  {
    int i = Logger.a(2, 34, -1340489811);
    this.e = 0;
    Log.v("fb-SplashScreenActivity", String.format("SSA.onDestroy ssaId:%x rhaId:%x", new Object[] { Long.valueOf(this.
    ((000)getApplicationContext()).y.remove(this);
    super.onDestroy();
    Logger.a(2, 35, -147085439, i);
  }

  public final void onPause()
  {
    int i = Logger.a(2, 34, 1688926624);
    this.e = 2;
```

SplashScreenActivity.class - Java Decompiler

facebook-dex2jar.jar

- X
- com
  - facebook
    - acra
    - analytics.appstatelogger
    - androidcompat
    - base
      - app
        - ApplicationLike.class
          - ApplicationLike
            - ApplicationLike()
            - a(int) : void
            - b() : void
            - c() : void
        - SplashScreenActivity.class
          - SplashScreenActivity
        - SplashScreenApplication$RedirectHack
      - init
    - breakpad
    - browser.lite
    - browserextensions.ipc
    - common
    - cpuprofiler
    - dalvikdistract
    - exoplayer.ipc
    - fbtrace
    - forker
    - intent.thirdparty
    - jni
    - katana.app
    - loom
    - nobreak
    - proxygen
    - quicklog
    - rti
    - secure.webkit
    - soloader

SplashScreenActivity.class

```
    localViewTreeObserver.addOnPreDrawListener(new 05t(this, localViewTreeObserver));
  }

  public final void onBackPressed()
  {
    setResult(1062849428);
    super.onBackPressed();
    ((000)getApplicationContext()).a(this, 7);
  }

  public void onCreate(Bundle paramBundle)
  {
    int i = Logger.a(2, 34, -1048628720);
    super.onCreate(null);
    paramBundle = getIntent();
    this.c = paramBundle.getLongExtra("com.facebook.base.app.splashId", 0L);
    this.d = paramBundle.getLongExtra("com.facebook.base.app.rhaId", 0L);
    Log.v("fb-SplashScreenActivity", String.format("SSA.onCreate ssaId:%x rhaId:%x", new Object[] { Long.valueOf(this.c)
    ((000)getApplicationContext()).a(this);
    if (!isFinishing()) {
      this.e = 1;
    }
    Logger.a(2, 35, 2009630515, i);
  }

  public void onDestroy()
  {
    int i = Logger.a(2, 34, -1340489811);
    this.e = 0;
    Log.v("fb-SplashScreenActivity", String.format("SSA.onDestroy ssaId:%x rhaId:%x", new Object[] { Long.valueOf(this.
    ((000)getApplicationContext()).y.remove(this);
    super.onDestroy();
    Logger.a(2, 35, -147085439, i);
  }

  public final void onPause()
  {
    int i = Logger.a(2, 34, 1688926624);
    this.e = 2;
```

FacebookSplashScreenActivity.class - Java Decompiler

facebook-dex2jar.jar

FacebookSplashScreenActivity.class

```
  ((FacebookApplication)getApplicationContext()).m.b("ColdStart/SplashScreenDisplay");
}

/* Error */
public final void onCreate(android.os.Bundle paramBundle)
{
  // Byte code:
  //   0: iconst_2
  //   1: bipush 34
  //   3: ldc 64
  //   5: invokestatic 70   com/facebook/loom/logger/Logger:a  (III)I
  //   8: istore_2
  //   9: aload_0
  //   10: invokevirtual 46 com/facebook/katana/app/FacebookSplashScreenActivity:getApplicationContext ()Landroid/content/Con
  //   13: checkcast 48     com/facebook/katana/app/FacebookApplication
  //   16: getfield 52 com/facebook/katana/app/FacebookApplication:m     LX/005;
  //   19: ldc 72
  //   21: ldc 73
  //   23: invokevirtual 78 X/006:a   (Ljava/lang/String;I)LX/00M;
  //   26: astore 5
  //   28: aconst_null
  //   29: astore 4
  //   31: aload_0
  //   32: aload_1
  //   33: invokespecial 80 com/facebook/base/init/GenericLogoSplashScreenActivity:onCreate  (Landroid/os/Bundle;)V
  //   36: aload_0
  //   37: invokespecial 82 com/facebook/katana/app/FacebookSplashScreenActivity:d ()V
  //   40: aload 5
  //   42: ifnull +12 -> 54
  //   45: iconst_0
  //   46: ifeq +25 -> 71
  //   49: aload 5
  //   51: invokevirtual 87 X/00M:close    ()V
  //   54: aload_0
  //   55: ldc 88
  //   57: iload_2
  //   58: invokestatic 93  X/01c:a   (Landroid/app/Activity;II)V
  //   61: return
  //   62: astore 1
```

- analytics.appstatelogger
- androidcompat
- base
- breakpad
- browser.lite
- browserextensions.ipc
- common
- cpuprofiler
- dalvikdistract
- exoplayer.ipc
- fbtrace
- forker
- intent.thirdparty
  - NativeThirdPartyUriHelper$Choos
    - NativeThirdPartyUriHelper$Ch
  - NativeThirdPartyUriHelper$Fbrpc
- jni
- katana.app
  - FacebookApplication.class
  - FacebookSplashScreenActivity.cl
    - FacebookSplashScreenActivity
- loom
- nobreak
- proxygen
- quicklog
- rti
- secure.webkit
  - WebView.class
- soloader
- sosource.bsod
- systrace
- tigon
- tools.dextr.runtime
- video.vps
- webpsupport
- xplat.fbglog
- xzdecoder

Reverse engineering using Android APK Tool

Reverse engineering using Android APK Tool

- Decompile directly to smali code.

Reverse engineering using Android APK Tool

- Decompile directly to smali code.
- Encode resource files.

Reverse engineering using Android APK Tool

- Decompile directly to smali code.
- Encode resource files.
- Recompile again to normal apk with updated code or resource.

Android bytecode introduction

Android bytecode datatype

V	Void
Z	Boolean
B	byte
S	short
C	char
I	int
J	long (64 bits)
F	float
D	double (64 bits)

Android bytecode datatype

Lpackage1/package2/className;

V	Void
Z	Boolean
B	byte
S	short
C	char
I	int
J	long (64 bits)
F	float
D	double (64 bits)

Android bytecode datatype

V	Void
Z	Boolean
B	byte
S	short
C	char
I	int
J	long (64 bits)
F	float
D	double (64 bits)

Lpackage1/package2/className;

```
package com.hqt.model;
class Person {
}
```

Android bytecode datatype

V	Void
Z	Boolean
B	byte
S	short
C	char
I	int
J	long (64 bits)
F	float
D	double (64 bits)

Lpackage1/package2/className;

```
package com.hqt.model;
class Person {
}
```

Lcom/hqt/model/Person;

Android bytecode datatype

V	Void
Z	Boolean
B	byte
S	short
C	char
I	int
J	long (64 bits)
F	float
D	double (64 bits)

Lpackage1/package2/className;

```
package com.hqt.model;
class Person {
}
```

Lcom/hqt/model/Person;

Android bytecode datatype

V	Void
Z	Boolean
B	byte
S	short
C	char
I	int
J	long (64 bits)
F	float
D	double (64 bits)

Lpackage1/package2/className;

```
package com.hqt.model;
class Person {
}
```

Lcom/hqt/model/ Person;

Android bytecode method

action {param1, param2}, LpackageName/ClassName->
methodName(paramType1; paramType2)ReturnType

Android bytecode method

action {param1, param2}, LpackageName/ClassName->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

const-string v1, "hqthao"

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

const-string v1, “hqthao”

const-string v2, “Hello World"

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

```
const-string v1, “hqthao”
const-string v2, “Hello World”

invoke-static {v1, v2},
```

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

```
const-string v1, “hqthao”
const-string v2, “Hello World”

invoke-static {v1, v2}, Landroid/util/Log;
```

Android bytecode method

action {param1, param2}, LpackageName/ClassName; -> **methodName(paramType1; paramType2)ReturnType**

Log.e(“hqthao”, “Hello World”)

const-string v1, “hqthao”
const-string v2, “Hello World”

invoke-static {v1, v2}, Landroid/util/Log; ->

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

```
const-string v1, “hqthao”
const-string v2, “Hello World”
invoke-static {v1, v2}, Landroid/util/Log;->

e(Ljava/lang/String;Ljava/lang/String;)I
```

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1 ; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

```
const-string v1, “hqthao”
const-string v2, “Hello World”
invoke-static {v1, v2}, Landroid/util/Log;->

e(Ljava/lang/String; Ljava/lang/String;)I
```

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

const-string v1, “hqthao”
const-string v2, “Hello World”
invoke-static {v1, v2}, Landroid/util/Log;->

e(Ljava/lang/String; Ljava/lang/String;)I

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

const-string v1, “hqthao”
const-string v2, “Hello World”
invoke-static {v1, v2}, Landroid/util/Log;->

e(Ljava/lang/String;Ljava/lang/String;)I

```
public static int e(String tag, String msg) {}
```

Android bytecode method

action {param1, param2}, LpackageName/ClassName;->
methodName(paramType1; paramType2)ReturnType

Log.e(“hqthao”, “Hello World”)

```
const-string v1, “hqthao”
const-string v2, “Hello World”
invoke-static {v1, v2}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
```

Android bytecode method

Static Function: com.utils.Utils.sum(long, int): long

Android bytecode method

Static Function: com.utils.Utils.sum(long, int): long

```
const v1, 4     // assign first param v1 to 4
```

Android bytecode method

Static Function: com.utils.Utils.sum(long, int): long

```
const v1, 4     // assign first param v1 to 4
const v3, 5     // assign second param v2 to 5
```

Android bytecode method

Static Function: com.utils.Utils.sum(long, int): long

```
const v1, 4     // assign first param v1 to 4
const v3, 5     // assign second param v2 to 5
invoke-static {v1, v3}, Lcom/utils/Utils;->sum(JI)I
```

Android bytecode method

Static Function: com.utils.Utils.sum(long, int): long

```
const v1, 4     // assign first param v1 to 4
const v3, 5     // assign second param v2 to 5
invoke-static {v1, v3}, Lcom/utils/Utils;->sum(JI)I
```

Android bytecode method

Static Function: com.utils.Utils.sum(long, int): long

```
const v1, 4     // assign first param v1 to 4
const v3, 5     // assign second param v2 to 5
invoke-static {v1, v3}, Lcom/utils/Utils;->sum(JI)I
invoke-static {v1, v2, v3}, Lcom/utils/Utils;->sum(JI)I
```

Android bytecode method

Method signature: methodName(paramType1, paramType2)

Android bytecode method

Method signature: methodName(paramType1, paramType2): paramType3

Android bytecode method

Method signature: methodName(paramType1, paramType2): paramType3

Landroid/util/Log;->e(Ljava/lang/String; L.java/lang/String):**I**

Landroid/util/Log;->e(Ljava/lang/String; L.java/lang/String):**V**

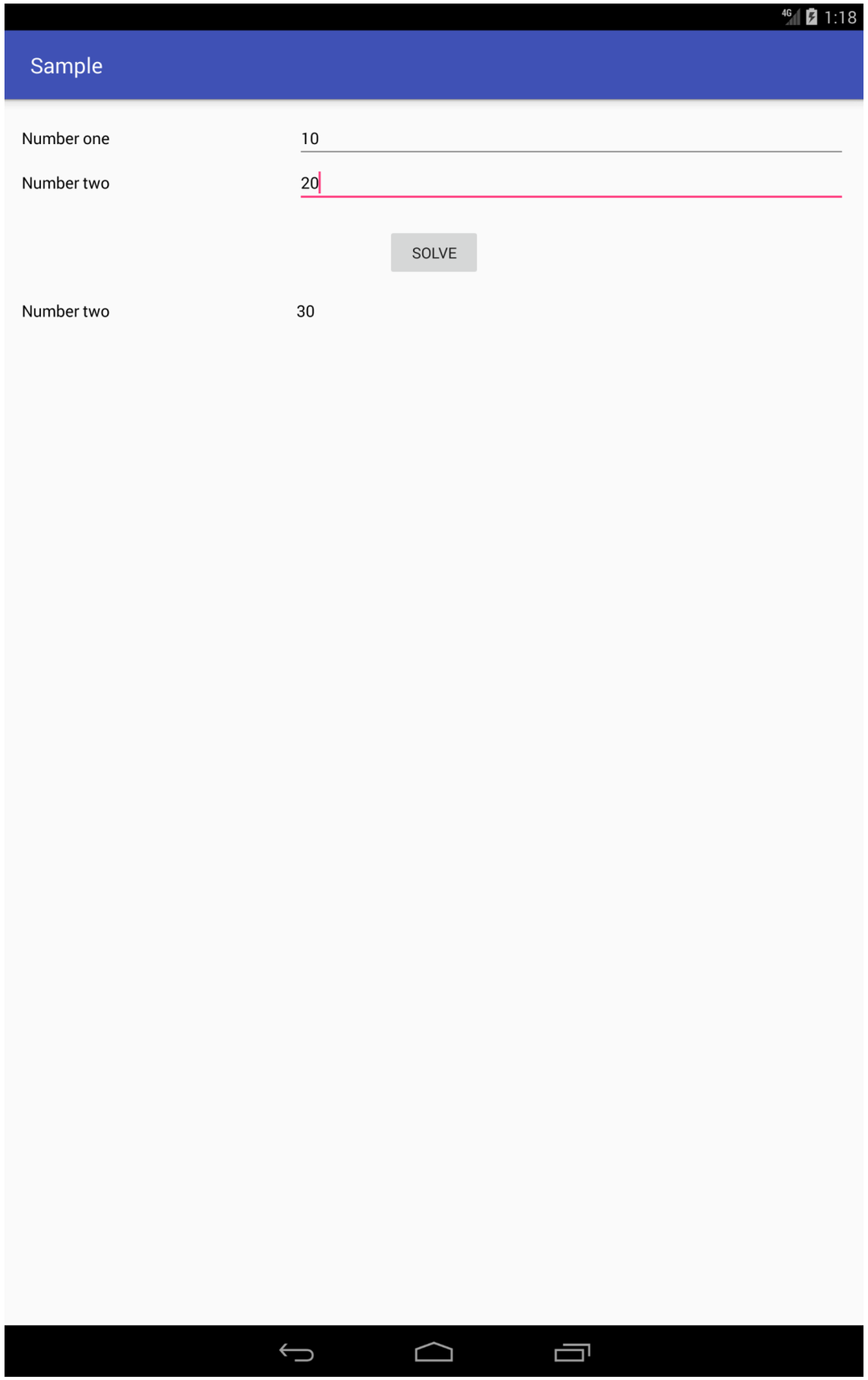
Android bytecode method

Method signature: methodName(paramType1, paramType2): paramType3

Landroid/util/Log;->e(Ljava/lang/String; Ljava/lang/String):**I**

Landroid/util/Log;->e(Ljava/lang/String; Ljava/lang/String):**V**

Demo


1:18
Sample
Number one
10
Number two
20
SOLVE
Number two
30

```
calcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int firstNumber = Integer.parseInt(firstNumberText.getText().toString());
        int secondNumber = Integer.parseInt(secondNumberText.getText().toString());
        ICalc calc = new CalcImpl();
        int result = calc.sum(firstNumber, secondNumber);
        resultText.setText(result + "");
    }
});
```

```
calcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int firstNumber = Integer.parseInt(firstNumberText.getText().toString());
        int secondNumber = Integer.parseInt(secondNumberText.getText().toString());
        ICalc calc = new CalcImpl();
        int result = calc.sum(firstNumber, secondNumber);
        resultText.setText(result + "");
    }
});
```

```java
calcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int firstNumber = Integer.parseInt(firstNumberText.getText().toString());
        int secondNumber = Integer.parseInt(secondNumberText.getText().toString());
        ICalc calc = new CalcImpl();
        int result = calc.sum(firstNumber, secondNumber);
        resultText.setText(result + "");
    }
});
```

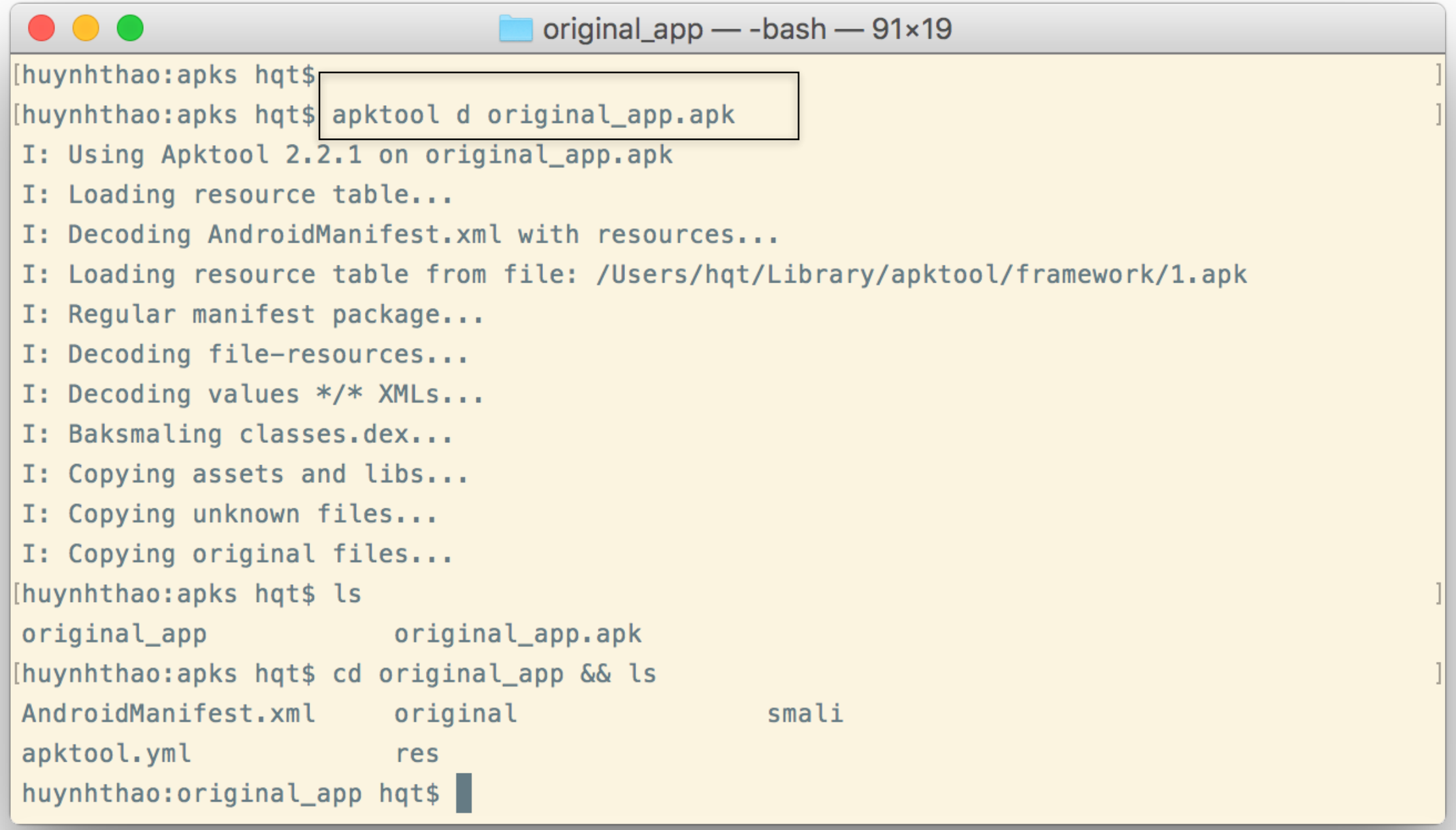
```java
public interface ICalc {
    int sum(int first, int second);
}
```

```java
calcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int firstNumber = Integer.parseInt(firstNumberText.getText().toString());
        int secondNumber = Integer.parseInt(secondNumberText.getText().toString());
        ICalc calc = new CalcImpl();
        int result = calc.sum(firstNumber, secondNumber);
        resultText.setText(result + "");
    }
});
```

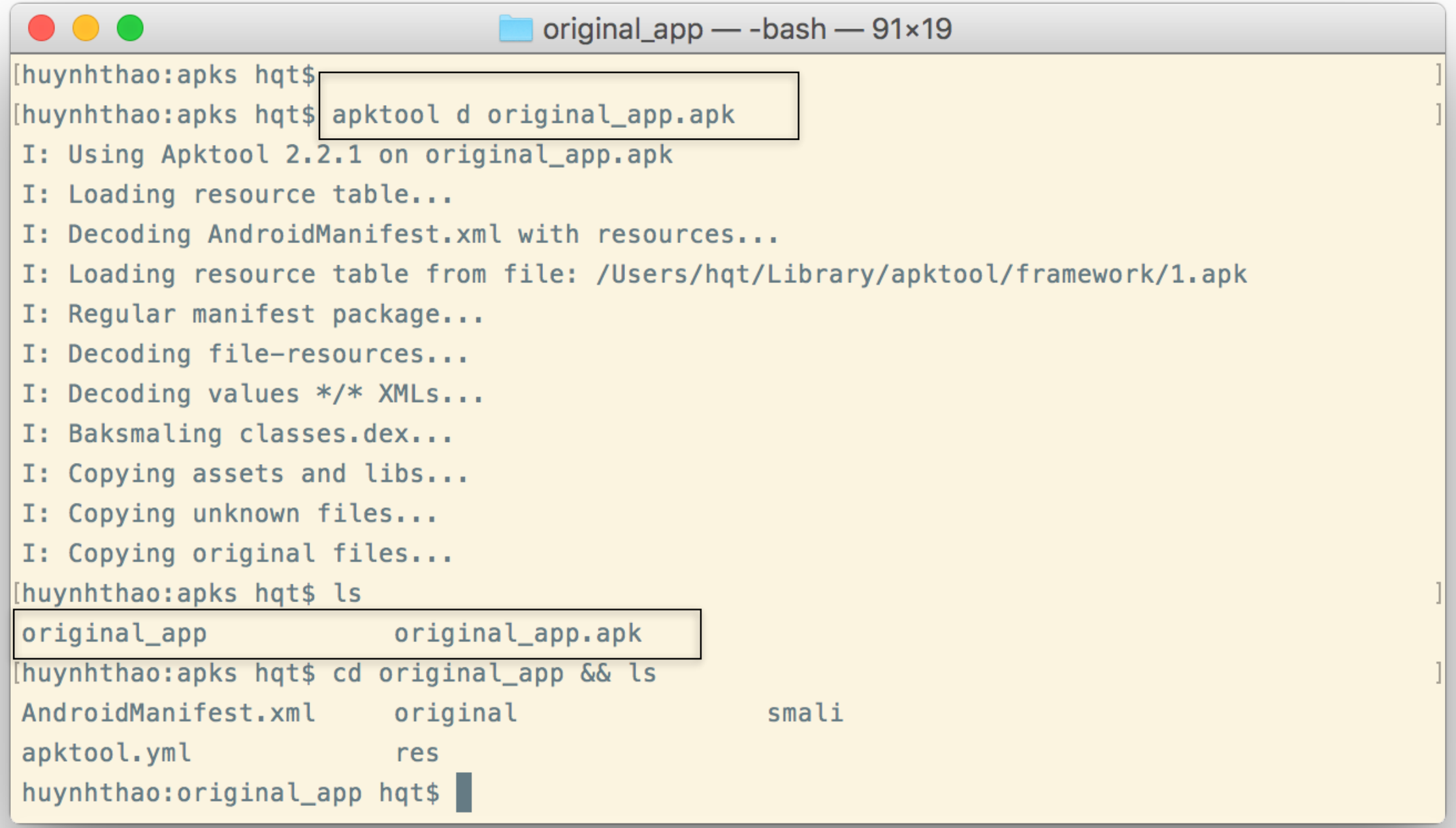
```java
public interface ICalc {
    int sum(int first, int second);
}
```

```java
public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

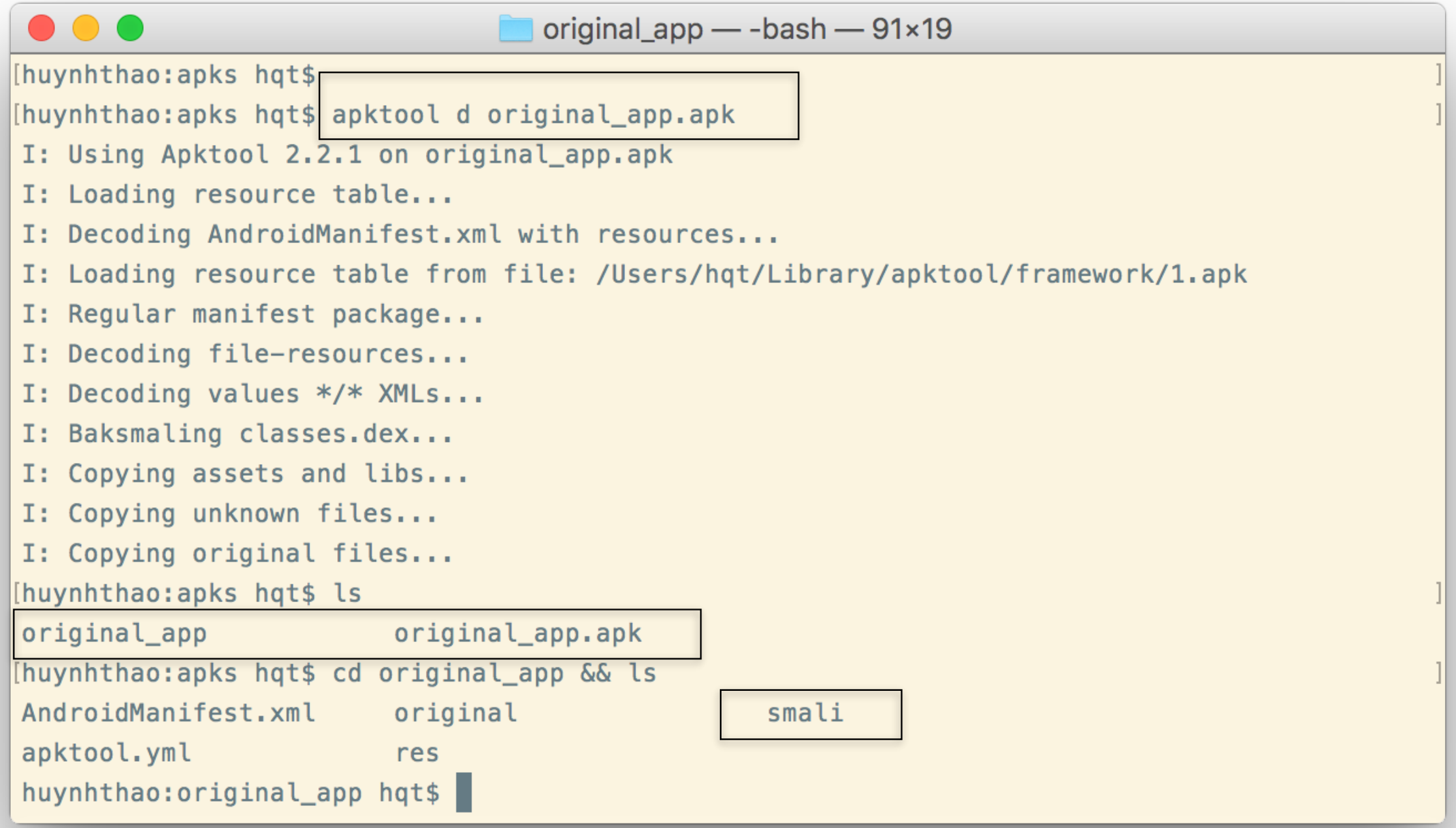
Step 2: decompile app
apktool d original_app.apk



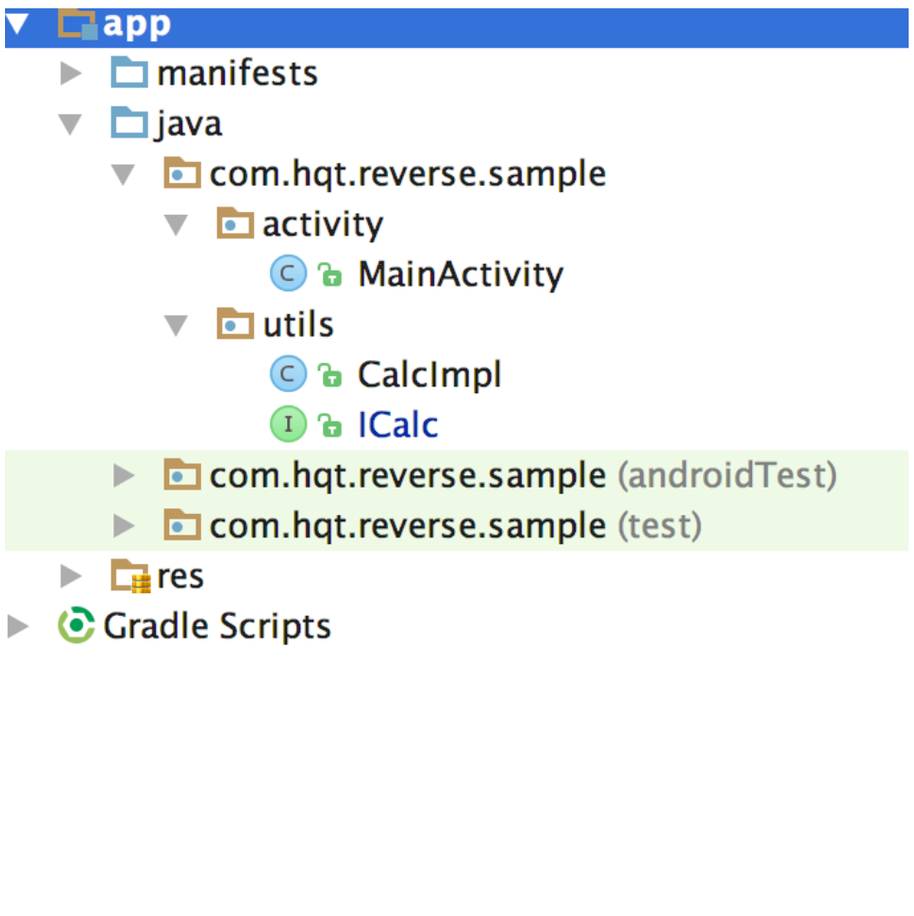
Step 2: decompile app
apktool d original_app.apk

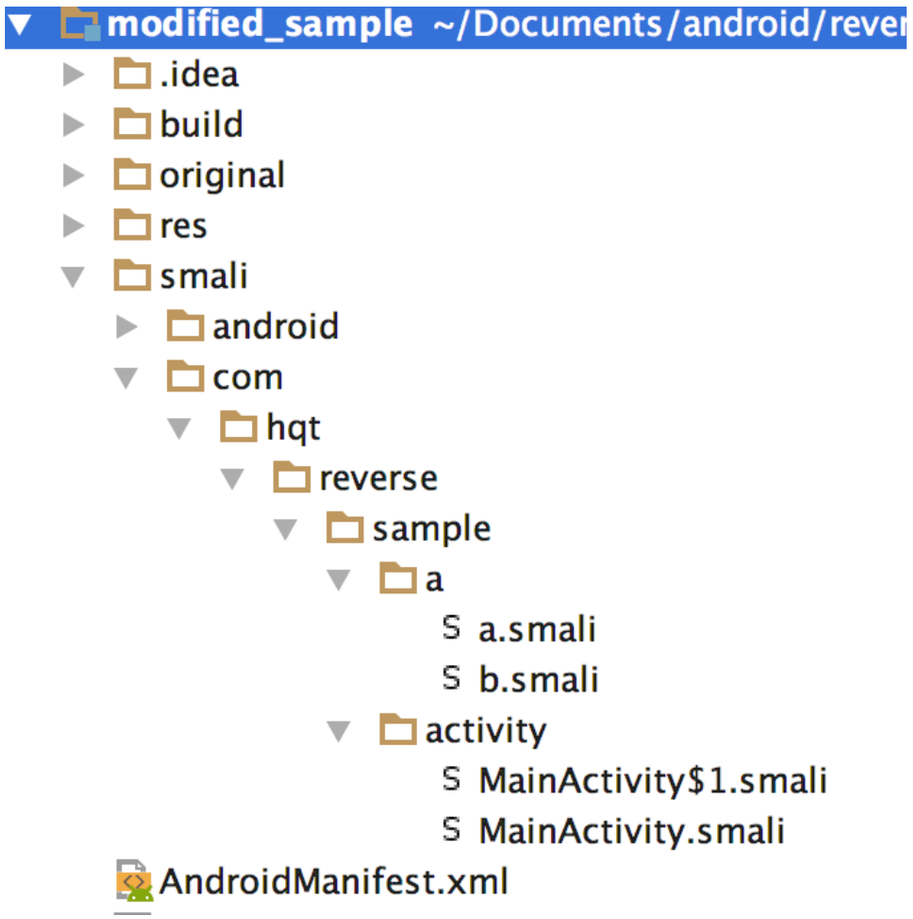


Step 2: decompile app
apktool d original_app.apk

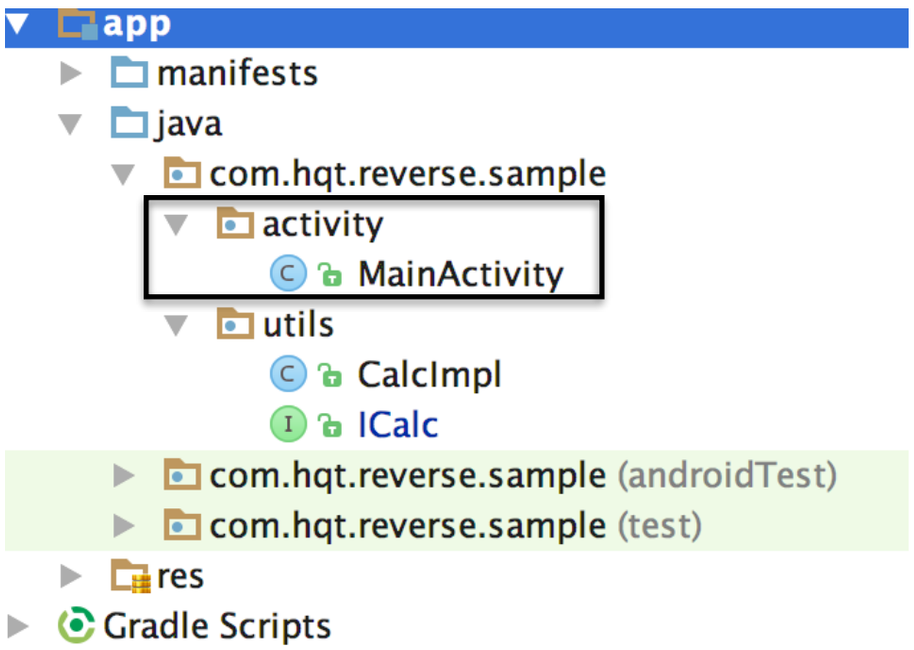


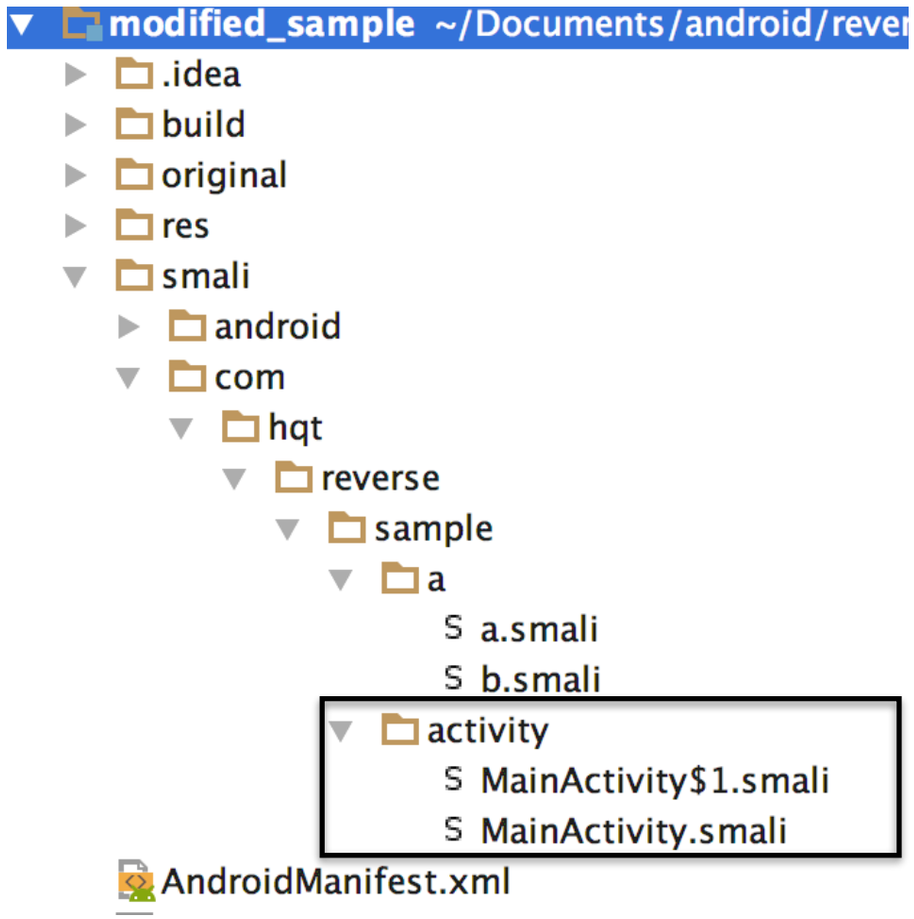
Step 3: Understand smali code





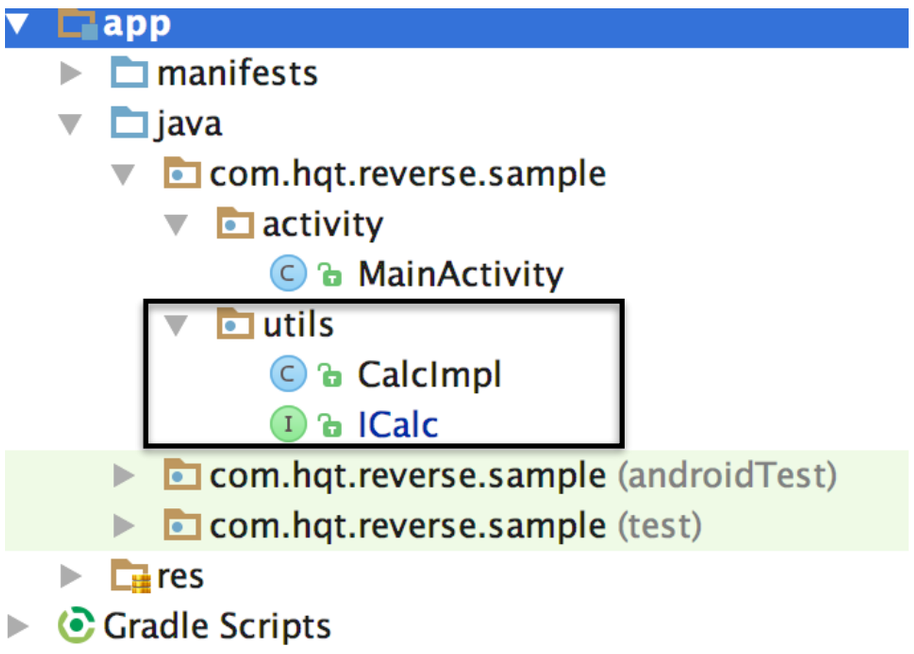
Step 3: Understand smali code

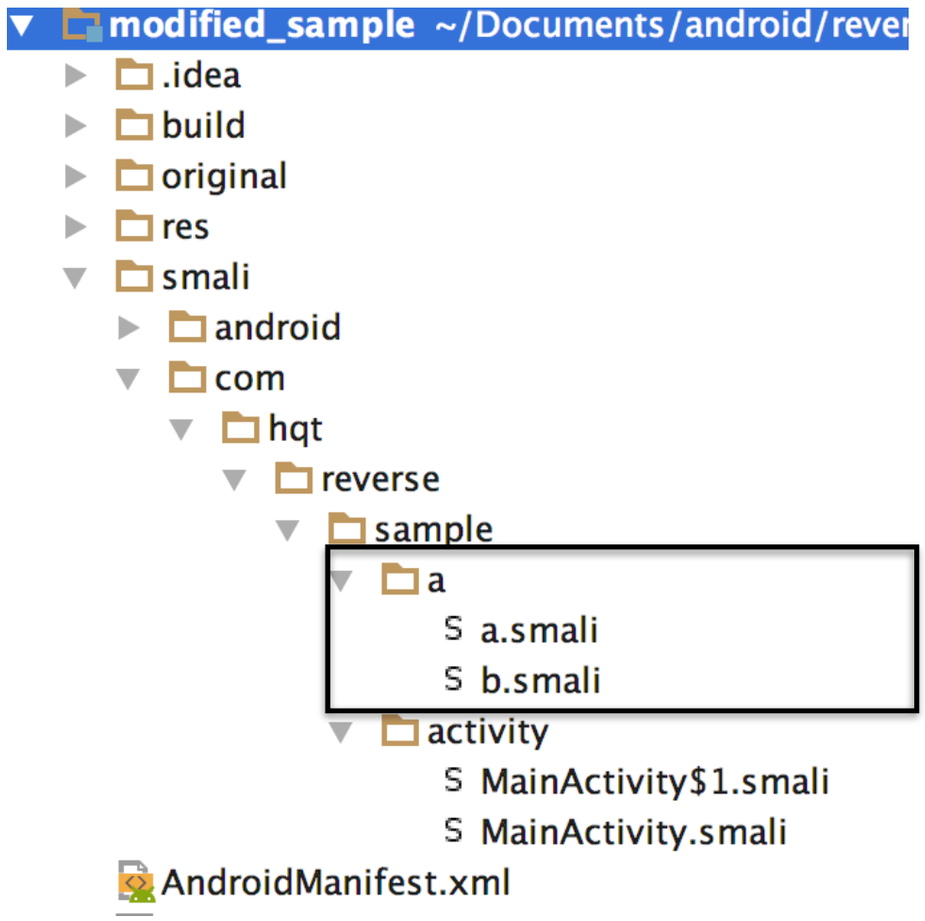




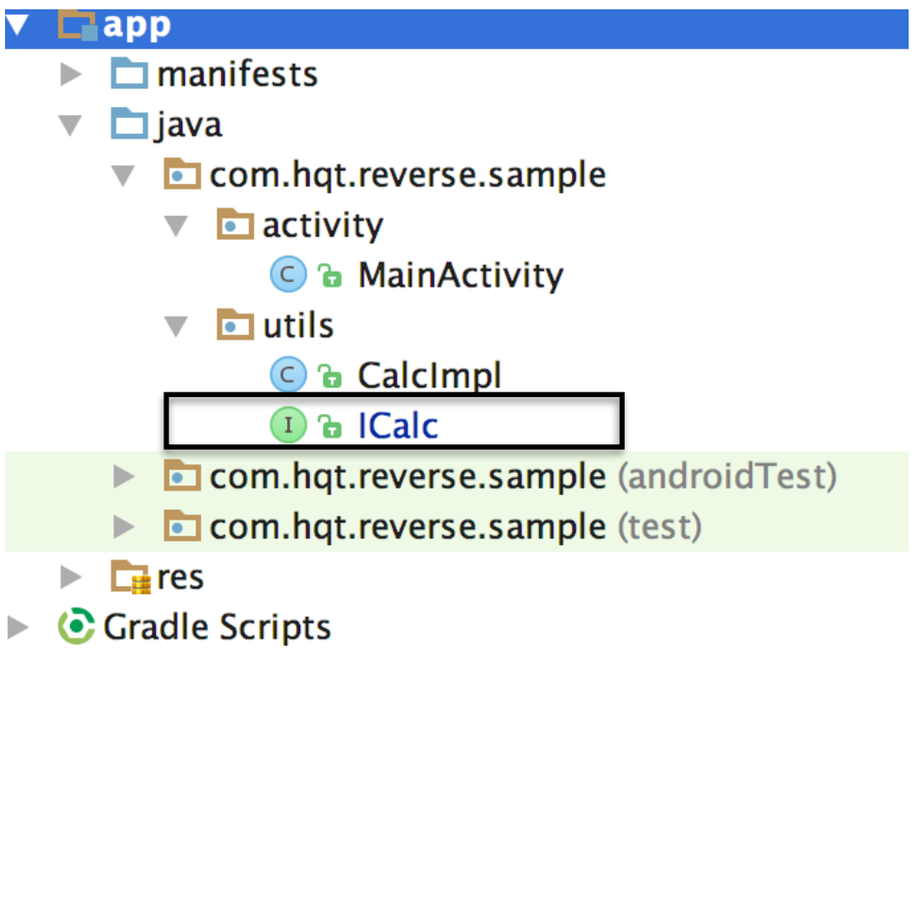

```
<activity android:name=".activity.MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" /
>
    </intent-filter>
</activity>
```

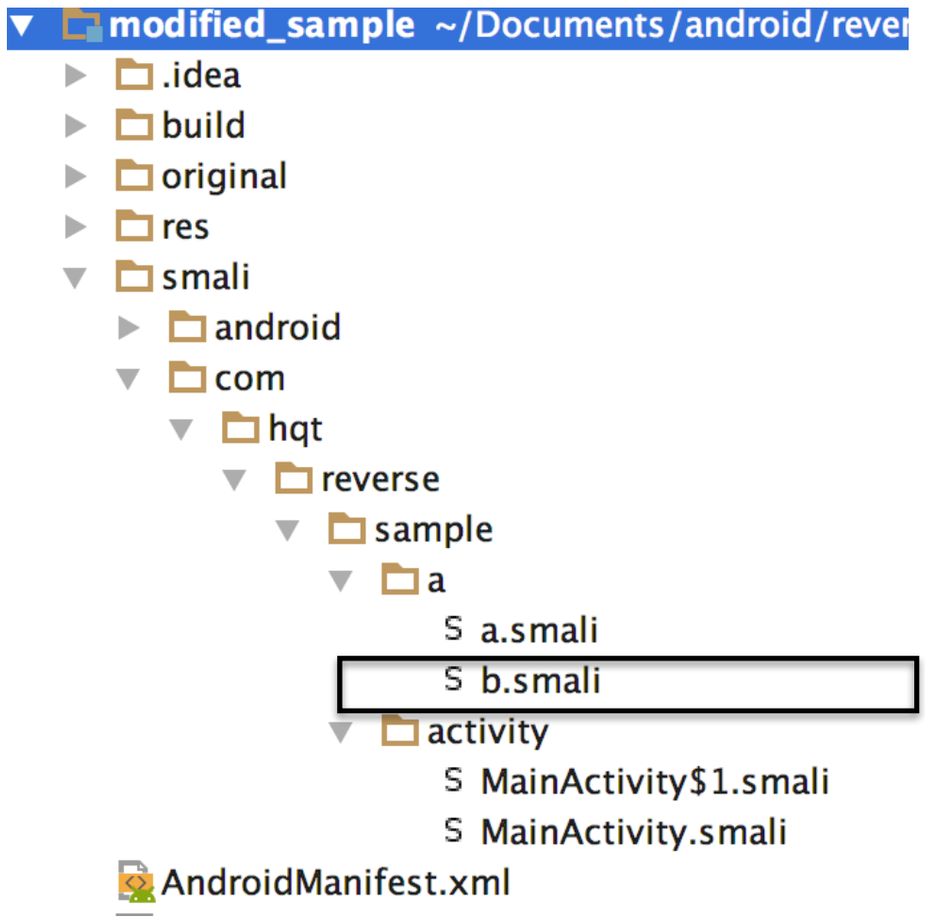
Step 3: Understand smali code



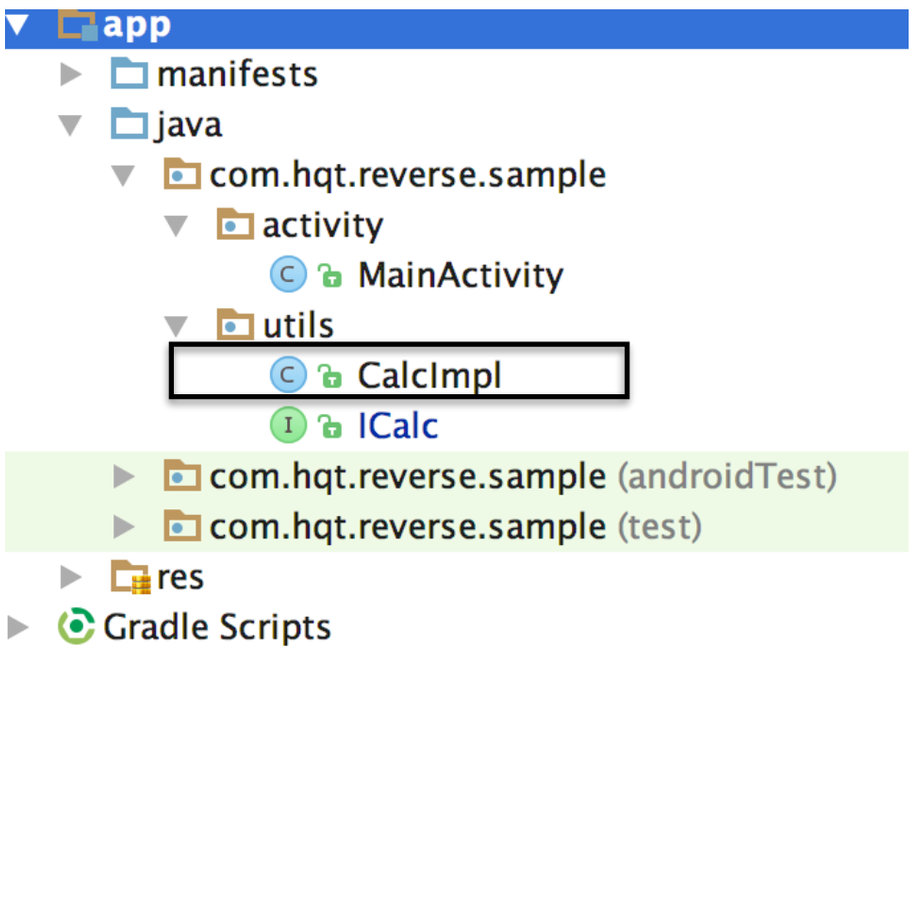


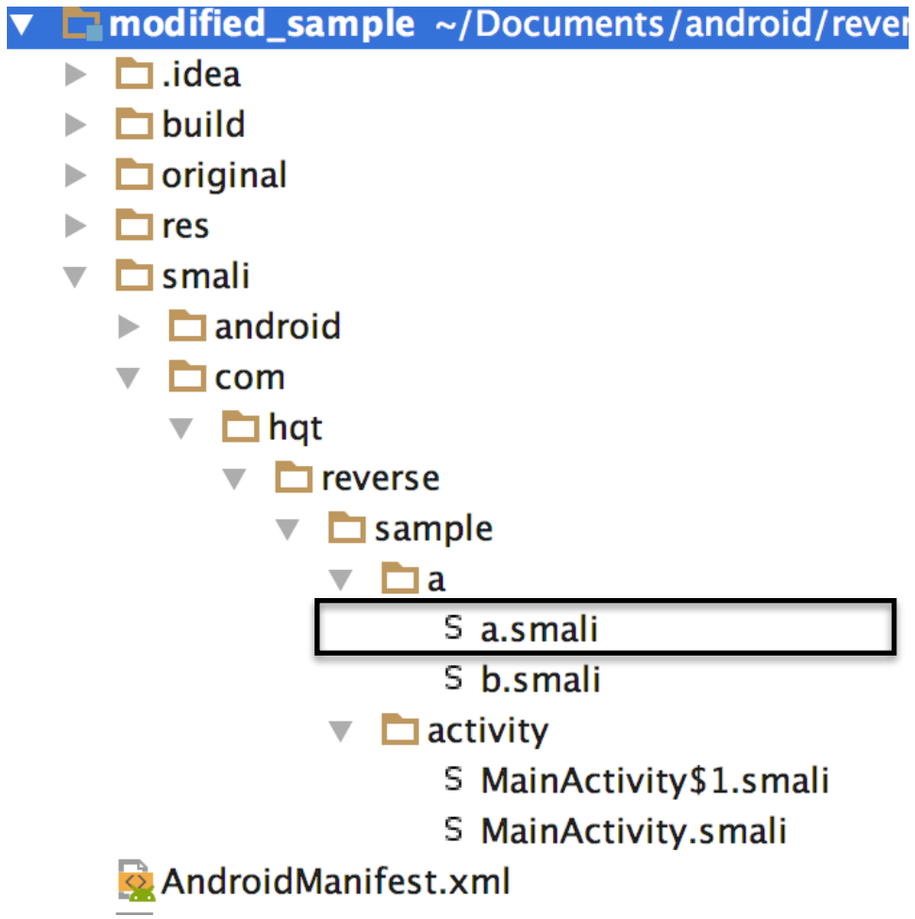
Step 3: Understand smali code





Step 3: Understand smali code





Step 3: Understand smali code

```
public interface ICalc {
    int sum(int first, int second);
}
```

```
.class public interface abstract Lcom/hqt/reverse/sample/a/b;
.super Ljava/lang/Object;


# virtual methods
.method public abstract a(II)I
.end method
```

Step 3: Understand smali code

```
public interface ICalc {
    int sum(int first, int second);
}
```

interface ICalc
became interface “b”

```
.class public interface abstract Lcom/hqt/reverse/sample/a/b;
.super Ljava/lang/Object;

# virtual methods
.method public abstract a(II)I
.end method
```

Step 3: Understand smali code

```java
public interface ICalc {
    int sum(int first, int second);
}
```

method sum became method “a”

```
.class public interface abstract Lcom/hqt/reverse/sample/a/b;
.super Ljava/lang/Object;

# virtual methods
.method public abstract a(II)I
.end method
```

Step 3: Understand smali code

```java
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

Step 3: Understand smali code

```java
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;


# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method

# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method
```

Step 3: Understand smali code

```
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

interface
com.hqt.reverse.sample.utils
became
com.hqt.reverse.a

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;


# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method

# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method
```

Step 3: Understand smali code

```java
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;


# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method

# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method
```

class CalcImpl became class a

Step 3: Understand smali code

```java
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

implements ICalc became implements Lcom/hqt/reverse/sample/a/b

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;


# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method


# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method
```

Step 3: Understand smali code

```java
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

Implicit constructor

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;
```

```
# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method
```

```
# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method
```

Step 3: Understand smali code

```java
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;


# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method
```

```
# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method
```

Step 3: Understand smali code

```java
public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
        return first + second;
    }
}
```

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;


# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method
```

using method add-int for sum

```
# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method
```

Step 4: Modifying smali code

```java
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;


# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method
```

z

```
# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method
```

using method mul-int for multiplication

```
.method public mul(II)I
    .locals 1
    mul-int v0, p1, p2
    return v0
.end method
```

Step 4: Modifying smali code

```java
package com.hqt.reverse.sample.utils;

public class CalcImpl implements ICalc {
    @Override
    public int sum(int first, int second) {
       return first + second;
    }
}
```

```
.class public Lcom/hqt/reverse/sample/a/a;
.super Ljava/lang/Object;

# interfaces
.implements Lcom/hqt/reverse/sample/a/b;


# direct methods
.method public constructor <init>()V
    .locals 0

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method
```

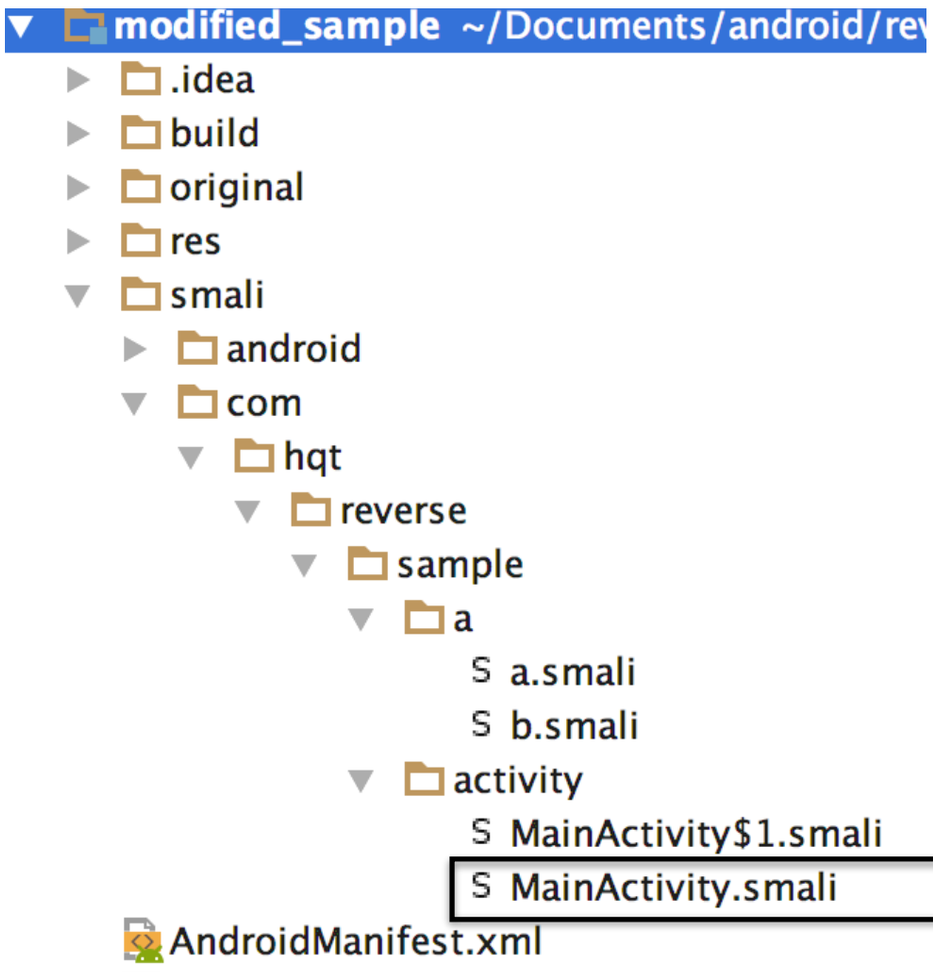
```
# virtual methods
.method public a(II)I
    .locals 1

    add-int v0, p1, p2

    return v0
.end method

.method public mul(II)I
    .locals 1

    mul-int v0, p1, p2

 return v0
.end method
```

Step 4: Modifying smali code

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    firstNumberText = (EditText) findViewById(R.id.first_number_text);
    secondNumberText = (EditText) findViewById(R.id.second_number_text);
    resultText = (TextView) findViewById(R.id.result_text);
    calcButton = (Button) findViewById(R.id.calc_button);

    calcButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            int firstNumber = Integer.parseInt(firstNumberText.getText().toString());
            int secondNumber = Integer.parseInt(secondNumberText.getText().toString());
            ICalc calc = new CalcImpl();
            int result = calc.sum(firstNumber, secondNumber);
            resultText.setText(result + "");
        }
    });
}
```

Step 4: Modifying smali code

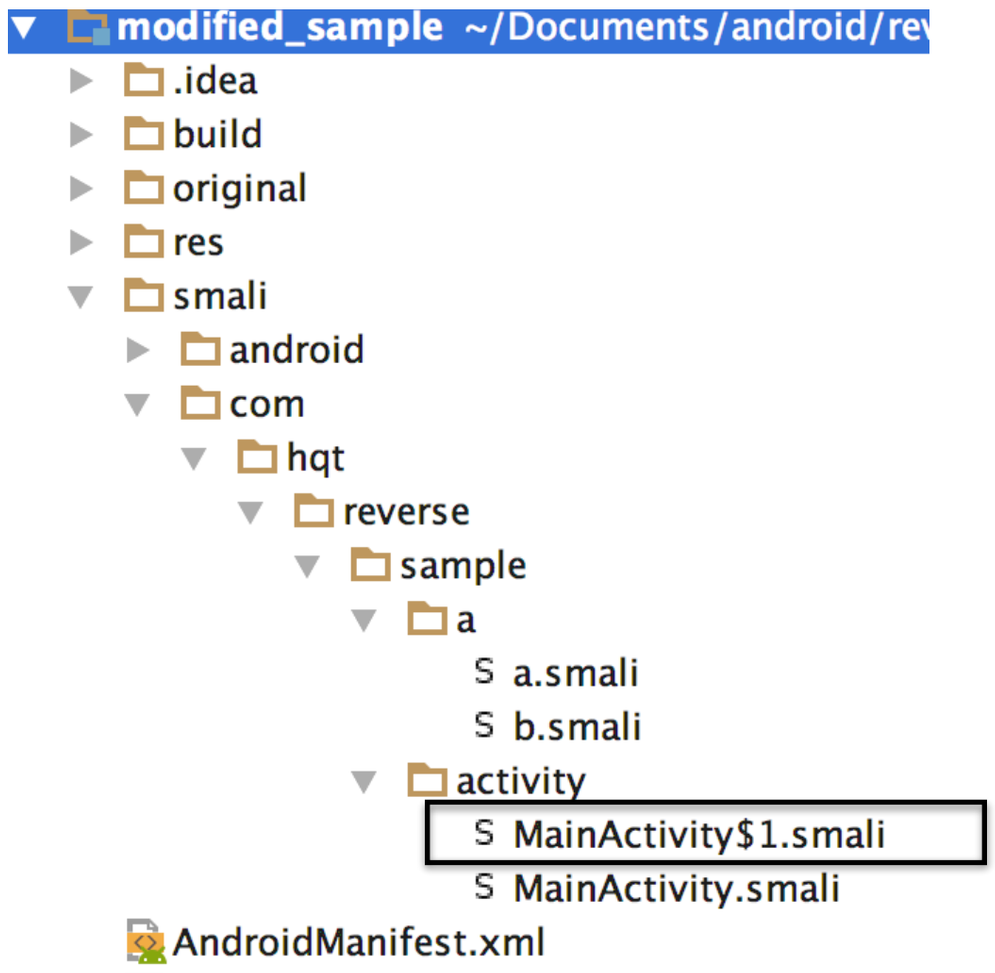
```
public class MainActivity extends AppCompatActivity {

    // variable definition

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // finding view

        calcButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                        }
        });
    }
}
```



Step 4: Modifying smali code

```
public class MainActivity extends AppCompatActivity {

    // variable definition

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // finding view

        calcButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
            }
        });

    }
}
```

Anonymous inner class became MainActivity$1

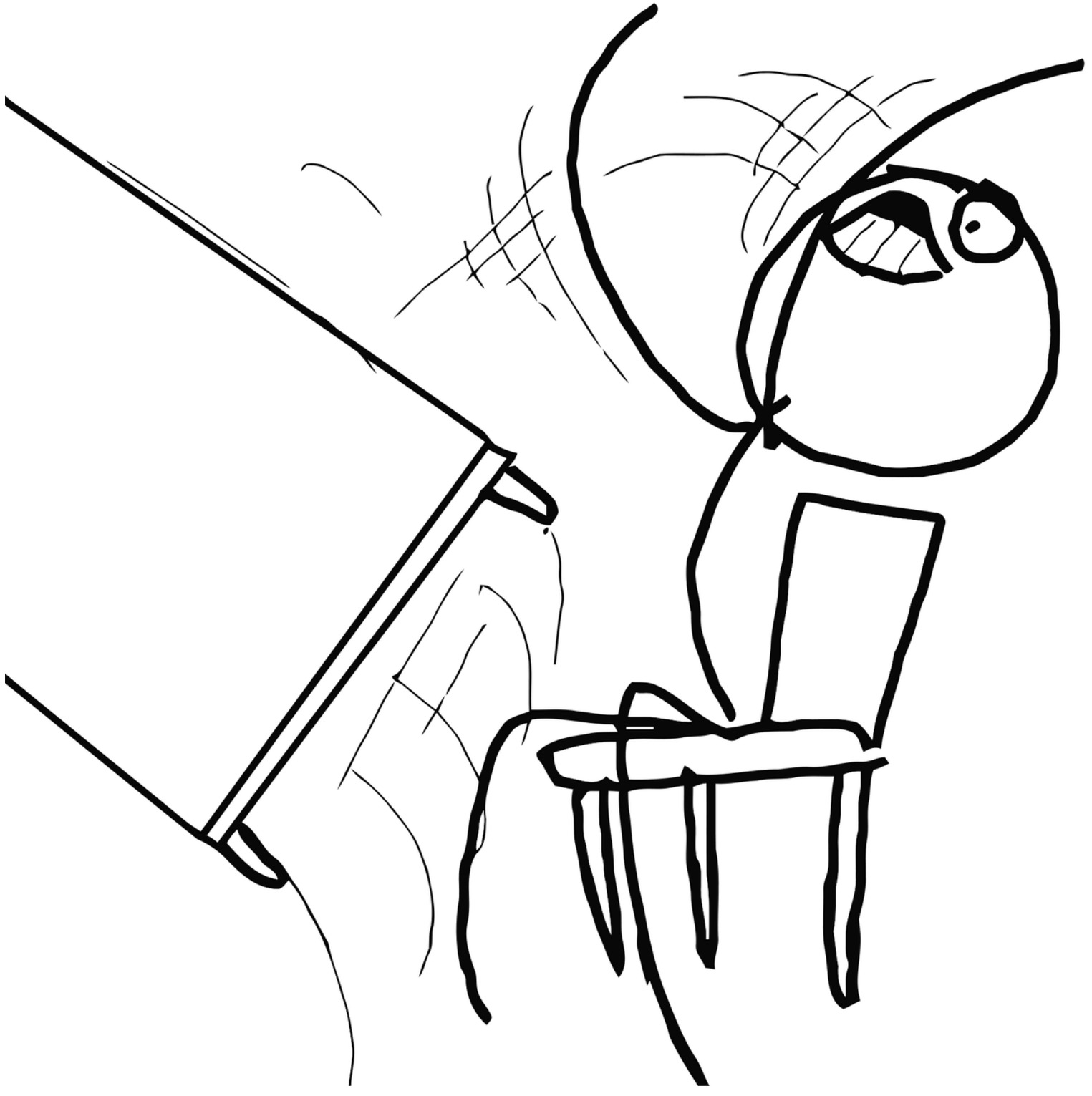


File $MainActivity$1.smali

```
# virtual methods
.method public onClick(Landroid/view/View;)V
    .locals 3

    iget-object v0, p0, Lcom/hqt/reverse/sample/activity/MainActivity$1;->a:Lcom/hqt/reverse/sample/activity/MainActivity;

    iget-object v0, v0, Lcom/hqt/reverse/sample/activity/MainActivity;->m:Landroid/widget/EditText;

    invoke-virtual {v0}, Landroid/widget/EditText;->getText()Landroid/text/Editable;

    move-result-object v0

    invoke-virtual {v0}, Ljava/lang/Object;->toString()Ljava/lang/String;

    move-result-object v0

    invoke-static {v0}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;)I

    move-result v0

    iget-object v1, p0, Lcom/hqt/reverse/sample/activity/MainActivity$1;->a:Lcom/hqt/reverse/sample/activity/MainActivity;

    iget-object v1, v1, Lcom/hqt/reverse/sample/activity/MainActivity;->n:Landroid/widget/EditText;

    invoke-virtual {v1}, Landroid/widget/EditText;->getText()Landroid/text/Editable;

    move-result-object v1

    invoke-virtual {v1}, Ljava/lang/Object;->toString()Ljava/lang/String;

    move-result-object v1

    invoke-static {v1}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;)I

    move-result v1

    new-instance v2, Lcom/hqt/reverse/sample/a/a;

    invoke-direct {v2}, Lcom/hqt/reverse/sample/a/a;-><init>()V

    invoke-interface {v2, v0, v1}, Lcom/hqt/reverse/sample/a/b;->mul(II)I

    move-result v0

    iget-object v1, p0, Lcom/hqt/reverse/sample/activity/MainActivity$1;->a:Lcom/hqt/reverse/sample/activity/MainActivity;
```

File $MainActivity$1.smali

```
# virtual methods
.method public onClick(Landroid/view/View;)V
    .locals 3

    iget-object v0, p0, Lcom/hqt/reverse/sample/activity/MainActivity$1;->a:Lcom/hqt/reverse/sample/activity/MainActivity;

    iget-object v0, v0, Lcom/hqt/reverse/sample/activity/MainActivity;->m:Landroid/widget/EditText;

    invoke-virtual {v0}, Landroid/widget/EditText;->getEditableText()Landroid/text/Editable;

    move-result-object v0

    invoke-virtual {v0}, Ljava/lang/Object;->toString()Ljava/lang/String;

    move-result-object v0

    invoke-static {v0}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;)I

    move-result v0

    iget-object v1, p0, Lcom/hqt/reverse/sample/activity/MainActivity$1;->a:Lcom/hqt/reverse/sample/activity/MainActivity;

    iget-object v1, v1, Lcom/hqt/reverse/sample/activity/MainActivity;->m:Landroid/widget/EditText;

    invoke-virtual {v1}, Landroid/widget/EditText;->getEditableText()Landroid/text/Editable;

    move-result-object v1

    invoke-virtual {v1}, Ljava/lang/Object;->toString()Ljava/lang/String;

    move-result-object v1

    invoke-static {v1}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;)I

    move-result v1

    new-instance v2, Lcom/hqt/reverse/sample/a/a;

    invoke-direct {v2}, Lcom/hqt/reverse/sample/a/a;-><init>()V

    invoke-interface {v2, v0, v1}, Lcom/hqt/reverse/sample/a/b;->mul(II)I

    move-result v0

    iget-object v1, p0, Lcom/hqt/reverse/sample/activity/MainActivity$1;->a:Lcom/hqt/reverse/sample/activity/MainActivity;
```



```java
calcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // some line before

        ICalc calc = new CalcImpl();
        int result = calc.sum(firstNumber, secondNumber);

        // some line after
    }
});
```

```
calcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // some line before

        ICalc calc = new CalcImpl();
        int result = calc.sum(firstNumber, secondNumber);

        // some line after
    }
});
```

Find <icalc_object>.sum(int, int)

```java
calcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // some line before

        ICalc calc = new CalcImpl();
        int result = calc.sum(firstNumber, secondNumber);

        // some line after
    }
});
```

Find <icalc_object>.sum(int, int)

- icalc_object -> Lcom/hqt/reverse/sample/a/b
- sum -> a
- (int, int) -> (I,I)

```
calcButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // some line before

        ICalc calc = new CalcImpl();
        int result = calc.sum(firstNumber, secondNumber);

        // some line after
    }
});
```

Find <icalc_object>.sum(int, int)

- icalc_object -> Lcom/hqt/reverse/sample/a/b
- sum -> a
- (int, int) -> (I,I)

Find Lcom/hqt/reverse/sample/a/b;->a(II)I

```
move-result-object v1

invoke-static {v1}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;)I

move-result v1

new-instance v2, Lcom/hqt/reverse/sample/a/a;

invoke-direct {v2}, Lcom/hqt/reverse/sample/a/a;-><init>()V

invoke-interface {v2, v0, v1}, Lcom/hqt/reverse/sample/a/b;->a(II)I

move-result v0
```

```
move-result-object v1
invoke-static {v1}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;)I
move-result v1
new-instance v2, Lcom/hqt/reverse/sample/a/a;
invoke-direct {v2}, Lcom/hqt/reverse/sample/a/a;-><init>()V
invoke-interface {v2, v0, v1}, Lcom/hqt/reverse/sample/a/b;->a(II)I
move-result v0
```

```
move-result v1
new-instance v2, Lcom/hqt/reverse/sample/a/a;
invoke-direct {v2}, Lcom/hqt/reverse/sample/a/a;-><init>()V
invoke-interface {v2, v0, v1}, Lcom/hqt/reverse/sample/a/b;->mul(II)I
move-result v0
```

```
move-result-object v1
invoke-static {v1}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;)I
move-result v1
new-instance v2, Lcom/hqt/reverse/sample/a/a;
invoke-direct {v2}, Lcom/hqt
invoke-interface {v2, v0, v1
move-result v0
```

```
move-result v1
new-instance v2, Lcom/h
invoke-direct {v2}, Lco
invoke-interface {v2, v
move-result v0
```

```
->mul(II)I
```

Step 4 Compile again apk file

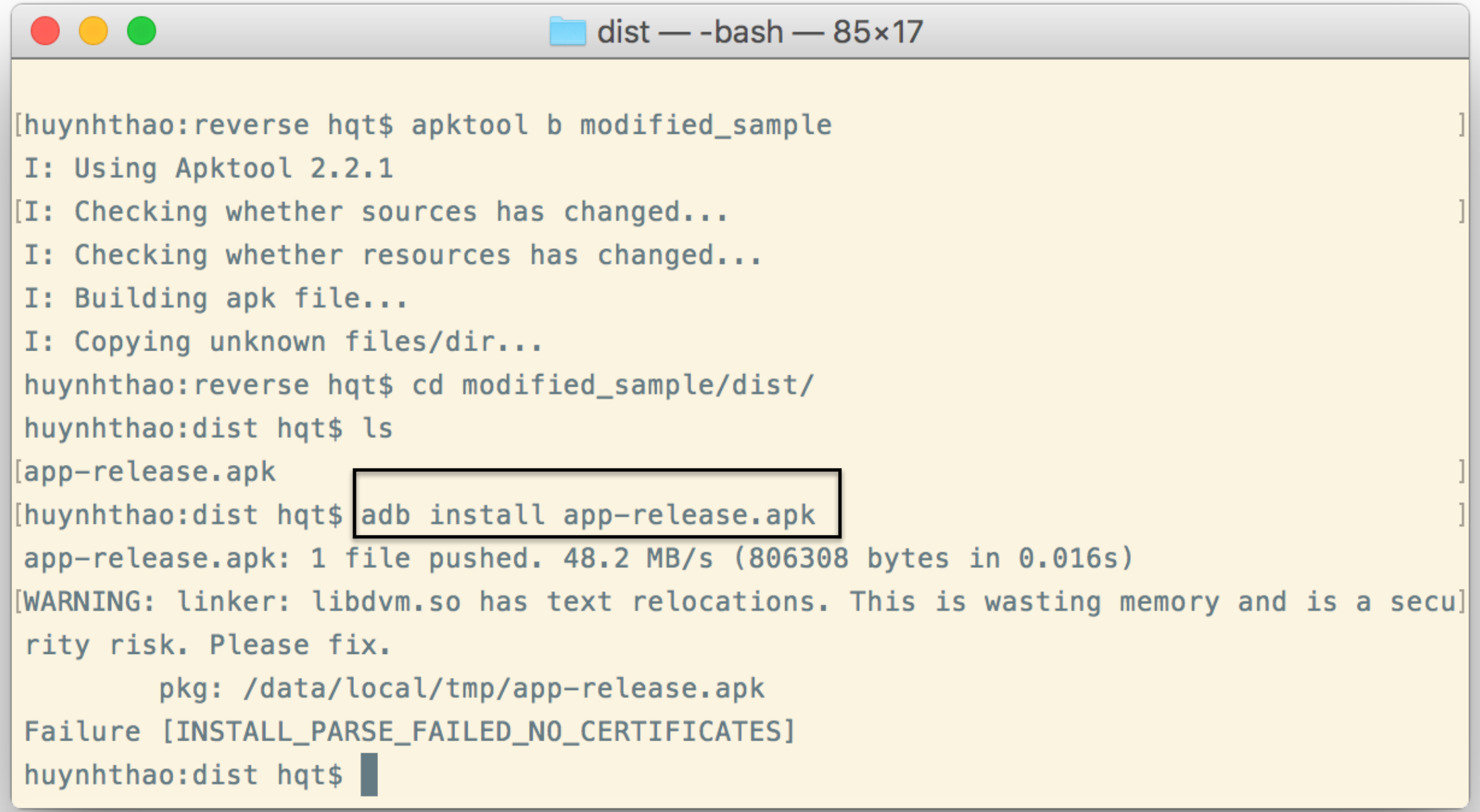
apktool b modified_apk_folder

Step 4 Compile again apk file

apktool b modified_apk_folder

Step 4 Compile again apk file

apktool b modified_apk_folder



Step 4 Compile again apk file

apktool b modified_apk_folder

Step 5 Sign again apk file

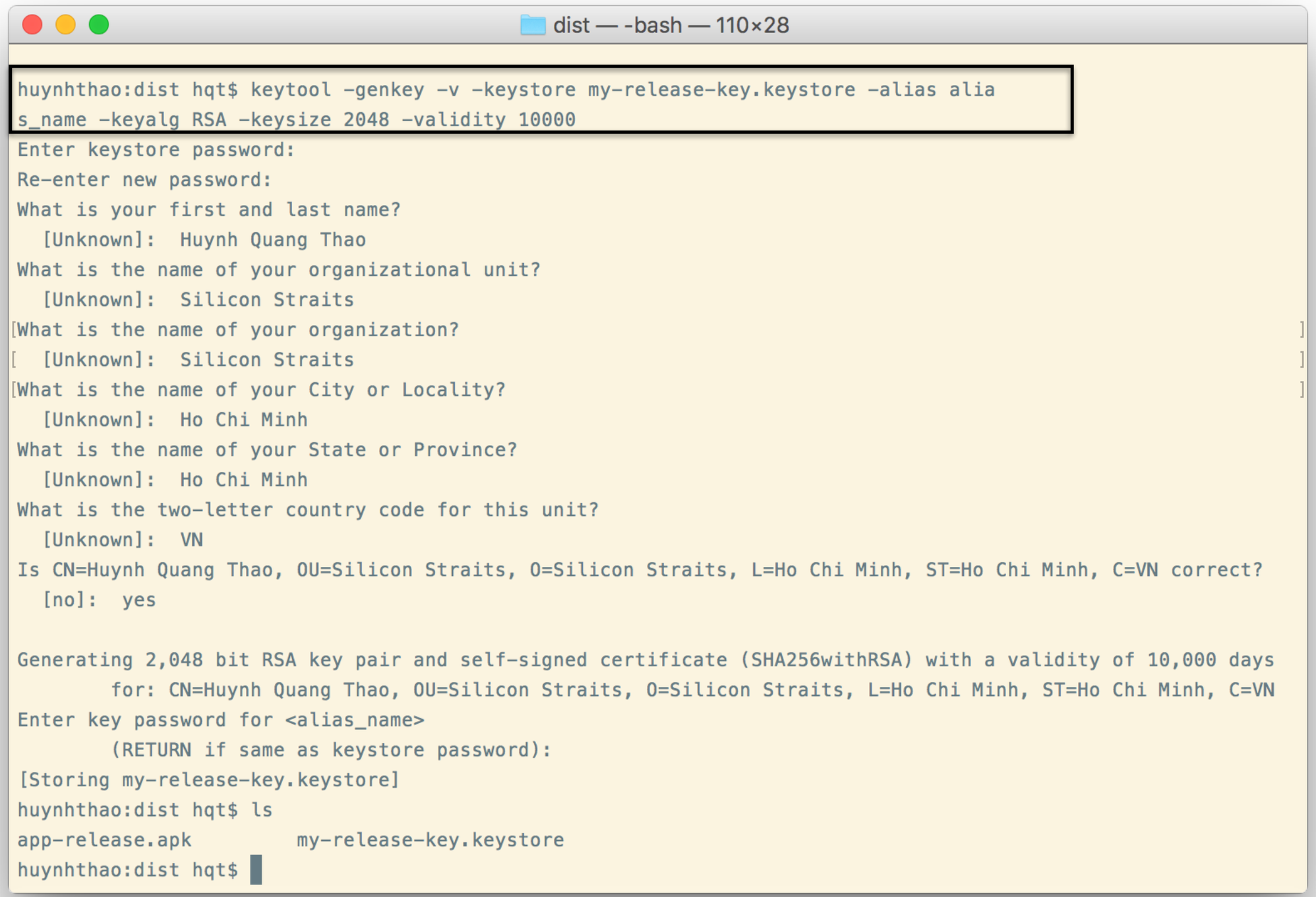
Step 1: Generate keystore:

keytool -genkey -v -keystore **my-release-key.keystore**
-alias alias_name -keyalg RSA -keysize 2048 -validity 10000

Step 5 Sign again apk file

Step 1: Generate keystore:

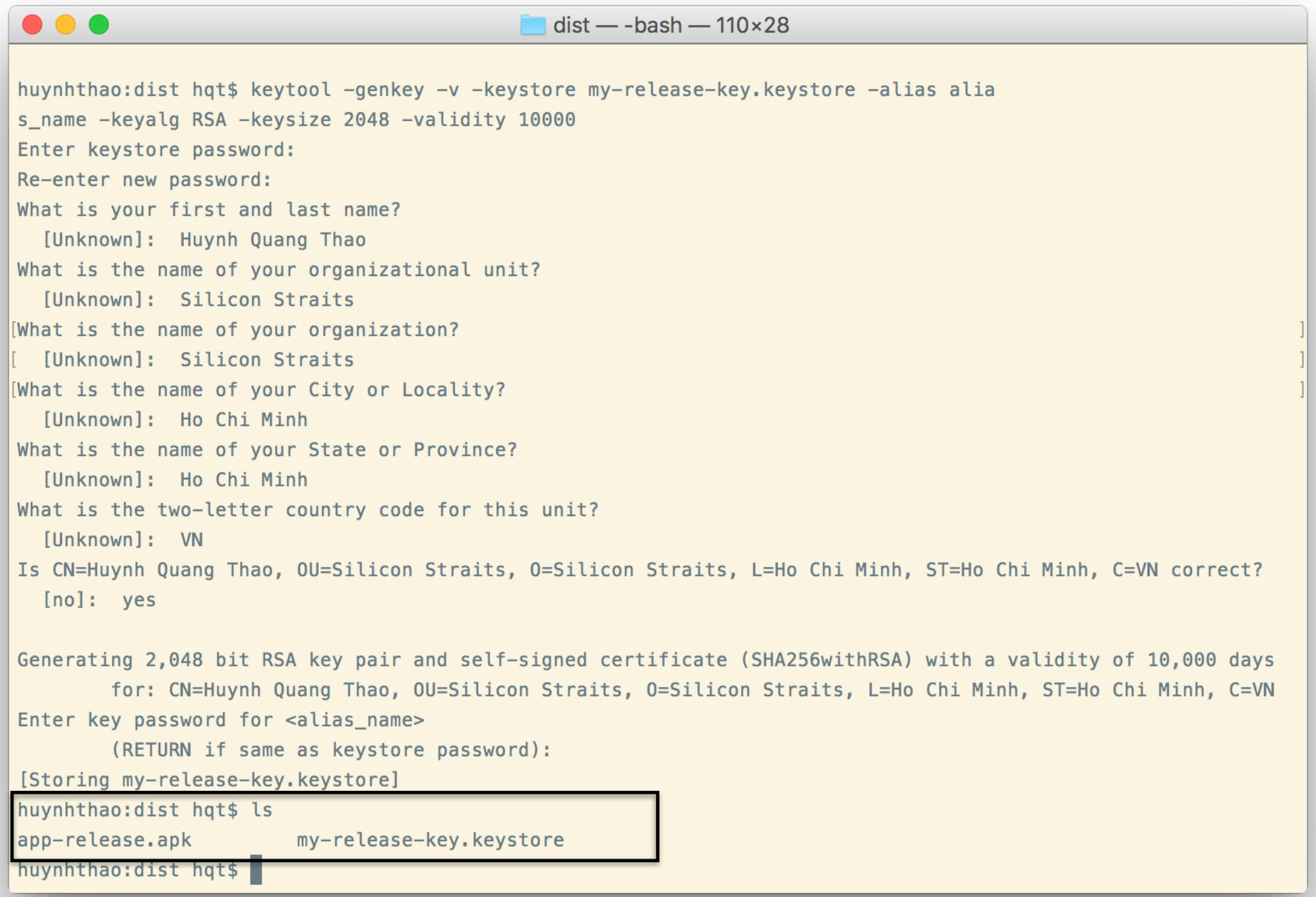
keytool -genkey -v -keystore **my-release-key.keystore**
-alias alias_name -keyalg RSA -keysize 2048 -validity 10000



Step 5 Sign again apk file

Step 1: Generate keystore:

keytool -genkey -v -keystore **my-release-key.keystore**
-alias alias_name -keyalg RSA -keysize 2048 -validity 10000



Step 5 Sign again apk file

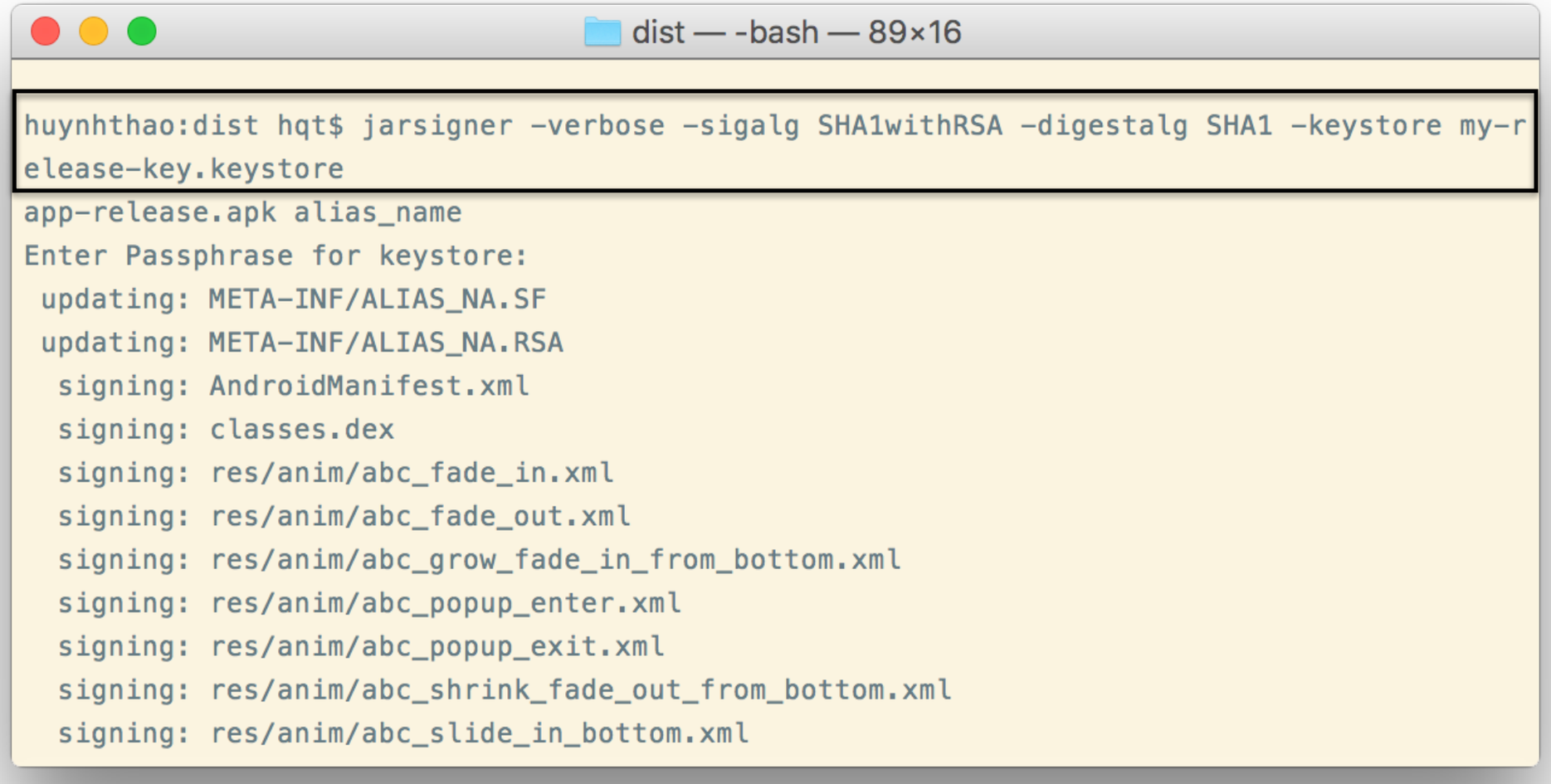
Step 2: Sign apk with generated keystone

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
-keystore **my-release-key.keystore**
my_application.apk alias_name

Step 5 Sign again apk file

Step 2: Sign apk with generated keystone

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
-keystore **my-release-key.keystore**
my_application.apk alias_name



Step 5 Sign again apk file

Step 2: Sign apk with generated keystone

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
-keystore **my-release-key.keystore**
my_application.apk alias_name

Step 5 Sign again apk file

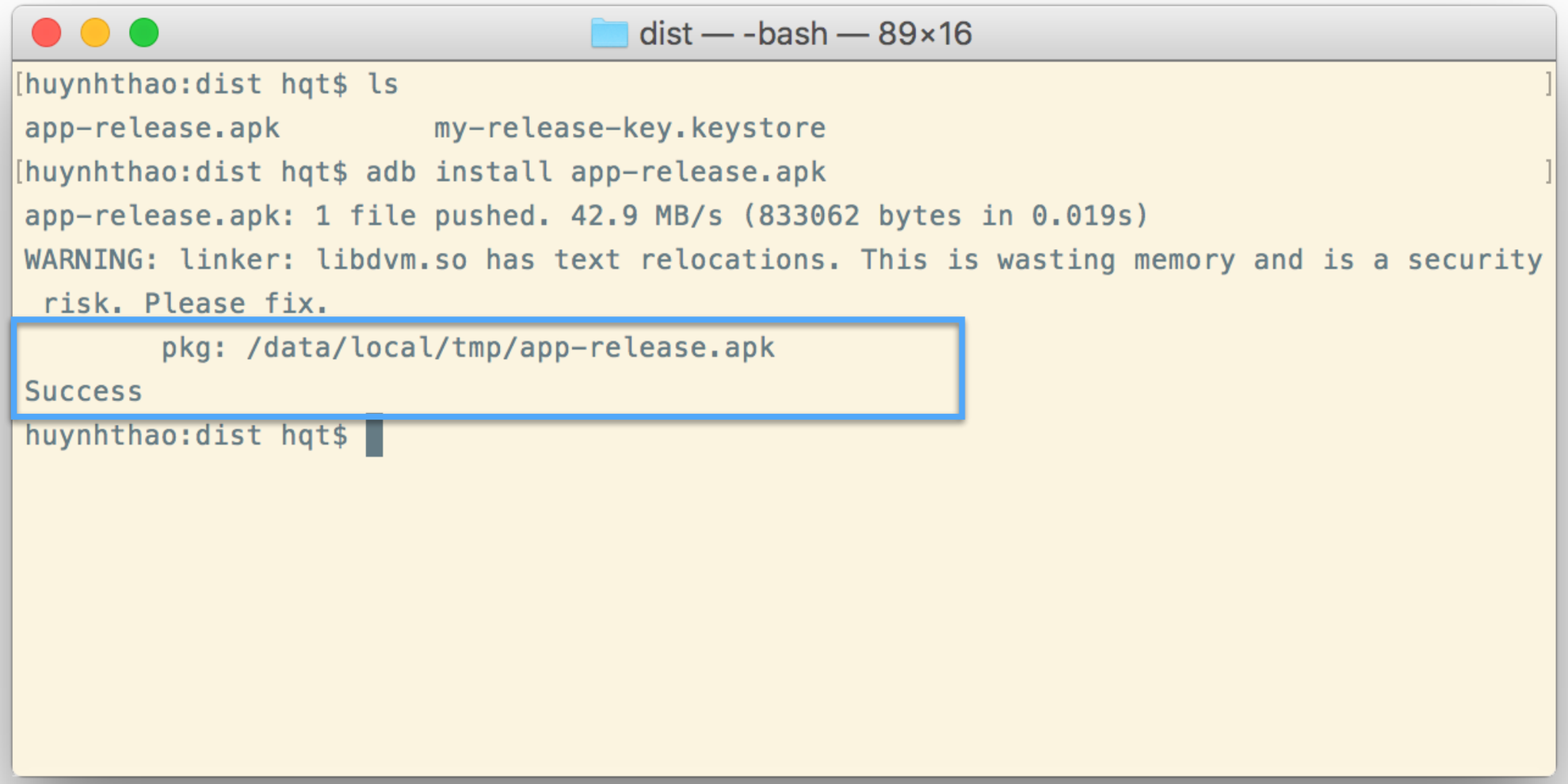
Step 2: Sign apk with generated keystone

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
-keystore **my-release-key.keystore**
my_application.apk alias_name

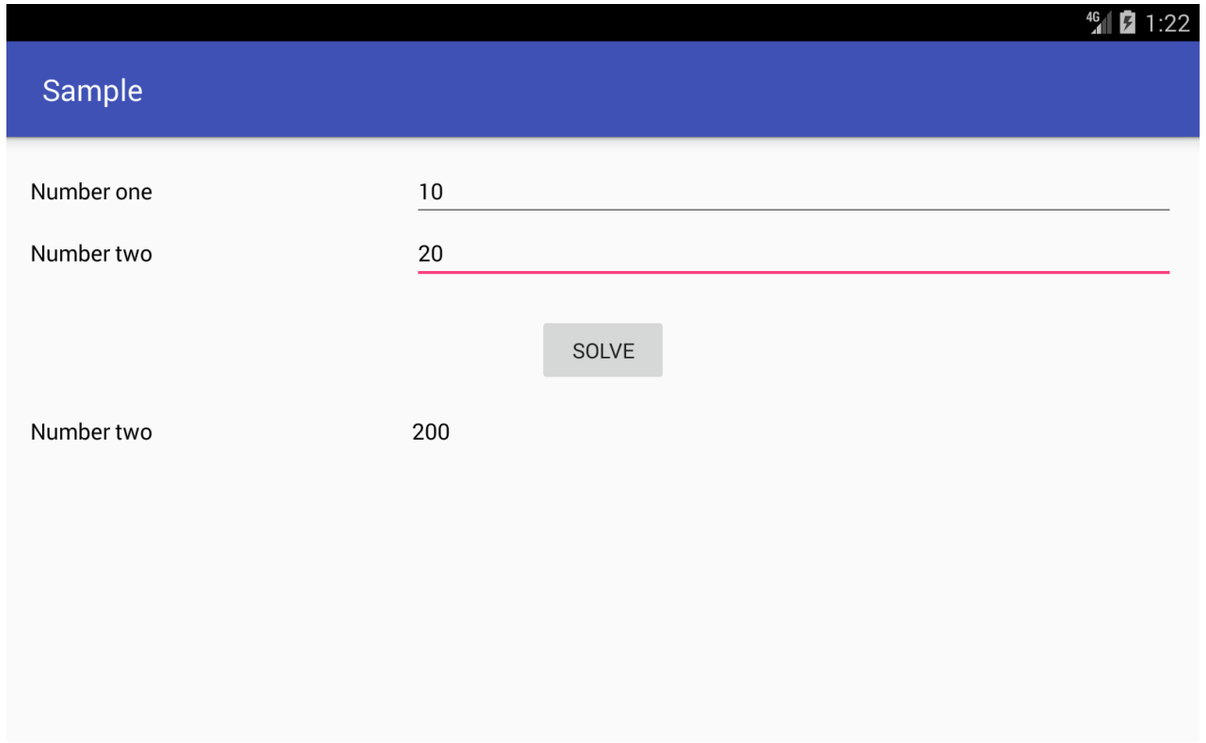
Step 5 Sign again apk file

Step 2: Sign apk with generated keystone

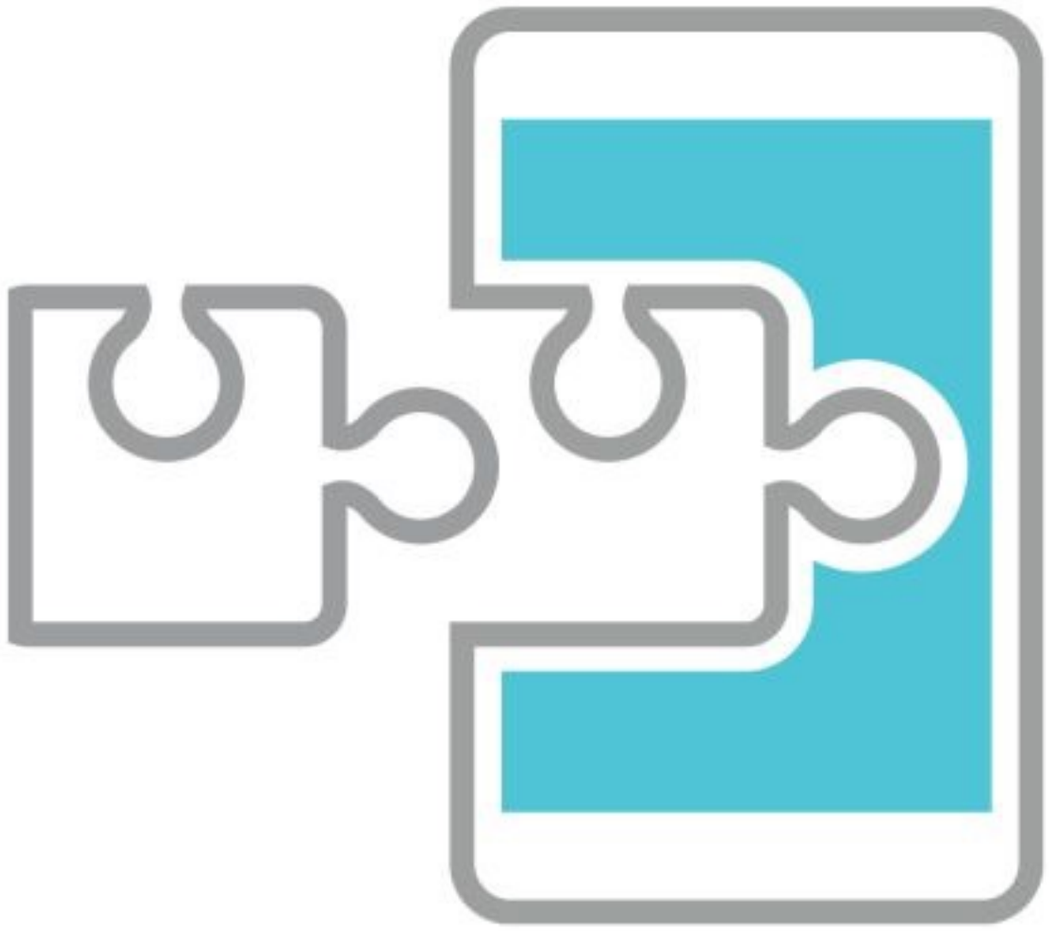
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
-keystore **my-release-key.keystore**
my_application.apk alias_name



Result



Additional Tools



Xposed

What is hooking

```
public int calc(int a, int b) {
    // logic code here
    if (a % 2 == 0) return 0;
    return a + b;
}
```

What is hooking

```
public int calc(int a, int b) {
    // logic code here
    if (a % 2 == 0) return 0;
    return a + b;
}
```

calc(10, 10) = 0
calc(5, 10) = 15

What is hooking

```
public int calc(int a, int b) {

        // logic code here
        if (a % 2 == 0) return 0;
        return a + b;

}
```

What is hooking

```
public int calc(int a, int b) {

    // logic code here
    if (a % 2 == 0) return 0;
    return a + b;

}
```

Hook before method:

What is hooking

```
public int calc(int a, int b) {

    // logic code here
    if (a % 2 == 0) return 0;
    return a + b;

}
```

Hook before method:
a = a + 1

What is hooking

```
public int calc(int a, int b) {

    // logic code here
    if (a % 2 == 0) return 0;
    return a + b;

}
```

Hook before method:
a = a + 1

calc(10, 10) = calc(11, 10) = 21
calc(5, 10) = calc(6, 10) = 0

What is hooking

```
public int calc(int a, int b) {

    // logic code here
    if (a % 2 == 0) return 0;
    return a + b;

}
```

Hook before method:
a = a + 1

calc(10, 10) = calc(11, 10) = 21
calc(5, 10) = calc(6, 10) = 0

Hook after method:

What is hooking

```
public int calc(int a, int b) {

    // logic code here
    if (a % 2 == 0) return 0;
    return a + b;

}
```

Hook before method:
a = a + 1

calc(10, 10) = calc(11, 10) = 21
calc(5, 10) = calc(6, 10) = 0

Hook after method:
return result + 1

What is hooking

```
public int calc(int a, int b) {

    // logic code here
    if (a % 2 == 0) return 0;
    return a + b;

}
```

Hook before method:
a = a + 1

calc(10, 10) = calc(11, 10) = 21
calc(5, 10) = calc(6, 10) = 0

Hook after method:
return result + 1

calc(10, 10) = 0 = 0 + 1 = 1
calc(5, 10) = 15 = 15 + 1 = 16

Xposed modules

- https://github.com/wanam/YouTubeAdAway

Xposed modules

- https://github.com/wanam/YouTubeAdAway
- https://github.com/pylerSM/YouTubeBackgroundPlayback

Xposed modules

- https://github.com/wanam/YouTubeAdAway
- https://github.com/pylerSM/YouTubeBackgroundPlayback
- https://github.com/devadvance/rootcloak

Xposed modules

- https://github.com/wanam/YouTubeAdAway
- https://github.com/pylerSM/YouTubeBackgroundPlayback
- https://github.com/devadvance/rootcloak
- https://github.com/M66B/XPrivacy

Demo

```
public class XposedHook implements IXposedHookLoadPackage {

    public void handleLoadPackage(final XC_LoadPackage.LoadPackageParam lpparam) {
        if (!lpparam.packageName.equals("com.hqt.reverse.sample")) return;

        Class<?> clazz = XposedHelpers.findClass("com.hqt.reverse.sample.a.a",
lpparam.classLoader);

        XposedHelpers.findAndHookMethod(clazz, "a",
                int.class, int.class,
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        Integer first = (Integer) param.args[0];
                        Integer second = (Integer) param.args[1];
                        Integer result = first * second;
                        param.setResult(result);
                    }
                });
    }
}
```

Demo

```
public class XposedHook implements IXposedHookLoadPackage {

    public void handleLoadPackage(final XC_LoadPackage.LoadPackageParam lpparam) {
        if (!lpparam.packageName.equals("com.hqt.reverse.sample")) return;

        Class<?> clazz = XposedHelpers.findClass("com.hqt.reverse.sample.a.a",
lpparam.classLoader);

        XposedHelpers.findAndHookMethod(clazz, "a",
                int.class, int.class,
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        Integer first = (Integer) param.args[0];
                        Integer second = (Integer) param.args[1];
                        Integer result = first * second;
                        param.setResult(result);
                    }
                });
    }
}
```

Demo

```
public class XposedHook implements IXposedHookLoadPackage {

    public void handleLoadPackage(final XC_LoadPackage.LoadPackageParam lpparam) {
        if (!lpparam.packageName.equals("com.hqt.reverse.sample")) return;

        Class<?> clazz = XposedHelpers.findClass("com.hqt.reverse.sample.a.a",
lpparam.classLoader);

        XposedHelpers.findAndHookMethod(clazz, "a",
                int.class, int.class,
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        Integer first = (Integer) param.args[0];
                        Integer second = (Integer) param.args[1];
                        Integer result = first * second;
                        param.setResult(result);
                    }
                });
    }
}
```

Demo

```
public class XposedHook implements IXposedHookLoadPackage {

    public void handleLoadPackage(final XC_LoadPackage.LoadPackageParam lpparam) {
        if (!lpparam.packageName.equals("com.hqt.reverse.sample")) return;

        Class<?> clazz = XposedHelpers.findClass("com.hqt.reverse.sample.a.a",
lpparam.classLoader);

        XposedHelpers.findAndHookMethod(clazz, "a",
                int.class, int.class,
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        Integer first = (Integer) param.args[0];
                        Integer second = (Integer) param.args[1];
                        Integer result = first * second;
                        param.setResult(result);
                    }
                });
    }
}
```

Demo

```
public class XposedHook implements IXposedHookLoadPackage {

    public void handleLoadPackage(final XC_LoadPackage.LoadPackageParam lpparam) {
        if (!lpparam.packageName.equals("com.hqt.reverse.sample")) return;

        Class<?> clazz = XposedHelpers.findClass("com.hqt.reverse.sample.a.a",
lpparam.classLoader);

        XposedHelpers.findAndHookMethod(clazz, "a",
                int.class, int.class,
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        Integer first = (Integer) param.args[0];
                        Integer second = (Integer) param.args[1];
                        Integer result = first * second;
                        param.setResult(result);
                    }
                });
    }
}
```

Demo

```
public class XposedHook implements IXposedHookLoadPackage {

    public void handleLoadPackage(final XC_LoadPackage.LoadPackageParam lpparam) {
        if (!lpparam.packageName.equals("com.hqt.reverse.sample")) return;

        Class<?> clazz = XposedHelpers.findClass("com.hqt.reverse.sample.a.a",
lpparam.classLoader);

        XposedHelpers.findAndHookMethod(clazz, "a",
                int.class, int.class,
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        Integer first = (Integer) param.args[0];
                        Integer second = (Integer) param.args[1];
                        Integer result = first * second;
                        param.setResult(result);
                    }
                });
    }
}
```

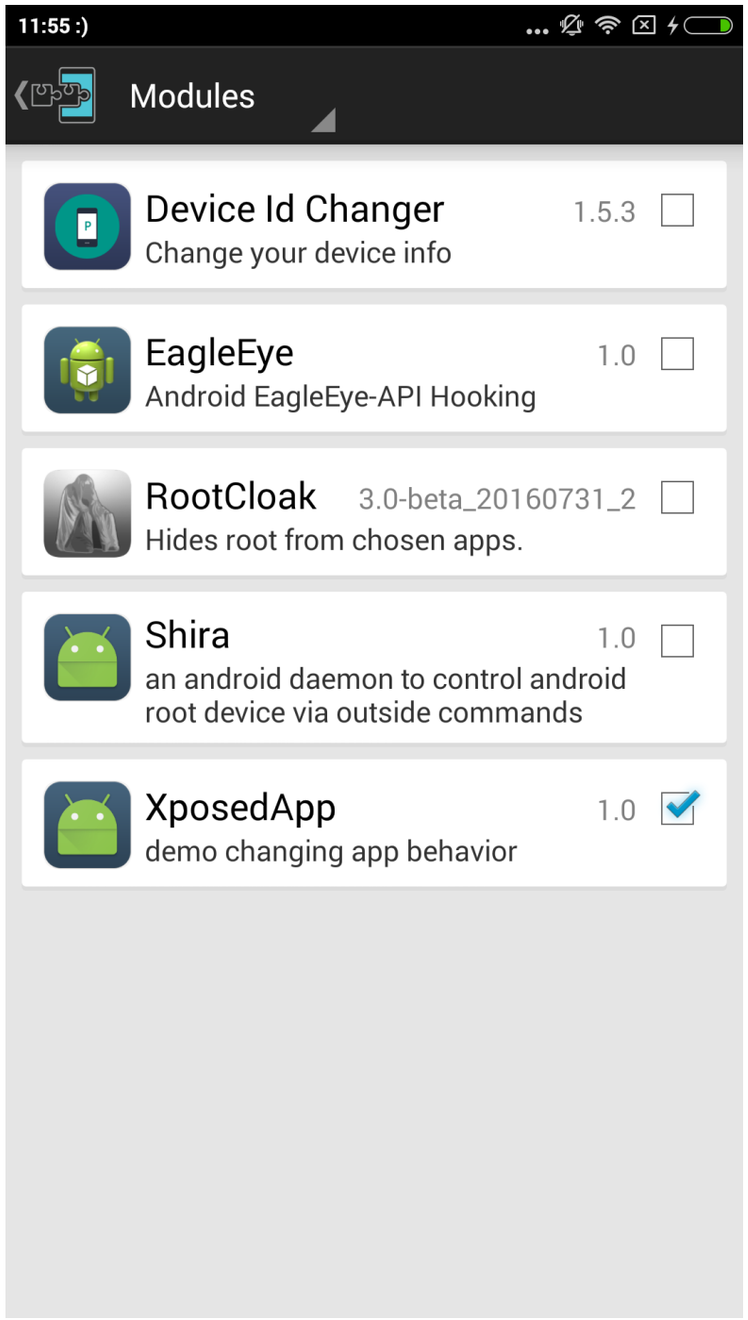
Demo

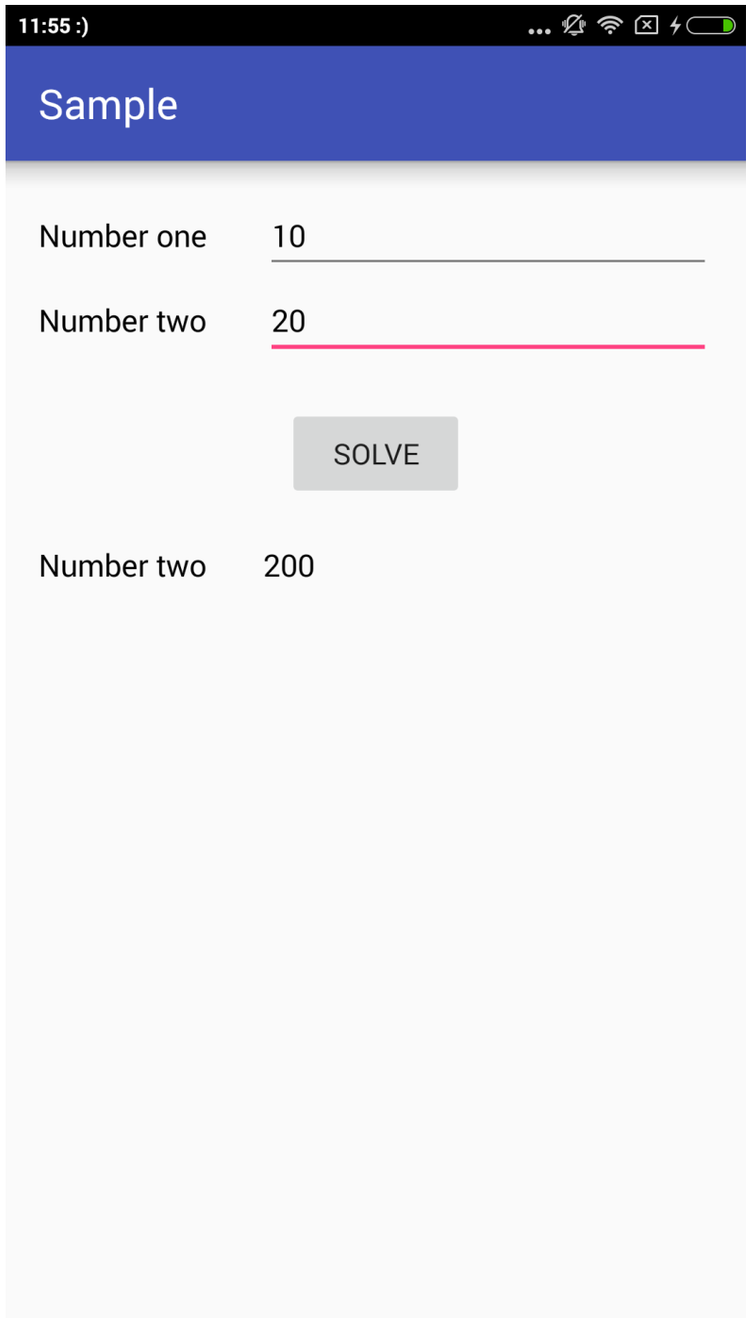
```
public class XposedHook implements IXposedHookLoadPackage {

    public void handleLoadPackage(final XC_LoadPackage.LoadPackageParam lpparam) {
        if (!lpparam.packageName.equals("com.hqt.reverse.sample")) return;

        Class<?> clazz = XposedHelpers.findClass("com.hqt.reverse.sample.a.a",
lpparam.classLoader);

        XposedHelpers.findAndHookMethod(clazz, "a",
                int.class, int.class,
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        Integer first = (Integer) param.args[0];
                        Integer second = (Integer) param.args[1];
                        Integer result = first * second;
                        param.setResult(result);
                    }
                });
    }
}
```

Demo

```
public class XposedHook implements IXposedHookLoadPackage {

    public void handleLoadPackage(final XC_LoadPackage.LoadPackageParam lpparam) {
        if (!lpparam.packageName.equals("com.hqt.reverse.sample")) return;

        Class<?> clazz = XposedHelpers.findClass("com.hqt.reverse.sample.a.a",
lpparam.classLoader);

        XposedHelpers.findAndHookMethod(clazz, "a",
                int.class, int.class,
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        Integer first = (Integer) param.args[0];
                        Integer second = (Integer) param.args[1];
                        Integer result = first * second;
                        param.setResult(result);
                    }
                });
    }
}
```

Demo


11:55 :)
Modules
Device Id Changer
1.5.3
Change your device info
EagleEye
1.0
Android EagleEye-API Hooking
RootCloak
3.0-beta_20160731_2
Hides root from chosen apps.
Shira
1.0
an android daemon to control android root device via outside commands
XposedApp
1.0
demo changing app behavior


11:55 :)
Sample
Number one
10
Number two
20
SOLVE
Number two
200

Reflection API

- change some internal value.

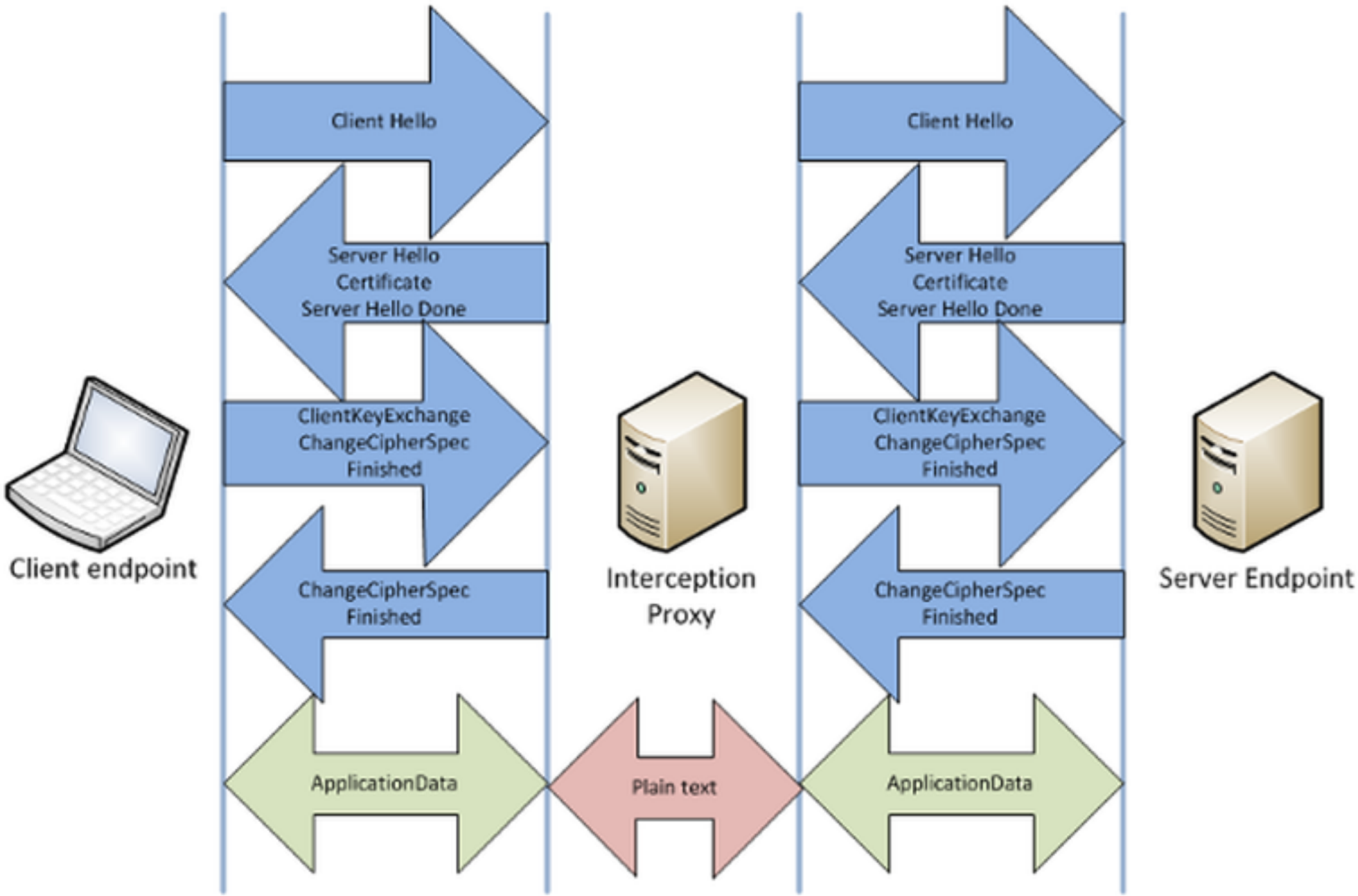
Reflection API

- change some internal value.
- call some private methods.

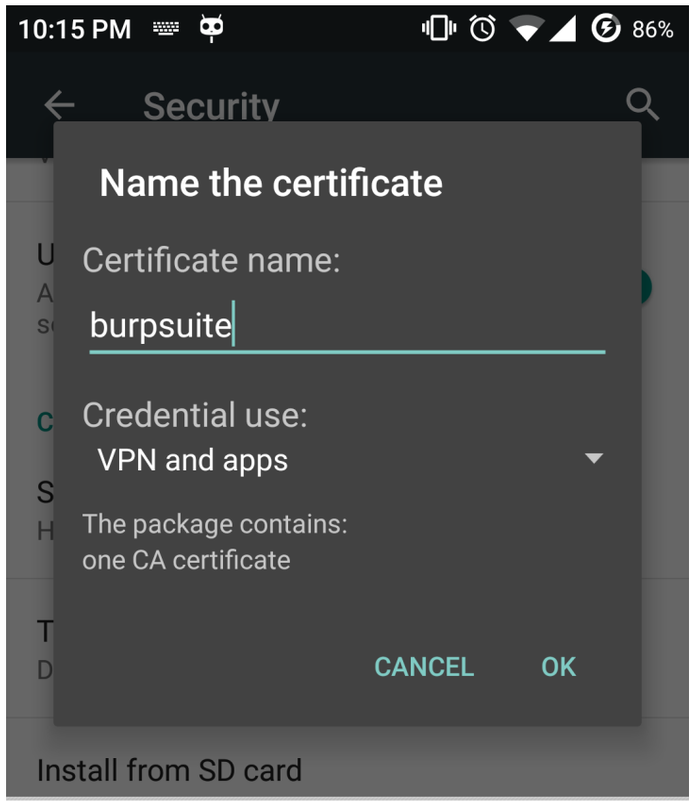
Reflection API

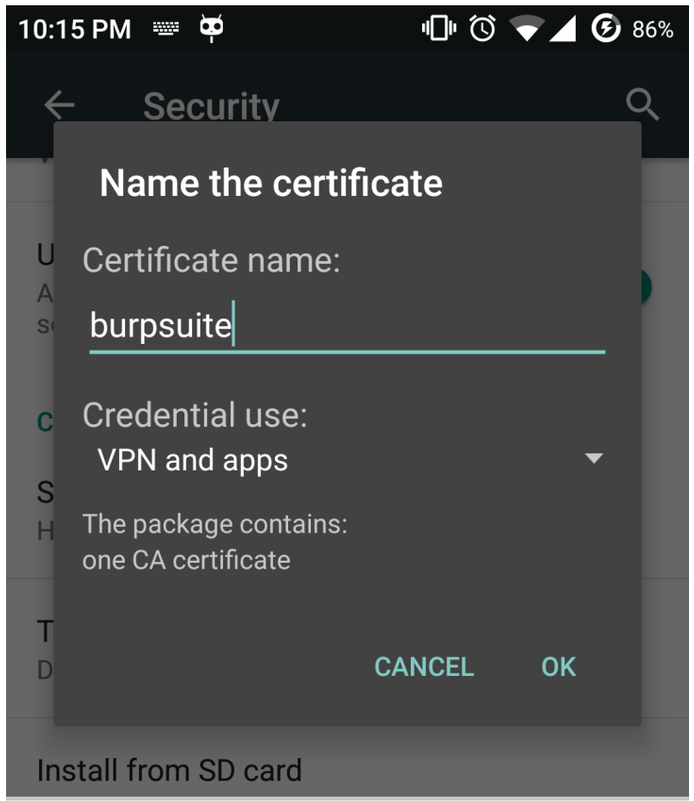
- change some internal value.
- call some private methods.
- Internal Android doesn’t obfuscate its source code.

Intercept HTTPS request



Intercept HTTPS request


10:15 PM
86%
Security
Name the certificate
Certificate name:
burpsuite
Credential use:
VPN and apps
The package contains:
one CA certificate
CANCEL
OK
Install from SD card



Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Options | Alerts

Intercept | HTTP history | WebSockets history | Options

Filter: Hiding CSS, image and general binary content

#	Host	Meth...	URL	Para...	Edited	Status	Length	MIME ty...	Ext
57	https://113.171.18.187	GET	/js/bg/nx8mdnyQUk4n_2x42sh...	☐	☐	200	10199	script	js
7	https://113.171.18.232	GET	/chrome-variations/seed?osna...	☑	☐	304	178		
60	https://113.171.18.55	GET	/s/roboto/v15/QHD8zigcbDB8a...	☐	☐	200	20278		ttf
34	https://113.171.246.123	GET	/embed/1hTkiNkf6-0?wmode=...	☑	☐	200	18418	HTML	
20	https://113.171.246.168	GET	/js/api.js	☐	☐	304	631	script	js
37	https://113.171.246.187	GET	/yts/jsbin/www-embed-player-n...	☐	☐	200	1834...	script	js
39	https://113.171.246.187	GET	/yts/jsbin/html5player-new-en_...	☐	☐	200	1038...	script	js
18	https://113.171.246.84	GET	/js/client:plusone.js	☐	☐	304	851	script	js
6	https://118.69.198.85	GET	/	☐	☐	200	1808...	HTML	
21	https://118.69.198.85	GET	/threads/tren-tay-ban-dung-st...	☐	☐	200	1614...	HTML	
63	https://118.69.198.85	POST	/login/login	☑	☐	200	67207	HTML	
64	https://118.69.198.85	GET	/css.php?css=Gritter,GritterE...	☑	☐			HTML	ph
35	https://31.13.70.1	GET	/fql?q=SELECT+total_count+F...	☑	☐	200	843	script	

Request | Response

Raw | Params | Headers | Hex

POST request to /login/login

Type	Name
Cookie	tt_session
Cookie	xf_vim|mudim-settings
Body	login
Body	password

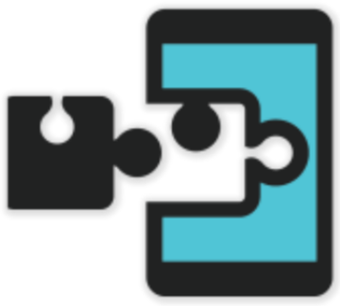
Not always work ...

Not always work ...

ProxyDroid

Not always work ...

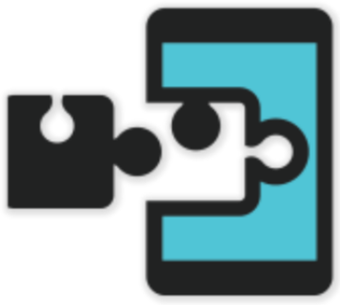
ProxyDroid



JustTrustMe

Not always work ...

ProxyDroid



JustTrustMe

using native hooking framework. (later)

Multidex Supporting

Multidex Introduction

Total number of methods that can be referenced within a single DEX file limit to **65,536**

Multidex Introduction

Total number of methods that can be referenced within a single DEX file limit to **65,536**

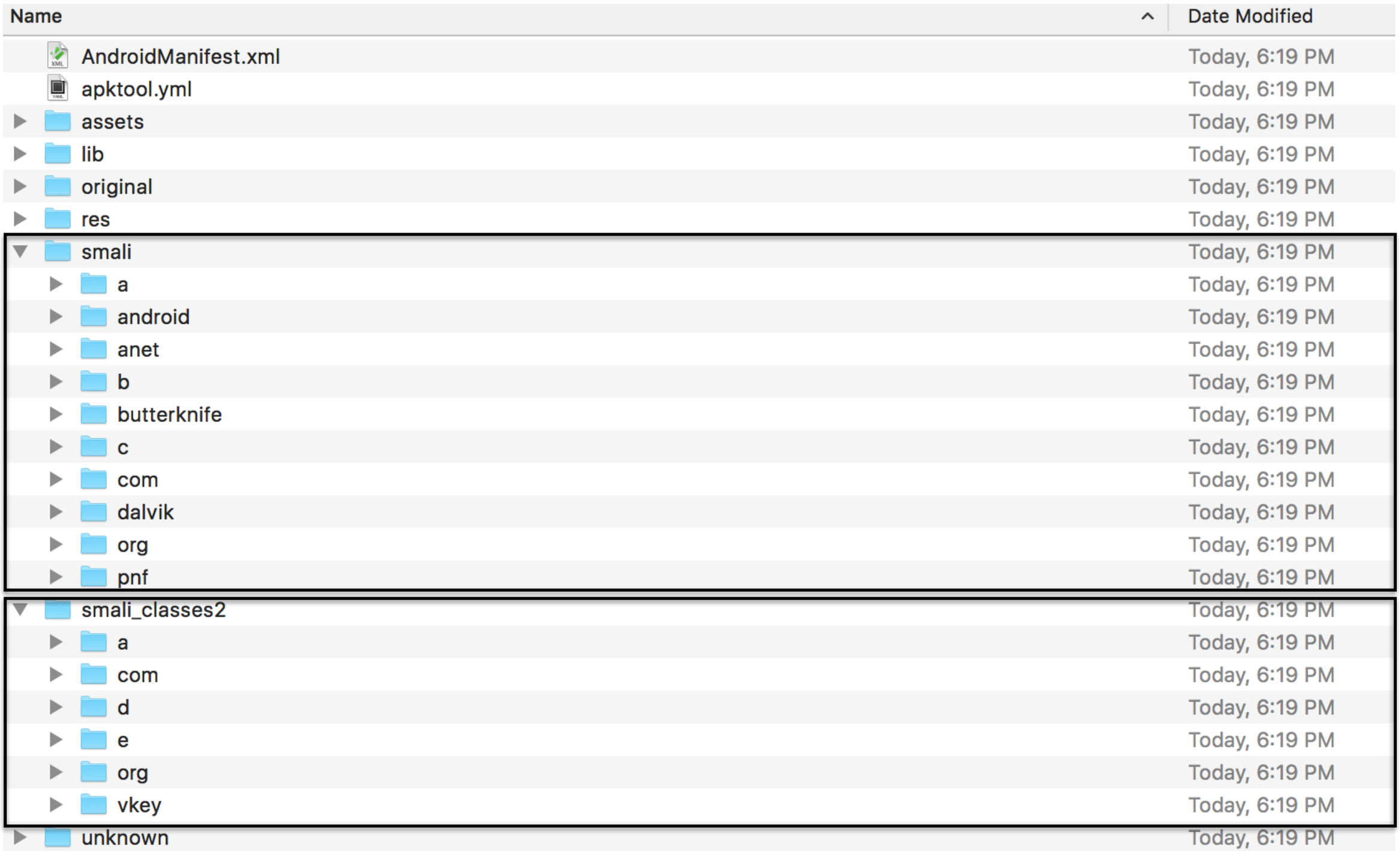
```
public class MyApplication extends Application {
    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(base);
        MultiDex.install(this);
    }
}
```

Multidex Introduction

Total number of methods that can be referenced within a single DEX file limit to **65,536**

```
public class MyApplication extends Application {
    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(base);
        MultiDex.install(this);
    }
}
```

Multidex Smali



Merge multi dex files

Baksmali: https://github.com/testwhat/SmaliEx

Command: java -jar baksmali.jar sample.apk

Merge multi dex files

Baksmali: https://github.com/testwhat/SmaliEx

Command: java -jar baksmali.jar sample.apk

Multidex Hooking

```
XposedHelpers.findAndHookMethod(“com.org.d.a.b”,
        paramLoadPackageParam.classLoader, "doingSomething", new
XC_MethodHook() {
            protected void afterHookedMethod(MethodHookParam
paramAnonymousMethodHookParam)
                    throws Throwable {
                super.afterHookedMethod(paramAnonymousMethodHookParam);
                paramAnonymousMethodHookParam.setResult("");
            }
        });
```

Multidex Hooking

```
XposedHelpers.findAndHookMethod(“com.org.d.a.b”,
        paramLoadPackageParam.classLoader, "doingSomething", new
XC_MethodHook() {
            protected void afterHookedMethod(MethodHookParam
paramAnonymousMethodHookParam)
                    throws Throwable {
                super.afterHookedMethod(paramAnonymousMethodHookParam);
                paramAnonymousMethodHookParam.setResult("");
            }
        });
```

- This hooking function will fail if this function lies on **second dex file**.

Multidex Hooking

```
XposedHelpers.findAndHookMethod("com.org.d.a.b",
        paramLoadPackageParam.classLoader, "doingSomething", new
XC_MethodHook() {
            protected void afterHookedMethod(MethodHookParam
paramAnonymousMethodHookParam)
                    throws Throwable {
                super.afterHookedMethod(paramAnonymousMethodHookParam);
                paramAnonymousMethodHookParam.setResult("");
            }
        });
```

- This hooking function will fail if this function lies on **second dex file**.
- We must hook all functions after **Multidex.install(this)** line.

Multidex Hooking

```
final Class<?>[] customClass = new Class<?>[1];
customClass[0] = XposedHelpers.findClass(“com.sample.MyApplication”,
        paramLoadPackageParam.classLoader);

XposedHelpers.findAndHookMethod(customClass[0], "onCreate",
    new XC_MethodHook() {
        @Override
        protected void afterHookedMethod(MethodHookParam param) {
            XposedBridge.log(“Hooked");
});
```

Multidex Hooking

```
final Class<?>[] customClass = new Class<?>[1];
customClass[0] = XposedHelpers.findClass(“com.sample.MyApplication”,
        paramLoadPackageParam.classLoader);

XposedHelpers.findAndHookMethod(customClass[0], "onCreate",
        new XC_MethodHook() {
            @Override
            protected void afterHookedMethod(MethodHookParam param) {
                XposedBridge.log(“Hooked");
    });
```

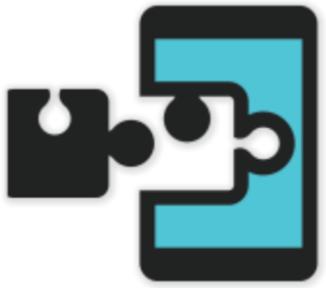
Multidex Hooking

```
final Class<?>[] customClass = new Class<?>[1];
customClass[0] = XposedHelpers.findClass(“com.sample.MyApplication”,
        paramLoadPackageParam.classLoader);

XposedHelpers.findAndHookMethod(customClass[0], "onCreate",
        new XC_MethodHook() {
            @Override
            protected void afterHookedMethod(MethodHookParam param) {
                XposedBridge.log(“Hooked");
    });
```

Multidex Hooking

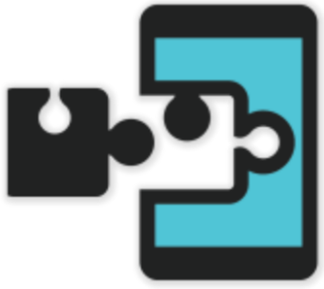
```
final Class<?> customClass = new Class<?>[1];
        customClass[0] = XposedHelpers.findClass("com.sample.MyApplication",
                paramLoadPackageParam.classLoader);
        XposedHelpers.findAndHookMethod(customClass[0], "onCreate",
                new XC_MethodHook() {
                    @Override
                    protected void afterHookedMethod(MethodHookParam param) {
                        XposedBridge.log("Hooked");

                        Class<?> nestedClass = XposedHelpers.findClass("com.org.d.a.b",
                                                                paramLoadPackageParam.classLoader);

                        XposedHelpers.findAndHookMethod(nestedClass, "doSomething",
                                new XC_MethodHook() {
                                    @Override
                                    protected void afterHookedMethod(MethodHookParam param){
                                        Log.e("hqthao", "hooked to different dex file");
                                    }
                                });

});
```

Hooking native layer

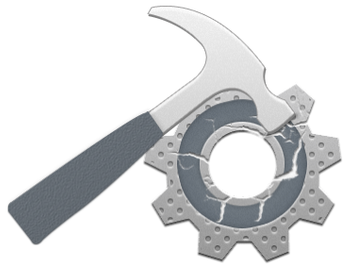


XPosed: only hook bridge methods.

Hooking native layer



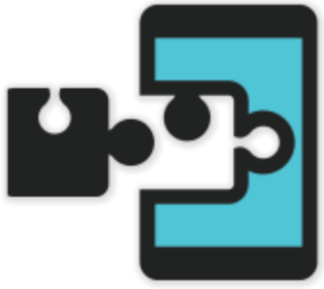
XPosed: only hook bridge methods.



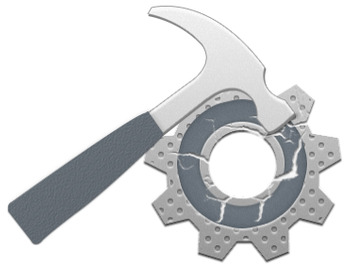
Cydia Substrate:

- similar for iOS reverse engineers.
- only for Android 4.3 or before.

Hooking native layer



XPosed: only hook bridge methods.



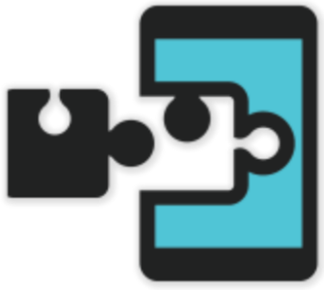
Cydia Substrate:

- similar for iOS reverse engineers.
- only for Android 4.3 or before.

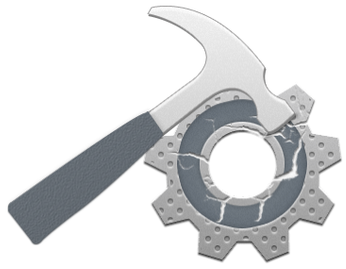
Frida:

- using javascript.
- fast for prototype.

Hooking native layer



XPosed: only hook bridge methods.



Cydia Substrate:

- similar for iOS reverse engineers.
- only for Android 4.3 or before.

Frida:

- using javascript.
- fast for prototype.

AndroidEagleEye:

- Good for implementing.
- Not good for testing prototype like Frida.

Others

IDA - Interactive Disassembler

Smalidea

Others

IDA - Interactive Disassembler

Smalidea

https://github.com/hqt/reverse-engineering

Q&A