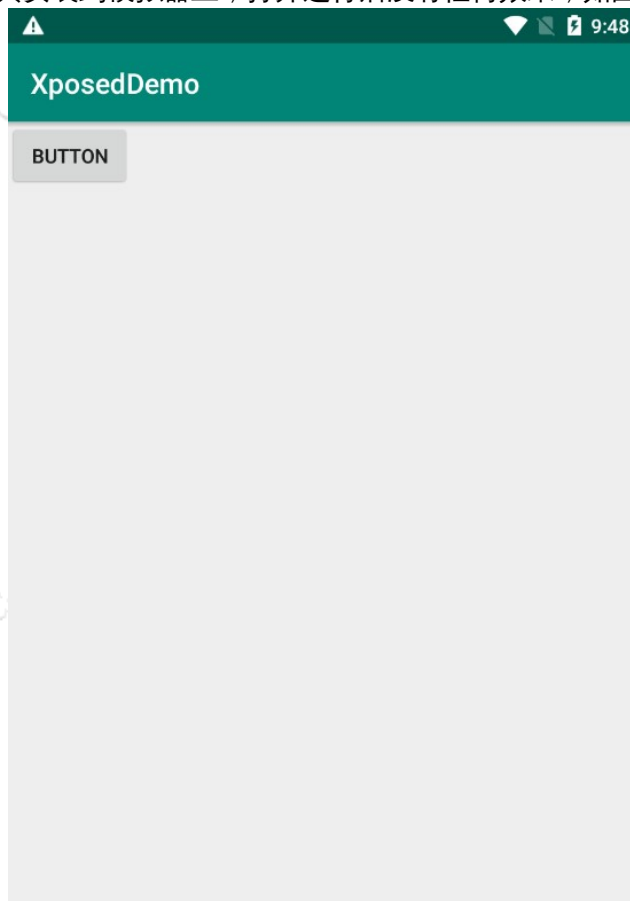


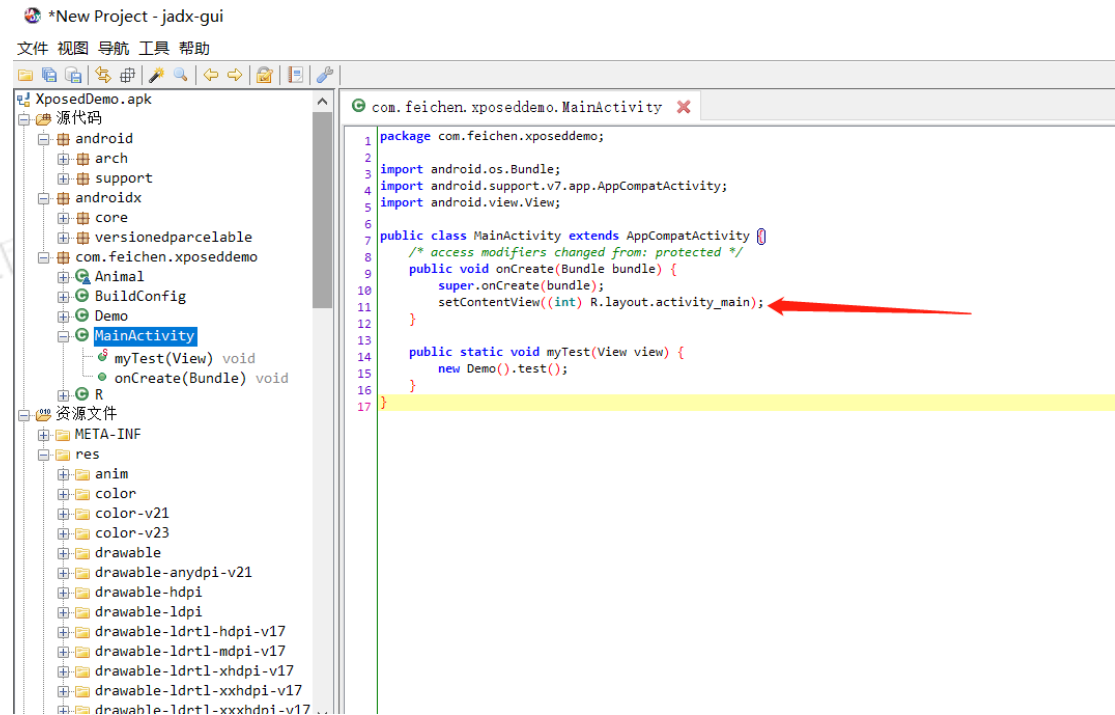
8.1.2 Xposed Hook-上

0x01 Hook 修改变量

在编写 hook 类的时候会去实现一个 IXposedHookLoadPackage 接口(加载应用程序, 即“ Android 软件包”时获得通知), 重写了 handleLoadPackage 方法(加载应用程序时将调用此方法), 该方法有一个参数 lpparam(有关该应用程序的信息), 这个方法向被实现的模块提供更多关于运行环境上下文的信息。首先我们实验的 app 是一款编写好的 XposedDemo, 将其安装到模拟器上, 打开运行后没有任何效果, 如图所示:

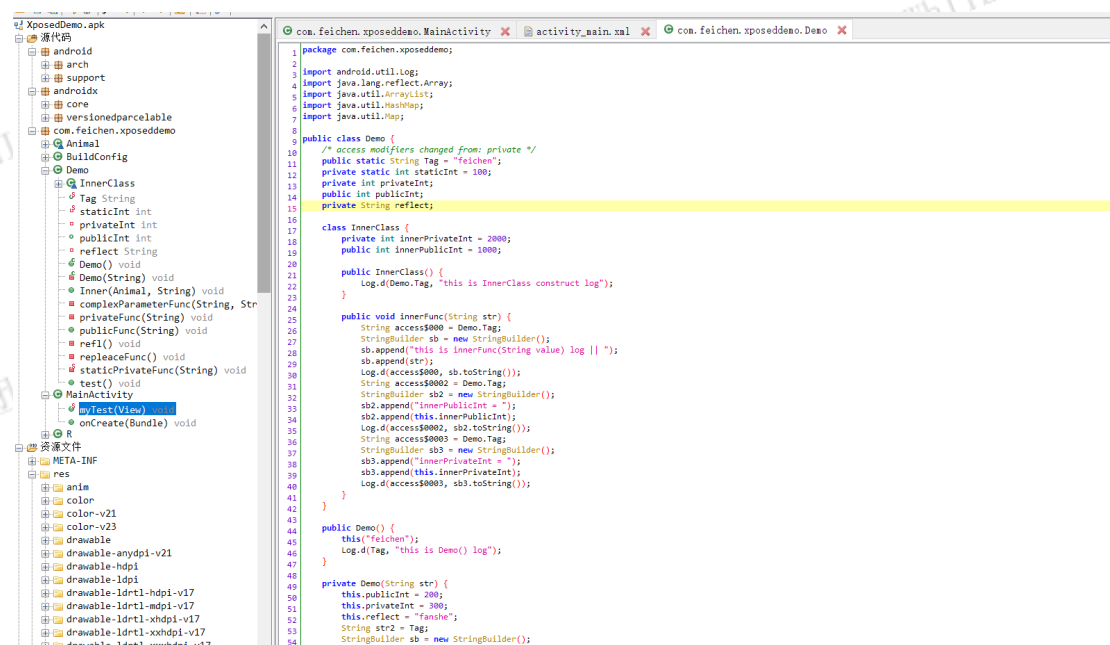
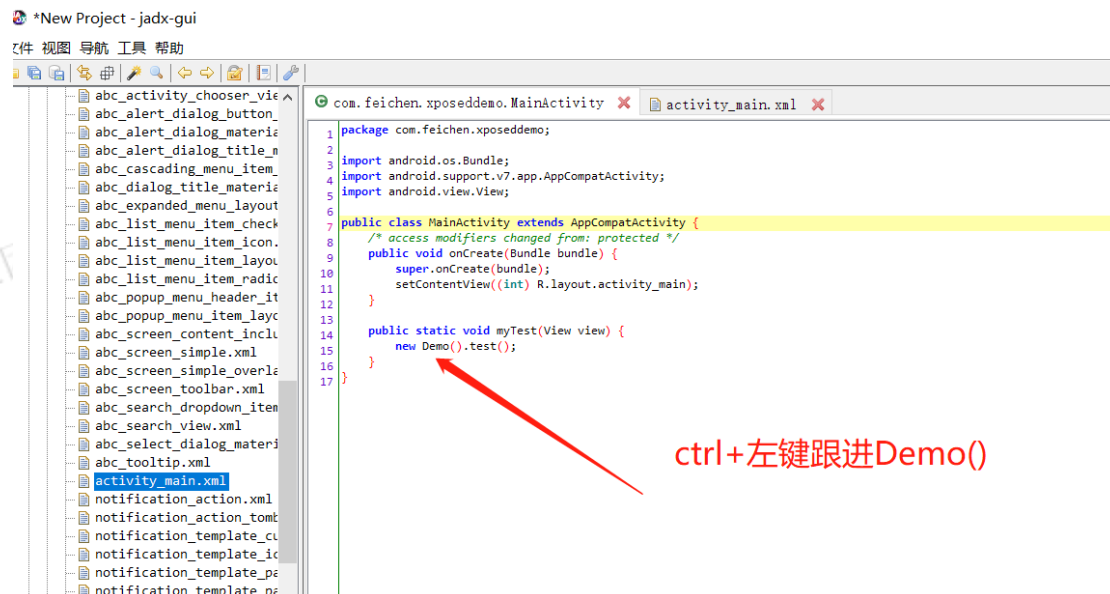


使用 jadx-gui 反编译工具查看其代码, 注意在一个 Activity 在启动的时候, 都会会在 onCreate () 方法中执行 setContentView(R.layout.activity_main)这行代码, 来将指定的资源 xml 文件加载到对应的 activity 中。

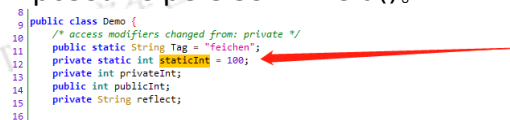


然后定位到 res/layout/activity_main.xml,可以发现当我们点击 button 的时候会触发 myTest 方法，回到 MainActivity，跟进 myTest() 的 Demo 类。





经过分析 test()方法可以发现，当我们点击按钮时会在日志中输出很多对应的日志信息，其中包括静态变量 staticInt = 100，注意静态全局变量 hook 的时候调用的是使用的 XposedHelpers.setStaticIntField()，若是全局普通变量用 XposedHelpers.setIntField()。



```

59
60 public void test() {
61     String str = Tag;
62     StringBuilder sb = new StringBuilder();
63     sb.append("staticInt = ");
64     sb.append(staticInt);
65     Log.d(str, sb.toString());
66     String str2 = Tag;
67     StringBuilder sb2 = new StringBuilder();
68     sb2.append("publicInt = ");
69     sb2.append(this.publicInt);
70     Log.d(str2, sb2.toString());
71     String str3 = Tag;
72     StringBuilder sb3 = new StringBuilder();
73     sb3.append("privateInt = ");
74     sb3.append(this.privateInt);
75     Log.d(str3, sb3.toString());
76     publicFunc("feichen");
77     String str4 = Tag;
78     StringBuilder sb4 = new StringBuilder();
79     sb4.append("publicInt = ");
80     sb4.append(this.publicInt);
81     Log.d(str4, sb4.toString());
82     String str5 = Tag;
83     StringBuilder sb5 = new StringBuilder();
84     sb5.append("privateInt = ");
85     sb5.append(this.privateInt);
86     Log.d(str5, sb5.toString());
87     privateFunc("feichen");
88     staticPrivateFunc("feichen");
89     String[][] strArr = (String[][] ) Array.newInstance(String.class, new int[]{1, 2});
90     HashMap hashMap = new HashMap();
91     hashMap.put("key", "value");
92     ArrayList arrayList = new ArrayList();
93     arraylist.add("listValue");
94 }

```

staticInt程序默认值是100,这里我们修改为520

我们再次运行 app 点击按钮，不过此次打开我们的 ddms 查看日志输出。

L...	Time	PID	TID	Application	Tag	Text
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is Demo(String) log feichen
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is Demo() log
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	staticInt = 100
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	publicInt = 200
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	privateInt = 300
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is publicFunc(String value) log feichen
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	publicInt = 200
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	privateInt = 300
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is privateFunc(String value) log feichen
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is staticPrivateFunc(String value) log feichen
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is complexParameter(String value) log feichen
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is replaceFunc log
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is Inner log feichen
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is eatFunc(String value) log feichen
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	AnimalInt = 400
D	09-16 10:36:32.379	2336	2336	com.feichen.xposed...	feichen	this is InnerClass construct log

未hook时，staticInt的值

然后编写 hook 代码如下：

```
package com.xposed;
```

```
import de.robv.android.xposed.IXposedHookLoadPackage;
```

```
import de.robv.android.xposed.XC_MethodHook;
```

```
import de.robv.android.xposed.XposedBridge;
```

```
import de.robv.android.xposed.XposedHelpers;
```

```
import de.robv.android.xposed.callbacks.XC_LoadPackage.LoadPackageParam;
```

```
import android.util.Log;
```

```
public class Hook implements IXposedHookLoadPackage {
```

```
    public void handleLoadPackage(final LoadPackageParam lpparam) throws
    Throwable {
```

```
        Log.d("feichen", "hook..."); //日志输出方式一
```

```
        XposedBridge.log("Loaded app: " + lpparam.packageName); //日志输出方
```

式二

```
        if (lpparam.packageName.equals("com.feichen.xposeddemo")){
```

```
            final
```

```
            Class
```

```
            clazz
```

```
=
```

```
XposedHelpers.findClass("com.feichen.xposeddemo.Demo",lpparam.classLoader);
```

```
XposedHelpers.setStaticIntField(clazz,"staticInt",520);
```

```
}
```

```
}
```

将写好的 xposed 编译安装到 xposed 后,勾选上写好的 xposed 模块,并重启手机,然后运行 app,打开 ddms,点击 button 按钮,查看 ddms 中的 staticInt 初始化值已经被我们 hook 修改为 520,如图:

Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.

L...	Time	PID	TID	Application	Tag	Text
D	09-17 00:19:44.434	2288	2288	com.feichen.xposeddemo	feichen	this is Demo(String) log feichen
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is Demo() log
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	staticInt = 520
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	publicInt = 200
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	privateInt = 300
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is publicFunc(String value) log feichen
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	publicInt = 200
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	privateInt = 300
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is privateFunc(String value) log feichen
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is staticPrivateFunc(Strin value) log feichen
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is complexParameter(Strin value) log feichen
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is replaceFunc log
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is Inner log feichen
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is eatFunc(String value) log feichen
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	AnimalInt = 400
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is InnerClass construct log
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	this is innerFunc(String value) log feichen
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	innerPublicInt = 1000
D	09-17 00:19:44.435	2288	2288	com.feichen.xposeddemo	feichen	innerPrivateInt = 2000

然后 hook 字符串变量的话使用 XposedHelpers.setStaticObjectField(clazz,"Tag","Lvmeng");这一条语句,具体的效果这里不再演示,有兴趣的小伙伴下去可以自己尝试。

0x02 Hook 普通方法

Hook 普通方法使用的是 XposedHelpers 下的 findAndHookMethod (类的字节码,方法名,回调函数)方法(用于 Hook 当前类下的所有方法),它有一个重载函数接收四个参数,

findAndHookMethod(类名全路径,类加载器,方法名,回调函数),其中回调函数除了使用

XC_MethodHook()之外,还有 XC_MethodReplacement()。对于有参数的函数需要带上参数的字节码。在 0x03 的地方就是四个参数的 findAndHookMethod。

因此,Hook 普通方法的代码如下:

```
XposedHelpers.findAndHookMethod(clazz, "test", new XC_MethodHook(){
    public void beforeHookedMethod(MethodHookParam param){
```

```
Log.d("Lvmeng","Lvmeng=====before");
```

```
}
```

```
public void afterHookedMethod(MethodHookParam param){
```

```
Log.d("Lvmeng","Lvmeng=====after");
```

```

    }
    });

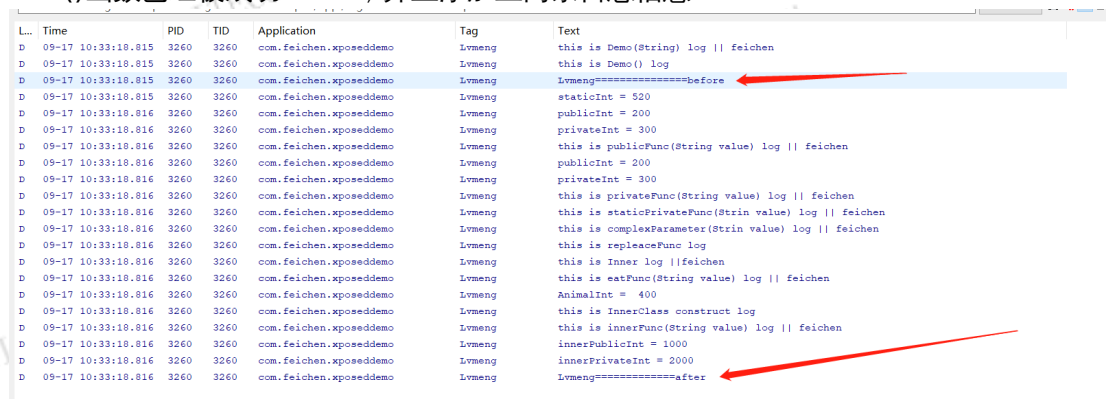
```

其中 beforeHookedMethod 会在调用原方法前执行，如果使用 setResult 则跳过原方法，并返回 setResult 参数中的值。

afterHookedMethod 会在调用原方法后执行，setResult 可改变返回值。

replaceHookedMethod 会完全替换原方法，即原方法不执行，且返回值可以直接 return，setResult 不生效。

然后将写好的 xposed 编译安装到 xposed 后，勾选上写好的 xposed 模块，并重启手机，然后运行 app,打开 ddms,点击 button 按钮,查看 ddms 中日志情况如下，可以发现 test()函数已经被成功 hook，并且添加上两条日志信息



L...	Time	PID	TID	Application	Tag	Text
D	09-17 10:33:18.815	3260	3260	com.feichen.xposeddemo	Lvmeng	this is Demo(String) log feichen
D	09-17 10:33:18.815	3260	3260	com.feichen.xposeddemo	Lvmeng	this is Demo() log
D	09-17 10:33:18.815	3260	3260	com.feichen.xposeddemo	Lvmeng	Lvmeng=====before
D	09-17 10:33:18.815	3260	3260	com.feichen.xposeddemo	Lvmeng	staticInt = 520
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	publicInt = 200
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	privateInt = 300
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is publicFunc(String value) log feichen
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	publicInt = 200
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	privateInt = 300
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is privateFunc(String value) log feichen
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is staticPrivateFunc(Strin value) log feichen
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is complexParameter(Strin value) log feichen
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is replaceFunc log
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is Inner log feichen
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is eatFunc(String value) log feichen
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	AnimalInt = 400
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is InnerClass construct log
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	this is innerFunc(String value) log feichen
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	innerPublicInt = 1000
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	innerPrivateInt = 2000
D	09-17 10:33:18.816	3260	3260	com.feichen.xposeddemo	Lvmeng	Lvmeng=====after

0x03 Hook 获取参数与返回值

Hook 获取参数是方法中要传入的参数，我们也是可以在 beforeHookedMethod 和 afterHookedMethod 方法中获取我们的参数值，其 hook 代码如下：

```

XposedHelpers.findAndHookMethod(clazz, "publicFunc",String.class, new
XC_MethodHook(){

```

```

    public void beforeHookedMethod(MethodHookParam param){

```

```

        Log.d("Lvmeng","Lvmeng=====before");

```

```

        Log.d("before-获取参数", ""+param.args[0]);

```

```

    }

```

```

    public void afterHookedMethod(MethodHookParam param){

```

```

        Log.d("Lvmeng","Lvmeng=====after");

```

```

        Log.d("after-获取参数", ""+param.args[0]);

```

```

    }

```

```

    });

```

其中，我们 hook 的方法是 publicFunc，查看代码可以发现该方法是接收参数的，如图所示：

```

111 }
112
113 public void publicFunc(String str) {
114     String str2 = Tag;
115     StringBuilder sb = new StringBuilder();
116     sb.append("this is publicFunc(String value) log || ");
117     sb.append(str);
118     Log.d(str2, sb.toString());
119 }
120

```

然后安装运行后的日志信息如下：

Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.						verbose
L	Time	PID	TID	Application	Tag	Text
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	this is Demo(String) log feichen
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	this is Demo() log
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	staticInt = 520
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	publicInt = 200
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	privateInt = 300
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	Lvmeng=====before
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	before-获取参数 feichen
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	this is publicFunc(String value) log feichen
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	Lvmeng=====after
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	after-获取参数 feichen
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	publicInt = 200
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	privateInt = 300
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	this is privateFunc(String value) log feichen
D	09-17 11:04:31.315	2254	2254	com.feichen.xposeddemo	Lvmeng	this is staticPrivateFunc(Strin value) log feichen
D	09-17 11:04:31.316	2254	2254	com.feichen.xposeddemo	Lvmeng	this is complexParameter(Strin value) log feichen
D	09-17 11:04:31.316	2254	2254	com.feichen.xposeddemo	Lvmeng	this is releaseFunc log

Hook 获取返回值一般都是在 afterHookedMethod 方法中，Hook 的代码如下：

```

public void afterHookedMethod(MethodHookParam param){
    Log.d("Lvmeng", ""+param.getResult());
}

```

在这里不再进行演示获取返回值，有兴趣的小伙伴可以下去自行测试。

0x04 Hook 构造函数

Hook 构造函数可分为有参构造函数前、无参构造函数前、有参构造函数后和无参构造函数后。这里 Hook 构造函数使用的是 XposedHelpers 下的 findAndHookConstructor，详细代码如下：

```

XposedHelpers.findAndHookConstructor(clazz, new XC_MethodHook() {
    public void beforeHookedMethod(MethodHookParam param) throws
    Throwable {
        Log.d("=====", "这是无参构造函数前");
    }

    public void afterHookedMethod(MethodHookParam param) throws
    Throwable {
        Log.d("=====", "这是无参构造函数后");
        XposedHelpers.setIntField(param.thisObject, "publicInt", 20000000);
    }
});

```



```

XposedHelpers.findAndHookConstructor(clazz, String.class, new
XC_MethodHook() {
    public void beforeHookedMethod(MethodHookParam param) throws
Throwable {
        Log.d("=====", "这是有参构造函数前");
        param.args[0] = "-";
    }

    public void afterHookedMethod(MethodHookParam param) throws
Throwable {
        Log.d("=====", "这是有参构造函数后");
    }
});

```

然后安装运行后的 Hook 日志如下：

L...	Time	PID	TID	Application	Tag	Text
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	=====	这是无参构造函数前
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	=====	这是有参构造函数前
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	this is Demo(String) log feichen
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	=====	这是有参构造函数后
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	this is Demo() log
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	=====	这是无参构造函数后
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	staticInt = 520
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	publicInt = 20000000
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	privateInt = 300
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	Lvmeng=====before
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	before-获取参数	feichen
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	this is publicFunc(String value) log feichen
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	Lvmeng=====after
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	after-获取参数	feichen
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	null
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	publicInt = 20000000
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	privateInt = 300
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	this is privateFunc(String value) log feichen
D	09-17 13:21:17.726	2253	2253	com.feichen.xposeddemo	Lvmeng	this is staticPrivateFunc(Strin value) log feichen
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	this is complexParameter(Strin value) log feichen
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	this is replaceFunc log
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	this is Inner log feichen
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	this is eatFunc(String value) log feichen
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	AnimalInt = 400
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	this is InnerClass construct log
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	this is innerFunc(String value) log feichen
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	innerPublicInt = 1000
D	09-17 13:21:17.727	2253	2253	com.feichen.xposeddemo	Lvmeng	innerPrivateInt = 2000