看雪3万课程笔记-FRIDA高级API实用方法：Frida Hook Java（六）hook动态dex

原创

kfyzjd2008 已于 2022-02-16 17:46:15 修改 179 收藏
分类专栏： 安卓 文章标签： 安卓逆向 frida 逆向 frida hook 看雪三万
于 2022-02-16 16:29:54 首次发布
版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/kfyzjd2008/article/details/122966007
版权


安卓 专栏收录该内容
19 篇文章 5 订阅
订阅专栏

本节接上一节课，APP进入第五关。

本节知识点为hook动态dex：

观察代码可以发现关键判断点处 getDynamicDexCheck() 的 check()方法在代码中无法找到。

根据代码发现在getDynamicDexCheck为null时会动态加载dex文件。

```java
private void loaddex() {
    File cacheDir = getCacheDir();
    if (!cacheDir.exists()) {
        cacheDir.mkdir();
    }
    String str = cacheDir.getAbsolutePath() + File.separator + "DynamicPlugin.dex";
    File file = new File(str);
    try {
        if (!file.exists()) {
            file.createNewFile();
            copyFiles(this, "DynamicPlugin.dex", file);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
        this.DynamicDexCheck = (AbstractC0000CheckInterface) new DexClassLoader(str, cacheDir.getAbsolutePath(), null, getClassL
        if (this.DynamicDexCheck == null) {
            Toast.makeText(this, "loaddex Failed!", 1).show();
        }
    } catch (Exception e2) {
        e2.printStackTrace();
    }
}

public AbstractC0000CheckInterface getDynamicDexCheck() {
    if (this.DynamicDexCheck == null) {
        loaddex();
    }
    return this.DynamicDexCheck;
}

/* access modifiers changed from: protected */
@Override // androidx.activity.ComponentActivity, androidx.core.app.ComponentActivity, com.example.androiddemo.Activity.BaseFrid
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    loaddex();
}

@Override // com.example.androiddemo.Activity.BaseFridaActivity
public void onCheck() {
    if (getDynamicDexCheck() == null) {
        Toast.makeText(this, "onClick loaddex Failed!", 1).show();
    } else if (getDynamicDexCheck().check()) {
        CheckSuccess();
        startActivity(new Intent(this, FridaActivity6.class));
        finishActivity(0);
    } else {
        super.CheckFailed();
```

当DynamicDexCheck为空时动态加载dex文件

关键判断点

CSDN @kfyzjd2008

首先我们要捋一下访问动态dex的流程：

1、使用选择器choose主动调用方法名（本例：getDynamicDexCheck），并获取方法的类名.$className

2、 利用enumerateClassLoaders遍历所有的类加载器

3、使用findClass在遍历时找到对应的className名称的类

4、使用Java.classFactory加载器加载实例

5、此时的类可以正常hook

代码如下：

```
function hook_dyn_dex(){
    //hook 动态dex
    Java.perform(function(){
        var FridaActivity5 = Java.use("com.example.androiddemo.Activity.FridaActivity5");

        Java.choose("com.example.androiddemo.Activity.FridaActivity5",{
            onMatch:function(instance){
                console.log(instance.getDynamicDexCheck().$className);
            },onComplete:function(){

            }
        });

        Java.enumerateClassLoaders({    //enumerateClassLoaders()枚举 Java VM 中存在的类加载器
            onMatch:function(loader){
                try {
                    if(loader.findClass("com.example.androiddemo.Dynamic.DynamicCheck")){
                        console.log(loader);
                        Java.classFactory.loader = loader;//加载器实例分配给Java.classFactory.loader
                    }
                } catch (error) {

                }

            },
            onComplete:function(){

            }
        });
        var DynamicCheck = Java.use("com.example.androiddemo.Dynamic.DynamicCheck");
        console.log(DynamicCheck);
        DynamicCheck.check.implementation = function(){
            console.log("DynamicCheck.check");
            return true;
        }
    });
}
```

然后在frida命令行中主动调用该函数即可通关。

hook_dyn_dex()