**前言**

Frida是一款基于python + javascript 的hook框架，通杀android\ios\linux\win\osx等各平台，由于是基于脚本的交互，因此相比xposed和substrace cydia更加便捷，本文重点介绍Frida在android下面的使用。Frida的官网为：[http://www.frida.re/](https://link.jianshu.com?t=http%3A%2F%2Fwww.frida.re%2F)

**安装和搭建Frida环境**

首先要保证你的android手机已经root。通过pip安装frida:

pip install frida

下载frida-server：

frida\_server的下载地址：[https://github.com/frida/frida/releases](https://link.jianshu.com?t=https%3A%2F%2Fgithub.com%2Ffrida%2Ffrida%2Freleases)

到android手机上并且运行

adb push frida-server /data/local/tmp/  
adb shell  
su  
cd /data/local/tmp/  
chmod 777 frida-server  
./frida-server

转发android TCP端口到本地：

adb forward tcp:27042 tcp:27042  
adb forward tcp:27043 tcp:27043

测试frida环境，如果出现android手机的进程列表说明搭建成功：

frida-ps -R  
PID Name

2700 acceleratord  
2713 adbd  
2798 agnsscontrol  
2799 agnsslog  
2195 akmd09911  
8078 android.process.acore  
31283 android.process.media  
2185 atcmdserver  
4939 chargelogcat  
2796 chr\_logd  
22856 com.android.browser  
7912 com.android.contacts  
22417 com.android.gallery3d  
....

**得到android手机当前最前端Activity所在的进程**

get\_front\_app.py

其中get\_front\_app.py的内容如下：

import frida

rdev = frida.get\_remote\_device()

front\_app = rdev.get\_frontmost\_application()

print front\_app

**枚举android手机所有的进程**

enum\_process.py

enum\_process.py内容如下：

import frida

rdev = frida.get\_remote\_device()

processes = rdev.enumerate\_processes()

for process in processes:

print process

**枚举某个进程加载的所有模块以及模块中的导出函数**

import frida

rdev = frida.get\_remote\_device()

session = rdev.attach("com.tencent.mm") #如果存在两个一样的进程名可以采用rdev.attach(pid)的方式

modules = session.enumerate\_modules()

for module in modules:

print module

export\_funcs = module.enumerate\_exports()

print "\tfunc\_name\tRVA"

for export\_func in export\_funcs:

print "\t%s\t%s"%(export\_func.name,hex(export\_func.relative\_address))

**hook android的native函数**

import frida

import sys

rdev = frida.get\_remote\_device()

session = rdev.attach("com.tencent.mm")

scr = """

Interceptor.attach(Module.findExportByName("libc.so" , "open"), {

onEnter: function(args) {

send("open("+Memory.readCString(args[0])+","+args[1]+")");

},

onLeave:function(retval){

}

});

"""

script = session.create\_script(scr)

def on\_message(message ,data):

print message

script.on("message" , on\_message)

script.load()

sys.stdin.read()

**hook android的java层函数**

如下代码为hook微信（测试版本为6.3.13，不同版本由于混淆名字的随机生成的原因或者代码改动导致类名不一样）  
**com.tencent.mm.sdk.platformtools.ay**类的随机数生成函数，让微信猜拳随机（type=2），而摇色子总是为6点（type=5）

import frida

import sys

rdev = frida.get\_remote\_device()

session = rdev.attach("com.tencent.mm")

scr = """

Java.perform(function () {

var ay = Java.use("com.tencent.mm.sdk.platformtools.ay");

ay.pu.implementation = function(){

var type = arguments[0];

send("type="+type);

if (type == 2)

{

return this.pu(type);

}

else

{

return 5;

}

};

});

"""

script = session.create\_script(scr)

def on\_message(message ,data):

print message

script.on("message" , on\_message)

script.load()

sys.stdin.read()

**通过frida向android进程注入dex**

import frida, sys, optparse, re

def on\_message(message, data):

if message['type'] == 'send':

print("[\*] {0}".format(message['payload']))

else:

print(message)

jscode = """

Java.perform(function () {

var currentApplication = Java.use("android.app.ActivityThread").currentApplication();

var context = currentApplication.getApplicationContext();

var pkgName = context.getPackageName();

var dexPath = "%s";

var entryClass = "%s";

Java.openClassFile(dexPath).load();

console.log("inject " + dexPath +" to " + pkgName + " successfully!")

Java.use(entryClass).%s("%s");

console.log("call entry successfully!")

});

"""

def checkRequiredArguments(opts, parser):

missing\_options = []

for option in parser.option\_list:

if re.match(r'^\[REQUIRED\]', option.help) and eval('opts.' + option.dest) == None:

missing\_options.extend(option.\_long\_opts)

if len(missing\_options) > 0:

parser.error('Missing REQUIRED parameters: ' + str(missing\_options))

if \_\_name\_\_ == "\_\_main\_\_":

usage = "usage: python %prog [options] arg\n\n" \

"example: python %prog -p com.android.launcher " \

"-f /data/local/tmp/test.apk " \

"-e com.parker.test.DexMain/main " \

"\"hello fridex!\""

parser = optparse.OptionParser(usage)

parser.add\_option("-p", "--package", dest="pkg", type="string",

help="[REQUIRED]package name of the app to be injected.")

parser.add\_option("-f", "--file", dest="dexPath", type="string",

help="[REQUIRED]path of the dex")

parser.add\_option("-e", "--entry", dest="entry", type="string",

help="[REQUIRED]the entry function Name.")

(options, args) = parser.parse\_args()

checkRequiredArguments(options, parser)

if len(args) == 0:

arg = ""

else:

arg = args[0]

pkgName = options.pkg

dexPath = options.dexPath

entry = options.entry.split("/")

if len(entry) > 1:

entryClass = entry[0]

entryFunction = entry[1]

else:

entryClass = entry[0]

entryFunction = "main"

process = frida.get\_usb\_device(1).attach(pkgName)

jscode = jscode%(dexPath, entryClass, entryFunction, arg)

script = process.create\_script(jscode)

script.on('message', on\_message)

print('[\*] Running fridex')

script.load()

sys.stdin.read()

**通过注入抛出异常代码实现跟踪程序调用栈**

在<<Android 软件安全与逆向分析>>这本书中第八章有介绍通过重打包写入异常代码进行栈跟踪，但是这样比较麻烦，使用frida注入更方便。

**frida的相关资源**

[https://github.com/dweinstein/awesome-frida](https://link.jianshu.com?t=https%3A%2F%2Fgithub.com%2Fdweinstein%2Fawesome-frida)  
[http://jaq.alibaba.com/community/art/show?articleid=816](https://link.jianshu.com?t=http%3A%2F%2Fjaq.alibaba.com%2Fcommunity%2Fart%2Fshow%3Farticleid%3D816)  
[https://koz.io/using-frida-on-android-without-root/](https://link.jianshu.com?t=https%3A%2F%2Fkoz.io%2Fusing-frida-on-android-without-root%2F)  
[http://www.ninoishere.com/frida-learn-by-example/](https://link.jianshu.com?t=http%3A%2F%2Fwww.ninoishere.com%2Ffrida-learn-by-example%2F)  
[https://github.com/TheCjw/Frida-Android-Scripts](https://link.jianshu.com?t=https%3A%2F%2Fgithub.com%2FTheCjw%2FFrida-Android-Scripts)

作者：parkerpeng  
链接：https://www.jianshu.com/p/ca8381d3e094  
來源：简书  
简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。