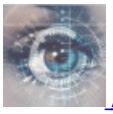
pingpong 攻防世界

原创

北风~ 于 2021-01-15 18:30:47 发布 3507 收藏

分类专栏： Android CTF

本文链接：https://blog.csdn.net/weixin_45055269/article/details/112644316



Android 同时被 2 个专栏收录

4 篇文章 1 订阅

订阅专栏

CTF

31 篇文章 3 订阅

订阅专栏

安卓题

JEB打开

主函数逻辑不难，1000000，ping一下pong一下减到0，输出flag，细看有两个反作弊，一个是`MainActivity.this.tt % 2 == 0`、`MainActivity.this.tt % 2 == 1`每对tt取余操作，满足条件就将tt重新归为1000000。所以tt目前的值决定你现在是ping还是pong；另一个是`MainActivity.this.p =MainActivity.this.ping(MainActivity.this.p, MainActivity.this.num);`、`MainActivity.this.p = MainActivity.this.pong(MainActivity.this.p, MainActivity.this.num);`，每次ping和pong都会调用so库里的ping()和pong()执行，所以直接将tt置0，中间不进行操作是没法将flag计算出来的。

```
    public MainActivity() {
        super();
        this.p = 0;
        this.num = 0;
        this.ttt = 1000000;
        this.tt = this.ttt;
        this.jping = new com.geekerchina.pingpongmachine.MainActivity$1(this);
        this.jpong = new com.geekerchina.pingpongmachine.MainActivity$2(this);
    }

    protected void onCreate(Bundle arg4) {
        super.onCreate(arg4);
        this.setContentView(0x7F04001B);
        this.findViewById(0x7F0B0057).setOnClickListener(this.jping);
        this.findViewById(0x7F0B0058).setOnClickListener(this.jpong);
    }

    public boolean onCreateOptionsMenu(Menu arg3) {
        this.getMenuInflater().inflate(0x7F0D0000, arg3);
        return 1;
    }

    public boolean onOptionsItemSelected(MenuItem arg3) {
        boolean v1 = arg3.getItemId() == 0x7F0B0070 ? true : super.onOptionsItemSelected(arg3);
        return v1;
    }

    public native int ping(int arg1, int arg2) {
    }

    public native int pong(int arg1, int arg2) {
    }
```


两个思路，可以frida调用so库里的函数模拟执行，或者第二种思路是将so文件拿出来，自己编一个新的主函数逻辑再整成一个新的apk

思路一：frida

参考链接：

安卓逆向之Frida Native层HOOK

【移动安全高级篇】————11、Frida篇

攻防世界 reverse pingpong

```
import frida, sys


def on_message(message, data):
    if message['type'] == 'send':
        print("[*] {0}".format(message['payload']))
    else:
        print(message)

jscode = """
setImmediate(function () {
    Java.perform(function () {
        console.log("start");
        //so层hook
        //导出函数
```

```
//var exports = Module.enumerateExportsSync("libpp.so");
//for(var i=0;i<exports.length;i++){
//    send("name:"+exports[i].name+"  address:"+exports[i].address);
// }

//遍历模块找基址
Process.enumerateModules({
    onMatch: function (exp) {
        if (exp.name == 'libpp.so') {
            send('enumerateModules find');
            send(exp.name + "|" + exp.base + "|" + exp.size + "|" + exp.path);
            send(exp);
            return 'stop';
        }
    },
    onComplete: function () {
        send('enumerateModules stop');
    }
});

//通过模块名直接查找基址
var soAddr = Module.findBaseAddress("libpp.so");
send("soAddr:" + soAddr);

var aping=0x1308+1
var apong=0x1564+1
//   hook导出函数 通过函数名

var fping=Module.findExportByName("libpp.so", "Java_com_geekerchina_pingpongmachine_MainActivity_ping")
send("findExportByName ping():" +fping );
fping=new NativePointer(soAddr).add(aping);
//NativePointer   简写ptr
send("findExportByName ping():" +fping );
var ping=new NativeFunction(fping, "int", ['pointer','pointer','int', 'int']);


//send("findExportByName edit():"+Module.findExportByName("libpp.so", "_ZL4editP7_JNIEnvP8_jobjecti"))
// Interceptor.attach(fping, {
//     onEnter: function (args) {
//         send("ping() began p:" + args[2] + ", num:" + args[3] );
//     },
//     onLeave: function (retval) {
//         send("ping() return:" + retval);
//     }
// });

//   hook导出函数 通过函数名
var fpong=Module.findExportByName("libpp.so", "Java_com_geekerchina_pingpongmachine_MainActivity_pong")
send("findExportByName pong():" +fpong );
fpong=new NativePointer(soAddr).add(apong);
send("findExportByName pong():" +fpong );
var pong=new NativeFunction(fpong, "int", ['pointer','pointer','int', 'int']);
//send("findExportByName edit():"+Module.findExportByName("libpp.so", "_ZL4editP7_JNIEnvP8_jobjecti"))
// Interceptor.attach(fpong, {
//     onEnter: function (args) {
//         send("pong() began p:" + args[2] + ", num:" + args[3] );
//     },
//     onLeave: function (retval) {
//         send("pong() return:" + retval);
//     }
```

```
        // });
        var env = Java.vm.getEnv();
        var obj=ptr(0);
        var check = 1000000;
        var beFlag = 0;
        var num = 0;
        while (true) {               //下面这部分代码参考 https://blog.csdn.net/jasalee/article/details/70242837
            if (check % 2 == 1) {
                --check;
                beFlag = pong(env,obj,beFlag,num);
                ++num;
                if(num >= 7) {
                    num = 0;
                }
            } else {
                --check;
                beFlag = ping(env,obj,beFlag,num);
                ++num;
                if(num >= 7){
                    num = 0;
                }
            }
            if (check == 0) {
                send("check:"+check+" num:"+num)
                send("FLAG : "+"BCTF{MagicNum" + beFlag+ "}");
                break;
            }
        }

    });
});
"""


# 运行中hook
process = frida.get_usb_device().attach('com.geekerchina.pingpongmachine')
script = process.create_script(jscode)
script.on('message', on_message)
print('[*] Running test')
script.load()
sys.stdin.read()


'''
[*] Running test
start
[*] enumerateModules find
[*] libpp.so|0xd4c48000|24576|/data/app/com.geekerchina.pingpongmachine-6gshbHpfeBmTBFRYBxQpNg==/lib/arm/libpp.s
o
[*] {'name': 'libpp.so', 'base': '0xd4c48000', 'size': 24576, 'path': '/data/app/com.geekerchina.pingpongmachine
-6gshbHpfeBmTBFRYBxQpNg==/lib/arm/libpp.so'}
[*] enumerateModules stop
[*] soAddr:0xd4c48000
[*] findExportByName ping():0xd34c9309
[*] findExportByName ping():0xd4c49309
[*] findExportByName pong():0xd34c9565
[*] findExportByName pong():0xd4c49565
[*] check:0 num:1
[*] FLAG : BCTF{MagicNum4500009}
```

```
'''
#以上嫖自DirWangK师傅，逻辑是找libpp.so的ping函数和pong函数，调用这两个函数模拟执行
#实际复现时，模拟器可以运行，但frida在模拟器里找不到libpp.so，网上查了查发现frida真机效果，模拟器经常出问题，但我没真机，直接开始哭
```

不过还是有点收获的：ping和pong函数的地址IDA里可以直接看，但是IDA是thumb模式，也就是所有汇编指令后面只跟了两个参数，所以指令下偏移地址+1

思路二：调用原apk的so文件，重新编译成新的apk

参考文章：

AS生成.so文件并在其它项目中进行引用，调用里面的方法

复现了一波没成功，先咕咕咕

BCTF{MagicNum4500009}