# ISCC2024 WriteUp

Whooops+李卓航+925798691@qq.com

### mobile+ChallengeMobile

### 解题思路

Jeb打开分析，Checker类＋isflag方法

解密ming文件调用a函数

转为dex文件

拿到dex文件的frida：

// 使用 Java.perform 确保脚本在正确的 Java 环境中执行

Java.perform(function() {

// 加载目标应用的 MainActivity 类

let MainActivity = Java.use("com.example.challengemobile.MainActivity");

// 劫持 MainActivity 类的 a 方法

MainActivity["a"].implementation = function (bArr) {

// 调用原始的 a 方法并获取返回的 dex 文件

let dexdexdexfffiiillleee = this['a'](bArr);

// 打印 dex 文件内容到控制台

console.log(dexdexdexfffiiillleee);

// 返回 dex 文件，保持原始方法的行为

return dexdexdexfffiiillleee;

};

});

之后跑出来的字节写入文件：

# 导入 struct 模块，用于处理字节数据的打包和解包

import struct

# 定义一个函数，将整数数组写入到一个二进制文件中

def write\_bytes\_to\_file(arr, file\_name):

# 以写二进制模式 ('wb') 打开文件

with open(file\_name, 'wb') as f:

# 遍历数组中的每个整数

for num in arr:

# 检查整数是否在 -127 到 127 之间

if -127 <= num <= 127:

# 使用 struct.pack 将整数打包成一个字节，并写入文件

f.write(struct.pack('b', num))

# 示例整数数组

arr = [1, -2, 3, -4, 5, -6, 7, -8, 9, -10, 11, -12, 13, -14, 15, -16, 17, -18, 19, -20]

# 指定输出文件名

file\_name = 'output.bin'

# 调用函数，将数组写入到指定的文件中

write\_bytes\_to\_file(arr, file\_name)

查看Checker类密文：



调用getKey函数并且输出返回值：

// 使用 Java.perform 确保脚本在正确的 Java 环境中执行

Java.perform(function () {

// 获取 java.lang.Class 类的引用

var class0001 = Java.use("java.lang.Class");

// 获取 dalvik.system.DexClassLoader 类的引用

var classloader000 = Java.use("dalvik.system.DexClassLoader");

// 劫持 DexClassLoader 类的 loadClass 方法，该方法有一个参数类型为 String

classloader000.loadClass.overload("java.lang.String").implementation = function (name) {

// 调用原始的 loadClass 方法并获取返回值

var retValue = this.loadClass(name, false);

// 目标类的名称

var hookname = "com.example.challengemobile.Checker";

// 检查当前加载的类名是否为目标类

if (name == hookname) {

// 获取加载的目标类的引用

var hookClass = retValue;

// 将目标类转换为 java.lang.Class 类型

var hook0001castsss = Java.cast(hookClass, class0001);

// 获取目标类中的 getKey 方法，该方法无参数

var classmethod000 = hook0001castsss.getMethod("getKey", []);

// 调用 getKey 方法，并获取返回值

var retValue = classmethod000.invoke(null, []);

// 打印 getKey 方法的返回值

console.log("retValue:", retValue);

// 返回 getKey 方法的返回值

return retValue;

}

// 返回原始 loadClass 方法的返回值

return retValue;

};

});

对密文进行base64解密然后xxtea解密

解密脚本：

# 导入 base64 模块，用于 Base64 编码和解码

import base64

# 导入 xxtea 模块，用于 XXTEA 加密和解密

import xxtea

# 定义一个 Base64 编码的字符串

str = b"sFPv3nYjIkmlBOiKOmk3vWOw2aXhFdLgy3uTN5NEjexEZ4oC"

# 使用 base64.b64decode 方法将字符串解码为字节数据

a = base64.b64decode(str)

# 打印解码后的字节数据

print(a)

# 打印解码后字节数据的长度

print(len(a))

# 定义用于 XXTEA 解密的密钥

key = b"sM51I634n137gH7~"

# 使用 xxtea.decrypt 方法进行解密

# 解密后的结果为字节数据

decrypted\_data = xxtea.decrypt(a, key)

# 打印解密后的数据

print(decrypted\_data)