2018-11 科来杯山东省省赛writeup

原创

木木or沐沐 于 2018-11-11 22:38:34 发布 723 收藏
文章标签： wp

这次省赛只做出两个题目来，还是自己太菜了。而且感觉自己这一阵子太过于浮躁，一直就静不下心来，正好通过这次比赛好好反省一下吧。

(2018年11月11日）

复现环境：http://47.105.148.65:4000/

crack it

给出了一个shadow文件，百度知道这是Linux下用来记录和存储密码信息的，/etc/shadow文件中的记录行与/etc/passwd中的一一对应，它由pwconv命令根据/etc/passwd中的数据自动产生。

可以使用kali下的John the ripper来尝试破解，具体命令为：john shadow(要破解的文件）显示破解的命令：john shadow --show

Mobile

1. sign_in

这是一个简单的apk逆向分析题，我们先在模拟器里运行一下这个apk，随便输入一下，发现弹出了“Try again.", 然后反编译成java代码，直接搜索这个字符串就可以定位到核心代码了。

```
if (str.equals(new String(Base64.decode(new StringBuffer(getFlag()).reverse().toString(), 0)))) {

        showMsgToast("Congratulations !");

 private String getFlag() {

      return getBaseContext().getString(R.string.toString);

}
```

分析上面的代码可以看出将输入的内容与getFlag的返回参数反序然后加密的结果进行比较，相等就返回Congratulations。所以，我们可以得到getFlag参数进行上面的处理。这里可以在反编译的资源文件里直接搜索toString方法，就可以得到那个参数。然后用python进行以下处理就可以了。

```
>>> s="991YiZWOz81ZhFjZfJXdwk3X1k2XzlXZlt3ZhxmZ"

>>> s[::-1]

'ZmxhZ3tlZXlzX2k1X3kwdXJfZjFhZ18zOWZiY199'

>>> import base64

>>> s="ZmxhZ3tlZXlzX2k1X3kwdXJfZjFhZ18zOWZiY199"

>>> base64.b64decode(s)
```

'flag{Her3_i5_y0ur_f1ag_39fbc_}'

1. fake-func

```
if(check.checkflag(((EditText) MainActivity.this.findViewById(R.id.editText)).getText().toString())) {
            Toast.makeText(MainActivity.this, "you are right~!", 1).show();
        } else {
            Toast.makeText(MainActivity.this, "wrong!", 1).show();
```

这里调用了so中的Check.checkflag函数来验证flag，我们定位到.so文件中来进行分析吧。发现与一个固定字符串来进行比较，把这个字符串进行base64解密看看吧。提交发现是错的。伤心ing。然后我就放弃了。后来看writeup，发现这里用了hook技术来处理。

1. andorid木马分析

题目描述：在某次安全检查中，工作人员在受害者手机中提取到了两个apk文件，其中有一个是木马，请找到木马文件，并分析出该木马的C&C通信服务器域名。flag的提交方式为：flag{域名}，如你分析出的结果是http://www.baidu.com ，那么flag就是flag{www.baidu.com}

C&C服务器，其全称为command and control server，远程命令和控制服务器，目标机器可以接收来自服务器的命令，从而达到服务器控制目标机器的目的。该方法常用于病毒木马控制被感染的机器。

通过分析题目信息可以知道flag应该是C&C服务器的url，直接搜索geturl，可以看到 InputStream is = getAssets().open("logo.png")，将url加密到了logo.png图片里，然后我们继续分析代码看看进行了什么样的加密。outBuffer[i] = (byte) (buffer[i] ^ PASS[i % PASS.length]);进行了异或操作。

```
public String c(String d) {
    int i;
    byte[] a = d.getBytes();//将文件字符串以字节的形式读入
    for (i = 0; i < d.length() / 2; i += 2) {//将后面的字和前面的字交换
        byte t = a[i];
        a[i] = a[(d.length() - 1) - i];
        a[(d.length() - 1) - i] = t;
    }
    String[] s = new String(a).split(",");
    String dx = "";
    for (i = 0; i < s.length; i += 2) {
        dx = new StringBuilder(String.valueOf(dx)).append((char) Integer.parseInt(s[i])).toString();
    }
    return dx; //将字节转成字符串
}
```

```
byte[] buffer = new byte[size];
        byte[] outBuffer = new byte[size];
        is.read(buffer);
        for (int i = 0; i < size; i++) {
            outBuffer[i] = (byte) (buffer[i] ^ PASS[i % PASS.length]);//对文件数据进行xor处理
        }
        url = new String(outBuffer);
```

Re

1. File

这个题目的核心算法是读取文件的内容与一个固定的字符串和一个固定的hex数组，转换成10进制和数组的长度进行异或，以为异或运算是可逆的，所以，我们可以得到文件的内容，将文件的内容转成16进制的数据，写到一个文件中，查看这个文件的md5就是这个题目的flag。

Pwn1 repeat

这道题主要考察了变量覆盖和格式化字符串漏洞和got表的改写。

首先用nc连接到远程的服务器，发现有一个简单的输入和输出的交互，随便输入几个%x，可以能把变量的十六进制的值打印出来，说明存在格式化字符串的漏洞。看到循环次数受totalcount的限制，然后我们想到用pwntools框架中的fmtstr_payload(4,{address(totalcount):change_value}来修改，同理，number的值也这样来改写。这里有一个GetFlag函数包含system(‘/bin/sh’)可以考虑改写put函数的got表到这个函数的地址，这里有个小技巧来获取put函数got表的地址就是：使用pwntools的elf模块，e=ELF('./pwnnner')printhex(e.got['puts'])

Affine

这是一个仿射密码的解密题目，比较简单，当时也没考虑很多，就写了一个暴力破解的脚本来做的，原理是把26个字符都加密一遍，如果和这个密文相等就输出这个字符，拼接成字符串就是flag。

日志分析

这个题目的日志分析是sqlmap二分法扫描的payload日志，我们可以写个脚本来筛选出成功注入的payload，并且把探测到的字符连接起来就是flag。