2017GCTF部分writeup

转载

weixin_30443895 于 2017-06-12 22:01:00 发布 138 收藏
文章标签： java c# python
原文链接：http://www.cnblogs.com/lllkh/p/6995258.html

0x00:热身题

渗透测试大法：第一招，扫端口；第二招，... 。

扫后台试试呗，用御剑扫到存在robots.txt,访问发现很多个Disallow:可能的试试，发现flag在/rob0t.php中

flag:GCTF{ae609880185f1d75}

0x01:reverseMe

下载下来的文件用winhex查看一下发现头部D9FF很眼熟啊，想起来和JPEG文件格式的尾部FFD9正好反过来而且题目是reverseMe，赶紧去看看尾部D8FF正好是JPEG文件头倒过来，那么就很明显了就是让我们把文件数据反过来，这里在windows下的python中总是读不全该文件，扔到linux中去执行就ok了，改成jpg后缀查看是个倒着的flag直接反着输入即可。

#python脚本

```
a=open("ae0c42b1-5e0d-4600-abc8-40f36a763061.reverseMe")
temp=a.read()
temp=temp[::-1]
b=open("1.jpg",'w+')
b.write(temp)
```

0x02:text.pyc

文件下载下来后是个pyc文件去在线pyc反编译：https://tool.lu/pyc/，发现有部分不能编译，又找到了NSCTF的re500和这题很类似：http://blog.nsfocus.net/wp-content/uploads/2015/09/reverse_500.pdf，去用uncompyle反编译找到报错的位置，这里专门去研究了一下pyc的字节码，开始报错的位置是3个NOP也就是对应字节码09，然后发现有三个LOAD_CONST并没有用到，猜测应该是把这几个字符串都相加起来，也就是修改成为头两个LOAD_CONST然后BINARY_ADD,再一个LOAD_CONST再BINARY_ADD的形式：

```
00 02 00 00 00 40 00 00  00 73 48 00 00 00 64 0D
00 64 03 00 17 64 04 00  17 64 05 00 17 64 06 00
17 5A 00 00 64 07 00 64  08 00 6C 01 00 5A 01 00
```

即64 0D 00 64 03 00 17 64 04 00 17 64 05 00 17 64 06 00 17

再把改了后的pyc拿去反编译可以看到前面的str变成了5个字符串相加，后面出错的地方为flag3的函数位置有4个00字节码，比较前面两个函数的字节码发现这四个字节码是多余的，于是删除了这四个00，但是发现删除后并不能运行，最后自己慢慢的研究了下字节码发现flag3函数的字节码和前两个函数基本相同，不同的就是多了个base64的解码，于是还原原py文件再运行flag3()拿到flag：

#python代码

```
str = '=cWbihGfyMzNIlzZ' +'0cjZzMW'+'N5cTM4Y'+'jYygTOy' + 'cmNycWNyYmM1Ujf'
import base64

def flag1():
code = str[::-3]
result = ''
for i in code:
ss = ord(i) - 1
result += chr(ss)
print result[::-1]


def flag2():
code = str[::-2]
result = ''
for i in code:
ss = ord(i) - 1
result += chr(ss)
print result[::-2]


def flag3():
code = str[::-1]
code = base64.b64decode(code)
result = ''
for i in code:
ss = ord(i) - 1
result += chr(ss)
print result[::-1]

flag3()
```

执行结果：

flag{126d8f36e2b486075a1781f51f41e144}结束后了解到删除那四个字节后还要改前面的总字节数

0x03：APK逆向

下载后dex2jar然后jd-gui去看代码，思路还是比较清晰的大体就是把字符串Tenshine，getbytes的md5值去执行tohexstring函数得出一个字符串和你输入的字符串比较，直接把两个函数拿出来写个java执行就行了：

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class one {
public static void main(String[] args) {
try{
String str="Tenshine";
MessageDigest localMessageDigest = MessageDigest.getInstance("MD5");
localMessageDigest.reset();
localMessageDigest.update(str.getBytes());
String str1 = toHexString(localMessageDigest.digest(), "");
StringBuilder localStringBuilder = new StringBuilder();
for (int i = 0; i < str1.length(); i += 2)
localStringBuilder.append(str1.charAt(i));
String str2 = localStringBuilder.toString();
System.out.println(str2);
}
catch (NoSuchAlgorithmException localNoSuchAlgorithmException)
{
localNoSuchAlgorithmException.printStackTrace();
}
}

private static String toHexString(byte[] paramArrayOfByte, String paramString)
{
StringBuilder localStringBuilder = new StringBuilder();
int i = paramArrayOfByte.length;
for (int j = 0; j < i; j++)
{
String str = Integer.toHexString(0xFF & paramArrayOfByte[j]);
if (str.length() == 1)
localStringBuilder.append('0');
localStringBuilder.append(str).append(paramString);
}
return localStringBuilder.toString();
}

}
```

0x04:debug.exe

用peid查看发现是.net程序，用ILSPY去查看程序，查到关键类，同样直接把函数拿出来写个C#即可：

```
private static int a(int A_0, int A_1)
{
return (new int[]
{
2,
3,
5,
7,
11,
13,
17,
19,
23,
29,
31,
37,
41,
43,
47,
53,
59,
61,
67,
71,
73,
79,
83,
89,
97,
101,
103,
107,
109,
113
})[A_1] ^ A_0;
}

private static string b(string A_0)
{
byte[] bytes = Encoding.ASCII.GetBytes(A_0);
return "flag{" + BitConverter.ToString(new MD5CryptoServiceProvider().ComputeHash(bytes)).Replace("-", "")
+ "}";
}
```

```
private static void c(string A_0, int A_1, ref string A_2)
{
int num = 0;
if (0 < A_0.Length)
{
do
{
char c = A_0[num];
int num2 = 1;
do
{
c = Convert.ToChar(a(Convert.ToInt32(c), num2));
num2++;
}
while (num2 < 15);
A_2 += c;
num++;
}
while (num < A_0.Length);
}
A_2 = b(A_2);
}



private void button1_Click(object sender, EventArgs e)
{
string str = null;
string value = string.Format("{0}", DateTime.Now.Hour + 1);
string a_ = "CreateByTenshine";
c(a_, Convert.ToInt32(value), ref str);
textBox1.Text = str;
}
}
```

把CreateByTenshine和当前小时+1去做函数c返回的str即为flag，这里好像发现最后答案不管在几点都一定。

flag{967DDDFBCD32C1F53527C221D9E40A0B}

转载于:https://www.cnblogs.com/lllkh/p/6995258.html