【BugKu-CTF论坛writeup(杂项)】就五层你能解开吗

原创

Kingyo12 于 2018-03-18 21:43:53 发布 3094 收藏 2
分类专栏： BugKu-CTF论坛writeup(杂项)

本文链接：https://blog.csdn.net/preserphy/article/details/79599397

就五层你能解开吗

300

链接: http://pan.baidu.com/s/1i4TQoz7 密码: w65m

提示：第一层：CRC32 碰撞

第二层：维吉尼亚密码

第三层：sha1 碰撞

第四层：md5 相同文件不同

第五层：RSA

在网盘里下载下来一个压缩包文件。然后开始根据提示来做。

==========

第一层：CRC32碰撞

借助了网上大神的脚本，附github链接https://github.com/theonlypwner/crc32

碰撞结果如下

```
4 bytes: {0x1c, 0x00, 0x1c, 0xa1}
verification checksum: 0x7c2df918 (OK)
alternative: 5EJeBD (OK)
alternative: 74bFvQ (OK)
alternative: D4WldU (OK)
alternative: Jvea5S (OK)
alternative: OSgAFe (OK)
alternative: WtUlWB (OK)
alternative: XgD1qA (OK)
alternative: _3n26b (OK)
alternative: _CRC32 (OK)
alternative: aSKHAn (OK)
alternative: dvIh2X (OK)
alternative: fJLvkE (OK)
alternative: hESFWK (OK)
alternative: l1r6Sx (OK)
alternative: pbakFl (OK)
alternative: uGcK5Z (OK)
alternative: vgh8vJ (OK)
alternative: xt8TKP (OK)
alternative: ytyePI (OK)
```

```
4 bytes: {0xad, 0xd5, 0xfa, 0x78}
verification checksum: 0xa58a1926 (OK)
alternative: 1Jnhwi (OK)
alternative: 3W5fG8 (OK)
alternative: LEDrYc (OK)
alternative: N4lQmv (OK)
alternative: Tbv_HD (OK)
alternative: ZmiotJ (OK)
alternative: _i5_n0 (OK)
alternative: bxy760 (OK)
alternative: jS1DST (OK)
alternative: kSpuHM (OK)
alternative: lJwKbf (OK)
alternative: rhL5Cg (OK)
alternative: s9oe4b (OK)
alternative: stBXYj (OK)
alternative: tmEfsA (OK)
alternative: zbZVOO (OK)
```



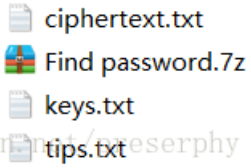
```
4 bytes: {0x1b, 0xd6, 0x38, 0xc2}
verification checksum: 0x4dad5967 (OK)
alternative: 9rNYn3 (OK)
alternative: Ay8sZC (OK)
alternative: QHSaFX (OK)
alternative: TmQA5n (OK)
alternative: VQT_ls (OK)
alternative: X28BT9 (OK)
alternative: _GLQzV (OK)
alternative: goMEPt (OK)
alternative: nyUKFQ (OK)
alternative: t_s4f3 (OK)
alternative: xQxVkx (OK)
alternative: yQ9gpa (OK)
```


从里面选取有意义的字符串连接起来。发现可以连成_CRC32_i5_n0t_s4f3

输入密码解压。解压成功。

第二层：维吉尼亚密码

继续解压里面的压缩包。得到下面几个文件



先看一下tips

```
你知道维吉尼亚密码吗？
我们给了keys.txt，唯一的密钥就在其中，那么解密ciphertext.txt里的密文吧！
解压密码就在明文里，祝你好运！

Do you know the Vigenére Ciphers?
We gave the keys.txt, Only have a key in it, So decrypts ciphertext.txt!
Unzip Password in plaintext, good luck to you!
```


百度了一下维吉尼亚密码。可以知道维吉尼亚密码就是由单一的凯撒密码演变过来的多表查询密码，具体的加解密方式百度百科就有，很简单，这里不再赘述。破解的脚本网上也有很多，也可以自己写，这里就不放出来了。

我们来看一下密文。开头是三个字母的单词，常用的三个字母的单词作为开头的就是the、her、she、his、but、and等等，我试了一下，假如rla对应的明文是the，那么它的密钥就是YEW。

找到了一串以YEW开头的密钥 YEWCQGEWCYBNHDHPXOYUBJJPQIRAPSOUIYEOMTSV ，试着用这个密钥破解一下。果然解出来了一段有意义的文字

```
vigenere cipher funny
```

用这个密码进行解密压缩包，成功。

第三层：sha1 碰撞

```
恭喜!

现在我们遇到一个问题,我们有一个zip文件,但我们不知道完整的解压密码。
幸好我们知道解压密码的一部分sha1值。
你能帮我们找到的密码吗?

不完整的密码："*7*5-*4*3?"  *代表可打印字符

不完整的sha1："619c20c*a4de755*9be9a8b*b7cbfa5*e8b4365*"  *代表可打印字符

人生苦短，我用Python。

Congratulations!
Now we run into a problem,We have a zip file, but we don't know the complete unzip password.
Fortunately, we know that part of the unzip password of sha1 value.
can you help us to find the password?

Incomplete password is "*7*5-*4*3?"   * in the range of ASCII printable characters

Incomplete sha1 is "619c20c*a4de755*9be9a8b*b7cbfa5*e8b4365*"    * in the range of ASCII printab

Life is short, you need Python.                                    //blog.csdn.net/preserphy
```

行吧，继续使用脚本进行解密。（所以这道杂项就是考python脚本的么？）

```
17~5-s4F3?
```

嗯，继续用这个密码解密压缩包。成功。

第四层：md5 相同文件不同

```
Hello World ;-)
MD5校验真的安全吗?
有没有两个不同的程序MD5却相同呢?
如果有的话另一个程序输出是什么呢?
解压密码为单行输出结果。

Hello World ;-)
MD5 check is really safe?
There are two different procedures MD5 is the same?
If so what is the output of another program?
The decompression password is a single-line output.
```

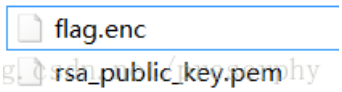
看到这段话让我很迷茫，所以就不得不去百度了一下。然后看到了安全客的一篇文章

> 小编发现，其实早在2011年，就有几位密码学家使用“构造前缀碰撞法”（chosen-prefix collisions）制作出了md5一样但运行结果不一样的exe可执行文件。
>
> **HelloWorld-colliding.exe**
>
> **GoodbyeWorld-colliding.exe**
>
> 下面这两个程序会在屏幕上打印出不同的字符，但是它们的 MD5 都是一样的。

把这两个可执行文件下载下来运行了一下，第一个输出了 `Hello World ;-)`，第二个输出了 `Goodbye World :-(`

所以我们尝试使用Goodbye World :-(作为密码解压压缩包。成功。

第五层：RSA



可以看到有个RSA公钥文件，一个名为flag的加密后的文件。

想哭，最后一步卡住了……先存草稿，改天继续……

同学说可以用openssl打开pem文件查看，所以我就查了一下openssl命令的使用方法，然后尝试打开

```
root@kali:~/桌面# openssl rsa -inform PEM -in rsa_public_key.pem -noout -modulus -
text -pubin
Public-Key: (1026 bit)
Modulus:
    02:8f:ff:9d:d3:e6:fe:97:81:64:9e:b7:fe:5e:93:
    03:cf:69:63:47:c4:11:0b:c4:ba:39:69:f0:b1:16:
    69:84:0c:51:d8:1a:68:42:b6:df:2b:09:0f:21:cd:
    76:d4:37:1a:8c:0e:47:04:8c:96:5e:ca:5b:46:91:
    3a:fb:b8:da:05:20:72:a0:56:6d:70:39:c6:18:ab:
    a9:06:57:59:b0:59:e2:9e:48:5d:c5:06:1a:16:ac:
    63:12:94:38:d9:35:4e:65:df:57:47:54:6b:85:db:
    3d:69:98:19:c4:b7:73:2d:f9:27:c7:08:4a:5d:52:
    d6:e6:d6:aa:c1:44:62:34:25
Exponent:
    01:f8:fb:a4:10:05:2d:f7:ed:a3:46:2f:1a:ac:d6:
    9e:40:76:04:33:ca:33:57:67:cd:73:05:a3:d0:90:
    80:5a:5f:d4:05:dd:6e:ea:70:e9:8f:0c:a1:e1:cf:
    25:47:48:67:1b:f0:c9:80:06:c2:0e:ee:1d:62:79:
    04:35:09:fe:7a:98:23:8b:43:91:60:a5:61:2d:a7:
    1e:90:45:14:e8:12:80:61:7e:30:7c:3c:d3:31:3f:
    a4:c6:fc:a3:31:59:d0:44:1f:bb:18:d8:3c:af:4b:
    d4:6f:6b:92:97:a8:0a:14:2d:d6:9b:f1:a3:57:cc:
    b5:e4:c2:00:b6:d9:0f:15:a3
Modulus=28FFF9DD3E6FE9781649EB7FE5E9303CF696347C4110BC4BA3969F0B11669840C51D81A684
2B6DF2B090F21CD76D4371A8C0E47048C965ECA5B46913AFBB8DA052072A0566D7039C618ABA906575
9B059E29E485DC5061A16AC63129438D9354E65DF5747546B85DB3D699819C4B7732DF927C7084A5D5
2D6E6D6AAC144623425
```

可以看到Exponent很大，Exponent就是指数，百度了一下，发现当指数很大的时候，可以使用维纳攻击。在github上找了一下维纳攻击的工具，果然有…github真的是个很可爱的地方/

```
d= 826466797229427501729333977237178332216882214947197683422108239340
9363691895
```

所以我们现在知道了私钥d那么就可以生成私钥文件来破解flag了。

先生成私钥文件【这里卡了我好久，我是在github上面找的rsatool可以在知道d,n,e的情况下生成私钥的pem文件，但是它依赖于gmpy一个很强大的数学库，然而我安装过程中各种报错各种百思不得其解…后来发现虚拟机没有联网呵呵呵……】

```
python rsatool.py -o rsapkey.pem -f PEM -e 0x01f8fba4
10052df7eda3462f1aacd69e40760433ca335767cd7305a3d090805a5fd405dd6eea70e98f0ca1e1
cf254748671bf0c98006c20eee1d6279043509fe7a98238b439160a5612da71e904514e81280617e
307c3cd3313fa4c6fca33159d0441fbb18d83caf4bd46f6b9297a80a142dd69bf1a357ccb5e4c200
b6d90f15a3 -n 0x028fff9dd3e6fe9781649eb7fe5e9303cf696347c4110bc4ba3969f0b1166984
0c51d81a6842b6df2b090f21cd76d4371a8c0e47048c965eca5b46913afbb8da052072a0566d7039
c618aba9065759b059e29e485dc5061a16ac63129438d9354e65df5747546b85db3d699819c4b773
2df927c7084a5d52d6e6d6aac144623425 -d 82646679722942750172933397723717833221688 2
21494719768342210823934093636918 95
Using (n, d) to initialise RSA instance

n =
28fff9dd3e6fe9781649eb7fe5e9303cf696347c4110bc4ba3969f0b11669840c51d81a6842b6df2
b090f21cd76d4371a8c0e47048c965eca5b46913afbb8da052072a0566d7039c618aba9065759b05
9e29e485dc5061a16ac63129438d9354e65df5747546b85db3d699819c4b7732df927c7084a5d52d
6e6d6aac144623425

e =
1f8fba410052df7eda3462f1aacd69e40760433ca335767cd7305a3d090805a5fd405dd6eea70e98
f0ca1e1cf254748671bf0c98006c20eee1d6279043509fe7a98238b439160a5612da71e904514e81
280617e307c3cd3313fa4c6fca33159d0441fbb18d83caf4bd46f6b9297a80a142dd69bf1a357ccb
5e4c200b6d90f15a3

d =
1245a2e4c321ada55905c249b7e09640f88a41cabd63c932b44e010d3788c977

p =
225ffc6f7c6d8959696663e4cdb90d4a250a99a2122bf6b3d9a4431a1a8d2e6cd7848af4e396b97e
c1109ea9cf30cf67af2215cc2cbd3a754ff0aa6407df9b49b

q =
13156850d0559551936f425e83ec24bf0905591f13e4a07857f8849f36b4f904431b50aa4509baf1
369583e32fff2b25af10e39de868cea7706efee8c36fb663f

Saving PEM as rsapkey.pem
```


接下来就是利用openssl进行解密了——

```
openssl rsautl -decrypt -in flag.enc -inkey rsapkey.pem
flag{W0rld_Of_Crypt0gr@phy}
```


好了，到这里就找到flag了！！！！！！

==========

心痛，折腾了这么久终于做出来了。不过，在做题的过程中学到了不少东西，还收获了好几个工具脚本嘿嘿嘿&…