picoCTF2018 Writeup之General skills篇

Stefano / 2018-10-17 18:33:42 / 浏览数 4400 安全技术 CTF 顶(1) 踩(0)

前言

接上篇，General-skills部分主要是一些linux命令的使用和小tricks。部分题目文件已打包为附件。

General Warmup 1 2 3

Question

> If I told you your grade was 0x41 in hexadecimal, what would it be in ASCII?
>
> Can you convert the number 27 (base 10) to binary (base 2)?
>
> What is 0x3D (base 16) in decimal (base 10).

Solution

三题签到题，差不多一个意思，就放一起了。

```
>>> chr(0x41)
'A'
>>> bin(27)[2:]
'11011'
>>> 0x3d
61
```

flag:

- picoCTF{A}
- picoCTF{11011}
- picoCTF{61}

Resources

Question

> We put together a bunch of resources to help you out on our website! If you go over there, you might even find a flag! https://picoctf.com/resources (link)

Hint

No hints available

Solution

打开网页下拉，就能看到flag

```
Thanks for reading the resources page! Here’s a flag for your time: picoCTF{xiexie_ni_lai_zheli}
```

flag:picoCTF{xiexie_ni_lai_zheli}

Grep 1

Question

> Can you find the flag in file? This would be really obnoxious to look through by hand, see if you can find a faster way. You can also find the file in /problems/grep-1_3_8d9cff3d178c231ab735dfef3267a1c2 on the shell server.

Hint

> grep tutorial

Solution

`grep`，(global search regular expression(RE) and print out the line,全面搜索正则表达式并把行打印出来)是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。

`cat file`会有一大堆乱码字符，使用`grep picoCTF file`可以把flag从乱码中提取出来。

```
finn@pico-2018-shell-2:/problems/grep-1_2_ee2b29d2f2b29c65db957609a3543418$ grep picoCTF file
picoCTF{grep_and_you_will_find_42783683}
```

net cat

Question

> Using netcat (nc) will be a necessity throughout your adventure. Can you connect to `2018shell1.picoctf.com` at port `49387` to get the flag?

Hint

nc tutorial

Solution

nc是一个简单、可靠的网络工具，可通过TCP或UDP协议传输读写数据。

通过nc连接题目的服务器得到flag。

```
❯ nc 2018shell2.picoctf.com 36356
That wasn't so hard was it?
picoCTF{NEtcat_iS_a_NEcESSiTy_9454f3e0}
```

flag:`picoCTF{NEtcat_iS_a_NEcESSiTy_9454f3e0}`

pipe

Question

> During your adventure, you will likely encounter a situation where you need to process data that you receive over the network rather than through a file. Can you find a way to save the output from this program and search for the flag? Connect with `2018shell1.picoctf.com 48696`.

Hint

> Remember the flag format is picoCTF{XXXX}
>
> Ever heard of a pipe? No not that kind of pipe... This kind

Solution

管道命令操作符是：|,它能处理经由前面一个指令传出的正确输出信息，也就是 standard output 的信息，然后作为标准的输入 standard input，传递给下一个命令。

连接服务器，配合`grep`得到flag。

```
❯ nc 2018shell2.picoctf.com 34532 |grep picoCTF
picoCTF{almost_like_mario_b797f2b3}
```

flag:`picoCTF{almost_like_mario_b797f2b3}`

Strings

Question

> Can you find the flag in this file without actually running it? You can also find the file in /problems/strings_2_b7404a3aee308619cb2ba79677989960 on the shell server.

Hint

> strings

Solution

`strings`命令可以打印文件中可打印的字符，使用`strings`命令配合`grep`命令可以把flag提取出来。

```
finn@pico-2018-shell-2:/problems/strings_4_40d221755b4a0b134c2a7a2e825ef95f$ strings strings |grep picoCTF
picoCTF{sTrIngS_sAVeS_Time_d3ffa29c}
```

flag:`picoCTF{sTrIngS_sAVeS_Time_d3ffa29c}`

grep 2

Question

> This one is a little bit harder. Can you find the flag in /problems/grep-2_3_826f886f547acb8a9c3fccb030e8168d/files on the shell server? Remember, grep is your friend.

Hint

> grep tutorial

Solution

目录下有许多个文件夹，每个文件夹下面又有文件夹和文件。可以使用`grep -r`选项来递归的搜寻文件。

```
finn@pico-2018-shell-2:/problems/grep-2_3_826f886f547acb8a9c3fccb030e8168d/files$ grep -r picoCTF
files2/file20:picoCTF{grep_r_and_you_will_find_556620f7}
```

flag:`picoCTF{grep_r_and_you_will_find_556620f7}`

Aca-Shell-A

Question

> It's never a bad idea to brush up on those linux skills or even learn some new ones before you set off on this adventure! Connect with nc `2018shell1.picoctf.com 27833`.

Hint

> Linux for Beginners

Solution

这题用到了以下的基础linux命令：

- `ls`
- `cd`
- `rm`
- `whoami`
- `cat`
- 如何执行二进制可执行文件

依照指示输入命令就可以了。

```
$ nc 2018shell1.picoctf.com 27833
Sweet! We have gotten access into the system but we aren't root.
It's some sort of restricted shell! I can't see what you are typing
but I can see your output. I'll be here to help you along.
If you need help, type "echo 'Help Me!'" and I'll see what I can do
There is not much time left!
~/$ ls
blackmail
executables
passwords
photos
secret
```

```
~/$ cd secret
Now we are cookin'! Take a look around there and tell me what you find!
~/secret$ ls
intel_1
intel_2
intel_3
intel_4
intel_5
profile_AipieG5Ua9aewei5ieSoh7aph
profile_Xei2uu5suwangohceedaifohs
profile_ahShaighaxahMooshuP1johgo
profile_ahqueith5aekongieP4ahzugi
profile_aik4hah9ilie9foru0Phoaph0
profile_bah9Ech9oa4xaicohphahfaiG
profile_ie7sheiP7su2At2ahw6iRikoe
profile_of0Nee4laith8odaeLachoonu
profile_poh9eij4Choophaweiwev6eev
profile_poo3ipohGohThi9Cohverai7e
Sabatoge them! Get rid of all their intel files!
~/secret$ rm intel*
Nice! Once they are all gone, I think I can drop you a file of an exploit!
Just type "echo 'Drop it in!' " and we can give it a whirl!
~/secret$ echo 'Drop it in!'
Drop it in!
I placed a file in the executables folder as it looks like the only place we can execute from!
Run the script I wrote to have a little more impact on the system!
~/secret$ cd ..
~/$ cd executables
~/executables$ ls
dontLookHere
~/executables$ ./dontLookHere
...
...
...
Looking through the text above, I think I have found the password. I am just having trouble with a username.
Oh drats! They are onto us! We could get kicked out soon!
Quick! Print the username to the screen so we can close are backdoor and log into the account directly!
You have to find another way other than echo!
~/executables$ whoami
l33th4x0r
Perfect! One second!
Okay, I think I have got what we are looking for. I just need to to copy the file to a place we can read.
Try copying the file called TopSecret in tmp directory into the passwords folder.
~/executables$ cp /tmp/TopSecret passwords
Server shutdown in 10 seconds...
Quick! go read the file before we lose our connection!
~/executables$ cd ..
~/$ ls
blackmail
executables
passwords
photos
secret
~/$ cd passwords
~/passwords$ ls
TopSecret
~/passwords$ cat TopSecret
Major General John M. Schofield's graduation address to the graduating class of 1879 at West Point is as follows: The discipli
picoCTF{CrUsHeD_It_9edaa84a}
```

flag:`picoCTF{CrUsHeD_It_9edaa84a}`

environ

Question

> Sometimes you have to configure environment variables before executing a program. Can you find the flag we've hidden in an environment variable on the shell server?

Hint

> unix env

Solution

考察linux系统环境变量，使用env命令可以列出系统中所有的环境变量，配合grep命令得到flag。

```
finn@pico-2018-shell-2:/problems/grep-2_3_826f886f547acb8a9c3fccb030e8168d/files$ env|grep pico
SECRET_FLAG=picoCTF{eNv1r0nM3nT_v4r14Bl3_fL4g_3758492}
```

flag:`picoCTF{eNv1r0nM3nT_v4r14Bl3_fL4g_3758492}`

ssh-keyz

Question

> As nice as it is to use our webshell, sometimes its helpful to connect directly to our machine. To do so, please add your own public key to ~/.ssh/authorized_keys, using the webshell. The flag is in the ssh banner which will be displayed when you login remotely with ssh to with your username.

Hint

> key generation tutorial
>
> We also have an expert demonstrator to help you along. link

Solution

公钥连接服务器相关，在shell服务器中使用`ssh-keygen -t rsa`生成自己的公私钥对，默认生成为~/.ssh/id_rsa.pub(公钥)和~/.ssh/id_rsa（私钥），将私钥下载到本地，然后用`ssh <username>@2018shell2.picoctf.com`连接服务器即可在欢迎信息的banner中看到flag。

这题也可以直接查看banner文件。

```
finn@pico-2018-shell-2:~$ cat /etc/ssh/sshd_config |grep banner
Banner /opt/ssh_banner
finn@pico-2018-shell-2:~$ cat /opt/ssh_banner
picoCTF{who_n33ds_p4ssw0rds_38dj21}
```

flag:`picoCTF{who_n33ds_p4ssw0rds_38dj21}`

what base is this?

Question

> To be successful on your mission, you must be able read data represented in different ways, such as hexadecimal or binary. Can you get the flag from this program to prove you are ready? Connect with `nc 2018shell1.picoctf.com 1225`.

Hint

> I hear python is a good means (among many) to convert things.
>
> It might help to have multiple windows open

Solution

三个不通的进制转换，分别是二进制、十六进制和八进制，转换为十进制，然后发送对应ascii码的单词，写个脚本处理一下就可以了。

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-


from pwn import *
import re

r = remote('2018shell2.picoctf.com', 31711)
```

```
binary = r.recvuntil('as a word.')
binary = re.findall(r'(\d+)', binary)
binary_word = ''.join([chr(int(i, 2)) for i in binary])
r.sendline(binary_word)

hexnum = r.recvuntil('as a word.')
hexnum = re.findall(r'([0-9a-f]+) as', hexnum)[0]
hexword = hexnum.decode('hex')
r.sendline(hexword)

octal = r.recvuntil('as a word.')
octal = re.findall(r'([0-9]+)', octal)
octal_word = ''.join([chr(int(i, 8)) for i in octal])
r.sendline(octal_word)

print r.recvuntil('}\n')

r.close()
```

```
$ python nc_convert.py
[+] Opening connection to 2018shell2.picoctf.com on port 31711: Done

Input:
You got it! You're super quick!
Flag: picoCTF{delusions_about_finding_values_68051dea}

[*] Closed connection to 2018shell2.picoctf.com port 31711
```

flag:`picoCTF{delusions_about_finding_values_68051dea}`

you can't see me

Question

> '...reading transmission... Y.O.U. .C.A.N.'.T. .S.E.E. .M.E. ...transmission ended...' Maybe something lies in /problems/you-can-t-see-me_3_1a39ec6c80b3f3a18610074f68acfe69.

Hint

> What command can see/read files?
>
> What's in the manual page of ls?

Solution

`ls -a`可以查看以`.`开头的隐藏文件。

```
finn@pico-2018-shell-2:/problems/you-can-t-see-me_2_cfb71908d8368e3062423b45959784aa$ ls -a
.  .    ..
```

其中一个`.`代表当前目录，另一个是一个文件，直接`cat .`会显示

```
finn@pico-2018-shell-2:/problems/you-can-t-see-me_2_cfb71908d8368e3062423b45959784aa$ cat .
cat: .: Is a directory
```

输入`cat`然后用tab键补全就能看到真正的文件名了，是一个`.`和两个■■

```
finn@pico-2018-shell-2:/problems/you-can-t-see-me_2_cfb71908d8368e3062423b45959784aa$ cat .\ \
picoCTF{j0hn_c3na_paparapaaaaaaa_paparapaaaaaa_093d6aff}
```

flag:`picoCTF{j0hn_c3na_paparapaaaaaaa_paparapaaaaaa_093d6aff}`

absolutely relative

Question

> In a filesystem, everything is relative ¯_(ツ)_/¯. Can you find a way to get a flag from this program? You can find it in /problems/absolutely-relative_1_15eb86fcf5d05ec169cc417d24e02c87 on the shell server. Source.

Hint

> Do you have to run the program in the same directory? (⊙.☉)7
>
> Ever used a text editor? Check out the program 'nano'

Solution

阅读程序源码

```
#include <stdio.h>
#include <string.h>

#define yes_len 3
const char *yes = "yes";

int main()
{
   char flag[99];
   char permission[10];
   int i;
   FILE * file;


   file = fopen("/problems/absolutely-relative_0_d4f0f1c47f503378c4bb81981a80a9b6/flag.txt" , "r");
   if (file) {
       while (fscanf(file, "%s", flag)!=EOF)
       fclose(file);
   }

   file = fopen( "./permission.txt" , "r");
   if (file) {
       for (i = 0; i < 5; i++){
           fscanf(file, "%s", permission);
       }
       permission[5] = '\0';
       fclose(file);
   }

   if (!strncmp(permission, yes, yes_len)) {
       printf("You have the write permissions.\n%s\n", flag);
   } else {
       printf("You do not have sufficient permissions to view the flag.\n");
   }

   return 0;
}
```

程序会判断当前目录下时候有一个名为`permission.txt`的文件，且文件内容为`yes`，如果满足，则输出flag。

由于flag文件打开的位置是绝对的，而permission文件的打开位置是相对的，所以只需要到一个自己有写权限的目录下创建一个permission.txt文件，写入`yes`，然后运行程

而我们在用户主目录下就拥有写权限，所以在用户主目录下写入并运行程序：

```
finn@pico-2018-shell-2:~$ echo -n "yes" > permission.txt
finn@pico-2018-shell-2:~$ /problems/absolutely-relative_0_d4f0f1c47f503378c4bb81981a80a9b6/absolutely-relative

You have the write permissions.
picoCTF{3v3r1ng_1$_r3l3t1v3_befc0ce1}
```

flag:`picoCTF{3v3r1ng_1$_r3l3t1v3_befc0ce1}`

in out error

Question

> Can you utlize stdin, stdout, and stderr to get the flag from this program? You can also find it in /problems/in-out-error_2_c33e2a987fbd0f75e78481b14bfd15f4 on the shell server

Hint

> Maybe you can split the stdout and stderr output?

Solution

linux输出重定向，可以参考Linux标准输入、输出和错误和文件重定向。

直接执行程序会把`Rick Roll'd`的歌词和`flag`混在一起输出，flag在`stderr`中，把`stdout`重定向到`/dev/null`就可以得到flag。

```
finn@pico-2018-shell-2:/problems/in-out-error_0_0f875f7714b995dad5946a15be6267a7$ ./in-out-error 1>/dev/null
Please may I have the flag?
picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoC
TF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1
p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_
1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_
7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng
_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6
fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}
picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoC
TF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1
p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_
1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_
7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng
_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6
fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}
picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoC
TF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1
p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}
```

flag:`picoCTF{p1p1ng_1S_4_7h1ng_85f6fd2c}`

learn gdb

Question

> Using a debugging tool will be extremely useful on your missions. Can you run this program in gdb and find the flag? You can find the file in /problems/learn-gdb_0_716957192e537ac769f0975c74b34194 on the shell server.

Hint

> Try setting breakpoints in gdb
>
> Try and find a point in the program after the flag has been read into memory to break on
>
> Where is the flag being written in memory?

Solution

执行程序会解密flag，但是不会输出。

```
finn@pico-2018-shell-2:/problems/learn-gdb_3_f1f262d9d48b9ff39efc3bc092ea9d7b$ ./run
Decrypting the Flag into global variable 'flag_buf'
.....................................
Finished Reading Flag into global variable 'flag_buf'. Exiting.
```

用gdb加载程序，反汇编`main`函数。

```
finn@pico-2018-shell-2:/problems/learn-gdb_3_f1f262d9d48b9ff39efc3bc092ea9d7b$ gdb run
...
(gdb) disassemble main
...
  0x00000000004008f1 <+40>:    callq  0x400650 <setvbuf@plt>
  0x00000000004008f6 <+45>:    mov    $0x4009d0,%edi
  0x00000000004008fb <+50>:    callq  0x400600 <puts@plt>
  0x0000000000400900 <+55>:    mov    $0x0,%eax
  0x0000000000400905 <+60>:    callq  0x400786 <decrypt_flag>
...
```

再反汇编decrypt_flag函数，在打印break-line直接下断点。

```
(gdb) disas decrypt_flag
...

  0x0000000000400896 <+272>:   mov    0x200b4b(%rip),%rdx        # 0x6013e8 <flag_buf>
  0x000000000040089d <+279>:   mov    -0x20(%rbp),%eax
  0x00000000004008a0 <+282>:   cltq
  0x00000000004008a2 <+284>:   add    %rdx,%rax
  0x00000000004008a5 <+287>:   movb   $0x0,(%rax)
  0x00000000004008a8 <+290>:   mov    $0xa,%edi
  0x00000000004008ad <+295>:   callq  0x4005f0 <putchar@plt>
  0x00000000004008b2 <+300>:   nop
...
(gdb) b *0x00000000004008a8
Breakpoint 1 at 0x4008a8
(gdb) r
Starting program: /problems/learn-gdb_3_f1f262d9d48b9ff39efc3bc092ea9d7b/run
Decrypting the Flag into global variable 'flag_buf'
.....................................
Breakpoint 1, 0x00000000004008a8 in decrypt_flag ()
```

`printf "%s", (char *) flag_buf`打印`flag_buf`变量，得到flag。

```
(gdb) printf "%s", (char *) flag_buf
picoCTF{gDb_iS_sUp3r_u53fuL_efaa2b29}
```

flag:`picoCTF{gDb_iS_sUp3r_u53fuL_efaa2b29}`

roulette

Question

> This Online Roulette Service is in Beta. Can you find a way to win $1,000,000,000 and get the flag? Source. Connect with `nc 2018shell1.picoctf.com 5731`

Hint

> There are 2 bugs!

Solution

查看源码，发现出问题的地方有两个。

```
long get_rand() {
 long seed;
 FILE *f = fopen("/dev/urandom", "r");
 fread(&seed, sizeof(seed), 1, f);
 fclose(f);
 seed = seed % 5000;
 if (seed < 0) seed = seed * -1;
 srand(seed);
 return seed;
}
...
int main(int argc, char *argv[]) {
 ...
 cash = get_rand();
 ...
}
```

这里直接用随机数种子作为初始cash值，所以可以用cash来预测后面的随机数。

第二个问题出现在`get_long()`函数。

```
long get_long() {
   printf("> ");
   uint64_t l = 0;
   char c = 0;
   while(!is_digit(c))
     c = getchar();
   while(is_digit(c)) {
     if(l >= LONG_MAX) {
       l = LONG_MAX;
       break;
     }
     l *= 10;
     l += c - '0';
     c = getchar();
   }
   while(c != '\n')
     c = getchar();
   return l;
}
```

这个函数会返回一个有符号长整型的值，但是，在中间的运算过程却使用了一个无符号的uint64_t。

所以当我们构造一个特别大的输入时，就可以造成溢出，使函数返回一个有符号的负数。

基于以上两个bug，我们就可以构造出满足以下要求的输入了。

```
if(cash > ONE_BILLION) { // cash■■■■
   printf("*** Current Balance: $%lu ***\n", cash);
   if (wins >= HOTSTREAK) {// ■■■■■■■■■■■
     puts("Wow, I can't believe you did it.. You deserve this flag!");
     print_flag();
     exit(0);
}
```

预测随机数用的c程序

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
   srand(atoi(argv[1]));
   int i = 0;
   for (i = 0; i < 8; i++) {
       long k = (rand() % 36) + 1;
       if (i % 2 == 0){
           printf("%ld ", k);
       }
   }
   puts("");
   return 0;
}
```

产生随机数序列

```
~ gcc generator.c -o random_predict
~ ./random_predict 611
12 23 34 26
~

$ nc 2018shell2.picoctf.com 48312
Welcome to ONLINE ROULETTE!
Here, have $611 to start on the house! You'll lose it all anyways >:)

How much will you wager?
Current Balance: $611    Current Wins: 0
> 611
Choose a number (1-36)
> 12

Spinning the Roulette for a chance to win $1222!

Roulette  :  12

You chose correct!
```

```
How much will you wager?
Current Balance: $1222   Current Wins: 1
> 1222
Choose a number (1-36)
> 23

Spinning the Roulette for a chance to win $2444!

Roulette  :  23

Wow, you won!

How much will you wager?
Current Balance: $2444   Current Wins: 2
> 2444
Choose a number (1-36)
> 34

Spinning the Roulette for a chance to win $4888!

Roulette  :  34

Congrats!

How much will you wager?
Current Balance: $4888   Current Wins: 3
> 3221225472
Choose a number (1-36)
> 25

Spinning the Roulette for a chance to win $2147483648!

Roulette  :  26

WRONG
If you keep it up, maybe you'll get the flag in 100000000000 years

*** Current Balance: $1073746712 ***
Wow, I can't believe you did it.. You deserve this flag!
picoCTF{1_h0p3_y0u_f0uNd_b0tH_bUg5_8fb4d984}
```

flag:`picoCTF{1_h0p3_y0u_f0uNd_b0tH_bUg5_8fb4d984}`

store

Question

> We started a little store, can you buy the flag? Source. Connect with `2018shell1.picoctf.com 53220`.

Hint

> Two's compliment can do some weird things when numbers get really big!

Solution

也是一个整数溢出的问题，购买一个小额的商品，造成负数溢出就可以使自己的钱数增加。

但是......出题人好像忘记去掉一些东西，直接把flag放在了二进制可执行文件里，所以直接`strings`就能得到flag了`=-=`。

```
~ strings store |grep pico
YOUR FLAG IS: picoCTF{numb3r3_4r3nt_s4f3_dbd42a50}
```

常规解法如下。

```
Welcome to the Store App V1.0
World's Most Secure Purchasing App

[1] Check Account Balance
```

```
[2] Buy Stuff

[3] Exit

Enter a menu selection
2
Current Auctions
[1] I Can't Believe its not a Flag!
[2] Real Flag
1
Imitation Flags cost 1000 each, how many would you like?
10000000000000000

Your total cost is: -1981284352

Your new balance: 1981285452

Welcome to the Store App V1.0
World's Most Secure Purchasing App

[1] Check Account Balance

[2] Buy Stuff

[3] Exit

Enter a menu selection
2
Current Auctions
[1] I Can't Believe its not a Flag!
[2] Real Flag
2
A genuine Flag costs 100000 dollars, and we only have 1 in stock
Enter 1 to purchase1
YOUR FLAG IS: picoCTF{numb3r3_4r3nt_s4f3_dbd42a50}
```

flag:picoCTF{numb3r3_4r3nt_s4f3_dbd42a50}

题目附件.zip (0.385 MB) 下载附件

点击收藏 | 0 关注 | 1

上一篇：picoCTF2018 Write... 下一篇：区块链安全—安全守卫者“哈希函数”...

1. 2 条回复

Pinging 2018-10-17 21:57:13

写的不错哦！

0 回复Ta

robInS0n 2018-10-21 21:04:30

0 回复Ta