网络攻防 WAR Game

2021 版

计算机科学与技术系

实验简介

我们提供的虚拟机镜像是一个基于 Linux 的漏洞挖掘的练习闯关平台。同学们需要从中找到漏洞并突破权限。在完成该平台的练习后，同学们将对漏洞挖掘的原理有一个较为透彻 的理解，并掌握漏洞挖掘的基本方法。

该平台的每一个关卡对应一个名为 levelXX 的账号，密码与账号名相同。在完成每一关的题目之前，你需要用对应的账号登录系统。每个关卡的题目对应的文件都放在 /home/flagXX 目录中。例如，第一关的账号名是 level01，密码也是 level01，如果这个关卡有需要攻击的包含漏洞的程序，那么相应的程序就放在/home/flag01 目录中。每个关卡的内容介绍和相关程序的源代码可以在本实验讲义后面对应的 Level 小节中获得。

/home/flagXX 目录中的程序具备 SUID 权限。例如在关卡 level01 中，用 level01 账号登录进系统，然后执行/home/flag01/flag01，程序将以 flag01 的身份运行。当你利用该程序的漏洞提升自己的权限后，你的身份就将变为 flag01。提权成功后，你需要执行/bin/getflagXX 程序（对本例来说，就是执行/bin/getflag01），如果你确实是以 flagXX 的身份运行该程序，就将获得提示： “Congratulation! The flag is xxx-xxx” ，否则获得的提示为： “Wrong, You are in a non-flag account” 。每个关卡的最终目的就是利用程序漏洞提升自己的权限，然后想办法执行/bin/getflagXX 程序并获得 flag。

每个关卡的闯关方法可能有多种，如果你能在提交的实验报告中提供多种闯关方法，将获得加分。

Level00

为完成这个关卡，你需要以账号名level00、密码level00登录进系统。 这个关卡需要你在系统中找到一个以flag00身份运行的SUID程序。 为完成这个关卡，你可能会用到 Linux 系统中的 find 命令。它的具体用法请使用 man find 命令查看。

Level01

为完成这个关卡，你需要以账号名level01、密码level01登录进系统。这个关卡的程序可以在/home/flag01 目录中找到。

该程序（源代码如下所示）存在一个漏洞，允许任意程序被执行，你能找到吗？

```
flag01.c

#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?");
}
```

Level02

为完成这个关卡，你需要以账号名level02、密码level02登录进系统。这个关卡的程序 可以在/home/flag02 目录中找到。

该程序（源代码如下所示）存在一个漏洞，允许任意程序被执行，你能找到吗？

```
flag02.c
```

```
#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"));

  system(buffer);
}
```

Level03

为完成这个关卡，你需要以账号名level03、密码level03登录进系统。这个关卡的文件可以在/home/flag03 目录中找到。

请检查/home/flag03 目录中的文件。有一个 crontab 文件每隔几分钟就会被调用执行该文件。

Level04

为完成这个关卡，你需要以账号名level04、密码level04登录进系统。这个关卡的文件可以在/home/flag04 目录中找到。

这个关卡需要你读取 token 文件的内容，但下面的代码限制了可以读取的文件。请找到一个方法来绕过它。

```
flag04.c
```

```
#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 can 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);
}
```

Level05

为完成这个关卡，你需要以账号名level05、密码level05登录进系统。这个关卡的文件可以在/home/flag05 目录中找到。

请仔细检查/home/flag05 目录以找到需要的文件。

Level06

为完成这个关卡，你需要以账号名level06、密码level06登录进系统。
要注意的是，账号 flag06 的密码是按照传统 UNIX 的方法储存的。

Level07

为完成这个关卡，你需要以账号名level07、密码level07登录进系统。这个关卡的文件可以在/home/flag07 目录中找到。

用户 flag07 用 perl 语言编写了一个 CGI 程序，它允许用户通过该 CGI 程序来 ping 某个主机，以检查该CGI 程序所在站点是否能访问指定主机。

```
index.cgi

#!/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>");

}

ping(param("Host"));
```

Level08

为完成这个关卡，你需要以账号名level08、密码level08登录进系统。这个关卡的文件可以在/home/flag08 目录中找到。

Level09

为完成这个关卡，你需要以账号名level09、密码level09登录进系统。这个关卡的文件可以在/home/flag09 目录中找到。

在/home/flag09目录中有一个用C语言实现的SUID程序，它调用了下面的PHP代码。

```
flag09.php

<?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;

?>
```

Level10

为完成这个关卡，你需要以账号名level10、密码level10登录进系统。这个关卡的文件可以在/home/flag10 目录中找到。

设置了SUID位的/home/flag10/flag10程序将上传在命令行上指定的文件，只要该文件满足 access()系统调用的需求。

flag10.c

```
#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 can 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:18888 .. ", 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(18888);

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

```
        exit(EXIT_FAILURE);
     }

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

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

     ffd = open(file, O_RDONLY);
     if(ffd == -1) {
        printf("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("write file done!\n");

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

Level11

为完成这个关卡，你需要以账号名level11、密码level11登录进系统。这个关卡的文件可以在/home/flag11 目录中找到。

/home/flag11/flag11 程序处理标准输入并执行一个 shell 命令。

```
flag11.c


#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),
      'a' + (random() % 26), 'A' + (random() % 26),
      '0' + (random() % 10), 'a' + (random() % 26));

  fd = open(*path, O_CREAT|O_RDWR, 0600);
  unlink(*path);
  return fd;
}

void process(char *buffer, int length)
{
  unsigned int key;
  int i;

  key = length & 0xff;

  for(i = 0; i < length; i++) {
      buffer[i] ^= key;
      key -= buffer[i];
  }

  system(buffer);
}
```

```
#define CL "Content-Length: "

int main(int argc, char **argv)
{
  char line[256];
  char buf[1024];
  char *mem;
  int length;
  int fd;
  char *path;

  if(fgets(line, sizeof(line), stdin) == NULL) {
     errx(1, "read from stdin");
  }

  if(strncmp(line, CL, strlen(CL)) != 0) {
     errx(1, "invalid header");
  }

  length = atoi(line + strlen(CL));

  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);
  }
}
```

Level12

为完成这个关卡，你需要以账号名level12、密码level12登录进系统。这个关卡的文件可以在/home/flag12 目录中找到。

系统中存在一个后门进程正在监听端口号50000。

```
flag12.lua

local socket = require("socket")
local server = assert(socket.bind("127.0.0.1", 50000))

function hash(password)
       prog = io.popen("echo "..password.." | sha1sum", "r")
       data = prog:read("*all")
       prog:close()

       data = string.sub(data, 1, 40)

       return data
end

while 1 do
       local client = server:accept()
       client:send("Password: ")
       client:settimeout(60)
       local line, err = client:receive()
       if not err then
              print("trying " .. line) -- log from where ;\
              local h = hash(line)

              if h ~= "4563a4f4bd5787fdsc33de887b9250a0691da546" then
                     client:send("Wrong, try next time\n");
              else
```

```
                    client:send("Congratulation, your token is
523**CARRIER LOST**\n")
             end
       end

       client:close()
end
```

Level13

为完成这个关卡，你需要以账号名level13、密码level13登录进系统。这个关卡的文件可以在/home/flag13 目录中找到。

下面的程序里存在一个安全检查 ,如果调用该程序的用户 id 不匹配一个指定的用户 id，程序就不会继续运行。

```
flag13.c

#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");
     exit(EXIT_FAILURE);
  }

  // some code are snipped

  printf("your token is %s\n", token);

}
```

Level14

为完成这个关卡，你需要以账号名level14、密码level14登录进系统。这个关卡的文件可以在/home/flag14 目录中找到。

/home/flag14/flag14 程序将对输入进行加密，并将密文写到标准输出。一个经过加密的token 文件也在该目录中。

Level15

为完成这个关卡，你需要以账号名level15、密码level15登录进系统。这个关卡的文件可以在/home/flag15 目录中找到。

使用 strace 命令跟踪/home/flag15/flag15 程序，看看你是否能发现一些异常的情况。

你可能需要学习一下如何在 Linux 系统中编译共享库，以及通过查看 dlopen 的手册页来详细了解共享库的加载和处理流程。

Level16

为完成这个关卡，你需要以账号名level16、密码level16登录进系统。这个关卡的文件可以在/home/flag16 目录中找到。

有一个 perl CGI 脚本运行在端口号 1818 上。

```
index.cgi

#!/usr/bin/env perl

use CGI qw{param};

print "Content-type: text/html\n\n";

sub login {
  $username = $_[0];
  $password = $_[1];

  $username =~ tr/a-z/A-Z/;    # conver to uppercase
  $username =~ s/\s.*//;       # strip everything after a space

  @output = `egrep "^$username" /home/flag16/userdb.txt 2>&1`;
  foreach $line (@output) {
      ($usr, $pw) = split(/:/, $line);
```

```
      if($pw =~ $password)
          { return 1;
      }
  }

  return 0;
}

sub htmlz {
  print("<html><head><title>Login result</title></head><body>");
  if($_[0] == 1) {
      print("Your login was accepted<br/>");
  } else {
      print("Your login failed<br/>");
  }
  print("Would you like a cookie?<br/><br/></body></html>\n");
}

htmlz(login(param("username"), param("password")));
```

Level17

为完成这个关卡，你需要以账号名level17、密码level17登录进系统。这个关卡的文件可以在/home/flag17 目录中找到。

有一个包含漏洞的 python 脚本运行在端口号 10008 上。

```
flag17.py

#!/usr/bin/python

import os
import pickle
import time
import socket
import 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', 10008))
skt.listen(10)

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

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

Level18

为完成这个关卡，你需要以账号名level18、密码level18登录进系统。这个关卡的文件可以在/home/flag18 目录中找到。

请分析本关卡给出的C 程序，查找程序中的漏洞。

```
flag18.c

#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)
```

```
{
  FILE *fp;

  fp = fopen(PWFILE, "r");
  if(fp) {
     char file[64];

     if(fgets(file, sizeof(file) - 1, fp) == NULL) {
        dprintf("Unable to read password file %s\n", PWFILE);
        return;
     }
     if(strcmp(pw, file) != 0) return;
  }
  dprintf("logged in successfully (with%s password file)\n",
     fp == NULL ? "out" : "");

  globals.loggedin = 1;
}

void notsupported(char *what)
{
  char *buffer = NULL;
  asprintf(&buffer, "--> [%s] is unsupported at this current time.\n", what);
  dprintf(what);
  free(buffer);
}

void setuser(char *user)
{
  char msg[128];

  sprintf(msg, "unable to set user to '%s' -- not supported.\n", user);
  printf("%s\n", msg);
}

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

  while((c = getopt(argc, argv, "d:v")) != -1) {
     switch(c) {
        case 'd':
           globals.debugfile = fopen(optarg, "w+");
           if(globals.debugfile == NULL) err(1, "Unable to open %s", optarg);
```

```
            setvbuf(globals.debugfile, NULL, _IONBF, 0);
            break;
        case 'v':
            globals.verbose++;
            break;
    }
}

dprintf("Starting up. Verbose level = %d\n", globals.verbose);

setresgid(getegid(), getegid(), getegid());
setresuid(geteuid(), geteuid(), geteuid());

while(1) {
    char line[256];
    char *p, *q;

    q = fgets(line, sizeof(line)-1, stdin);
    if(q == NULL) break;
    p = strchr(line, '\n'); if(p) *p = 0;
    p = strchr(line, '\r'); if(p) *p = 0;

    dvprintf(2, "got [%s] as input\n", line);

    if(strncmp(line, "login", 5) == 0) {
        dvprintf(3, "attempt to login\n");
        login(line + 6);
    } else if(strncmp(line, "logout", 6) == 0) {
        globals.loggedin = 0;
    } else if(strncmp(line, "shell", 5) == 0) {
        dvprintf(3, "attempt to start shell\n");
        if(globals.loggedin) {
            execve("/bin/sh", argv, envp);
            err(1, "unable to execve");
        }
        dprintf("Permission denied\n");
    } else if(strncmp(line, "exit", 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;
}
```

Level19

为完成这个关卡，你需要以账号名level19、密码level19登录进系统。这个关卡的文件可以在/home/flag19 目录中找到。

这个关卡的程序在运行时存在一个漏洞。

```
flag19.c

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

int main(int argc, char **argv, char **envp)
{
  pid_t pid;
  char buf[256];
  struct stat statbuf;

  /* Get the parent's /proc entry, so we can verify its user id */

  snprintf(buf, sizeof(buf)-1, "/proc/%d", getppid());

  if(stat(buf, &statbuf) == -1) {
     printf("Unable to check parent process\n");
     exit(EXIT_FAILURE);
  }

  /* check the owner id */

  if(statbuf.st_uid == 0) {
     /* If root started us, it is ok to start the shell */

     execve("/bin/sh", argv, envp);
```

```
    err(1, "Unable to execve");
  }

  printf("You are unauthorized to run this program\n");
}
```

Level20

为完成这个关卡，你需要以账号名level20、密码level20登录进系统。这个关卡的文件 可以在/home/flag20 目录中找到。

```
flag20.c

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

int main(int argc, char **argv)
{
  volatile int modified;
  char buffer[64];

  modified = 0;
  gets(buffer);

  if(modified != 0)
     { execl("/bin/getflag20","getflag20",NULL);
  } else {
     printf("Try again?\n");
  }
}
```

Level21

为完成这个关卡，你需要以账号名level21、密码level21登录进系统。这个关卡的文件 可以在/home/flag21 目录中找到。

```
flag21.c

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

```
#include <err.h>

int main(int argc, char **argv)
{
  volatile int modified;
  char buffer[64];

  if(argc == 1) {
      errx(1, "please specify an argument\n");
  }

  modified = 0;
  strcpy(buffer, argv[1]);

  if(modified == 0x56575859)
      { execl("/bin/getflag21","getflag21",NULL);
  } else {
      printf("Try again, you got 0x%08x\n", modified);
  }
}
```

Level22

为完成这个关卡，你需要以账号名level22、密码level22登录进系统。这个关卡的文件 可以在/home/flag22 目录中找到。

```
flag22.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <err.h>

int main(int argc, char **argv)
{
  volatile int modified;
  char buffer[64];
  char *variable;

  variable = getenv("NJUCS");

  if(variable == NULL) {
      errx(1, "please set the NJUCS environment variable\n");
```

```
  }

  modified = 0;

  strcpy(buffer, variable);

  if(modified == 0x0d0a0d0a)
      { execl("/bin/getflag22","getflag22",NULL);
  } else {
      printf("Try again, you got 0x%08x\n", modified);
  }
}
```

Level23

为完成这个关卡，你需要以账号名level23、密码level23登录进系统。这个关卡的文件可以在/home/flag23 目录中找到。

```
flag23.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

void win()
{
  execl("/bin/getflag23","getflag23",NULL);
}

int main(int argc, char **argv)
{
  volatile int (*fp)();
  char buffer[64];

  fp = 0;

  gets(buffer);

  if(fp) {
      printf("calling function pointer, jumping to 0x%08x\n", fp);
      fp();
  }
}
```

Level24

为完成这个关卡，你需要以账号名level24、密码level24登录进系统。这个关卡的文件可以在/home/flag24 目录中找到。

在/home/flag24 目录中有两个可执行程序：flag24 和 flag24.odd。在解决了 flag24 之后，可以尝试解决 flag24.odd。

```
flag24.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

void win()
{
  execl("/bin/getflag24","getflag24",NULL);
}

int main(int argc, char **argv)
{
  char buffer[64];

  gets(buffer);
}
```

Level25

为完成这个关卡，你需要以账号名level25、密码level25登录进系统。这个关卡的文件可以在/home/flag25 目录中找到。

```
flag25.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int main(int argc, char **argv)
{
  char buffer[64];

  gets(buffer);
```

```
}
```

Level26

为完成这个关卡，你需要以账号名level26、密码level26登录进系统。这个关卡的文件可以在/home/flag26 目录中找到。

```
flag26.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

void getpath()
{
  char buffer[64];
  unsigned int ret;

  printf("input path please: "); fflush(stdout);

  gets(buffer);

  ret =   builtin_return_address(0);

  if((ret & 0xbf000000) == 0xbf000000)
      { printf("can not get here (%p)\n", ret);
      _exit(1);
  }

  printf("got path %s\n", buffer);
}

int main(int argc, char **argv)
{
  getpath();
}
```

Level27

为完成这个关卡，你需要以账号名level27、密码level27登录进系统。这个关卡的文件可以在/home/flag27 目录中找到。

```
flag27.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

char *getpath()
{
  char buffer[64];
  unsigned int ret;

  printf("input path please: "); fflush(stdout);

  gets(buffer);

  ret =   builtin_return_address(0);

  if((ret & 0xb0000000) == 0xb0000000)
      { printf("bzzzt (%p)\n", ret);
      _exit(1);
  }

  printf("got path %s\n", buffer);
  return strdup(buffer);
}

int main(int argc, char **argv)
{
  getpath();
}
```

Level28

为完成这个关卡，你需要以账号名level28、密码level28登录进系统。这个关卡的文件可以在/home/flag28 目录中找到。

```
flag28.c

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

```
void vuln(char *string)
{
  volatile int target;
  char buffer[64];

  target = 0;

  sprintf(buffer, string);

  if(target == 0xdeadbeef)
     { execl("/bin/getflag28","getflag28",NULL
     );
  }
}

int main(int argc, char **argv)
{
  vuln(argv[1]);
}
```

Level29

为完成这个关卡，你需要以账号名level29、密码level29登录进系统。这个关卡的文件可以在/home/flag29 目录中找到。

```
flag29.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int target;

void vuln()
{
  char buffer[512];

  fgets(buffer, sizeof(buffer), stdin);
  printf(buffer);

  if(target == 64)
     { execl("/bin/getflag29","getflag29",NULL);
  } else {
     printf("target is %d :(\n", target);
```

```
  }
}

int main(int argc, char **argv)
{
  vuln();
}
```

Level30

为完成这个关卡，你需要以账号名level30、密码level30登录进系统。这个关卡的文件可以在/home/flag30 目录中找到。

```
flag30.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int target;

void printbuffer(char *string)
{
  printf(string);
}

void vuln()
{
  char buffer[512];

  fgets(buffer, sizeof(buffer), stdin);

  printbuffer(buffer);

  if(target == 0x01025544)
      { execl("/bin/getflag30","getflag30",NULL);
  } else {
      printf("target is %08x :(\n", target);
  }
}

int main(int argc, char **argv)
{
```

```
  vuln();
}
```

Level31

为完成这个关卡，你需要以账号名level31、密码level31登录进系统。这个关卡的文件可以在/home/flag31 目录中找到。

```
flag31.c

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int target;

void hello()
{
  execl("/bin/getflag31","getflag31",NULL);
  _exit(1);
}

void vuln()
{
  char buffer[512];

  fgets(buffer, sizeof(buffer), stdin);

  printf(buffer);

  exit(1);
}

int main(int argc, char **argv)
{
  vuln();
}
```

Level32

为完成这个关卡，你需要以账号名level32、密码level32登录进系统。这个关卡的文件

可以在/home/flag32 目录中找到。

```
flag32.c

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

struct data
  { char name[64];
};

struct fp
  { int
  (*fp)();
};

void winner()
{
  execl("/bin/getflag32","getflag32",NULL);
}

void nowinner()
{
  printf("level has not been passed\n");
}

int main(int argc, char **argv)
{
  struct data *d;
  struct fp *f;

  d = malloc(sizeof(struct data));
  f = malloc(sizeof(struct fp));
  f->fp = nowinner;

  printf("data is at %p, fp is at %p\n", d, f);

  strcpy(d->name, argv[1]);

  f->fp();
}
```

Level33

为完成这个关卡，你需要以账号名level33、密码level33登录进系统。这个关卡的文件 可以在/home/flag33 目录中找到。

```
flag33.c

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



struct internet
  { int priority;
  char *name;
};

void winner()
{
  execl("/bin/getflag33","getflag33",NULL);
}

int main(int argc, char **argv)
{
  struct internet *i1, *i2, *i3;

  i1 = malloc(sizeof(struct internet)); i1-
  >priority = 1;
  i1->name = malloc(8);

  i2 = malloc(sizeof(struct internet)); i2-
  >priority = 2;
  i2->name = malloc(8);

  strcpy(i1->name, argv[1]);
  strcpy(i2->name, argv[2]);

  printf("and that's a wrap folks!\n");
}
```

Level34

为完成这个关卡，你需要以账号名level34、密码level34登录进系统。这个关卡的文件可以在/home/flag34 目录中找到。

```
flag34.c

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

struct auth
  { char
  name[32]; int
  auth;
};

struct auth *auth;
char *service;

int main(int argc, char **argv)
{
  char line[128];

  while(1) {
     printf("[ auth = %p, service = %p ]\n", auth, service);

     if(fgets(line, sizeof(line), stdin) == NULL) break;

     if(strncmp(line, "auth ", 5) == 0)
        { auth = malloc(sizeof(auth));
        memset(auth, 0, sizeof(auth));
        if(strlen(line + 5) < 31)
           { strcpy(auth->name, line +
           5);
        }
     }
     if(strncmp(line, "reset", 5) == 0)
        { free(auth);
     }
     if(strncmp(line, "service", 6) == 0)
        { service = strdup(line + 7);
     }
     if(strncmp(line, "login", 5) == 0) {
```

```
        if(auth->auth)
           { execl("/bin/getflag34","getflag34",NULL);
        } else {
           printf("please enter your password\n");
        }
     }
  }
}
```

Level35

为完成这个关卡，你需要以账号名level35、密码level35登录进系统。这个关卡的文件可以在/home/flag35 目录中找到。

```
flag35.c

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

void winner()
{
  execl("/bin/getflag35","getflag35",NULL);
}

int main(int argc, char **argv)
{
  char *a, *b, *c;

  a = malloc(32);
  b = malloc(32);
  c = malloc(32);

  strcpy(a, argv[1]);
  strcpy(b, argv[2]);
  strcpy(c, argv[3]);

  free(c);
  free(b);
  free(a);

  printf("dynamite failed?\n");
```

```
}
```