OverTheWire:Bandit通关WriteUp（2019.01.17完）

原创

置顶 cynthrial 于 2018-12-25 14:53:02 发布 11603 收藏 65
分类专栏： Linux 文章标签： WarGame
版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/cynthrial/article/details/85231979
版权

Linux 专栏收录该内容
12 篇文章 0 订阅
订阅专栏

OverTheWire:Bandit通关全攻略WriteUp

背景

OverTheWire:Bandit是一个学习linux命令的WarGame，通过闯关的模式，不断的学习新的命令，对于学习安全和Linux的朋友是一个很好的练习游戏，网址是 http://overthewire.org/wargames/bandit/ 。
这个游戏目前有34关，从Level0—Level34。游戏形式是通过ssh连接游戏服务器，通过各种命令行读取下一关的游戏服务器密钥，然后连接下一关的服务器继续读取，直到通关。

```
SSH Information
Host: bandit.labs.overthewire.org
Port: 2220
```

通关过程

Level 0

The goal of this level is for you to log into the game using SSH. The host to which you need to connect is bandit.labs.overthewire.org, on port 2220. The username is **bandit0** and the password is **bandit0**. Once logged in, go to the Level 1 page to find out how to beat Level 1.
这一关主要是让你选择一个合适ssh工具开始远程，**这一关的用户名和密码均为bandit0**常见的有secureCRT，Xshell, Putty, 不过我最近发现一款免费而且不比Xshell功能少的SSH工具叫MobaXterm，个人推荐。
Linux下更为方便

```
ssh bandit0@bandit.labs.overthewire.org -p 2220
密码：bandit0
```

Level 0–>Level 1

he password for the next level is stored in a file called readme located in the home directory. Use this password to log into **bandit1** using SSH. Whenever you find a password for a level, use SSH (on port 2220) to log into that level and continue the game.
Commands you may need to solve this level
ls, cd, cat, file, du, find
其中du命令是用来查看令也是查看使用空间的，但是与df命令不同的是Linux du命令是查看当前指定文件或目录(会递归显示子目录)占用磁盘空间大小，还是和df命令有一些区别的

```
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
boJ9jbbUNNfktd78OOpsqOltutMc3MY1
```

得到下一关用户名bandit1，密码为boJ9jbbUNNfktd78OOpsqOltutMc3MY1，之后用户名依次类推，不做赘述

Level 1 - Level 2

Level Goal
The password for the next level is stored in a file called - located in the home directory

Commands you may need to solve this level
ls, cd, cat, file, du, find
ls发现文件名是一个-，但是这个在linux中有特殊意义导致直接cat不好用

```
bandit1@bandit:~$ ls
-
bandit1@bandit:~$ cat -
^Z
[1]+  Stopped                 cat -
bandit1@bandit:~$ pwd
/home/bandit1
bandit1@bandit:~$ cat /home/bandit1/-
CV1DtqXWVFXTvM2F0k09SHz0YwRINYA9
```

直接输入绝对路径读取

Level 2 - Level 3

Level Goal
The password for the next level is stored in a file called spaces in this filename located in the home directory

Commands you may need to solve this level
ls, cd, cat, file, du, find

Helpful Reading Material
Google Search for “spaces in filename”
文件名有空格的读取

```
bandit2@bandit:~$ cat spaces\ in\ this\ filename
UmHadQclWmgdLOKQ3YNgjWxGoRMb5luK
```

用cat命令，然后Tab按键补齐，自动将空格转义，实现了密钥读取，或者给文件名加上双引号也可以读取。

Level 3 → Level 4

Level Goal
The password for the next level is stored in a hidden file in the inhere directory.

Commands you may need to solve this level
ls, cd, cat, file, du, find
密钥写在一个隐藏文件里面，通过ls -a参数可以找到隐藏文件

```
bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere/
bandit3@bandit:~/inhere$ ls
bandit3@bandit:~/inhere$ ls -la
total 12
drwxr-xr-x 2 root    root    4096 Oct 16 14:00 .
drwxr-xr-x 3 root    root    4096 Oct 16 14:00 ..
-rw-r----- 1 bandit4 bandit3   33 Oct 16 14:00 .hidden
bandit3@bandit:~/inhere$ cat .hidden
pIwrPrtPN36QITSp3EQaw936yaFoFgAB
```

Level 4 → Level 5

Level Goal
The password for the next level is stored in the only human-readable file in the inhere directory. Tip: if your terminal is messed up, try the “reset” command.

Commands you may need to solve this level

ls, cd, cat, file, du, find

文件说在人类能读懂的文件里面，可以看到当前目录有9个文件，通过file命令可以用于辨识文件类型。

```
bandit4@bandit:~$ ls
inhere
bandit4@bandit:~$ cd inhere/
bandit4@bandit:~/inhere$ ls -a
.   -file00  -file02  -file04  -file06  -file08
..  -file01  -file03  -file05  -file07  -file09
bandit4@bandit:~/inhere$ file ./*
./-file00: data
./-file01: data
./-file02: data
./-file03: data
./-file04: data
./-file05: data
./-file06: data
./-file07: ASCII text
./-file08: data
./-file09: data
bandit4@bandit:~/inhere$ cat ./-file07
koReBOKuIDDepwhWk7jZC0RTdopnAYKh
bandit4@bandit:~/inhere$
```

Level 5 → Level 6

Level Goal

The password for the next level is stored in a file somewhere under the inhere directory and has all of the following properties:

human-readable

1033 bytes in size

not executable

Commands you may need to solve this level

ls, cd, cat, file, du, find

一看有这么多文件夹

```
bandit5@bandit:~$ ls
inhere
bandit5@bandit:~$ cd inhere/
bandit5@bandit:~/inhere$ ls -a
.            maybehere02  maybehere06  maybehere10  maybehere14  maybehere18
..           maybehere03  maybehere07  maybehere11  maybehere15  maybehere19
maybehere00  maybehere04  maybehere08  maybehere12  maybehere16
maybehere01  maybehere05  maybehere09  maybehere13  maybehere17
```

根据特征我们可以用find 命令，找到一个符合条件的文件

```
bandit5@bandit:~/inhere$ find . -type f -size 1033c
./maybehere07/.file2
bandit5@bandit:~/inhere$ cat ./maybehere07/.file2
DXjZPULLxYr17uwoI01bNLQbtFemEgo7
```

附find参数解析
-size n[cwbkMG] : 档案大小 为 n 个由后缀决定的数据块。其中后缀含义为：
b: 代表 512 位元组的区块（如果用户没有指定后缀，则默认为 b）
c: 表示字节数
k: 表示 kilo bytes （1024字节）
w: 字 （2字节）
M:兆字节（1048576字节）
G: 千兆字节 （1073741824字节）
-type c : 档案类型是 c 。
d: 目录
c: 字型装置档案
b: 区块装置档案
p: 具名贮列
f: 一般档案
l: 符号连结
s: socket

Level 6 → Level 7

Level Goal
The password for the next level is stored somewhere on the server and has all of the following properties:

owned by user bandit7
owned by group bandit6
33 bytes in size
Commands you may need to solve this level
ls, cd, cat, file, du, find, grep
又是找文件，那么依然可以使用find命令，只不过参数稍稍的改变

```
bandit6@bandit:~$ find / -size 33c -user bandit7 -group bandit6 2>/dev/null
/var/lib/dpkg/info/bandit7.password
bandit6@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
HKBPTKQnIay4Fw76bEy8PVxKEDQRKTzs
```

后面的`2>/dev/null`因为find命令在根目录下查找会经常有很多权限的报错信息，所有在linux中通常用这种方式将错误信息重定向到“黑洞中”

Level 7 → Level 8

Level Goal
The password for the next level is stored in the file data.txt next to the word millionth

Commands you may need to solve this level
grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

根据提示data.txt中在密钥在millionth中，可以通过grep命令查看

```
bandit7@bandit:~$ ls
data.txt
bandit7@bandit:~$ cat data.txt |grep millionth
millionth       cvX2JJa4CFALtqS87jk27qwqGhBM9plV
```

Level 8 → Level 9

Level Goal
The password for the next level is stored in the file data.txt and is the only line of text that occurs only once

Commands you may need to solve this level
grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

Helpful Reading Material
The unix commandline: pipes and redirects
这题是要找到出现一次的那个行，肯定用uniq命令了，但是使用之前需要用sort命令对文本进行排序，因为uniq命令是通过判断上下两行是否一样来判断的，所以用sort排序一下然后在uniq就能找到唯一出现的那一行了

sort data.txt|uniq -u
sort data.txt|uniq -c
这题找到两种解法，一个是直接-u获取，还有就是-c列出出现的次数，然后从中找到是1的那一行即可

```
bandit8@bandit:~$ sort data.txt |uniq -u
UsvVyFSfZZWbi6wgC7dAFyFuR6jQQUhR
```

Level 9 → Level 10

Level Goal
The password for the next level is stored in the file data.txt in one of the few human-readable strings, beginning with several '=' characters.

Commands you may need to solve this level
grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

直接使用cat命令是很多很杂乱的东西，可以通过string命令查看文件中的字符串，根据提示信息可得下一关密钥以若干个"="开头，可以找到下一关的密钥truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk

```
bandit9@bandit:~$ ls
data.txt
bandit9@bandit:~$ strings data.txt
.MBB
`B6ha
t8lHX u
========== password
NHGu
5xhH
========== truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk
W.uO7
i$2w
epg~
```

Level 10 → Level 11

Level Goal
The password for the next level is stored in the file data.txt, which contains base64 encoded data

Commands you may need to solve this level
grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

Helpful Reading Material
Base64 on Wikipedia
题目提示密钥信息用了base64解码，我们解码即可

```
bandit10@bandit:~$ base64 -d data.txt
The password is IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR
```

Level 11 → Level 12

Level Goal
The password for the next level is stored in the file data.txt, where all lowercase (a-z) and uppercase (A-Z) letters have been rotated by 13 positions

Commands you may need to solve this level
grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

Helpful Reading Material
Rot13 on Wikipedia
tr用来从标准输入中通过替换或删除操作进行字符转换。tr主要用于删除文件中控制字符或进行字符转换。使用tr时要转换两个字符串：字符串1用于查询，字符串2用于处理各种转换。tr刚执行时，字符串1中的字符被映射到字符串2中的字符，然后转换操作开始。
带有最常用选项的tr命令格式为：
tr -c -d -s [“string1_to_translate_from”][“string2_to_translate_to”] < input-file

Rot13是一种特殊的凯撒密码转换，根据题目所说的字母的的顺序旋转了13个位置，就相当去26个字母的前13个位置与后13个位置调换了。那么我们就是用tr命令进行调换

```
bandit11@bandit:~$ cat data.txt
Gur cnffjbeq vf 5Gr8L4qetPEsPk8htqjhRK8XSP6x2RHh
bandit11@bandit:~$ cat data.txt |tr 'a-zA-Z' 'n-za-mN-ZA-M'
The password is 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu
```

Level 12 → Level 13

Level Goal
The password for the next level is stored in the file data.txt, which is a hexdump of a file that has been repeatedly compressed. For this level it may be useful to create a directory under /tmp in which you can work using mkdir. For example: mkdir /tmp/myname123. Then copy the datafile using cp, and rename it using mv (read the manpages!)

Commands you may need to solve this level
grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd, mkdir, cp, mv

Helpful Reading Material
Hex dump on Wikipedia
这是一道比较麻烦的题目，需要我们解压很多层。

```
bandit12@bandit:~$ ls
data.txt
bandit12@bandit:~$ file data.txt
data.txt: ASCII text
bandit12@bandit:~$ xxd -r data.txt > data.bin
-bash: data.bin: Permission denied
```

可以看到这本来是一个文本类型的文件，尝试用xxd转成bin提示权限不够，我们先复制一遍。

```
bandit12@bandit:~$ mkdir /tmp/c1911
bandit12@bandit:~$ cp data.txt /tmp/c1911
bandit12@bandit:~$
bandit12@bandit:~$ cd /tmp/c1911
bandit12@bandit:/tmp/c1911$ ls
data.txt
bandit12@bandit:/tmp/c1911$ xxd  -r data.txt > data.bin
bandit12@bandit:/tmp/c1911$ file data.bin
data.bin: gzip compressed data, was "data2.bin", last modified: Tue Oct 16 12:00:23 2018, max compression, from
Unix
```

复制完是gzip格式，改文件名，解压。

```
bandit12@bandit:/tmp/c1911$ mv data.bin data.gz
bandit12@bandit:/tmp/c1911$ gzip -d data.gz
bandit12@bandit:/tmp/c1911$ ls
data  data.txt
bandit12@bandit:/tmp/c1911$ file data
data: bzip2 compressed data, block size = 900k
```

还有一层bzip2 ???, 继续解压

```
bandit12@bandit:/tmp/c1911$ mv data data.bz2
bandit12@bandit:/tmp/c1911$ bunzip2 -d data.bz2
bandit12@bandit:/tmp/c1911$ file data
data: gzip compressed data, was "data4.bin", last modified: Tue Oct 16 12:00:23 2018, max compression, from Unix
```

还有没有解压的文件，继续搞搞吧！！！一直一直查看文件类型，重命名，解压。直到第八层压缩。。。。。

```
bandit12@bandit:/tmp/c1911$ mv data data.gz
bandit12@bandit:/tmp/c1911$ gzip -d data.gz
bandit12@bandit:/tmp/c1911$ file data
data: POSIX tar archive (GNU)
bandit12@bandit:/tmp/c1911$ mv data data.tar
bandit12@bandit:/tmp/c1911$ tar xvf data.tar
data5.bin
bandit12@bandit:/tmp/c1911$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/c1911$ mv data5.bin data5.tar
bandit12@bandit:/tmp/c1911$ tar xvf data5.tar
data6.bin
bandit12@bandit:/tmp/c1911$ file data6
data6: cannot open `data6' (No such file or directory)
bandit12@bandit:/tmp/c1911$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/c1911$ bunzip2 -d data6.bin
bunzip2: Can't guess original name for data6.bin -- using data6.bin.out
bandit12@bandit:/tmp/c1911$ file data6.bin.out
data6.bin.out: POSIX tar archive (GNU)
bandit12@bandit:/tmp/c1911$ mv data6.bin.out data.tar
bandit12@bandit:/tmp/c1911$ tar xvf data.tar
data8.bin
bandit12@bandit:/tmp/c1911$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: Tue Oct 16 12:00:23 2018, max compression, from
 Unix
bandit12@bandit:/tmp/c1911$ mv data8.bin data8.gz
bandit12@bandit:/tmp/c1911$ gzip -d data8.gz
bandit12@bandit:/tmp/c1911$ ls
data5.tar  data8  data.tar  data.txt
bandit12@bandit:/tmp/c1911$ cat data8
The password is 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL
```

Level 13 → Level 14

Level Goal

The password for the next level is stored in /etc/bandit_pass/bandit14 and can only be read by user bandit14. For this level, you don't get the next password, but you get a private SSH key that can be used to log into the next level. Note: localhost is a hostname that refers to the machine you are working on

Commands you may need to solve this level

ssh, telnet, nc, openssl, s_client, nmap

Helpful Reading Material

SSH/OpenSSH/Keys

这一关告诉我们下一关的密码存放在/etc目录下，且只有bandit14用户可读，我们当前目录下只有一个私钥文件，可以考虑用私钥文件去连接bandit14, 用bandit14 读取用户文件。

```
bandit13@bandit:~$ ls
sshkey.private
bandit13@bandit:~$ ssh -i sshkey.private bandit14@127.0.0.1
Could not create directory '/home/bandit13/.ssh'.
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes


bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
```

Level 14 → Level 15

Level Goal
The password for the next level can be retrieved by submitting the password of the current level to port 30000 on localhost.

Commands you may need to solve this level
ssh, telnet, nc, openssl, s_client, nmap

Helpful Reading Material
How the Internet works in 5 minutes (YouTube) (Not completely accurate, but good enough for beginners)
IP Addresses
IP Address on Wikipedia
Localhost on Wikipedia
Ports
Port (computer networking) on Wikipedia
这关说只要把本关的密钥提交即可得到反馈，看来我直接从bandit13 ssh连接到的bandit14 可以说并不是算过了这一关，还是要拿到这一关的密钥信息才能进行下一关，这也是这个游戏设计的一个巧妙之处吧。

```
bandit14@bandit:~$ telnet localhost 30000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
Correct!
BfMYroe26WYalil77FoDi9qh59eK5xNr

Connection closed by foreign host.
bandit14@bandit:~$
```

Level 15 → Level 16

Level Goal
The password for the next level can be retrieved by submitting the password of the current level to port 30001 on localhost using SSL encryption.

Helpful note: Getting “HEARTBEATING” and “Read R BLOCK”? Use -ign_eof and read the “CONNECTED COMMANDS” section in the manpage. Next to ‘R’ and ‘Q’, the ‘B’ command also works in this version of that command…

Commands you may need to solve this level
ssh, telnet, nc, openssl, s_client, nmap

Helpful Reading Material

Secure Socket Layer/Transport Layer Security on Wikipedia

OpenSSL Cookbook - Testing with OpenSSL

这题说是要通过ssl发送本关密码才可以的获得下一关的密钥信息。需要用到openssl。

```
bandit15@bandit:~$ openssl s_client -connect localhost  -port 30001
CONNECTED(00000003)
depth=0 CN = localhost
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = localhost
verify return:1
---
Certificate chain
 0 s:/CN=localhost
   i:/CN=localhost
---
Server certificate
-----BEGIN CERTIFICATE-----
MIICBjCCAW+gAwIBAgIESUpi7DANBgkqhkiG9w0BAQUFADAUMRIwEAYDVQQDDAls
b2NhbGhvc3QwHhcNMTgxMjExMTAwMTQyWhcNMTkxMjExMTAwMTQyWjAUMRIwEAYD
VQQDDAlsb2NhbGhvc3QwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMTezWZz
cd9EgMAz0HkacYFj/cRYHpakzE4SPuflAE+rn0rXNihs8Ium69kaQv+EkTAriLAT
qI2FlHT3qP1BsPPn7XGzhGJLHELpKThVJ3dcc7iC8mP5JMEr3Ysd64atu+7EU0iG
+bL56omnhjGWAwR57l/WP2N/ftaxwGVI3SqdAgMBAAGjZTBjMBQGA1UdEQQNMAuC
CWxvY2FsaG9zdDBLBglghkgBhvhCAQ0EPhY8QXV0b21hdGljYWxseSBnZW5lcmF0
ZWQgYnkgTmNhdC4gU2VlIGh0dHBzOi8vbm1hcC5vcmcvbmNhdC8uMA0GCSqGSIb3
DQEBBQUAA4GBAAO24zz8pAGH+VRu/zcztoxyu03edRTe2ofL20DXXkLaMychnux6
1V928fMcG938ErbjVmx6Bq5x0vL/EGL4A1tOa2jmnJcG5vgoeewzlTNcE+s/B2AO
1CKhVi94nLmsRRYpwrrgghN6YtU5akCQjYEeINfjpS4rbYoTUn/x0k1Z
-----END CERTIFICATE-----
subject=/CN=localhost
issuer=/CN=localhost
---
No client certificate CA names sent
Peer signing digest: SHA512
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1019 bytes and written 269 bytes
Verification error: self signed certificate
---
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 1024 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID: F76B1E6D4649F3CE8772262DFA926F6BF02E5DD581FE1AB59421003DA6BBD961
    Session-ID-ctx:
    Master-Key: E14BDEB9B5ACB1BAA7AC3BDA67C819E8125EBEB32E33BE5D14FEAE160B67DCF346A442B4F5C58BF2356248E7E50C51D8
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
    0000 - 64 68 30 37 ad 56 84 7c-c1 99 6e d2 02 05 fa fe   dh07.V.|..n.....
```

```
    0010 - af ec 8d 76 60 4d db 77-2c bd 5e b8 1c 9b 5c a6   ...v`M.w,.^...\.
    0020 - eb 2e 05 8d c7 3d bd bf-1d ae 9c e1 c3 3f 97 7d   .....=.......?.}
    0030 - d1 83 43 ff d1 a9 e9 23-ee b2 6e 6e b1 cb 91 4a   ..C....#..nn...J
    0040 - cf 29 af b3 8a 2a 24 fa-69 87 fa 31 03 11 9a 81   .)...*$.i..1....
    0050 - db 82 c3 5f 58 47 54 53-b1 71 26 5b 96 c2 5c 9f   ..._XGTS.q&[..\.
    0060 - 72 be 5f 55 f4 cd 1f ee-74 76 53 6c fb da f3 e3   r._U....tvSl....
    0070 - aa b4 c1 85 3b a0 64 d6-ef 2e 79 ce 9a 68 46 03   ....;.d...y..hF.
    0080 - bb 91 c3 f8 77 88 f3 44-c0 5f 01 b6 e0 19 d9 09   ....w..D._......
    0090 - ff 76 e9 eb 67 32 04 ee-83 0c b5 41 14 3c b6 7f   .v..g2.....A.<..

    Start Time: 1546574414
    Timeout   : 7200 (sec)
    Verify return code: 18 (self signed certificate)
    Extended master secret: yes
---
BfMYroe26WYalil77FoDi9qh59eK5xNr
Correct!
cluFn7wTiGryunymYOu4RcffSxQluehd

closed
```

Level 16 → Level 17

Level Goal

The credentials for the next level can be retrieved by submitting the password of the current level to a port on localhost in the range 31000 to 32000. First find out which of these ports have a server listening on them. Then find out which of those speak SSL and which don't. There is only 1 server that will give the next credentials, the others will simply send back to you whatever you send to it.

Commands you may need to solve this level

ssh, telnet, nc, openssl, s_client, nmap

Helpful Reading Material

Port scanner on Wikipedia

这一题说开放的端口在31000和32000中间的某一个开放了ssl服务的端口上，肯定要使用到端口扫描程序，这里我们就使用nmap, 扫描一个端口范围，找到我们应该使用的端口号

```
bandit16@bandit:~$ nmap -sV localhost -p 31000-32000

Starting Nmap 7.40 ( https://nmap.org ) at 2019-01-04 05:27 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00022s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE     VERSION
31518/tcp open  ssl/echo
31790/tcp open  ssl/unknown
1 service unrecognized despite returning data. If you know the service/version, please submit the following fing
erprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port31790-TCP:V=7.40%T=SSL%I=7%D=1/4%Time=5C2EE0B3%P=x86_64-pc-linux-gn
SF:u%r(GenericLines,31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20cur
SF:rent\x20password\n")%r(GetRequest,31,"Wrong!\x20Please\x20enter\x20the\
SF:x20correct\x20current\x20password\n")%r(HTTPOptions,31,"Wrong!\x20Pleas
SF:e\x20enter\x20the\x20correct\x20current\x20password\n")%r(RTSPRequest,3
SF:1,"Wrong!\x20Please\x20enter\x20the\x20correct\x20current\x20password\n
SF:")%r(Help,31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20current\x2
SF:0password\n")%r(SSLSessionReq,31,"Wrong!\x20Please\x20enter\x20the\x20c
SF:orrect\x20current\x20password\n")%r(TLSSessionReq,31,"Wrong!\x20Please\
SF:x20enter\x20the\x20correct\x20current\x20password\n")%r(Kerberos,31,"Wr
SF:ong!\x20Please\x20enter\x20the\x20correct\x20current\x20password\n")%r(
SF:FourOhFourRequest,31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20cu
SF:rrent\x20password\n")%r(LPDString,31,"Wrong!\x20Please\x20enter\x20the\
SF:x20correct\x20current\x20password\n")%r(LDAPSearchReq,31,"Wrong!\x20Ple
SF:ase\x20enter\x20the\x20correct\x20current\x20password\n")%r(SIPOptions,
SF:31,"Wrong!\x20Please\x20enter\x20the\x20correct\x20current\x20password\
SF:n");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 88.39 seconds
```

可以看到31518端口和31790端口开放了ssl服务，我们继续连接这个端口发送本关密钥。发现31518端口会将我们发送的内容直接返回，31790才是返回密码的正确端口。

```
bandit16@bandit:~$ openssl s_client -connect localhost -port 31790
CONNECTED(00000003)
depth=0 CN = localhost
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = localhost
verify return:1
---
Certificate chain
 0 s:/CN=localhost
   i:/CN=localhost
---
Server certificate
-----BEGIN CERTIFICATE-----
MIICBjCCAW+gAwIBAgIENT6X8jANBgkqhkiG9w0BAQUFADAUMRIwEAYDVQQDDAls
b2NhbGhvc3QwHhcNMTgxMjExMTAwMTQyWhcNMTkxMjExMTAwMTQyWjAUMRIwEAYD
VQQDDAlsb2NhbGhvc3QwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJuYYSnx
pA49LOi31RUGpW+JNJvStNuBSiMx17bhMuNlijN+b19LuSV1mWOAmo+zzIsBq5Yv
CbvXfCKrjJnxEGuP+XtPmC3trplmej2jlRa/sRmKDIuV74Ze0GjzO25TY6a5XW+J
lC0fqLCH/YsculqmLp8atEYYSaduS5vrz8ILAgMBAAGjZTBjMBQGA1UdEQQNMAuC
CWxvY2FsaG9zdDBLBglghkgBhvhCAQ0EPhY8QXV0b21hdGljYWxseSBnZW5lcmF0
ZWQgYnkgTmNhdC4gU2VlIGh0dHBzOi8vbm1hcC5vcmcvbmNhdC8uMA0GCSqGSIb3
DQEBBQUAA4GBAACqyQVna9ckIFWR3EzUKXl7JgkCN0BK9Wy2rqzgPiplvuhjd4lC
```

```
mIIDeEy7VnZoOEXj+YX4OIhnajkXi6Xf1pBYP+/RPo27n5PgBT3ywJkWwVKWgaPa
BFH7qSw6FBJZkiN5i25FSYXdg4/JT+/C6SZxDy5YGKLFoA3dEGCZ8hh7
-----END CERTIFICATE-----
subject=/CN=localhost
issuer=/CN=localhost
---
No client certificate CA names sent
Peer signing digest: SHA512
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1019 bytes and written 269 bytes
Verification error: self signed certificate
---
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 1024 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID: FC46E2669B162F04B5C370807C1B9E92FCEA3B123059C3E4701A30C1E749B661
    Session-ID-ctx:
    Master-Key: 54B4F5C34BE5CE0F3324991730088949988 1A634B2D13715698130E69E07A1F92B55E3AE86074A7ED3E0DAE14264F3DD
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
    0000 - 2e 71 2f 27 00 eb 72 19-5f 50 c2 d0 8e 8e 6f 16   .q/'..r._P....o.
    0010 - f3 18 94 00 14 11 41 35-a2 b9 c9 d9 d1 a3 87 cb   ......A5........
    0020 - d1 d5 9c 82 4e 31 5a e2-ec 49 a3 1e 37 eb 8d fe   ....N1Z..I..7...
    0030 - c9 ce cc c0 72 26 b8 42-70 86 71 5e 0a d6 35 77   ....r&.Bp.q^..5w
    0040 - 3d 49 07 54 d7 e7 17 d1-b6 20 0d 9c 62 bf 7c db   =I.T..... ..b.|.
    0050 - d3 a5 de bf 89 33 f4 c3-21 b1 88 7b dc 3b e4 11   .....3..!..{.;..
    0060 - fe 0a 43 d0 62 c2 b2 7c-94 62 cf 98 a3 b3 64 5e   ..C.b..|.b....d^
    0070 - 70 c1 9f fe 6d 2f 2d 40-36 6c f6 79 72 cb 30 d0   p...m/-@6l.yr.0.
    0080 - 89 e8 f9 83 5b 7d 65 0d-b2 ed 17 68 ad ae 6b 68   ....[}e....h..kh
    0090 - ad 5f ce 31 7f b3 ec e0-36 c9 f0 e6 de 0c 24 9a   ._.1....6.....$.

    Start Time: 1546576890
    Timeout   : 7200 (sec)
    Verify return code: 18 (self signed certificate)
    Extended master secret: yes
---
cluFn7wTiGryunymYOu4RcffSxQluehd
Correct!
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZl87ORiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzL0VUYbW
JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl
J9nOM8OJ0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asvlpmS8A
vLY9r60wYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
```

```
+TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT
8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd
HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A
R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDi
Ttiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu
L8ktHMPvodBwNsSBULpG0QKBgBAplTfC1HOnWiMGOU3KPwYWt0O6CdTkmJOmL8Ni
blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU
YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM
77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b
dxviW8+TFVEBl1O4f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9GOtt9JPsX8MBTakzh3
vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY=
-----END RSA PRIVATE KEY-----
closed
```

返回的是一段ssh私钥，不难猜想这是下一关连接的私钥信息，先存起来再说，直接在当前目录写发现没有权限，这样我们就需要写道/tmp目录下了

```
bandit16@bandit:~$ mkdir /tmp/ssh_conn
bandit16@bandit:~$ vim /tmp/ssh_conn/rsa.priv
```

用这个私钥去连接第17关。

```

bandit16@bandit:~$ ssh -i /tmp/ssh_conn/rsa.priv bandit17@localhost
Could not create directory '/home/bandit16/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/bandit16/.ssh/known_hosts).
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for '/tmp/ssh_conn/rsa.priv' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/tmp/ssh_conn/rsa.priv": bad permissions
bandit17@localhost's password:

```

提示权限太开放了，把权限改600再试。

```
bandit16@bandit:/tmp/ssh_conn$ ssh -i rsa.priv bandit17@localhost
Could not create directory '/home/bandit16/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes
。
。
。

bandit17@bandit:~$ cat /etc/bandit_pass/bandit17
xLYVMN9WE5zQ5vHacb0sZEVqbrp7nBTn

```

Level 17 → Level 18

Level Goal
There are 2 files in the homedirectory: passwords.old and passwords.new. The password for the next level is in passwords.new and is the only line that has been changed between passwords.old and passwords.new

NOTE: if you have solved this level and see 'Byebye!' when trying to log into bandit18, this is related to the next level, bandit19

Commands you may need to solve this level
cat, grep, ls, diff
diff 比较两个文件的不同, 然后passwd.new不同行行号密码对应的密码为bandit18

```
bandit17@bandit:~$ ls
passwords.new  passwords.old
bandit17@bandit:~$ diff passwords.old passwords.new
42c42
< hlbSBPAWJmL6WFDb06gpTx1pPButblOA
---
> kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
```

得到密钥kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd

Level 18 → Level 19

Level Goal
The password for the next level is stored in a file readme in the homedirectory. Unfortunately, someone has modified .bashrc to log you out when you log in with SSH.

Commands you may need to solve this level
ssh, ls, cat
用上面的密码，一上来就告诉我byebye,然后自动logout了，搞得我一脸懵逼，题目说是.bashrc文件自动登出的。那我们不分配伪终端就可以了，意思是说禁止分配伪终端。当用ssh或telnet等登录系统时，系统分配给我们的终端就是伪终端。如果ssh使用此选项登录系统时，由于禁用，将无法获得终端；但仍能够获得shell，只不过看起来像在本地，也没有很多应有的环境变量，例如命令提示符，PS1等。当使用命令ps -ef|grep [b]ash时看到root 22082 22080 0 11:51 ? 00:00:00 -bash显示终端那里是一个问号。

```
bandit17@bandit:~$ ssh bandit18@localhost -T
.........
..........此处省略粘贴的一大堆东西

  Enjoy your stay!

cat readme
IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x
```

读取readme,得到19关的密码。

Level 19 → Level 20

Level Goal
To gain access to the next level, you should use the setuid binary in the homedirectory. Execute it without arguments to find out how to use it. The password for this level can be found in the usual place (/etc/bandit_pass), after you have used the setuid binary.

Helpful Reading Material
setuid on Wikipedia
先看看家目录下的文件的权限

```
bandit19@bandit:~$ ls -l
total 8
-rwsr-x--- 1 bandit20 bandit19 7296 Oct 16 14:00 bandit20-do
```

属主的权限为rws, s是特殊权限位，允许一般用户用root权限执行这个文件。
通过文件名是想我们用bandit20这个用户执行这个命令读取密码，通过id 命令查看到bandit20用户的uid为11020，运行这个文件–help 命令查看用法可得用法，最后读取密码

```
bandit19@bandit:~$ ./bandit20-do --help
Usage: env [OPTION]... [-] [NAME=VALUE]... [COMMAND [ARG]...]
Set each NAME to VALUE in the environment and run COMMAND.

Mandatory arguments to long options are mandatory for short options too.
  -i, --ignore-environment  start with an empty environment
  -0, --null            end each output line with NUL, not newline
  -u, --unset=NAME      remove variable from the environment
      --help     display this help and exit
      --version  output version information and exit

A mere - implies -i.  If no COMMAND, print the resulting environment.

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/env>
or available locally via: info '(coreutils) env invocation'
bandit19@bandit:~$ ./bandit20-do NAME=11020 cat /etc/bandit_pass/bandit
bandit0   bandit12  bandit16  bandit2   bandit23  bandit27  bandit30  bandit4   bandit8
bandit1   bandit13  bandit17  bandit20  bandit24  bandit28  bandit31  bandit5   bandit9
bandit10  bandit14  bandit18  bandit21  bandit25  bandit29  bandit32  bandit6
bandit11  bandit15  bandit19  bandit22  bandit26  bandit3   bandit33  bandit7
bandit19@bandit:~$ ./bandit20-do NAME=11020 cat /etc/bandit_pass/bandit20
GbKksEFF4yrVs6il55v6gwY5aVje5f0j
```

Level 20 → Level 21

Level Goal
There is a setuid binary in the homedirectory that does the following: it makes a connection to localhost on the port you specify as a commandline argument. It then reads a line of text from the connection and compares it to the password in the previous level (bandit20). If the password is correct, it will transmit the password for the next level (bandit21).

NOTE: Try connecting to your own network daemon to see if it works as you think

Commands you may need to solve this level
ssh, nc, cat, bash, screen, tmux, Unix ‘job control’ (bg, fg, jobs, &, CTRL-Z, …)
`screen命令的说明` https://www.ibm.com/developerworks/cn/linux/l-cn-screen/
tmux是多开终端的命令，job controls经常用就不说了。
这题说是开放一个监听的端口，然后suconnect 文件访问这个端口如果得到和这关相同的密码就会返回下一关的密码，我们就用nc将本关的密码反馈给连接端口命令如下

```
\bandit20@bandit:~$ nc -lv <  /etc/bandit_pass/bandit20 &
[6] 11816
bandit20@bandit:~$ listening on [any] 34957 ...

bandit20@bandit:~$ ./suconnect 34957
connect to [127.0.0.1] from localhost [127.0.0.1] 46028
Read: GbKksEFF4yrVs6il55v6gwY5aVje5f0j
Password matches, sending next password
gE269g2h3mw3pwgrj0Ha9Uoqen1c9DGr
[6]   Done                    nc -lv < /etc/bandit_pass/bandit20
```

成功返回下一关的密码

Level 21 → Level 22

Level Goal
A program is running automatically at regular intervals from cron, the time-based job scheduler. Look in /etc/cron.d/ for the configuration and see what command is being executed.

Commands you may need to solve this level
cron, crontab, crontab(5) (use “man 5 crontab” to access this)
cron介绍可以参考这篇文章 https://www.cnblogs.com/longjshz/p/5779215.html
先按照提示看看当前目录下有什么，可以看到这是一个执行了一个脚本，然后打开这个脚本看看这是一个定时将22关密码写到/tmp目录下的一个脚本，我们读取这个临时文件就知道了下一关的密码。

```
bandit21@bandit:~$ cd /etc/cron.d
bandit21@bandit:/etc/cron.d$ ls
cronjob_bandit22  cronjob_bandit23  cronjob_bandit24
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null

bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv

bandit21@bandit:/etc/cron.d$ cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI
```

Level 22 → Level 23

Level Goal
A program is running automatically at regular intervals from cron, the time-based job scheduler. Look in /etc/cron.d/ for the configuration and see what command is being executed.

NOTE: Looking at shell scripts written by other people is a very useful skill. The script for this level is intentionally made easy to read. If you are having problems understanding what it does, try executing it to see the debug information it prints.

Commands you may need to solve this level
cron, crontab, crontab(5) (use “man 5 crontab” to access this)
先来看看这关所说的定时脚本是什么，如下

```
bandit22@bandit:~$ cd /etc/cron.d
bandit22@bandit:/etc/cron.d$ ls
cronjob_bandit22  cronjob_bandit23  cronjob_bandit24
bandit22@bandit:/etc/cron.d$ cat cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh  &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh  &> /dev/null
bandit22@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
```

实现的功能是取当前用户名，然后计算 I am user $当前用户名 的md5值，将bandit22密码的复制到tmp目录下的对应的md5值的文件中,读取

```
bandit22@bandit:/etc/cron.d$ /bin/bash /usr/bin/cronjob_bandit23.sh
Copying passwordfile /etc/bandit_pass/bandit22 to /tmp/8169b67bd894ddbb4412f91573b38db3
bandit22@bandit:/etc/cron.d$ cat /tmp/8169b67bd894ddbb4412f91573b38db3
Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI
```

读取这个文件，这是本关密码啊，依次类推，I am user bandit23的hash值就是下一关密码。

```
bandit22@bandit:~$ echo I am user bandit23| md5sum
8ca319486bfbbc3663ea0fbe81326349  -

bandit22@bandit:~$ cat /tmp/8ca319486bfbbc3663ea0fbe81326349
jc1udXuA1tiHqjIsL8yaapX5XIAI6i0n
```

Level 23 → Level 24

Level Goal
A program is running automatically at regular intervals from cron, the time-based job scheduler. Look in /etc/cron.d/ for the configuration and see what command is being executed.

NOTE: This level requires you to create your own first shell-script. This is a very big step and you should be proud of yourself when you beat this level!

NOTE 2: Keep in mind that your shell script is removed once executed, so you may want to keep a copy around…

Commands you may need to solve this level
cron, crontab, crontab(5) (use “man 5 crontab” to access this)
老办法，还是先看看这个定时脚本写了什么

```
bandit23@bandit:~$ cat /etc/cron.d/cronjob_bandit24
@reboot bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
* * * * * bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
bandit23@bandit:~$ cat /usr/bin/cronjob_bandit24.sh
#!/bin/bash

myname=$(whoami)

cd /var/spool/$myname
echo "Executing and deleting all scripts in /var/spool/$myname:"
for i in * .*;
do
    if [ "$i" != "." -a "$i" != ".." ];
    then
        echo "Handling $i"
        timeout -s 9 60 ./$i
        rm -f ./$i
    fi
done
```

/var/spool/cron/ 这个目录下存放的是每个用户包括root的crontab任务，每个任务以创建者的名字命名，比如tom建的crontab任务对应的文件就是/var/spool/cron/tom。一般一个用户最多只有一个crontab文件。
我们在/var/spool/bandit24目录下就可以运行bandit24的定时任务
创建一个放在改目录下的脚本就可以执行了
`vim getpass.sh`进入vim编辑模式输入脚本
写入
cat /etc/bandit_pass/bandit24 > /tmp/bandit24pass
：wq保存退出

```
bandit23@bandit:/var/spool/bandit24$ vim getpass
bandit23@bandit:/var/spool/bandit24$ chmod 777 getpass

bandit23@bandit:/var/spool/bandit24$
```

这时候在/var/spool/bandit24目录下不一定能看见你写的脚本，就像前面的定时任务脚本里面写的，执行完脚本这个就任务就删除了，所以没看到也不要奇怪。
这个时候说明我们的脚本已经执行了，可以去/tmp目录查看我们的密码了

```
bandit23@bandit:/var/spool/bandit24$ cat /tmp/bandit24pass
UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ
```

下一关的密钥已经写好了

Level 24 → Level 25

Level Goal
A daemon is listening on port 30002 and will give you the password for bandit25 if given the password for bandit24 and a secret numeric 4-digit pincode. There is no way to retrieve the pincode except by going through all of the 10000 combinations, called brute-forcing.

根据python的pwntools写个脚本跑密码就好了，注意，在其他目录下我们是没有写权限的，这个脚本只能在/tmp目录下创建。如果用的是我下面这种receive line方法，有些破坏输出的结果我要多接收一行过滤掉，

`vim /tmp/conn.py`

创建脚本如下：

```
#!  /usr/bin/python
from pwn import *

conn = remote('localhost', '30002')
badline = conn.recvline()
for i in range(1000):
    tmp = str(i).zfill(4)
    print '[+] Trying pincode: ' + str(tmp)
    conn.sendline('UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ ' + tmp)
    response = conn.recvline()
    print response
    if "Wrong" not in response;
       print "Got Pincode: " + str(tmp)
       response = conn.recvline()
       print response
       exit(0)
```

终端运行`python /tmp/conn.py`

```
[+] Trying pincode: 0377
Wrong! Please enter the correct pincode. Try again.

[+] Trying pincode: 0378
Correct!

Got Pincode: 0378
The password of user bandit25 is uNG9O58gUE7snukf3bvZ0rxhtnjzSGzG
```

输出结果如上所示，前面其他猜解过程就不贴上来了。

Level 25 → Level 26

Level Goal

Logging in to bandit26 from bandit25 should be fairly easy… The shell for user bandit26 is not /bin/bash, but something else. Find out what it is, how it works and how to break out of it.

Commands you may need to solve this level

ssh, cat, more, vi, ls, id, pwd

登录上去可以看到家目录上面有一个bandit26.sshkey, 可以像之前一样用这个私钥文件去连接远程的主机, `ssh -i bandit26.sshkey bandit26@localhost` , 发现连接直接被远程关闭了，加上-T 参数也没有用，题目也提示说这个用的是其他shell, 查看其某用户用的什么shell 可以查看/etc/passwd。

```
bandit25@bandit:~$ cat /etc/passwd|grep bandit26
bandit26:x:11026:11026:bandit level 26:/home/bandit26:/usr/bin/showtext
```

passwd文件的格式为:

账号名称：即登陆时的用户名

密码：早期UNIX系统的密码是放在这个文件中的，但因为这个文件的特性是所有程序都能够读取，所以，这样很容易造成数据被窃取，因此后来就将这个字段的密码数据改放到/etc/shadow中了

UID：用户ID，每个账号名称对应一个UID，通常UID=0表示root管理员

GID：组ID，与/etc/group有关，/etc/group与/etc/passwd差不多，是用来规范用户组信息的

用户信息说明栏： 用来解释这个账号是干什么的

家目录：home目录，即用户登陆以后跳转到的目录，以root用户为例，/root是它的家目录，所以root用户登陆以后就跳转到/root目录这里

Shell：用户使用的shell，通常使用/bin/bash这个shell，这也就是为什么登陆Linux时默认的shell是bash的原因，就是在这里设置的，如果要想更改登陆后使用的shell，可以在这里修改。另外一个很重要的东西是有一个shell可以用来替代让账号无法登陆的命令，那就是/sbin/nologin。

那bandit26用户用到的shell就是/usr/bin/showtext

```
bandit25@bandit:~$ cat /usr/bin/showtext
#!/bin/sh

export TERM=linux

more ~/text.txt
exit 0
```

系统关闭连接的原因是这个exit 0, 在这个exit 之前执行我们想要的命令就可以达到我们想要的效果了。
在more 命令执行之前可以执行命令即可，把会话的终端缩小，然后用文件连接bandit26，这样可以出发自动more, 在more命令还没有结束的时候按v进入vim编辑模式。再就是用vim特有的:e file，vim模式下的e命令可以导入文件到编辑器内，我们知道密码的所在，因此就可以用e命令来导入密码文件

```
: e  /etc/bandit_pass/bandit26
```

然后26关的密钥就被导入到终端可读取了，密钥为
5czgV9L3Xx8JPOyRbXh6lQbmIOWvPT6Z

Level 26 → Level 27

Level Goal
Good job getting a shell! Now hurry and grab the password for bandit27!

Commands you may need to solve this level
ls
这一关使用密码ssh登陆之后也是直接断开了，所以跟上一关套路一样，进入more模式，利用vim模式执行命令，这次不能用e来读取文件了，因为权限不够。!command也不行，!sh也不行，后来查看资料发现vim还有一种需要先设置shell的目录才行
vim模式下

```
:set shell=/bin/sh
:sh
```

然后设置完成上去就可以登录了。ls一下

```
bandit26@bandit:~$ ls
bandit27-do  text.txt
```

有个bandit27-do文件，执行这个文件读取bandit27就可以了。

```
bandit26@bandit:~$ ./bandit27-do cat /etc/bandit_pass/bandit27
3ba3118a22e93127a4ed485be72ef5ea
```

Level 27 → Level 28

Level Goal
There is a git repository at ssh://bandit27-git@localhost/home/bandit27-git/repo. The password for the user bandit27-git is the same as for the user bandit27.

Clone the repository and find the password for the next level.

Commands you may need to solve this level
git
这题是主要是克隆项目的命令，直接在当前目录是新建不了新文件的，所以我们在临时目录下创建目录即可，具体步骤如下，发现这个项目的里面的README就是存储的的密钥

```
bandit27@bandit:~$ git clone ssh://bandit27-git@localhost/home/bandit27-git/repo
fatal: could not create work tree dir 'repo': Permission denied
bandit27@bandit:~$ mkdir /tmp/conn
bandit27@bandit:~$ cd /tmp/conn
bandit27@bandit:/tmp/conn$ git clone ssh://bandit27-git@localhost/home/bandit27-git/repo
Cloning into 'repo'...
Could not create directory '/home/bandit27/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/bandit27/.ssh/known_hosts).
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

bandit27-git@localhost's password:
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
bandit27@bandit:/tmp/conn$ ls
repo
bandit27@bandit:/tmp/conn$ cd repo/
bandit27@bandit:/tmp/conn/repo$ ls
README
bandit27@bandit:/tmp/conn/repo$ cat README
The password to the next level is: 0ef186ac70e04ea33b4c1853d2526fa2
```

Level 28 → Level 29

Level Goal
There is a git repository at ssh://bandit28-git@localhost/home/bandit28-git/repo. The password for the user bandit28-git is the same as for the user bandit28.

Clone the repository and find the password for the next level.

Commands you may need to solve this level
git
克隆项目的过程和之前一样

```
bandit28@bandit:/tmp/conn28/repo$ cat README.md
# Bandit Notes
Some notes for level29 of bandit.

## credentials

- username: bandit29
- password: xxxxxxxxxx
```

题目告诉我们这次的密码是写在某个文件里面了,git log查看提交历史，然后对应版本提交id, 查找区别，得出密码。

```
bandit28@bandit:/tmp/conn28/repo$ git log
commit 073c27c130e6ee407e12faad1dd3848a110c4f95
Author: Morla Porla <morla@overthewire.org>
Date:   Tue Oct 16 14:00:39 2018 +0200

    fix info leak

commit 186a1038cc54d1358d42d468cdc8e3cc28a93fcb
Author: Morla Porla <morla@overthewire.org>
Date:   Tue Oct 16 14:00:39 2018 +0200

    add missing data

commit b67405defc6ef44210c53345fc953e6a21338cc7
Author: Ben Dover <noone@overthewire.org>
Date:   Tue Oct 16 14:00:39 2018 +0200

    initial commit of README.md
bandit28@bandit:/tmp/conn28/repo$ git diff 186a 073c
diff --git a/README.md b/README.md
index 3f7cee8..5c6457b 100644
--- a/README.md
+++ b/README.md
@@ -4,5 +4,5 @@ Some notes for level29 of bandit.
 ## credentials

 - username: bandit29
-- password: bbc96594b4e001778eee9975372716b2
+- password: xxxxxxxxxx
```

Level 29 → Level 30

Level Goal
There is a git repository at ssh://bandit29-git@localhost/home/bandit29-git/repo. The password for the user bandit29-git is the same as for the user bandit29.

Clone the repository and find the password for the next level.

Commands you may need to solve this level
git
git show命令，git log命令还有git diff命令查看git 提交历史，利用git branch -a 命令可以查询分支，发现总共有四个分支。

```
bandit29@bandit:/tmp/conn29/repo$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/dev
  remotes/origin/master
  remotes/origin/sploits-dev
```

git checkout 可以切换分支，当切换到dev查看gitlog 可以发现，最新的版本里面有个data needed for development

```
bandit29@bandit:/tmp/conn29/repo$ git checkout dev
Switched to branch 'dev'
Your branch is up-to-date with 'origin/dev'.
bandit29@bandit:/tmp/conn29/repo$ git log
commit 33ce2e95d9c5d6fb0a40e5ee9a2926903646b4e3
Author: Morla Porla <morla@overthewire.org>
Date:   Tue Oct 16 14:00:41 2018 +0200

    add data needed for development

commit a8af722fccd4206fc3780bd3ede35b2c03886d9b
Author: Ben Dover <noone@overthewire.org>
Date:   Tue Oct 16 14:00:41 2018 +0200

    add gif2ascii

commit 84abedc104bbc0c65cb9eb74eb1d3057753e70f8
Author: Ben Dover <noone@overthewire.org>
Date:   Tue Oct 16 14:00:41 2018 +0200

    fix username

commit 9b19e7d8c1aadf4edcc5b15ba8107329ad6c5650
Author: Ben Dover <noone@overthewire.org>
Date:   Tue Oct 16 14:00:41 2018 +0200

    initial commit of README.md
```

然后在这个版本里面的README发现密码

```
bandit29@bandit:/tmp/conn29/repo$ cat README.md
# Bandit Notes
Some notes for bandit30 of bandit.

## credentials

- username: bandit30
- password: 5b90576bedb2cc04c86a9e924ce42faf
```

Level 30 → Level 31

Level Goal

There is a git repository at ssh://bandit30-git@localhost/home/bandit30-git/repo. The password for the user bandit30-git is the same as for the user bandit30.

Clone the repository and find the password for the next level.

Commands you may need to solve this level
git
git show-ref可以现实本地存储库的所有可用的引用以及关联的提交ID

```
bandit30@bandit:/tmp/conn30/repo$ git show-ref
3aa4c239f729b07deb99a52f125893e162daac9e refs/heads/master
3aa4c239f729b07deb99a52f125893e162daac9e refs/remotes/origin/HEAD
3aa4c239f729b07deb99a52f125893e162daac9e refs/remotes/origin/master
f17132340e8ee6c159e0a4a6bc6f80e1da3b1aea refs/tags/secret
bandit30@bandit:/tmp/conn30/repo$ git show f171
47e603bb428404d265f59c42920d81e5
```

Level 31 → Level 32

Level Goal
There is a git repository at ssh://bandit31-git@localhost/home/bandit31-git/repo. The password for the user bandit31-git is the same as for the user bandit31.

Clone the repository and find the password for the next level.

Commands you may need to solve this level
git
这题是让我们提交到远程仓库

```
bandit31@bandit:/tmp/conn31/repo$ cat README.md
This time your task is to push a file to the remote repository.

Details:
    File name: key.txt
    Content: 'May I come in?'
    Branch: master
```

```
bandit31@bandit:/tmp/conn31/repo$ vim key.txt
bandit31@bandit:/tmp/conn31/repo$ ls
key.txt  README.md
bandit31@bandit:/tmp/conn31/repo$ git add key.txt
The following paths are ignored by one of your .gitignore files:
key.txt
Use -f if you really want to add them.
bandit31@bandit:/tmp/conn31/repo$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean
bandit31@bandit:/tmp/conn31/repo$ git add -f key.txt
bandit31@bandit:/tmp/conn31/repo$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   key.txt
```

```
bandit31@bandit:/tmp/conn31/repo$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   key.txt

bandit31@bandit:/tmp/conn31/repo$ git commit -m 'add key.txt'
[master 7eff4e3] add key.txt
 1 file changed, 1 insertion(+)
 create mode 100644 key.txt
bandit31@bandit:/tmp/conn31/repo$ git push origin master
Could not create directory '/home/bandit31/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/bandit31/.ssh/known_hosts).
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

bandit31-git@localhost's password:
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 324 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote: ### Attempting to validate files... ####
remote:
remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.
remote:
remote: Well done! Here is the password for the next level:
remote: 56a9bf19c63d650ce78e6ec0354ee45e
remote:
remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.
remote:
To ssh://localhost/home/bandit31-git/repo
 ! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'ssh://bandit31-git@localhost/home/bandit31-git/repo'
bandit31@bandit:/tmp/conn31/repo$
```

得到下一关的密钥56a9bf19c63d650ce78e6ec0354ee45e

Level 32 → Level 33

After all this git stuff its time for another escape. Good luck!

Commands you may need to solve this level
sh, man
连接的最后直接给了你一个大写的终端。怎么办呢，我还没思路

```
  For support, questions or comments, contact us through IRC on
  irc.overthewire.org #wargames.

  Enjoy your stay!

WELCOME TO THE UPPERCASE SHELL
>>
```

----------2019.5.13 更新----------

感谢评论区小伙伴提醒$0可以进入正常终端

```
WELCOME TO THE UPPERCASE SHELL
>> $0
$ ls
uppershell
$ whoami
bandit33
$ cat /etc/bandit_pass/bandit33
c9c3199ddf4121b10cf581a98d51caee
```