# 翻网线战争游戏 bandit关卡记录

原创

西溪丸 于 2020-03-21 14:46:25 发布 198 收藏

文章标签： linux shell 安全



本文链接：https://blog.csdn.net/zyhtheonly/article/details/105008508



之前玩的卡关了，尝新于是开始玩了下这个0ver_the-wire这个网站的war-game，但是他貌似是不让写writeup的，但是又想记录下怕以后忘掉，就直译一下叫翻网线战争游戏好了。

bandit难度主要是linux一些常用命令的使用，大多是如何用合适的命令快速定位到符合要求的文件。

## level 1-3

都是Linux最基础的命令，明文密码，找到位置直接cat查看即可。

## level 4

提示密码在唯一可读文件中，因此可以用file命令查看文件类型：

```
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$
```



## level 5

提示密码所在文件要同时符合大小1033b和不可执行，可以用find命令搞定：

```
bandit5@bandit:~/inhere$ find . -size 1033c
./maybehere07/.file2
bandit5@bandit:~/inhere$ ls -l maybehere07/.file2
-rw-r----- 1 root bandit5 1033 Oct 16  2018 maybehere07/.file2
bandit5@bandit:~/inhere$
```

## level 6

提示密码在服务器上某一个文件，满足大小33b和特定用户组，find命令+管道或者find直接搞定：

```
bandit6@bandit:~$ find / -size 33c 2>/dev/null | xargs ls -l | grep bandit7 | gr
ep bandit6
ls: cannot access '/home/bandit2/spaces': No such file or directory
ls: cannot access 'in': No such file or directory
ls: cannot access 'this': No such file or directory
ls: cannot access 'filename': No such file or directory
-rw-r----- 1 bandit7  bandit6  33 Oct 16  2018 /var/lib/dpkg/info/bandit7.passwo
rd
bandit6@bandit:~$ find / -user bandit7 -group bandit6 -size 32c 2>/dev/null
bandit6@bandit:~$ find / -user bandit7 -group bandit6 -size 33c 2>/dev/null
/var/lib/dpkg/info/bandit7.password
```



看了下level7开始可能会有比较tricky的或者会上三剑客命令之类的来解决，今天先到这了改天再玩。。

继续。

## level 7

提示在millions关键字旁边，grep看一下直接搞定了:

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

## level 8

提示在文本中只出现了一次，sort + uniq（如果改成特定次数的话可以用awk写脚本也可以uniq -c | grep x）：

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

## level 9

提示在可读string中，以多个‘=’开头：

```
bandit9@bandit:~$ strings data.txt | grep -E =+
```

## level 10

base64加密，直接解密即可：

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

## level 11

rot13加密，tr解密即可：

```
bandit11@bandit:~$ cat data.txt | tr [N-ZA-Mn-za-m] [A-MN-Za-mn-z]
The password is 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu
```

## level 12

本关开始稍微有些难度了；
提示文件是被多次压缩的hexdump，先按提示在tmp目录下mkdir cp mv，省略图；
然后用file查看文件发现是ASCII text，很尴尬，打开一看是字面意义的hexdump格式。。

```
00000940: 2d2e 2e77 4174 2e2e 5b2e 2e0a 3030 3030  -..wAt..[...0000
00000950: 3032 3330 3a20 3535 3530 2066 3234 3720  0230: 5550 f247
00000960: 3930 3433 2035 3039 3720 6436 3236 2033  9043 5097 d626 3
00000970: 6131 3620 6461 3332 2063 3231 3320 2055  a16 da32 c213  U
00000980: 502e 472e 4350 2e2e 263a 2e2e 322e 2e0a  P.G.CP..&:..2...
00000990: 3030 3030 3032 3430 3a20 3261 6364 2032  00000240: 2acd 2
000009a0: 3938 6120 3563 3861 2066 3063 3120 6239  98a 5c8a f0c1 b9
000009b0: 3966 2065 3265 6520 3438 6137 2030 6131  9f e2ee 48a7 0a1
000009c0: 3220 202a 2e29 2e5c 2e2e 2e2e 2e2e 2e48  2  *.).\.......H
000009d0: 2e2e 2e0a 3030 3030 3032 3530 3a20 3033  ....00000250: 03
000009e0: 6235 2035 6362 3320 3030 3337 2063 6563  b5 5cb3 0037 cec
```



看到提示中有xxd命令，想到这个貌似可以把hexdump转二进制，于是：

```
xxd -r target > real_target
```

再file，可以看到是gzip了:

```
real_target: gzip compressed data, was "data2.bin", last modified: Tue Oct 16
:00:23 2018, max compression, from Unix
```

解压之，当然得先改后缀为gz不然gzip不认：

```
gzip -d real_target.gz
```

再file，看到是bzip2：

```
real_target: bzip2 compressed data, block size = 900k
```

继续解压之，再file，看到又是gzip...（省略图）
重复上面的步骤，终于变成tar了，这可真是老千层饼了。。
tar再解压，解压出了个data5.bin，再一看又是个tar。。。。

```
data5.bin: POSIX tar archive (GNU)
```

继续解压这个data5.bin, 又出来个data6.bin，file一看是bzip2：

```
data6.bin: bzip2 compressed data, block size = 900k
```

继续重复上面步骤，出来了data8.bin，是gzip，再继续解压，这可终于是到头了：

```
data8.bin: ASCII text
```

（幸好cat一看这不是个hexdump，不然我要给作者寄刀片了）

## level 13

提示让你直接ssh过去，照办搞定：

```
$ ssh -i sshkey.private bandit14@localhost
```

进去了顺便可以拿个密码，下次就不用从bandit13 ssh了，今天就先到这了，被千层饼玩坏了。。

继续。

## level 14

提示向localhost 30000端口提交当前密码，telnet搞定；

## level 15

提示ssl上传到30001端口，知识盲区了，先man openssl看一下大概怎么玩，然后面向搜索引擎找到解法：

```
bandit15@bandit:~$ openssl
```

```
OpenSSL> s_client -connect localhost:30001
```

## level 16

提示同上，只不过是在31000-32000中某一端口，需要拿nmap先扫出来：

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

Starting Nmap 7.40 ( https://nmap.org ) at 2020-03-23 08:17 CET
Initiating Ping Scan at 08:17
Scanning localhost (127.0.0.1) [2 ports]
Completed Ping Scan at 08:17, 0.00s elapsed (1 total hosts)
Initiating Connect Scan at 08:17
Scanning localhost (127.0.0.1) [1001 ports]
Discovered open port 31790/tcp on 127.0.0.1
Completed Connect Scan at 08:17, 1.21s elapsed (1001 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00028s latency).
Not shown: 999 closed ports
PORT      STATE    SERVICE
31518/tcp filtered unknown
31790/tcp open     unknown
```



扫出来两个，试下open的那个，返回了一个rsa-key，这是让ssh进下一关了，参见level 13即可；
然后ssh遇到了问题，说文件权限too open:

```
Permissions 0644 for 'key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
```

权限改成400试试，搞定了！

## level 17

提示密码是两个文件中有差异的那行，直接diff即可；

## level 18

用17得到的密码登不进去，打开18一看提示说.bashrc被改了一登录就会被强制登出；
面向搜索引擎找找如何可以不执行bashrc登录，解决方案：

```
bandit17@bandit:~$ ssh bandit18@localhost "bash --norc"
```

没有命令行提示符，但是根据提示密码就在readme里面，于是：

```
ls
readme
cat readme
IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x
```

## level 19

给了个可执行的二进制，让你用他来查看20的密码，看了下就是setuid越权执行：

```
bandit19@bandit:~$ ./bandit20-do
Run a command as another user.
  Example: ./bandit20-do id
bandit19@bandit:~$ cat /etc/bandit_pass/bandit20
cat: /etc/bandit_pass/bandit20: Permission denied
bandit19@bandit:~$ ./bandit20-do /etc/bandit_pass/bandit20
env: '/etc/bandit_pass/bandit20': Permission denied
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
```

## level 20

提示要开一个新端口，用一个新的sh来连接（比较复杂）；
先扫一下不可用端口：

```
bandit20@bandit:~$ nmap -v -p 30000-31000 localhost

Starting Nmap 7.40 ( https://nmap.org ) at 2020-03-27 10:35 CET
Initiating Ping Scan at 10:35
Scanning localhost (127.0.0.1) [2 ports]
Completed Ping Scan at 10:35, 0.00s elapsed (1 total hosts)
Initiating Connect Scan at 10:35
Scanning localhost (127.0.0.1) [1001 ports]
Discovered open port 30001/tcp on 127.0.0.1
Discovered open port 30002/tcp on 127.0.0.1
Discovered open port 30000/tcp on 127.0.0.1
Completed Connect Scan at 10:35, 0.04s elapsed (1001 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00025s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
30000/tcp open  ndmps
30001/tcp open  pago-services1
30002/tcp open  pago-services2
```



用nc命令在30003开个监听：

```
bandit20@bandit:~$ nc -l -p 30003
bandit20@bandit:~$
bandit20@bandit:~$
bandit20@bandit:~$
bandit20@bandit:~$ nc -l -p 30003 < /etc/bandit_pass/bandit20
```

新开一个shell，运行可执行文件，获取密码：

```
bandit20@bandit:~$ ./suconnect 30003
Read: GbKksEFF4yrVs6il55v6gwY5aVje5f0j
Password matches, sending next password
```

## level 21

让在cron配置里看执行了什么，在/etc/cron.d/下找到bandit22相关：

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

看看这个脚本是在干什么，发现他定期把密码写到一个tmp下的文件里：

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

查看这个tmp文件获取密码即可；

## level 22

同21，不同的是要执行一下脚本看下写到tmp的文件名：

```
#!/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
bandit22@bandit:~$ /usr/bin/cronjob_bandit23.sh
Copying passwordfile /etc/bandit_pass/bandit22 to /tmp/8169b67bd894ddbb4412f9157
3b38db3
```



上面执行的时候$myname是bandit22，要手动换成23看下:

```
bandit22@bandit:~$ echo "I am user bandit23" | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbe81326349
```

## level 23

同上，查看定期的脚本做了些什么：

```
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
```



就是定期执行/var/spool/bandit24下的脚本，执行完就删除；
那么我们写一个查看密码的看看能不能放到/var/spool/bandit24下：

```
#!/bin/bash

cat /etc/bandit_pass/bandit24 >> /tmp/tmp.vI2oNTpMYa
~
```

```
t:~$
t:~$ cp /tmp/tmp.vI2oNTpMYa /var/spool/bandit24
t:~$
```

## level 24

提示用brute-force从1-10000里找到一个pin作为password的后缀，连接到30002端口来获取下一级的密码；
先连下30002端口看下：

```
bandit24@bandit:~$ nc localhost 30002
I am the pincode checker for user bandit25. Please enter the password for user b
andit24 and the secret pincode on a single line, separated by a space.
```

爆破：

```
bandit24@bandit:~$ for i in `seq 1 10000`; do real_password=UoMYTrfrBFHyQXmg6gzc
tqAwOmw1IohZ' '$i; echo $real_password | nc localhost 30002 >> /tmp/tmp.frDOx5uO
kO & done
```

查看：

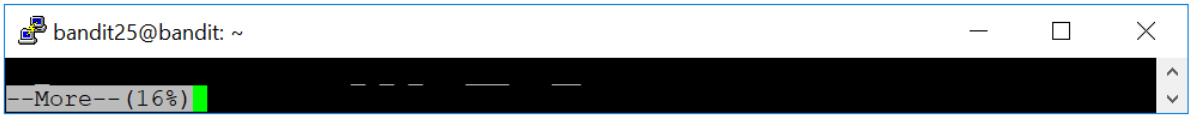
```
$ cat /tmp/tmp.frDOx5uOkO | sort | uniq -u
```

## level 25

提示26不是用bash进去的，也就是他不会运行bashrc；
主目录下有sshkey，先ssh试一下，登录被切断了；
按照提示，查看/etc/passwd，找到bandit27运行的脚本：

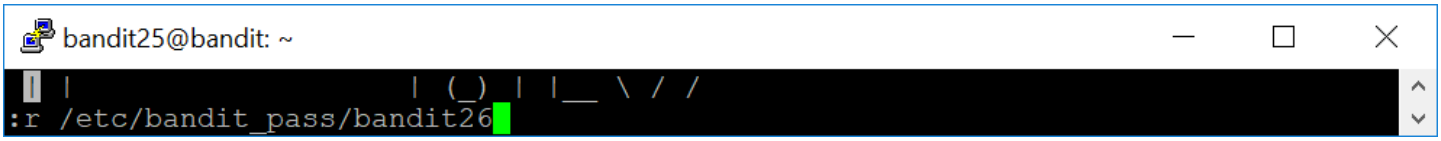
```
bandit25@bandit:~$ cat /usr/bin/showtext
#!/bin/sh

export TERM=linux

more ~/text.txt
exit 0
bandit25@bandit:~$
```

more可以进vim，然后是可以输入命令的，可以把控制台拉的只有一行，让more停住：






敲v进vim，输命令：




然后就可以找到密码了；