OverTheWire Bandit Writeup （11-20）

原创

合天网安实验室 于 2018-08-08 20:00:00 发布 125 收藏 1

昨天小编分享了OverTheWire Bandit Writeup （1-10），今天继续我们没说完的故事.....

> OverTheWire 是一个 wargame 网站。其中 **Bandit** 是一个适合学习Linux指令的游戏，主要是考察一些基本的 Linux 命令行操作 。规则是每一关利用提供的主机加端口和上一关得到的密码通过ssh进入指定的环境，按照要求拿到指定的key，而得到的key又作为下一关的密码。
>
> 网 站：
>
> http://overthewire.org/wargames/bandit/

上回我们已经到了Level 10 ，得到的密码是 `truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk`让我们继续，这部分的难度是中等。

Level 10 →11

描述：下一关的密码存储在 **`data.txt`**文件中，并且包含 **`base64`**编码数据

```
bandit10@bandit:~$ cat ./data.txt
VGhlIHBhc3N3b3JkIGlzIElGdWt3S0dzRlc4TU9xM0lSRnFyeEUxaHhUTkViVVBSCg==
```

得到一串base64加密的数据，这里需要我们解密，使用系统自带的 **`base64`**命令即可

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

Level 11 →12

描述：密码存储在 **`data.txt`**文件中，但是里面的英文字母字符（A-Z/a-z）都被旋转了13位
这里涉及到一个非常古老的置换密码算法 ROT13，简单来说就是把原字母用它13位之后对应的字母代替，超过时则重新绕回26英文字母开头。A换成N、B换成O、依此类推到M换成Z，然后序列反转：N换成A、O换成B、最后Z换成M。

在linux中，使用 **`tr`**命令即可完成

```
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 →13

描述：密码存储在 **data.txt**文件中，是一个通过hexdump转换过的经过多重压缩过的二进制文件数据，也就是是一个16进制文件。这题主要考察的是linux下各种压缩文件命令的用法

提示可能用到的命令有：“grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd, mkdir, cp, mv“

并且提示我们可以在/tmp下新建一个目录，把文件复制过去进行操作，可能是要操作步骤有点多吧~

第一步，先看看文件长什么样子

```
bandit12@bandit:~$ cat data.txt
00000000: 1f8b 0808 ecf2 445a 0203 6461 7461 322e  ......DZ..data2.
00000010: 6269 6e00 0149 02b6 fd42 5a68 3931 4159  bin..I...BZh91AY
```

第二步，复制一份到//tmp/level13/(这名字随便你自己取了)

```
bandit12@bandit:~$ mkdir /tmp/level13
bandit12@bandit:~$ cp data.txt /tmp/level13/
bandit12@bandit:~$ cd /tmp/level13/
```

第三步，把16进制文件转回二进制文件，用 **xxd**命令

```
bandit12@bandit:/tmp/level13$ xxd -r data.txt data
```

第三步，使用 **file**命令确定文件的类型，然后使用相应的命令解压文件

```
bandit12@bandit:/tmp/level13$ file data
data: gzip compressed data, was "data2.bin", last modified: Thu Dec 28 13:34:36 2017, max compression, from
bandit12@bandit:/tmp/level13$ mv data data.gz
andit12@bandit:/tmp/level13$ gzip -d data.gz
gzip: data.gz: decompression OK, trailing garbage ignored
```

第四步，重复上一步

```
bandit12@bandit:/tmp/level13$ file data
data: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/level13$ bzip2 -d data
bzip2: Can't guess original name for data -- using data.out
bandit12@bandit:/tmp/level13$ file data.out
data.out: gzip compressed data, was "data4.bin", last modified: Thu Dec 28 13:34:36 2017, max compression,
bandit12@bandit:/tmp/level13$ zcat data.out > data2
bandit12@bandit:/tmp/level13$ file data2
data2: POSIX tar archive (GNU)
bandit12@bandit:/tmp/level13$ tar -xvf data2
data5.bin
bandit12@bandit:/tmp/level13$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/level13$ tar -xvf data5.bin
data6.bin
bandit12@bandit:/tmp/level13$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/level13$ bzip2 -d data6.bin
bzip2: Can't guess original name for data6.bin -- using data6.bin.out
bandit12@bandit:/tmp/level13$ file data6.bin.out
data6.bin.out: POSIX tar archive (GNU)
bandit12@bandit:/tmp/level13$ tar -xvf data6.bin.out
data8.bin
bandit12@bandit:/tmp/level13$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: Thu Dec 28 13:34:36 2017, max compression,
bandit12@bandit:/tmp/level13$ zcat data8.bin > data9.bin
bandit12@bandit:/tmp/level13$ file data9.bin
data9.bin: ASCII text
bandit12@bandit:/tmp/level13$ cat data9.bin
The password is 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL
```

真是折腾啊~终于得到密码了

Level 13 →14

描述：进入下一关的密码存储在 **/etc/bandit_pass/bandit14**中，但是这个文件只有用户 **bandit14** 才能读取，在这一关，没有密码，但你可以通过ssh私钥登录到bandit14获得可以进入下一关的密码。

可能用到的命令：ssh, telnet, nc, openssl, s_client, nmap

从这一关开始考察linux网络管理方面的命令，比如ssh远程登录等

来看看有没有私钥，然后登录到 bandit14吧

```
bandit13@bandit:~$ ls -l
total 4
-rw-r----- 1 bandit14 bandit13 1679 Dec 28  2017 sshkey.private
bandit13@bandit:~$ ssh -i sshkey.private bandit14@localhost
```

登录成功，查看进入下一关的密码

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

Level 14→15

描述：将当前的密码提交到 localhot 的30000端口，就能获得下一关的密码

这很简单了，用telnet 登录 localhost 的30000端口，然后提交当前密码就行

```
bandit14@bandit:~$ telnet localhost 30000
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
Correct!
BfMYroe26WYalil77FoDi9qh59eK5xNr
Connection closed by foreign host.
```

Level 15→16

描述：通过ssl加密传输当前密码，然后提交到 localhost 的30001端口就能获得下一关的密码

和前一关差不多啦，只是多了一个ssl加密传输，而且不能用telnet了，因为telnet是明文传输啊~

这里我们使用openssl 的sclient命令，sclient是一个SSL/TLS客户端程序，与sserver对应，它不仅能与sserver进行通信，也能与任何使用ssl协议的其他服务程序进行通信。

```
bandit15@bandit:~$ openssl s_client -connect localhost:30001 -ign_eof
```

-ign_eof：当输入文件到达文件尾的时候并不断开连接。

然后提交当前密码，得到进入下一关的密码

```
    Verify return code: 18 (self signed certificate)
---
BfMYroe26WYalil77FoDi9qh59eK5xNr
Correct!
cluFn7wTiGryunymYOu4RcffSxQluehd
closed
bandit15@bandit:~$
```

Level 16→17

描述：将当前密码提交到 localhost 的 31000端口到 32000端口其中的一个端口，得到进入下一关的凭证。但只有其中一个端口开启了监听服务，并且需要通过ssl加密传输。

这一关看起来有点麻烦，难道一个个端口去尝试？这不符合我们的风格啊。

是时候祭出 **nmap**这个神器了。

先进行端口服务识别吧

```
bandit16@bandit:~$ nmap -A localhost -p31000-32000
......
31790/tcp open  ssl/unknown
| ssl-cert: Subject: commonName=bandit
.......
```

我们发现 31790 开启了监听，而且是ssl服务

给它密码吧

```
bandit16@bandit:~$ openssl s_client -connect localhost:31790
......
cluFn7wTiGryunymYOu4RcffSxQluehd
Correct!
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon
......
```

返回了一段RSA私钥，这个就是进入下一关的凭证，我们把它复制下来，保存为 **sshkey.private**文件

```
bandit16@bandit:/tmp$ mkdir /tmp/bandit16
bandit16@bandit:/tmp$ cd /tmp/bandit16
bandit16@bandit:/tmp/bandit16$ vim sshkey.private
```

然后使用私钥登录 bandit17

```
bandit16@bandit:/tmp/bandit16$ chmod 600 sshkey.private
bandit16@bandit:/tmp/bandit16$ ssh -i sshkey.private bandit17@localhost
```

Tips：这里必须要改私钥的权限，不然不让你登录的

Level 17→18

描述：有两个文件，分别是passwords.old 和 passwords.new，进入下一关的密码在 passwords.new 中，而且是 passwords.old中唯一被更改的一行。如果你已经解决了这个级别并且在尝试登录bandit18时看到'Byebye！'，这与下一级别有关。

可能用到的命令：cat, grep, ls, diff

考察点又回到了文件操作了？

很简单了，用 **diff**命令就可以了

diff是Unix系统的一个很重要的工具程序。它用来比较两个文本文件的差异

```
bandit17@bandit:~$ diff passwords.new passwords.old
42c42
< kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
---
> 6vcSC74ROI95NqkKaeEC2ABVMDX9TyUr
```

kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd 即为被修改了那行

```
bandit17@bandit:~$ ssh bandit18@localhost
......
Byebye !
Connection to localhost closed.
......
```

还真是诚不欺我啊，提示了Byebye ！接着看下一关怎么说

Level 18→19

描述：密码存储在家目录的 **readme**文件中，但是，但是，当使用SSH登录时，有人修改了".bashrc" 文件，导致你退出了。就是上一关提示的出现 Byebye ！。谁这么坑~~

~/.bashrc：该文件包含专用于你的bash shell的bash信息,当登录时以及每次打开新的shell时,该文件被读取.

估计是因为没有打开bash ，登录后没法为远程登录分配伪终端，所以导致退出了。

那要怎么登录呢？我们查找 **ssh**命令的帮助，找了 **-T**这个参数，这个参数的意思是"禁止分配伪终端"，意思就是不需要远程主机分配伪终端，来试试吧

```
bandit17@bandit:~$ ssh -T bandit18@localhost
id
uid=11018(bandit18) gid=11018(bandit18) groups=11018(bandit18)
```

成功登录了

```
ls
cat readme
IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x
```

Level 19→20

描述：要访问下一关，你必须使用家目录下的setuid 可执行程序，在使用setuid 文件后，可以在 / etc / bandit_pass /中找到密码。

这一关考察的是linux文件权限的知识。如果一个二进制可执行程序拥有 SUID权限，那么其他用户执行这个程序的时候就拥有和文件所有者一样的权限。

```
bandit19@bandit:~$ ls -l
total 8
-rwsr-x--- 1 bandit20 bandit19 7408 Dec 28  2017 bandit20-do
```

这个"bandit20-do"的权限是"rwsr-x---"，说明它就是那个拥有SUID权限的程序。

我们再看看" /etc/bandit_pass/bandit20" 文件的权限

```
bandit19@bandit:~$ cat /etc/bandit_pass/bandit20
cat: /etc/bandit_pass/bandit20: Permission denied
bandit19@bandit:~$ ls -l /etc/bandit_pass/bandit20
-r-------- 1 bandit20 bandit20 33 Dec 28  2017 /etc/bandit_pass/bandit20
```

发现只有“bandit20”用户可以读取，所以我们要借助“bandit20-do” 去调用“cat”命令查看文件内容

```
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
GbKksEFF4yrVs6il55v6gwY5aVje5f0j
-------------------
login bandit20
bandit19@bandit:~$ ssh bandit20@localhost
```

未完待续.....

长按二维码　识别加关注