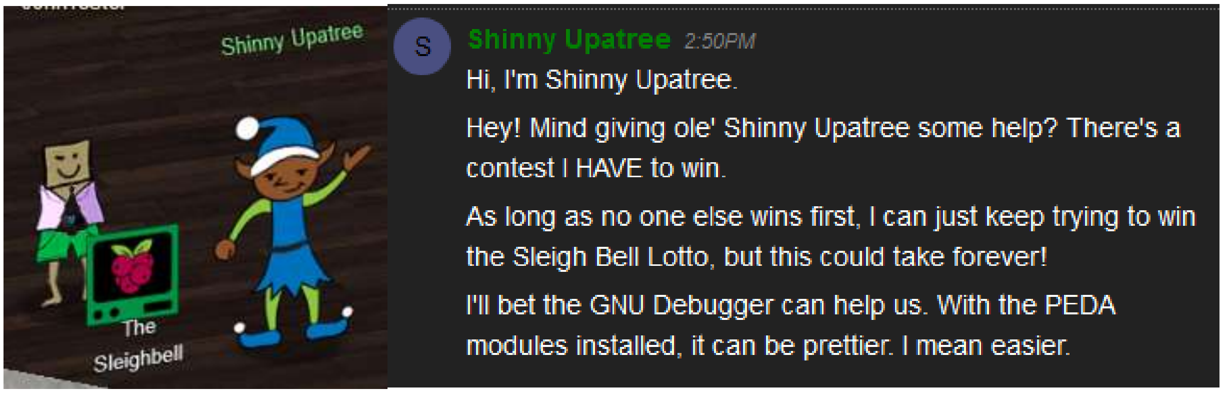
Terminal--The Sleigh bell

This terminal followed the method in the hint almost completely. because it is so simple, this lesson will just be a walk through. Feel free to do it on your own with only Shinny's hint to help you.

Getting Started

Shinny Upatree and The Sleighbell terminal are on the right side of the second floor, near the stairs.



S **Shinny Upatree** *2:50PM*

Hi, I'm Shinny Upatree.

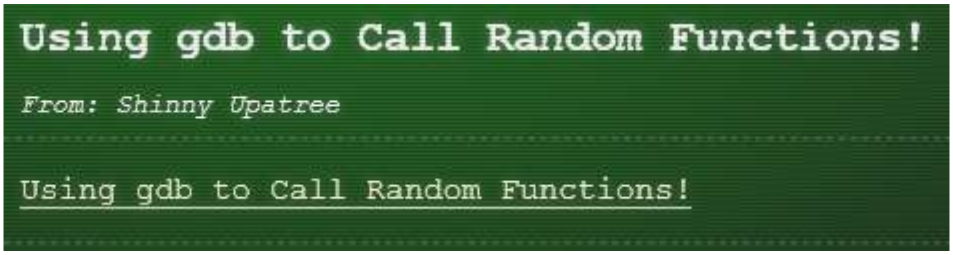
Hey! Mind giving ole' Shinny Upatree some help? There's a contest I HAVE to win.

As long as no one else wins first, I can just keep trying to win the Sleigh Bell Lotto, but this could take forever!

I'll bet the GNU Debugger can help us. With the PEDA modules installed, it can be prettier. I mean easier.

Hint

Shinny gives you the following link which connects to a SANS Pentest blog. The link to the blog is https://pen-testing.sans.org/blog/2018/12/11/using-gdb-to-call-random-functions.



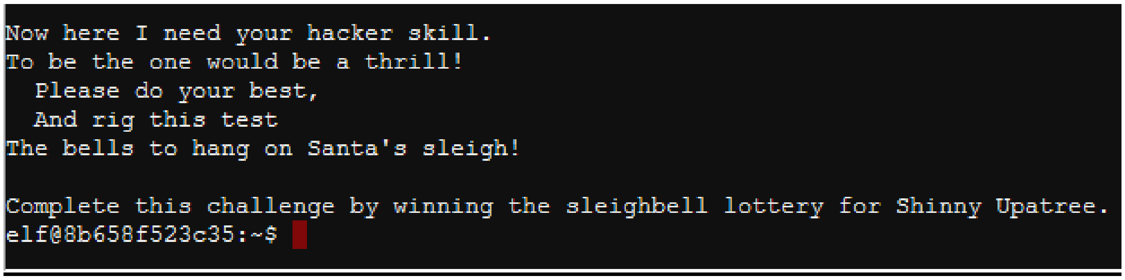
Using gdb to Call Random Functions!

From: Shinny Upatree

Using gdb to Call Random Functions!

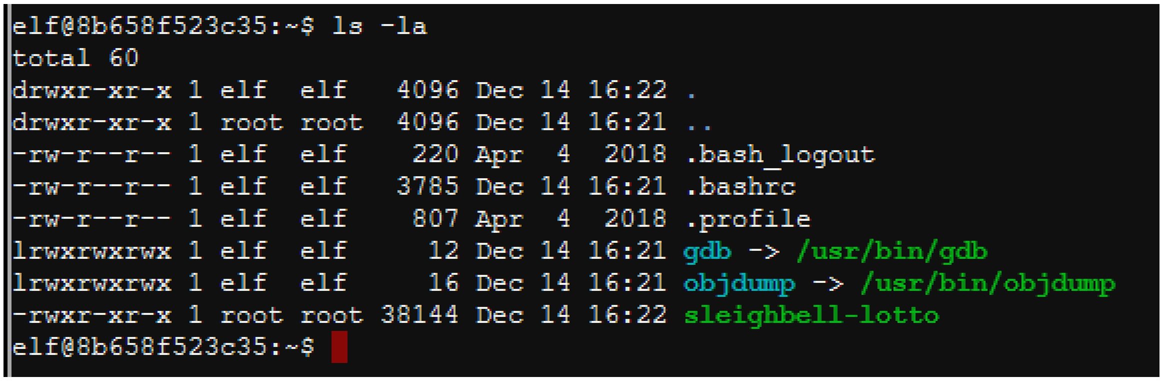
Solution

Here's the terminal.



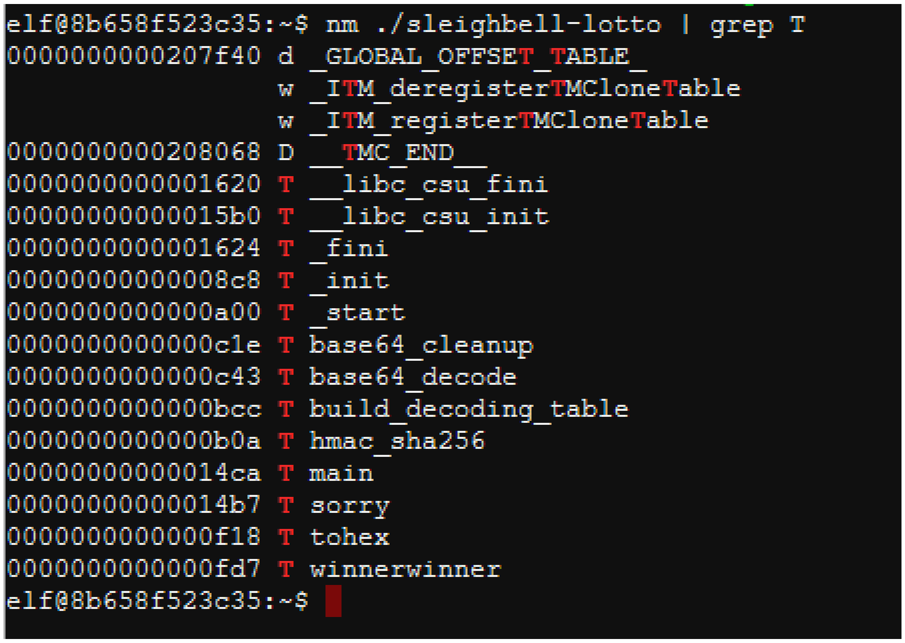
```
Now here I need your hacker skill.
To be the one would be a thrill!
  Please do your best,
  And rig this test
The bells to hang on Santa's sleigh!

Complete this challenge by winning the sleighbell lottery for Shinny Upatree.
elf@8b658f523c35:~$
```

It is always good to look around. It is nice that they left us a link to gdb.



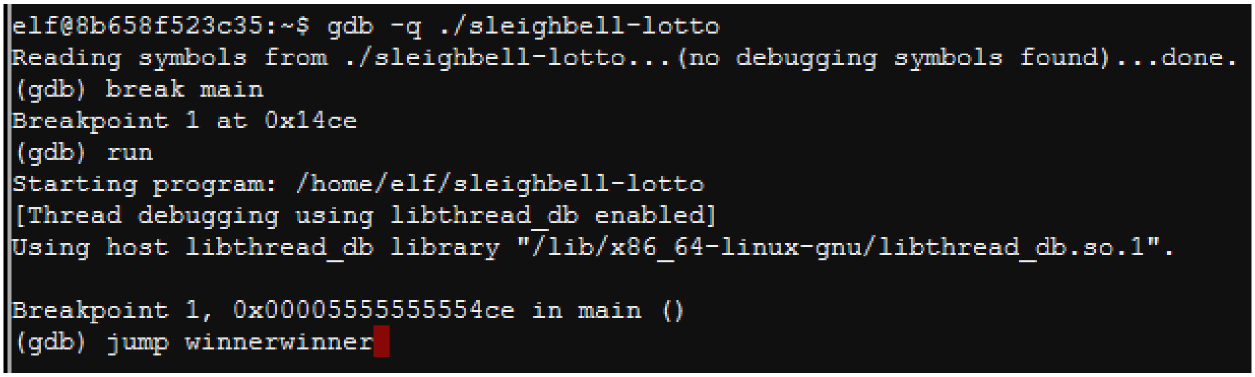
```
elf@8b658f523c35:~$ ls -la
total 60
drwxr-xr-x 1 elf  elf   4096 Dec 14 16:22 .
drwxr-xr-x 1 root root  4096 Dec 14 16:21 ..
-rw-r--r-- 1 elf  elf    220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 elf  elf   3785 Dec 14 16:21 .bashrc
-rw-r--r-- 1 elf  elf    807 Apr  4  2018 .profile
lrwxrwxrwx 1 elf  elf     12 Dec 14 16:21 gdb -> /usr/bin/gdb
lrwxrwxrwx 1 elf  elf     16 Dec 14 16:21 objdump -> /usr/bin/objdump
-rwxr-xr-x 1 root root 38144 Dec 14 16:22 sleighbell-lotto
elf@8b658f523c35:~$
```

From the hint, first run `nm`. It had a lot of output, so piping to `grep T` made it cleaner. Perhaps the function `winnerwinner` is what we want…



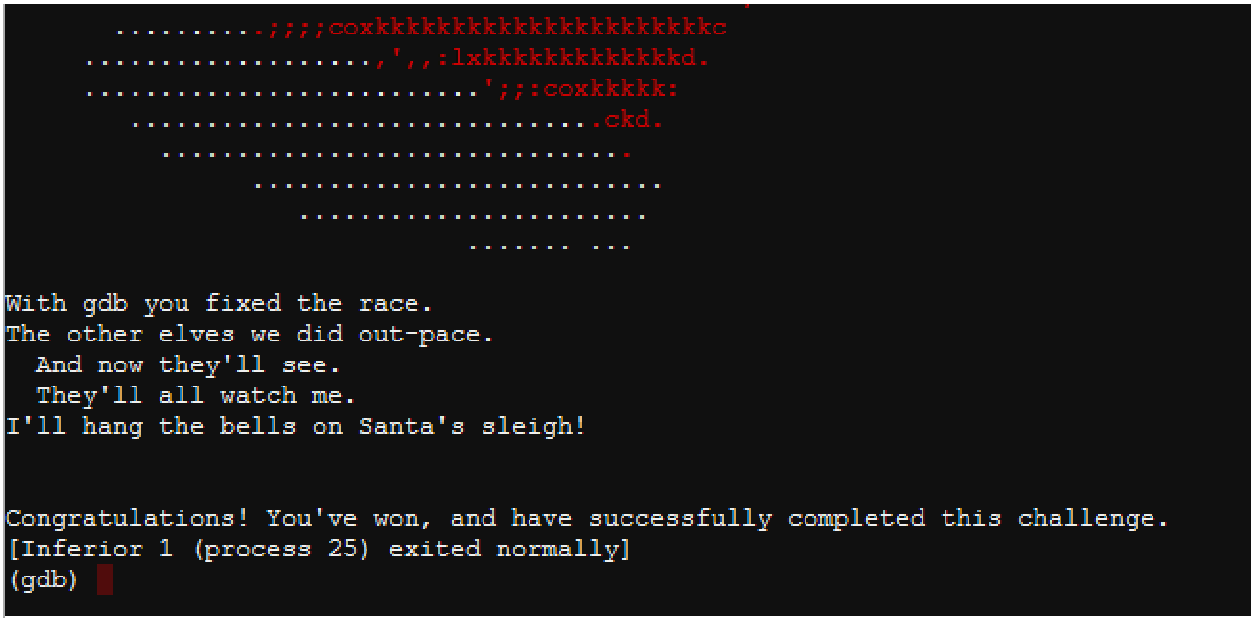
```
elf@8b658f523c35:~$ nm ./sleighbell-lotto | grep T
0000000000207f40 d _GLOBAL_OFFSET_TABLE_
                 w _ITM_deregisterTMCloneTable
                 w _ITM_registerTMCloneTable
0000000000208068 D __TMC_END__
0000000000001620 T __libc_csu_fini
00000000000015b0 T __libc_csu_init
0000000000001624 T _fini
00000000000008c8 T _init
0000000000000a00 T _start
0000000000000c1e T base64_cleanup
0000000000000c43 T base64_decode
0000000000000bcc T build_decoding_table
0000000000000b0a T hmac_sha256
00000000000014ca T main
00000000000014b7 T sorry
0000000000000f18 T tohex
0000000000000fd7 T winnerwinner
elf@8b658f523c35:~$
```

The next step in the blog is to run `gdb` on the target file, `sleighbell-lotto` in this case. Then set a break point and run the program. Finally, jump to `winnerwinner`.



```
elf@8b658f523c35:~$ gdb -q ./sleighbell-lotto
Reading symbols from ./sleighbell-lotto...(no debugging symbols found)...done.
(gdb) break main
Breakpoint 1 at 0x14ce
(gdb) run
Starting program: /home/elf/sleighbell-lotto
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, 0x00005555555554ce in main ()
(gdb) jump winnerwinner
```

And that’s it!



```
       ..........;;;;coxkkkkkkkkkkkkkkkkkkkkkkkc
     ...................,',,:lxkkkkkkkkkkkkkd.
     ..........................';;:coxkkkk:
        ...............................ckd.
          ...............................
                ...........................
                   .......................
                              ....... ...

With gdb you fixed the race.
The other elves we did out-pace.
  And now they'll see.
  They'll all watch me.
I'll hang the bells on Santa's sleigh!


Congratulations! You've won, and have successfully completed this challenge.
[Inferior 1 (process 25) exited normally]
(gdb)
```