

```
root@0x4261622792058:~/reverseCtf/loginKey/chal# ls -l
total 20
-rwxr-xr-x 1 root root 18528 Sep 10 17:49 CTFChallenge
root@0x4261622792058:~/reverseCtf/loginKey/chal# ./CTFChallenge

*****
*           Welcome to 0xDoc Sharing Server           *
*****

0x999 ~ Please enter in your Login Key: 0x123456789
Login failed!
root@0x4261622792058:~/reverseCtf/loginKey/chal#
```

Verify file in directory

Testing file

Then open file in GDB

```
root@0x4261622792058:~/reverseCtf/loginKey/chal# gdb ./CTFChallenge
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./CTFChallenge...
(gdb)
```

After opening the file in GDB, execute the command layout next. This command sets up the GDB interface to display the assembly output, registers.

Then Type run into the new window to run the program while gdb in analyzing it

```
0x555555551c9 <main>          endbr64
0x555555551cd <main+4>        push    %rbp
0x555555551ce <main+5>        mov     %rsp,%rbp
0x555555551d1 <main+8>        sub     $0x50,%rsp
0x555555551d5 <main+12>       mov     %fs:0x28,%rax
0x555555551de <main+21>       mov     %rax,-0x8(%rbp)
0x555555551e2 <main+25>       xor     %eax,%eax
0x555555551e4 <main+27>       lea     0xe1d(%rip),%rax      # 0x555555556008
0x555555551eb <main+34>       mov     %rax,%rdi
0x555555551ee <main+37>       mov     $0x0,%eax
0x555555551f3 <main+42>       call   0x5555555550b0 <printf@plt>
0x555555551f8 <main+47>       lea     0xe41(%rip),%rax      # 0x555555556040
0x555555551ff <main+54>       mov     %rax,%rdi
0x55555555202 <main+57>       mov     $0x0,%eax
0x55555555207 <main+62>       call   0x5555555550b0 <printf@plt>
0x5555555520c <main+67>       lea     0xe65(%rip),%rax      # 0x555555556078
0x55555555213 <main+74>       mov     %rax,%rdi
0x55555555216 <main+77>       call   0x555555555090 <puts@plt>
0x5555555521b <main+82>       lea     0xe8e(%rip),%rax      # 0x5555555560b0
0x55555555222 <main+89>       mov     %rax,%rdi
0x55555555225 <main+92>       mov     $0x0,%eax
0x5555555522a <main+97>       call   0x5555555550b0 <printf@plt>
0x5555555522f <main+102>      lea     -0x40(%rbp),%rax
0x55555555233 <main+106>      mov     %rax,%rsi

exec No process In:
(gdb) run
Starting program: /root/reverseCtf/loginKey/chal/CTFChallenge
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

*****
*           Welcome to 0xDoc Sharing Server           *
*****

0x999 ~ Please enter in your Login Key:
```

Enter in random input

Notice that something was lea from a external module named users.

```
0x55555555275 <main+172>    shl     $0x2,%rax
0x55555555279 <main+176>    lea     0x2da0(%rip),%rdx      # 0x555555558020 <users>
0x55555555280 <main+183>    add     %rax,%rdx
0x55555555283 <main+186>    lea     -0x40(%rbp),%rax
0x55555555287 <main+190>    mov     %rdx,%rsi
0x5555555528a <main+193>    mov     %rax,%rdi
0x5555555528d <main+196>    call    0x555555550c0 <strcmp@plt>
0x55555555292 <main+201>    test    %eax,%eax
0x55555555294 <main+203>    jne     0x5555555529f <main+214>
0x55555555296 <main+205>    movl    $0x1,-0x48(%rbp)
0x5555555529d <main+212>    jmp     0x555555552ae <main+229>
0x5555555529f <main+214>    addl    $0x1,-0x44(%rbp)
0x555555552a3 <main+218>    mov     0x2ddb(%rip),%eax      # 0x555555558084 <num_users>
0x555555552a9 <main+224>    cmp     %eax,-0x44(%rbp)
0x555555552ac <main+227>    jl      0x5555555525a <main+145>
0x555555552ae <main+229>    cmpl    $0x0,-0x48(%rbp)

exec No process In:
multi-thre No process In:
Starting program: /root/reverseCtf/loginKey/chal/CTFChallenge
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

*****
(gdb) 2 *      Welcome to 0xDoc Sharing Server      *
Undefined command: "2".  Try "help".*****
Undefined command: "2".  Try "help".
(gdb)
```

Using the memory address in hex 0x555555558020 from module users

You can display what is in that memory address by using command : `x/s 0x555555558020`

```
(gdb) x/s 0x555555558020
0x555555558020 <users>: "0x426162792058"
(gdb)
```

Now you have the login key which is my discord user name...

****Now you can copy and paste this into the code running in a netcat server to get the flag**