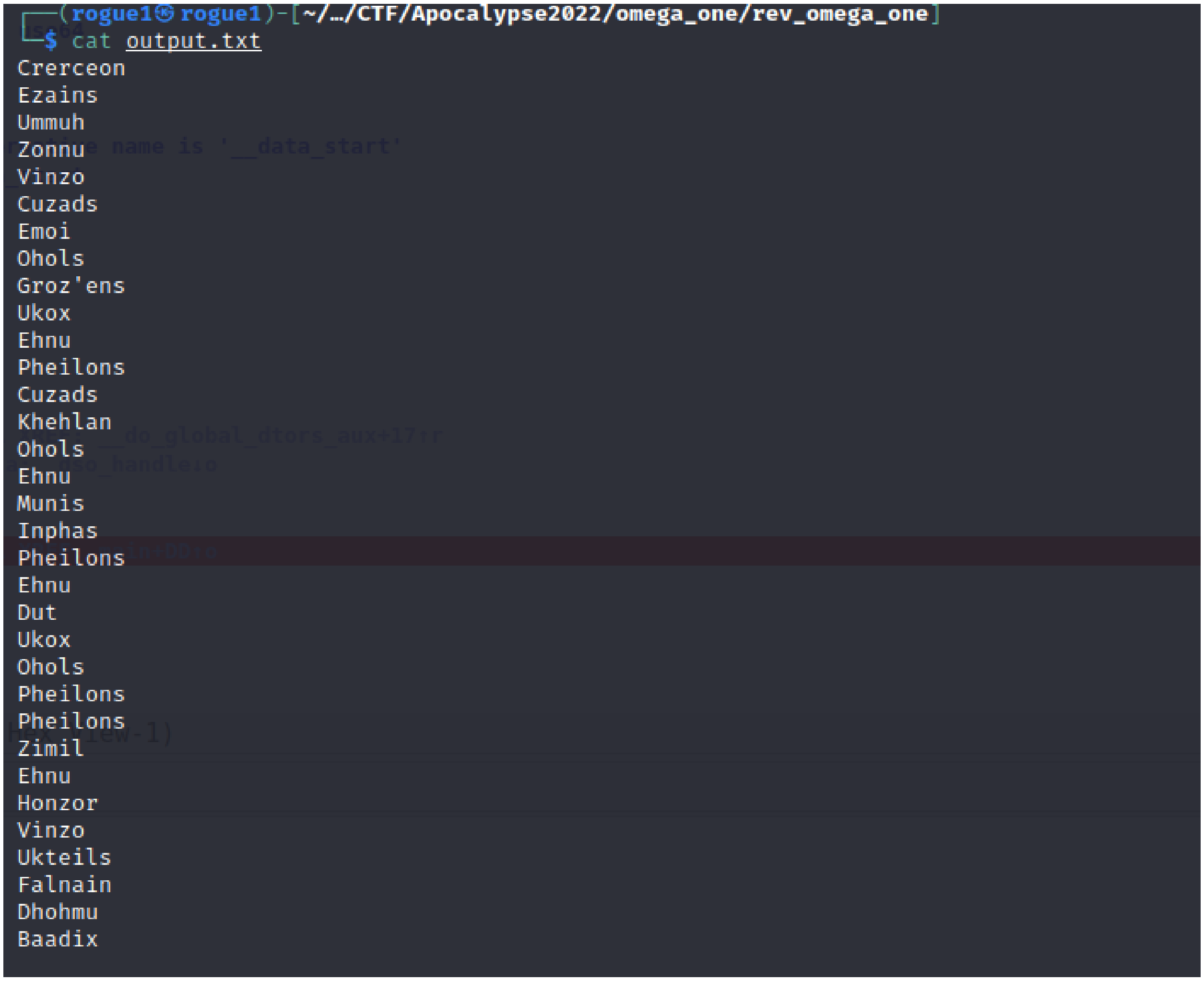
HTB Reverse Engineering: Omega One

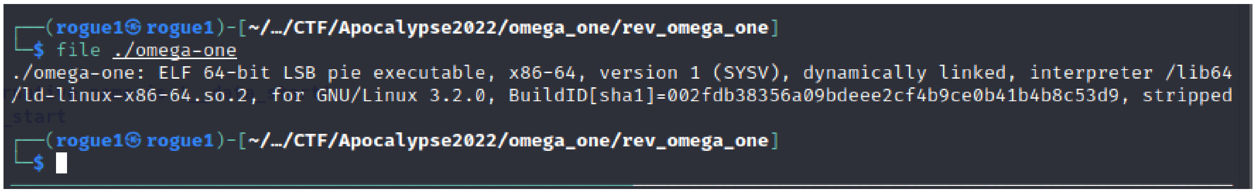
Tools: IDA

Given 2 files from a zip
./Omega-one and output.txt



```
┌──(rogue1㉿rogue1)-[~/…/CTF/Apocalypse2022/omega_one/rev_omega_one]
└─$ cat output.txt
Crerceon
Ezains
Ummuh
Zonnu
Vinzo
Cuzads
Emoi
Ohols
Groz'ens
Ukox
Ehnu
Pheilons
Cuzads
Khehlan
Ohols
Ehnu
Munis
Inphas
Pheilons
Ehnu
Dut
Ukox
Ohols
Pheilons
Pheilons
Zimil
Ehnu
Honzor
Vinzo
Ukteils
Falnain
Dhohmu
Baadix
```

1. ./omega-one does not run
running $file ./omega shows this output.



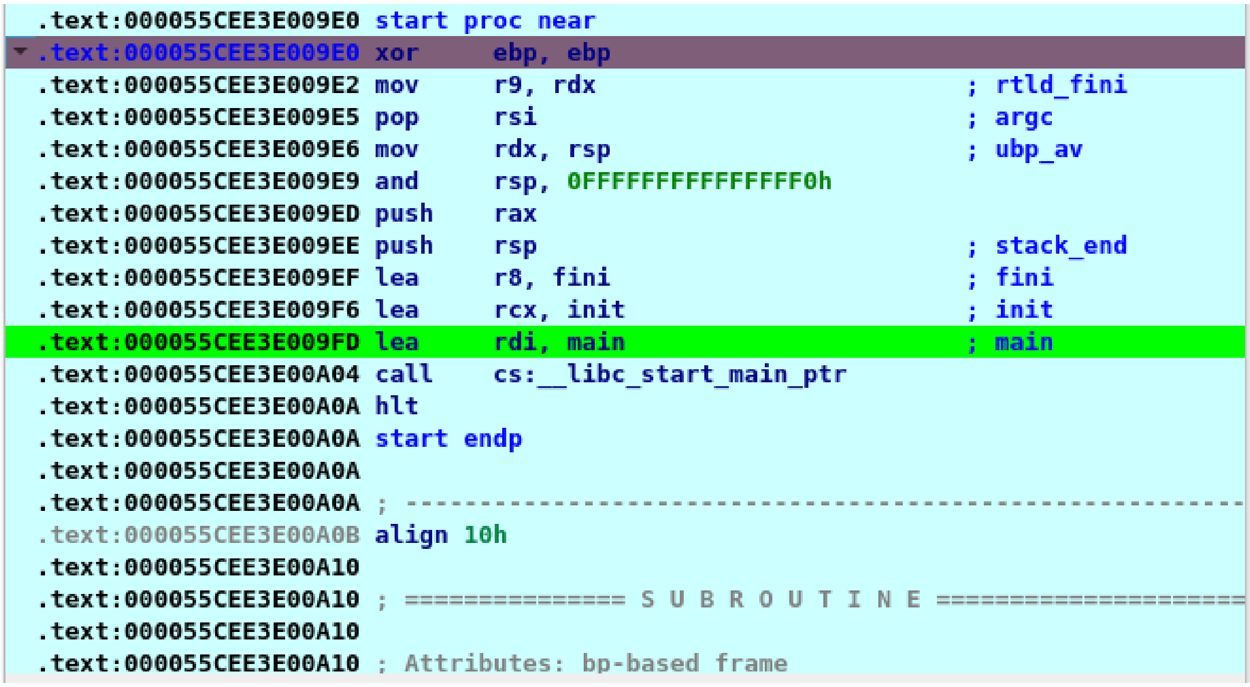
```
┌──(rogue1㉿rogue1)-[~/…/CTF/Apocalypse2022/omega_one/rev_omega_one]
└─$ file ./omega-one
./omega-one: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64
/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=002fdb38356a09bdeee2cf4b9ce0b41b4b8c53d9, stripped

┌──(rogue1㉿rogue1)-[~/…/CTF/Apocalypse2022/omega_one/rev_omega_one]
└─$ 
```

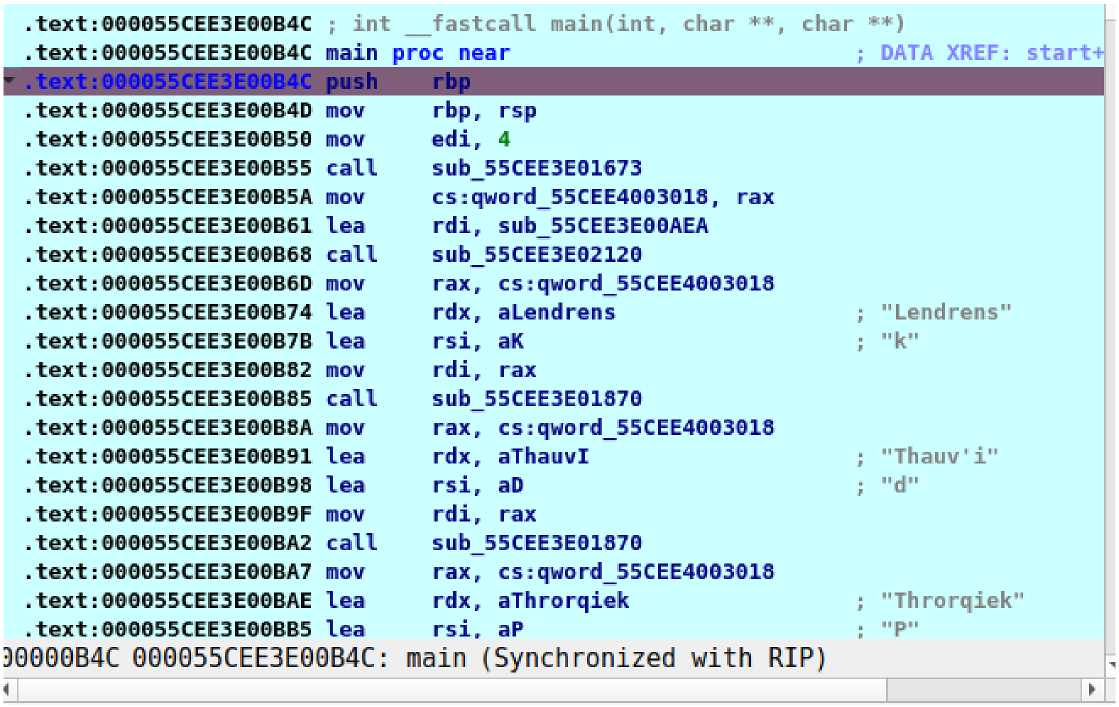
We can open the file in IDA or Ghidra.

In IDA navigating to exports we can see the start function which is also the entry point of the file. Just a few lines down we see Call to the libc_start_main_ptr. This is what we need.

I set Breakpoints to navigate the debugger and it will take me directly to the call file libc_start_main_ptr

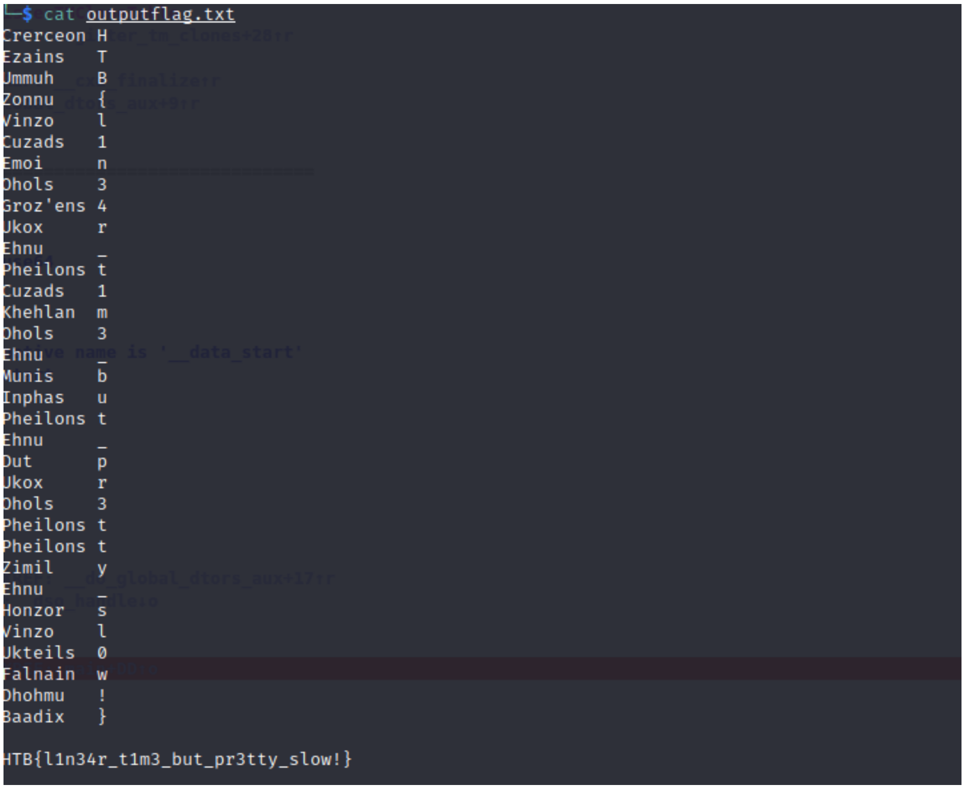


```
.text:000055CEE3E009E0 start proc near
.text:000055CEE3E009E0 xor     ebp, ebp
.text:000055CEE3E009E2 mov     r9, rdx                          ; rtld_fini
.text:000055CEE3E009E5 pop     rsi                              ; argc
.text:000055CEE3E009E6 mov     rdx, rsp                         ; ubp_av
.text:000055CEE3E009E9 and     rsp, 0FFFFFFFFFFFFFFF0h
.text:000055CEE3E009ED push    rax
.text:000055CEE3E009EE push    rsp                              ; stack_end
.text:000055CEE3E009EF lea     r8, fini                         ; fini
.text:000055CEE3E009F6 lea     rcx, init                        ; init
.text:000055CEE3E009FD lea     rdi, main                        ; main
.text:000055CEE3E00A04 call    cs:__libc_start_main_ptr
.text:000055CEE3E00A0A hlt
.text:000055CEE3E00A0A start endp
.text:000055CEE3E00A0A
.text:000055CEE3E00A0A ; ---------------------------------------------------------
.text:000055CEE3E00A0B align 10h
.text:000055CEE3E00A10
.text:000055CEE3E00A10 ; =============== S U B R O U T I N E ====================
.text:000055CEE3E00A10
.text:000055CEE3E00A10 ; Attributes: bp-based frame
```



```
.text:000055CEE3E00B4C ; int __fastcall main(int, char **, char **)
.text:000055CEE3E00B4C main proc near                           ; DATA XREF: start+
.text:000055CEE3E00B4C push    rbp
.text:000055CEE3E00B4D mov     rbp, rsp
.text:000055CEE3E00B50 mov     edi, 4
.text:000055CEE3E00B55 call    sub_55CEE3E01673
.text:000055CEE3E00B5A mov     cs:qword_55CEE4003018, rax
.text:000055CEE3E00B61 lea     rdi, sub_55CEE3E00AEA
.text:000055CEE3E00B68 call    sub_55CEE3E02120
.text:000055CEE3E00B6D mov     rax, cs:qword_55CEE4003018
.text:000055CEE3E00B74 lea     rdx, aLendrens                   ; "Lendrens"
.text:000055CEE3E00B7B lea     rsi, aK                          ; "k"
.text:000055CEE3E00B82 mov     rdi, rax
.text:000055CEE3E00B85 call    sub_55CEE3E01870
.text:000055CEE3E00B8A mov     rax, cs:qword_55CEE4003018
.text:000055CEE3E00B91 lea     rdx, aThauvI                     ; "Thauv'i"
.text:000055CEE3E00B98 lea     rsi, aD                          ; "d"
.text:000055CEE3E00B9F mov     rdi, rax
.text:000055CEE3E00BA2 call    sub_55CEE3E01870
.text:000055CEE3E00BA7 mov     rax, cs:qword_55CEE4003018
.text:000055CEE3E00BAE lea     rdx, aThrorqiek                  ; "Throrqiek"
.text:000055CEE3E00BB5 lea     rsi, aP                          ; "P"
00000B4C 000055CEE3E00B4C: main (Synchronized with RIP)
```

Now we can see names and a curious letter or symbol underneath. After some time I realized this must be a code to be deciphered by the output.txt file.After matching the letters with the corresponding names in the output.txt file I cracked the code.



```
└─$ cat outputflag.txt
Crerceon H
Ezains   T
Ummuh    B
Zonnu    {
Vinzo    l
Cuzads   1
Emoi     n
Ohols    3
Groz'ens 4
Ukox     r
Ehnu     _
Pheilons t
Cuzads   1
Khehlan  m
Ohols    3
Ehnu     _
Munis    b
Inphas   u
Pheilons t
Ehnu     _
Dut      p
Ukox     r
Ohols    3
Pheilons t
Pheilons t
Zimil    y
Ehnu     _
Honzor   s
Vinzo    l
Ukteils  0
Falnain  w
Dhohmu   !
Baadix   }

HTB{l1n34r_t1m3_but_pr3tty_slow!}
```