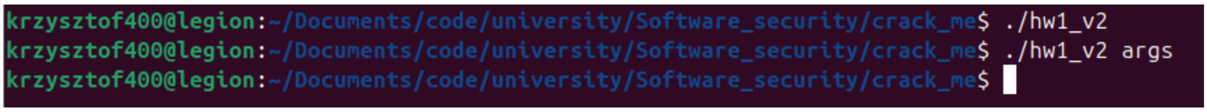
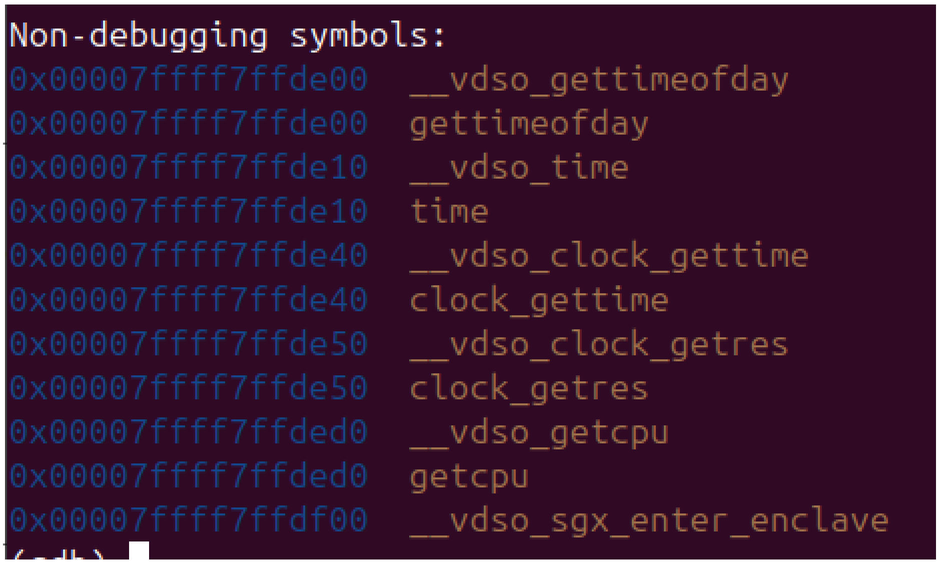
hw1:



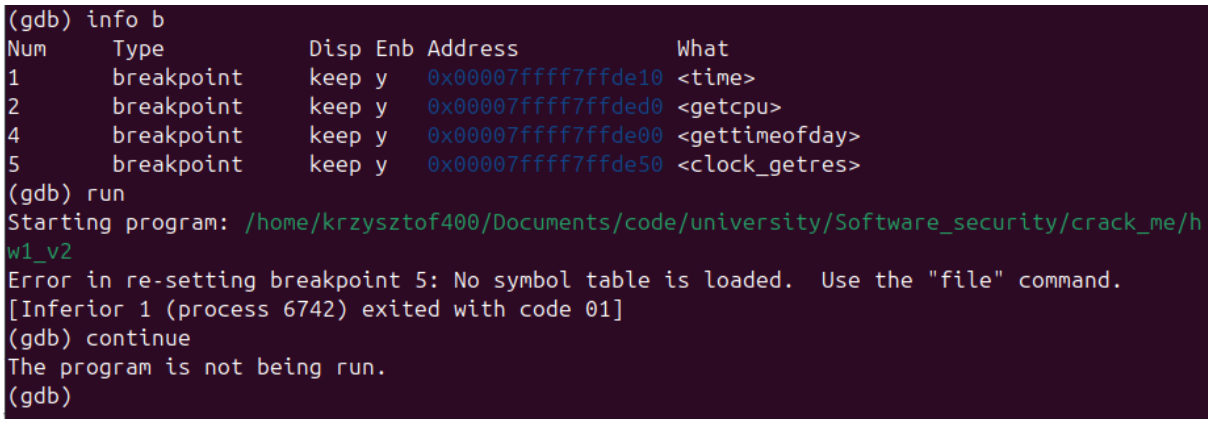
```
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$ ./hw1_v2
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$ ./hw1_v2 args
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$ 
```

after running command we see that nothing is displayed adding args doesnt change anything



```
Non-debugging symbols:
0x00007ffff7ffde00  __vdso_gettimeofday
0x00007ffff7ffde00  gettimeofday
0x00007ffff7ffde10  __vdso_time
0x00007ffff7ffde10  time
0x00007ffff7ffde40  __vdso_clock_gettime
0x00007ffff7ffde40  clock_gettime
0x00007ffff7ffde50  __vdso_clock_getres
0x00007ffff7ffde50  clock_getres
0x00007ffff7ffded0  __vdso_getcpu
0x00007ffff7ffded0  getcpu
0x00007ffff7ffdf00  __vdso_sgx_enter_enclave
(gdb) 
```

we dont know what the program is doing, judging from the functions and strings it gets time, hour and most likely stores it somewhere, most likely in elf file.



```
(gdb) info b
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00007ffff7ffde10 <time>
2       breakpoint     keep y   0x00007ffff7ffded0 <getcpu>
4       breakpoint     keep y   0x00007ffff7ffde00 <gettimeofday>
5       breakpoint     keep y   0x00007ffff7ffde50 <clock_getres>
(gdb) run
Starting program: /home/krzysztof400/Documents/code/university/Software_security/crack_me/h
w1_v2
Error in re-setting breakpoint 5: No symbol table is loaded.  Use the "file" command.
[Inferior 1 (process 6742) exited with code 01]
(gdb) continue
The program is not being run.
(gdb)
```

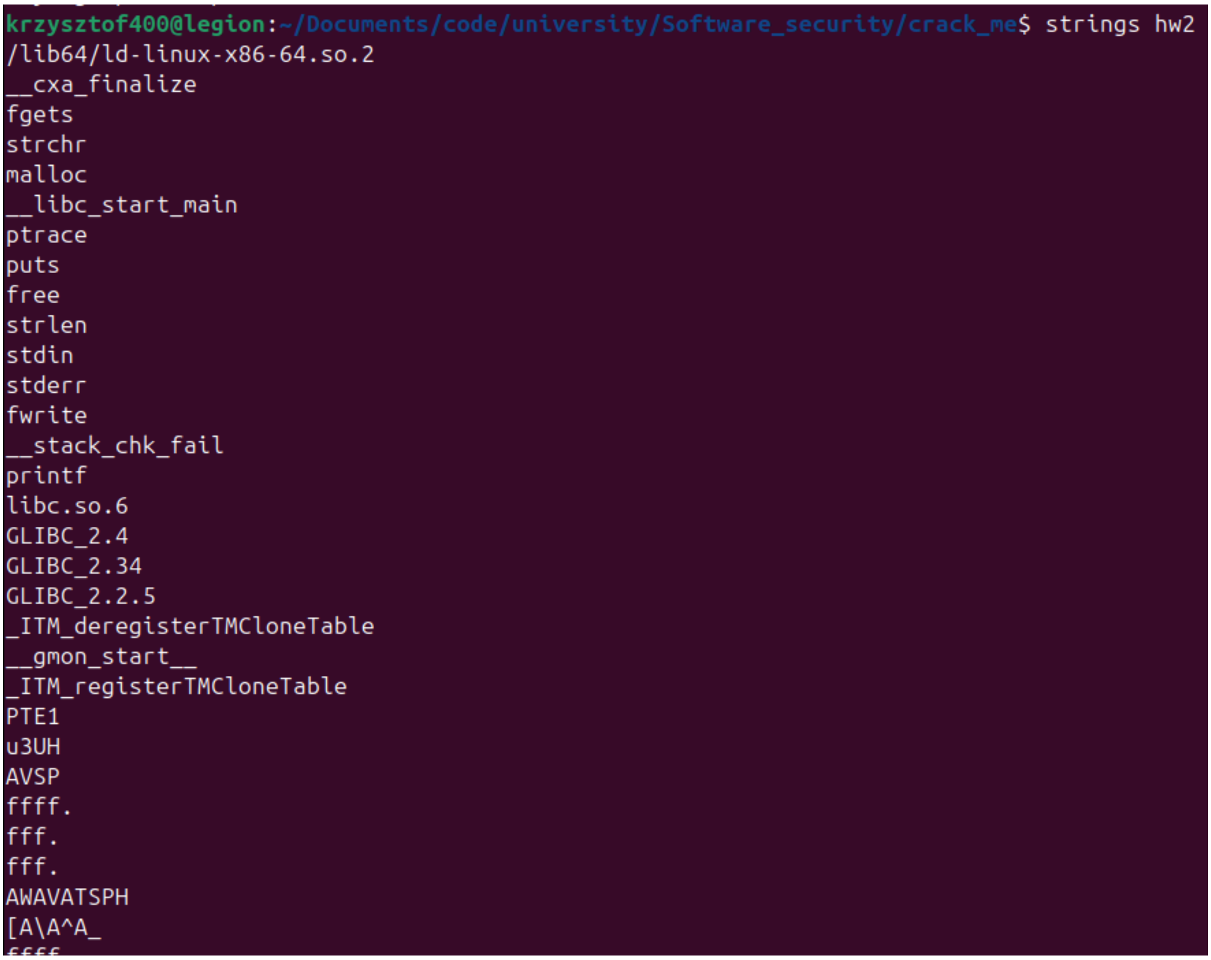
after setting breakpoints in each function program doesnt stop on any of them

listing variables, arguments doesnt tell me anything.

i cannot break this program i cannot even guess what the program is doing. got problems when tries to run it on ghidra.

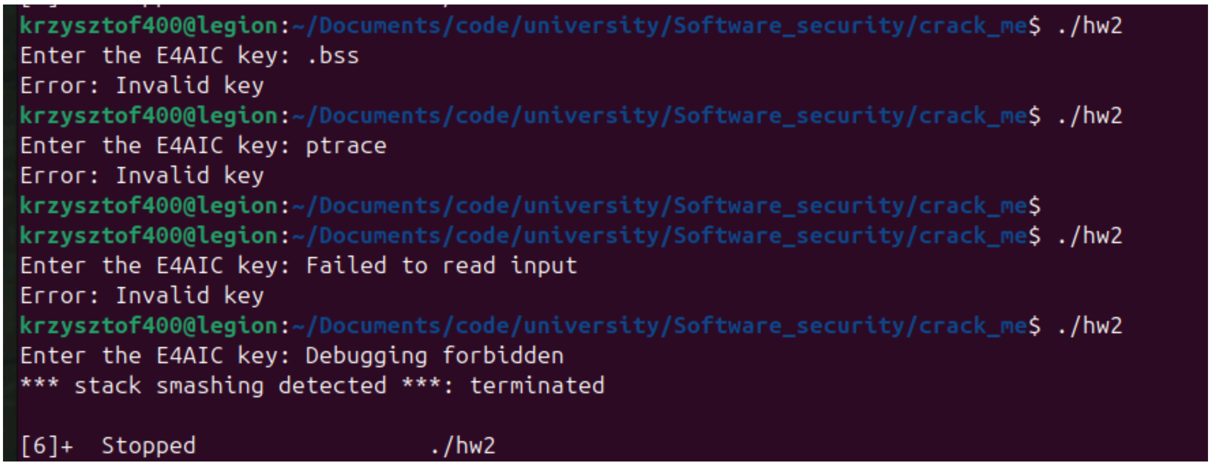
hw2:

started with displaying strings



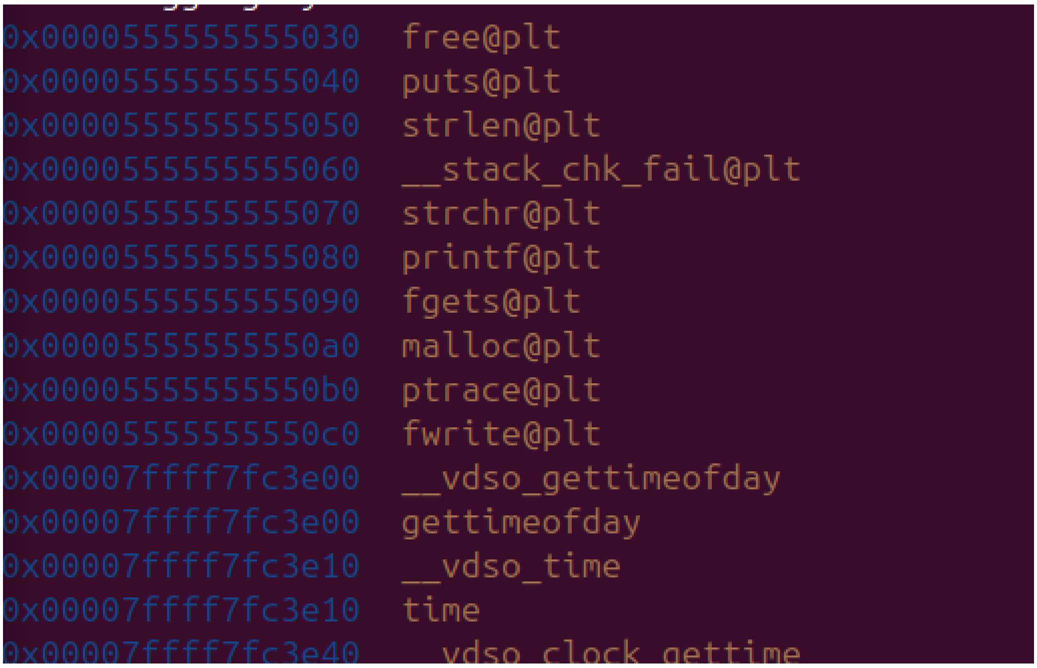
```
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$ strings hw2
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
fgets
strchr
malloc
__libc_start_main
ptrace
puts
free
strlen
stdin
stderr
fwrite
__stack_chk_fail
printf
libc.so.6
GLIBC_2.4
GLIBC_2.34
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
u3UH
AVSP
ffff.
fff.
fff.
AWAVATSPH
[A\A^A_
```

tried to enter given strings as keys



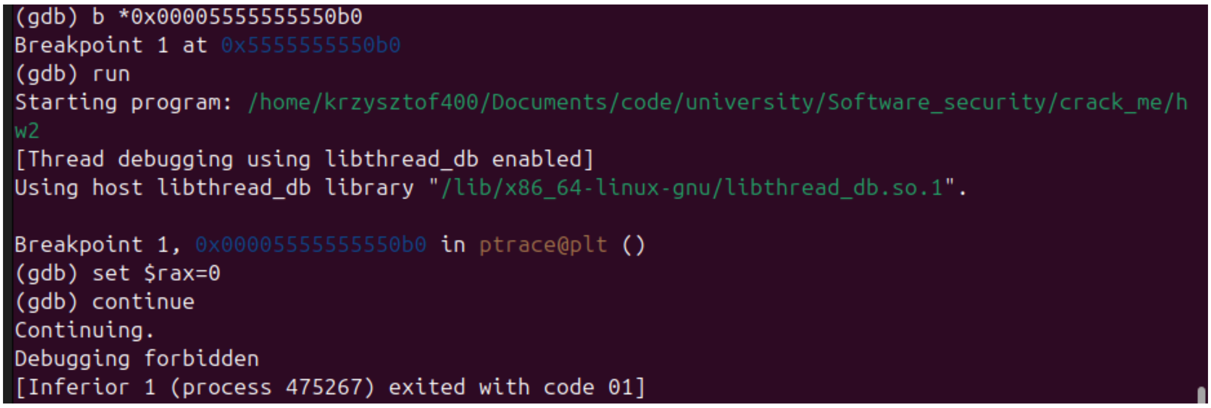
```
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$ ./hw2
Enter the E4AIC key: .bss
Error: Invalid key
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$ ./hw2
Enter the E4AIC key: ptrace
Error: Invalid key
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$ ./hw2
Enter the E4AIC key: Failed to read input
Error: Invalid key
krzysztof400@legion:~/Documents/code/university/Software_security/crack_me$ ./hw2
Enter the E4AIC key: Debugging forbidden
*** stack smashing detected ***: terminated

[6]+  Stopped                 ./hw2
```

didnt manage to achieve anything but got interesting results for Debugging forbidden input we can see functions:

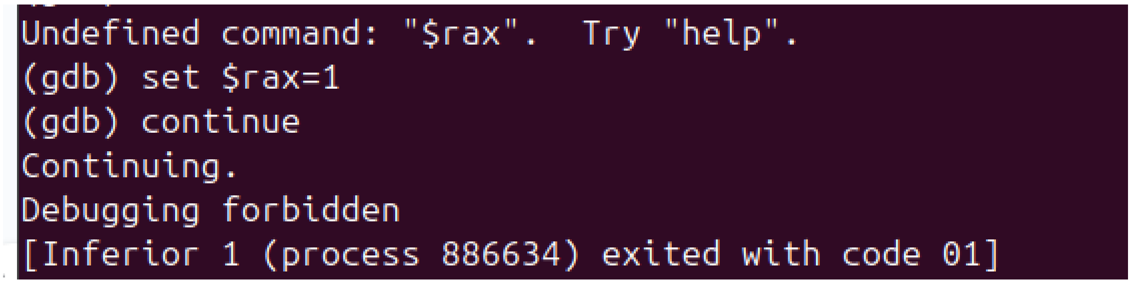


```
0x0000555555555030  free@plt
0x0000555555555040  puts@plt
0x0000555555555050  strlen@plt
0x0000555555555060  __stack_chk_fail@plt
0x0000555555555070  strchr@plt
0x0000555555555080  printf@plt
0x0000555555555090  fgets@plt
0x00005555555550a0  malloc@plt
0x00005555555550b0  ptrace@plt
0x00005555555550c0  fwrite@plt
0x00007ffff7fc3e00  __vdso_gettimeofday
0x00007ffff7fc3e00  gettimeofday
0x00007ffff7fc3e10  __vdso_time
0x00007ffff7fc3e10  time
0x00007ffff7fc3e40  __vdso_clock_gettime
```

ptrace is used to observe other threads therefor program can discover that it is debugged to prevent this we set the breakpoint for ptrace run a program and set the $rax=0 to fake succesful ptrace call

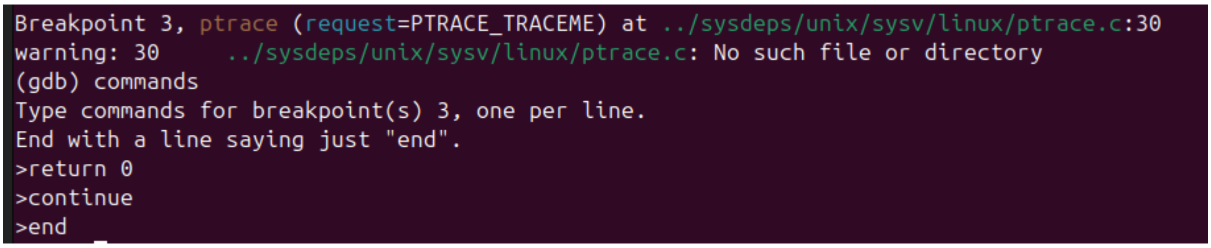


```
(gdb) b *0x00005555555550b0
Breakpoint 1 at 0x5555555550b0
(gdb) run
Starting program: /home/krzysztof400/Documents/code/university/Software_security/crack_me/h
w2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, 0x00005555555550b0 in ptrace@plt ()
(gdb) set $rax=0
(gdb) continue
Continuing.
Debugging forbidden
[Inferior 1 (process 475267) exited with code 01]
```



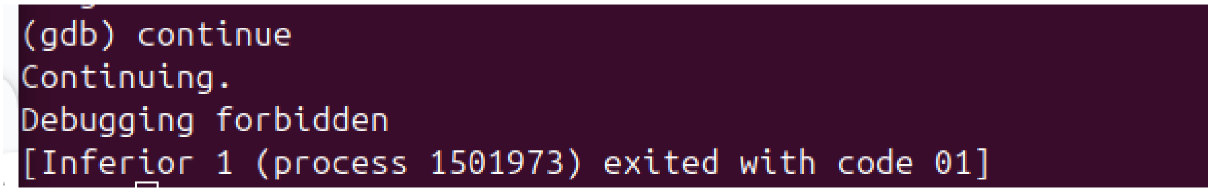
```
Undefined command: "$rax".  Try "help".
(gdb) set $rax=1
(gdb) continue
Continuing.
Debugging forbidden
[Inferior 1 (process 886634) exited with code 01]
```

we are unable to access program further by setting rax as 0. ptrace detects we are debugging program and disallows further investigation.



```
Breakpoint 3, ptrace (request=PTRACE_TRACEME) at ../sysdeps/unix/sysv/linux/ptrace.c:30
warning: 30     ../sysdeps/unix/sysv/linux/ptrace.c: No such file or directory
(gdb) commands
Type commands for breakpoint(s) 3, one per line.
End with a line saying just "end".
>return 0
>continue
>end
```

this method also doesnt work:



```
(gdb) continue
Continuing.
Debugging forbidden
[Inferior 1 (process 1501973) exited with code 01]
```

when we enter lay asm we can observe few thousand lines of code way beyond our abbilities to observe it all

function we are interested in is fgets since it is most likely for getting users input. What we would like to do is set breakpoint in this function and then change the program so that input would always be considered as correct. We cant do it because ptrace is blocking us.
We could also find a key somewhere in assembly but it is most likely well hidden prgram is using strchr not for example strcmp so it has (most likely) more elaborated way of comparing input to a key.