Room Name: Reverse Engineering

This room focuses on teaching the basics of assembly through reverse engineering

Task 2: crackme1

We can solve task2 via many process like strings command , ltrace ,strace or r2

------------------ 1st Process Via strings command -----------------

Step 1 : Download target file and check file format via file command

Step 2: Target file bin format so , target file execution file

Step 3: Use strings command and analyze file content

Command:

```
strings crackme1.bin
```

Step 4: After , analyze content of file via strings command we got password of crackme1.bin
File like this ha***

----------------- 2nd Process Via ltrace command --------------------

Step 1: Change permission of target file via chmod command .

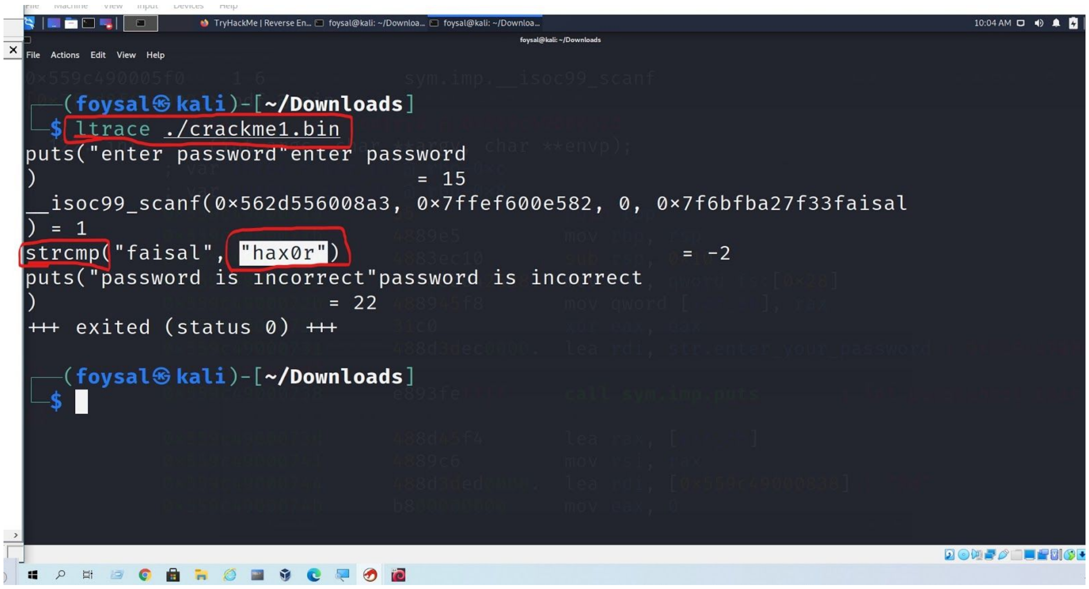
Command :

```
chmod  +x crackeme1.bin
```

Step 2: Open target file Via ltrace command

Commnad:

ltrace ./crackeme1.bin



```
┌──(foysal㉿kali)-[~/Downloads]
└─$ ltrace ./crackme1.bin
puts("enter password"enter password
)                                       = 15
__isoc99_scanf(0x562d556008a3, 0x7ffef600e582, 0, 0x7f6bfba27f33faisal
) = 1
strcmp("faisal", "hax0r")                              = -2
puts("password is incorrect"password is incorrect
)                               = 22
+++ exited (status 0) +++

┌──(foysal㉿kali)-[~/Downloads]
└─$ 
```

Step 3: Then , enter some strings and see output properly you will get strcmp for comparing two variable

Step 4: After all , we get password of our target file like this ha***

Task 3 : crackme2

Step 1: Download target file and analize via file, strings ,ltrace command

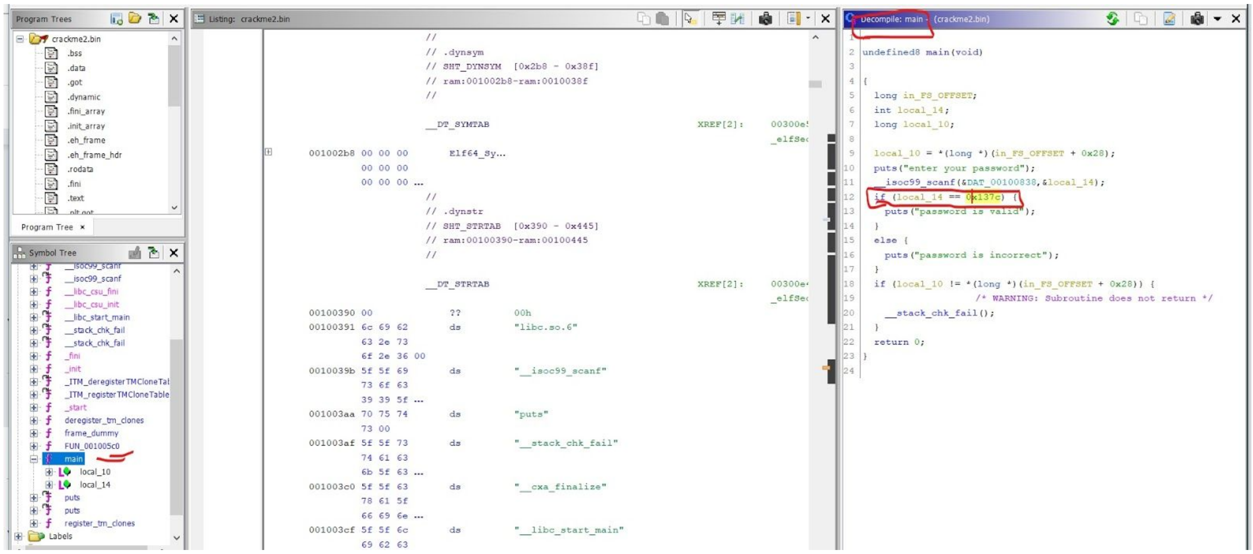
Step 2 : After using previous command like strings ,ltrace,strace we couldn't find any password For target file .

Step 3: As we couldn't find any password we tried another process via **"ghidra " Reverse Engineering Tools** .

Step 4: Open Target file via **"ghidra "** and go to main function of target file

Please , if you have no knowledge about uses of 'ghidra tools' you must learn first how to use "ghidra " >> It is so easy and so important tools for Reverse Engineering :)

Step 5 :See at decompile part of "ghidra " and you will find like this **" if (local_14 == 0x137c)"**



```
                             //
                             // .dynsym
                             // SHT_DYNSYM  [0x2b8 - 0x38f]
                             // ram:001002b8-ram:0010038f
                             //
                             __DT_SYMTAB                          XREF[2]:     00300e5
                                                                               _elfSe
        001002b8 00 00 00        Elf64_Sy...
                 00 00 00
                 00 00 00 ...
                             //
                             // .dynstr
                             // SHT_STRTAB  [0x390 - 0x445]
                             // ram:00100390-ram:00100445
                             //
                             __DT_STRTAB                          XREF[2]:     00300e4
                                                                               _elfSe
        00100390 00              ??         00h
        00100391 6c 69 62        ds         "libc.so.6"
                 63 2e 73
                 6f 2e 36 00
        0010039b 5f 5f 69        ds         "__isoc99_scanf"
                 73 6f 63
                 39 39 5f ...
        001003aa 70 75 74        ds         "puts"
                 73 00
        001003af 5f 5f 73        ds         "__stack_chk_fail"
                 74 61 63
                 6b 5f 63 ...
        001003c0 5f 5f 63        ds         "__cxa_finalize"
                 78 61 5f
                 66 69 6e ...
        001003cf 5f 5f 6c        ds         "__libc_start_main"
                 69 62 63
```

```

undefined8 main(void)

{
  long in_FS_OFFSET;
  int local_14;
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  puts("enter your password");
  __isoc99_scanf(&DAT_00100838,&local_14);
  if (local_14 == 0x137c) {
    puts("password is valid");
  }
  else {
    puts("password is incorrect");
  }
  if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
  }
  return 0;
}

```

Step 6: Convert **0x137c into decimal** you will get the password of the target file .

Password :49**

Task 4 :crackme3

Step 1 : Download target file and try previous process for find out the password of target file

Step 2 : Via previous process we couldn’t find password of Target file so try another process .

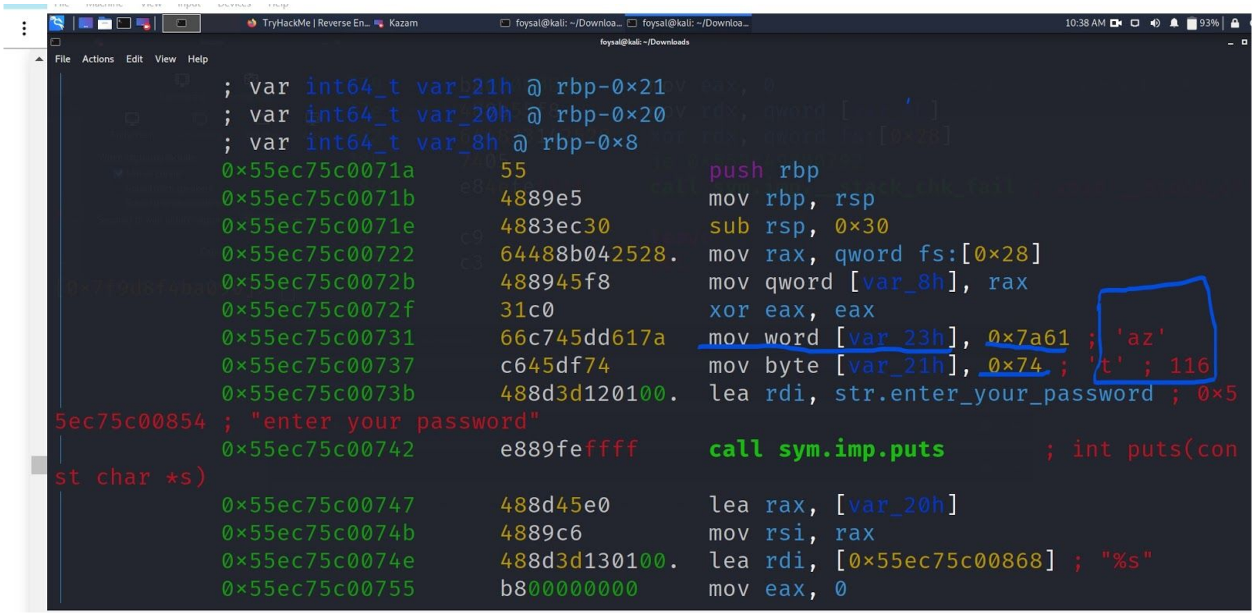
Step 3: Open target file Via **r2 (radere 2) command**.

R2 -d crackme3.bin

Step4 : Use **basic Command of Radare 2** :

1.aaa ------------for analyze all
2.afl   ------------for see all function list in target file
3.pdf @main --- for print disassembly main function

Step 5: After complete ,above all step we will see like this :


TryHackMe | Reverse En...
Kazam
foysal@kali: ~/Downloa...
foysal@kali: ~/Downloa...
10:38 AM
93%
foysal@kali: ~/Downloads
File
Actions
Edit
View
Help

```
            ; var int64_t var_21h @ rbp-0×21
            ; var int64_t var_20h @ rbp-0×20
            ; var int64_t var_8h @ rbp-0×8
            0×55ec75c0071a      55             push rbp
            0×55ec75c0071b      4889e5         mov rbp, rsp
            0×55ec75c0071e      4883ec30       sub rsp, 0×30
            0×55ec75c00722      64488b042528.  mov rax, qword fs:[0×28]
            0×55ec75c0072b      488945f8       mov qword [var_8h], rax
            0×55ec75c0072f      31c0           xor eax, eax
            0×55ec75c00731      66c745dd617a   mov word [var_23h], 0×7a61 ; 'az'
            0×55ec75c00737      c645df74       mov byte [var_21h], 0×74 ; 't' ; 116
            0×55ec75c0073b      488d3d120100.  lea rdi, str.enter_your_password ; 0×5
5ec75c00854 ; "enter your password"
            0×55ec75c00742      e889feffff     call sym.imp.puts        ; int puts(con
st char *s)
            0×55ec75c00747      488d45e0       lea rax, [var_20h]
            0×55ec75c0074b      4889c6         mov rsi, rax
            0×55ec75c0074e      488d3d130100.  lea rdi, [0×55ec75c00868] ; "%s"
            0×55ec75c00755      b800000000     mov eax, 0
```

Hare, symbolic letters are the first three letters of password

Password: “azt123”

Note : You can also get password via Break point