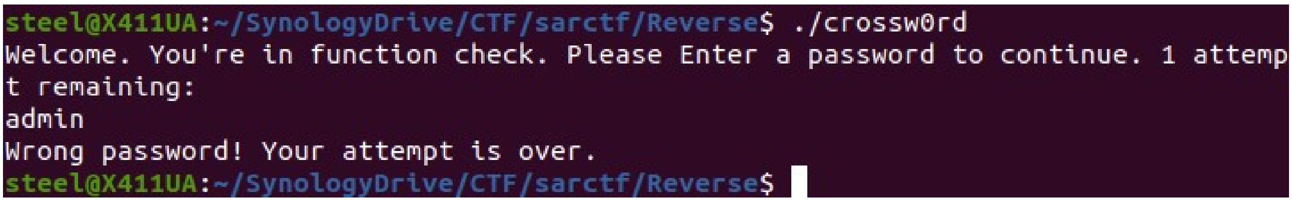
SarCTF

Reverse : Crossw0rd

Description : While the children were playing toys, Sherlock was solving crosswords in large volumes.

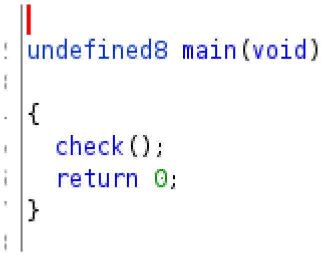
Solution :

A program is provided with this challenge. If we try to run it, it asks for a password.



```
steel@X411UA:~/SynologyDrive/CTF/sarctf/Reverse$ ./crossw0rd
Welcome. You're in function check. Please Enter a password to continue. 1 attemp
t remaining:
admin
Wrong password! Your attempt is over.
steel@X411UA:~/SynologyDrive/CTF/sarctf/Reverse$ 
```

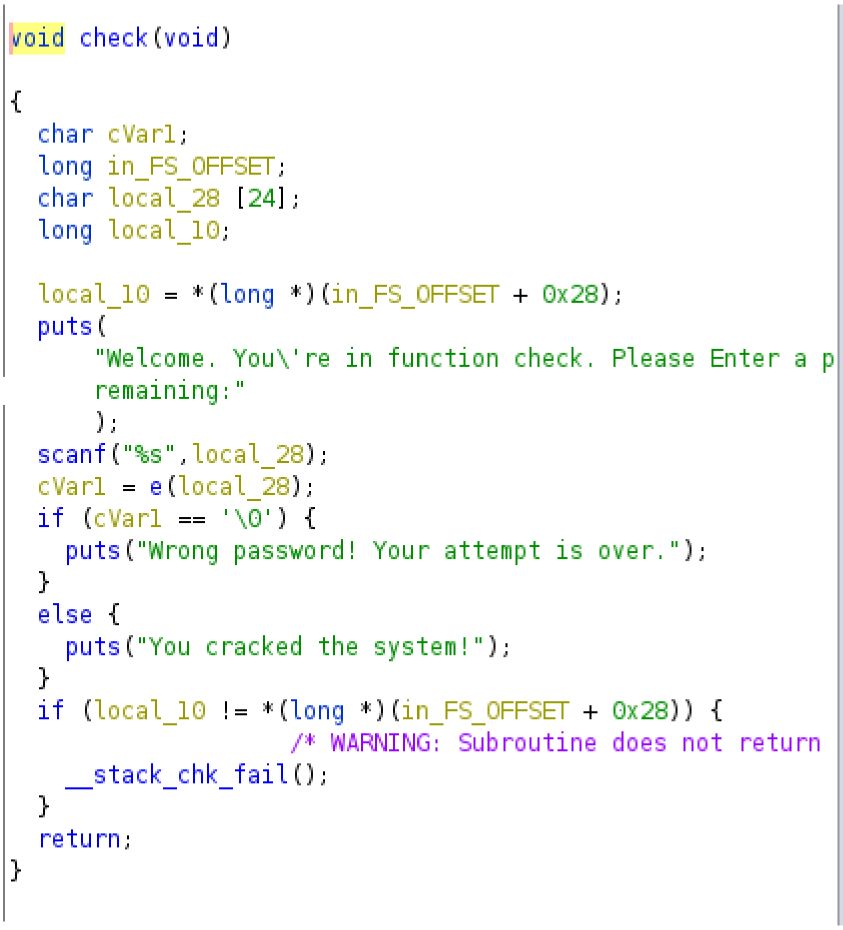
Let’s reverse it to get the password.
We open the file with ghidra. The main function only calls a function check.



```
undefined8 main(void)

{
  check();
  return 0;
}
```

The check function asks for the password, then call a function named *e* to check if it’s the right one.

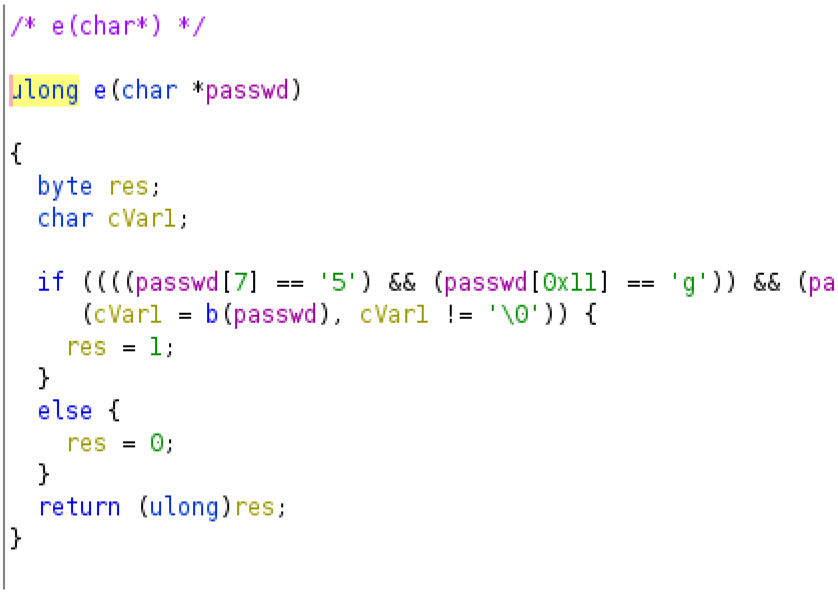


```
void check(void)

{
  char cVar1;
  long in_FS_OFFSET;
  char local_28 [24];
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  puts(
      "Welcome. You\'re in function check. Please Enter a p
      remaining:"
      );
  scanf("%s",local_28);
  cVar1 = e(local_28);
  if (cVar1 == '\0') {
    puts("Wrong password! Your attempt is over.");
  }
  else {
    puts("You cracked the system!");
  }
  if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return
    __stack_chk_fail();
  }
  return;
}
```

The e function checks 4 characters of the input, then calls a function called *b*.



```
/* e(char*) */

ulong e(char *passwd)

{
  byte res;
  char cVar1;

  if ((((passwd[7] == '5') && (passwd[0x11] == 'g')) && (pa
     (cVar1 = b(passwd), cVar1 != '\0')) {
    res = 1;
  }
  else {
    res = 0;
  }
  return (ulong)res;
}
```

There are six functions like this one, from *a* to *f*. Each one checks a different character of the input. By looking at all of the functions, we can get the password, which is the flag.

FLAG{3a5yr3v3r5ing}