








MC 문제풀이

 .gdb_history	2024-07-18 오후 3:15	GDB_HISTORY 파일	1KB
 Dockerfile	2024-07-18 오후 3:03	파일	1KB
 flag	2024-07-18 오후 3:14	파일	1KB
 ld-linux-x86-64.so.2	2024-07-18 오후 3:03	2 파일	236KB
 libc.so.6	2024-07-18 오후 3:03	6 파일	2,165KB
 mc_thread	2024-07-18 오후 3:03	파일	17KB
 mc_thread.c	2024-07-18 오후 3:03	C 파일	1KB

폴더를 열어보니 내가 못 푼 다른 문제랑 구성이 비슷했다.

내가 도커 파일에 관해서 많은 시간을 쏟았지만 알아낸 게 없으므로 일단 패스하고 C 파일 코드부터 봤다.

```
// Name: mc_thread.c
// Compile: gcc -o mc_thread mc_thread.c -pthread -no-pie
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void giveshell() { execve("/bin/sh", 0, 0); }

void init() {
    setvbuf(stdin, 0, 2, 0);
    setvbuf(stdout, 0, 2, 0);
}

void read_bytes(char *buf, int size) {
    int i;
    for (i = 0; i < size; i++)
        if (read(0, buf + i*8, 8) < 8)
            return;
}

void thread_routine() {
    char buf[256];
    int size = 0;
    printf("Size: ");
    scanf("%d", &size);
    printf("Data: ");
    read_bytes(buf, size);
}

int main() {
    pthread_t thread_t;
    init();
    if (pthread_create(&thread_t, NULL, (void *)thread_routine, NULL) < 0) {
        perror("thread create error.");
        exit(0);
    }
    pthread_join(thread_t, 0);
    return 0;
}
```

이 코드는 다중 스레드를 사용하여 입력을 처리하는 C 프로그램인 것 같다.

그리고 코드 해석 관련해서 서치하다보니 버퍼오버플로우 문제가 있는 것 같았다.

버퍼 오버플로 공격은 사용자 제어 데이터가 메모리에 기록되는 버퍼 오버플로 취약성을 의도적으로 악용하는 일반적인 사이버 공격입니다. 할당된 메모리 블록에 들어갈 수 있는 것보다 더 많은 데이터를 전송하면 공격자는 메모리의 다른 부분에 있는 데이터를 덮어쓸 수 있습니다.

아까 열어본 코드에도 그 위험성이 있는 코드인 것 같던데 왜 보안이 취약한 코드가 들어있을까?

gdb history 라는 파일을 메모장으로 열어보니 이렇게 나왔다.

```
q  
b *main  
r
```

그 외 파일들도 여러 프로그램을 이용해서 열어보려고 했는데 이 방법이 아닌 것 같은 느낌이 들었다.