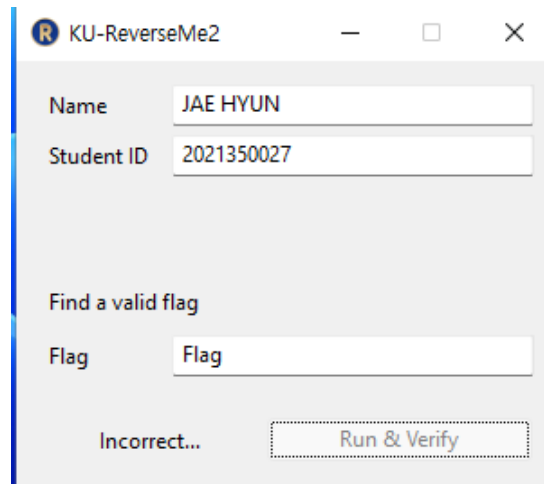


## [4] Identify instructions that delay verification processes.

- Run&Verify 버튼을 누르면 아래와 같이 Unverified에서 Incorrect가 뜨기까지 딜레이가 꽤 오래 생긴다.



### 1. 문자열 “Incorrect” 검색

- 먼저 Incorrect가 뜨는 과정을 알아봐야겠다.
- 마찬가지로 IDA에서는 검색이 안되지만, Ghidra에서는 검색이 된다,

Defined Strings - 2 items (of 713)			
Location	String Value	String Representation	Data Type
14001ff30	Correct!	u"Correct!"	unicode
14001ff48	Incorrect...	u"Incorrect..."	unicode

- 따라서 이를 참조하는 함수는 `FUN_140008d40` 임을 알았다.

### 2. FUN\_140008d40

```
void FUN_140008d40(longlong param_1)

{
    int iVar1;
    HWND *ppHVar2;
    longlong lVar3;
    undefined1 *puVar4;
    undefined *puVar5;
    undefined auStack_88 [32];
    longlong local_68;
    undefined8 local_60;
    undefined8 local_58;
    undefined local_50 [48];
    ulonglong local_20;

    local_20 = DAT_14002d080 ^ (ulonglong)auStack_88;
    local_68 = param_1;
    ppHVar2 = (HWND *)FUN_14000a5d0((HWND *)(param_1 + 8),&local_60,0x3eb);
```

```

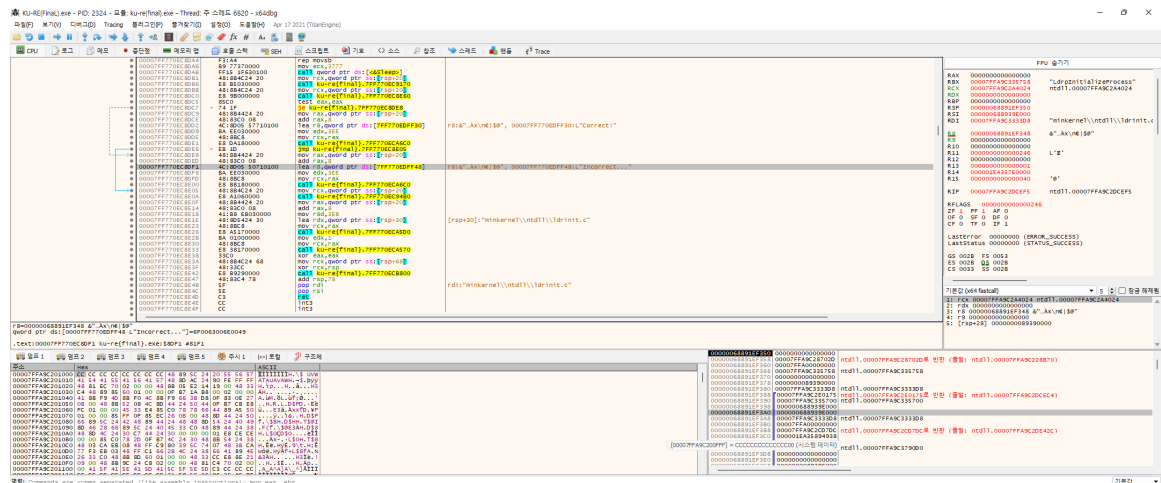
FUN_14000a570(ppHVar2,0);
puVar4 = &DAT_14001ff00;
puVar5 = local_50;
for (lVar3 = 0x2e; lVar3 != 0; lVar3 = lVar3 + -1) {
    *puVar5 = *puVar4;
    puVar4 = puVar4 + 1;
    puVar5 = puVar5 + 1;
}
Sleep(0x3777);
FUN_140009170();
iVar1 = FUN_140008e60(local_68);
if (iVar1 == 0) {
    FUN_14000a6c0((HWND *)(local_68 + 8),0x3ee,L"Incorrect...");
}
else {
    FUN_14000a6c0((HWND *)(local_68 + 8),0x3ee,L"Correct!");
}
FUN_1400094b0();
ppHVar2 = (HWND *)FUN_14000a5d0((HWND *)(local_68 + 8),&local_58,0x3eb);
FUN_14000a570(ppHVar2,1);
FUN_14000b800(local_20 ^ (ulonglong)auStack_88);
return;
}

```

- 함수 안의 `Sleep(0x3777);`로 인해 딜레이가 됨을 알 수 있다.

### 3. x64dbg로 해당 구간 확인

- 문자열 검색을 통해 Incorrect가 있는 부분을 검색해, `FUN_140008d40` 위치를 찾았다.



- 해당 sleep함수를 call하는 부분을 NULL로 채웠다.



