

# 리버싱 스터디 발표

이윤승

December 21, 2021

- 실행파일을 직접 까보자

# Preface

- 시작 인원 : 7  $\rightarrow$  3
- 이윤승(스터디장) / 명현창 / 성호
- 이유는 모르겠지만 동아리 그만둔다고 나가는 사람이 매우 많았음.

# 스터디 진행

- 딱 6번 만남. 9/29 11/24
- 디스코드 모임.
- 편차가 좀 있었지만 본인 기준 약 주당 5시간정도 소요
- 다음주 분량을 정하고 이에 대해서 각자 연습한후 (모르면 카톡방에서 질문) 모임날짜에 모여서 문제 푼것들 다시 복습.
- 연습하고 문제풀때는 화면공유하면서 다같이 진행
- 상대적으로 지식이 많은 스터디장이 최대한 관련지식을 알려주려고 노력

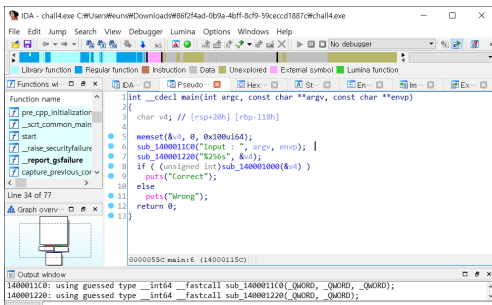
# 이동안 한것 (처음)

- 드림핵의 reversing ver.1 강의로 시작
- 조원들이 어셈을 아는 사람들이 없었기에 일단 이 부분부터 강의보면서 공부함.
- 처음엔 악으로 강으로 x86 debug 파일 쓰면서 진짜로 어셈보면서 코드 읽어봤음.

```
00000001400012F0 | $ 40:57          push rdi
00000001400012F2 | . 48:83EC 30     sub rsp,30
00000001400012F6 | . B9 10000000     mov ecx,10
00000001400012F8 | . FF15 A71D0000   call qword ptr ds:[<smallloc>]
0000000140001301 | . 48:894424 20     mov qword ptr ss:[rsp+20],rax
0000000140001306 | . 48:8B7C24 20     mov rdi,qword ptr ss:[rsp+20]
0000000140001308 | . 33C0            xor eax,eax
000000014000130D | . B9 10000000     mov ecx,10
0000000140001312 | . F3:AA          rep stosb
0000000140001314 | . 48:8D0D 351F0000 lea rcx,qword ptr ds:[140003250]
0000000140001318 | . E8 40DFFFFF     call <easy-crackme2.sub_140001060>
0000000140001320 | . FF15 6A1E0000   call qword ptr ds:[<getchar>]
0000000140001326 | . B9 01000000     mov ecx,1
000000014000132B | . 48:6BC9 00      imul rcx,rcx,0
000000014000132F | . 48:8B5424 20     mov rdx,qword ptr ss:[rsp+20]
0000000140001334 | . 8B040A          mov byte ptr ds:[rdx+rcx],al
0000000140001337 | . B8 01000000     mov eax,1
000000014000133C | . 48:6BC0 01      imul rax,rax,1
0000000140001340 | . 48:8B4C24 20     mov rcx,qword ptr ss:[rsp+20]
0000000140001345 | . C60401 64       mov byte ptr ds:[rcx+rax],64
0000000140001349 | . B8 01000000     mov eax,1
000000014000134E | . 48:6BC0 02      imul rax,rax,2
0000000140001352 | . 48:8B4C24 20     mov rcx,qword ptr ss:[rsp+20]
0000000140001357 | . C60401 77       mov byte ptr ds:[rcx+rax],77
000000014000135B | . B8 01000000     mov eax,1
0000000140001360 | . 48:6BC0 03      imul rax,rax,3
0000000140001364 | . 48:8B4C24 20     mov rcx,qword ptr ss:[rsp+20]
0000000140001369 | . C60401 73       mov byte ptr ds:[rcx+rax],73
000000014000136D | . B8 01000000     mov eax,1
0000000140001372 | . 48:6BC0 04      imul rax,rax,4
```

# 이동안 한것 (중간)

- ver.1 다음에 reversing ver.2 가 있어서 다음은 이걸로 함.
- ida 도입으로 C 디컴파일 기능을 맛보고나서 신세계 체험중
- C언어로 편하게 읽으면서 문제 난이도가 많이내려가서 꽤 어려운것도 할 수 있었음.
- 예제 문제로 실행파일을 직접 수정하는 문제도있었음.
- 몰랐는데 ver.1이 어려우니 쉬운걸로 나온게 ver.2더라



```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char v4; // [rsp+20h] [rbp-110h]
4
5     memset(&v4, 0, 0x100ui64);
6     sub_1400011C0("Input: ", argv, envp); |
7     sub_140001220("%256s", &v4);
8     if ( (unsigned int)sub_140001000(&v4) )
9         puts("Correct");
10    else
11        puts("Wrong");
12    return 0;
13 }
```

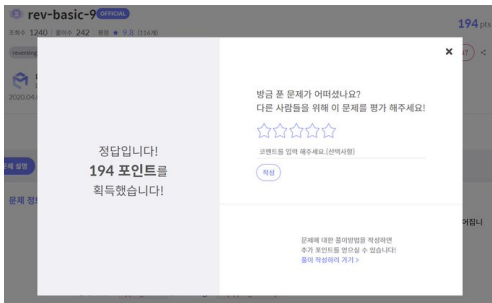
0000055C: main:6 (14000115C)

Output window

```
1400011C0: using guessed type __int64 __fastcall sub_1400011C0(_QWORD, _QWORD, _QWORD);
140001220: using guessed type __int64 __fastcall sub_140001220(_QWORD, _QWORD);
```

# 이동안 (끝)

- 실제 문제풀이
- CTF파트에 단계별 풀기 중 리버싱 문제를 순서대로 품.
- 여러 문제를 베타적으로 분배해서 각자 풀고 모여서 푼사람이 해설하면서 다른 사람이 다같이 푸는식으로 진행.



# 결과

- 총 11문제를 풀었음.
- 노력하면 어셈을 어느정도 읽을 수 있는 수준까진 도달



# 아쉬운점

- 직접 푼 모든 문제가 프로그램에 들어가는 입력값을 뽑아내는 문제였음
- 다양한 유형의 문제를 풀지 (못)않은것 정도?