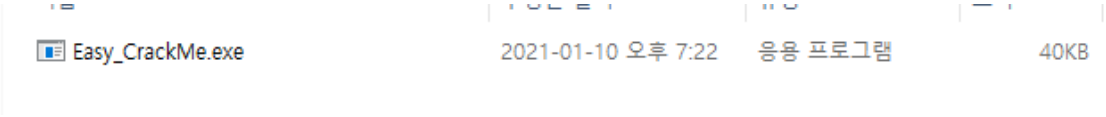
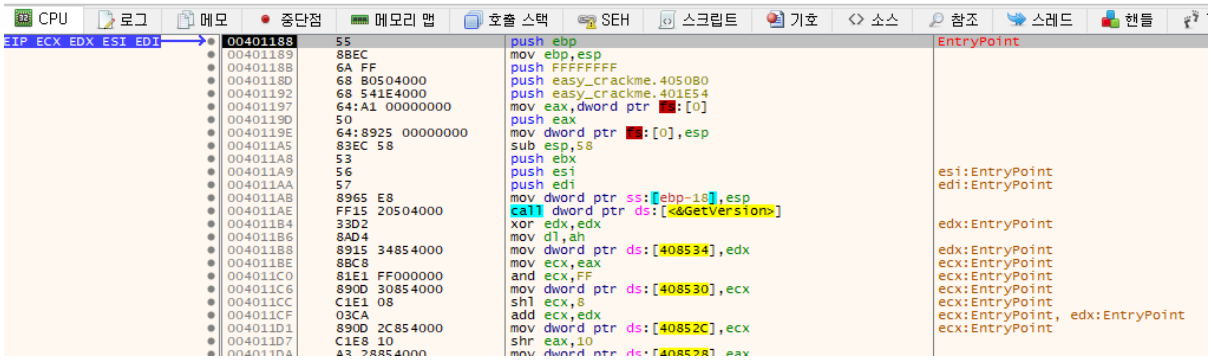
Reversing.kr – Easy Crack

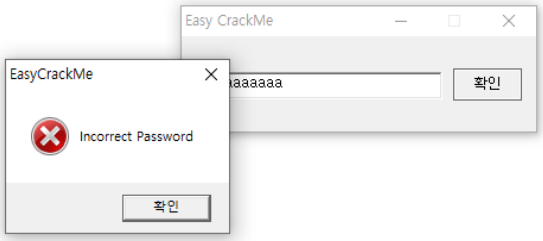
주어진 파일



X32dbg로 주어진 실행 파일 분석

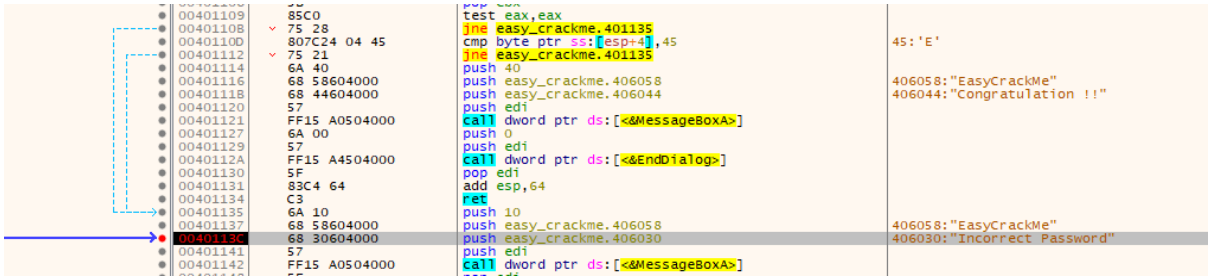


실행시킨 후 임의의 문자열 입력한 후 결과화면

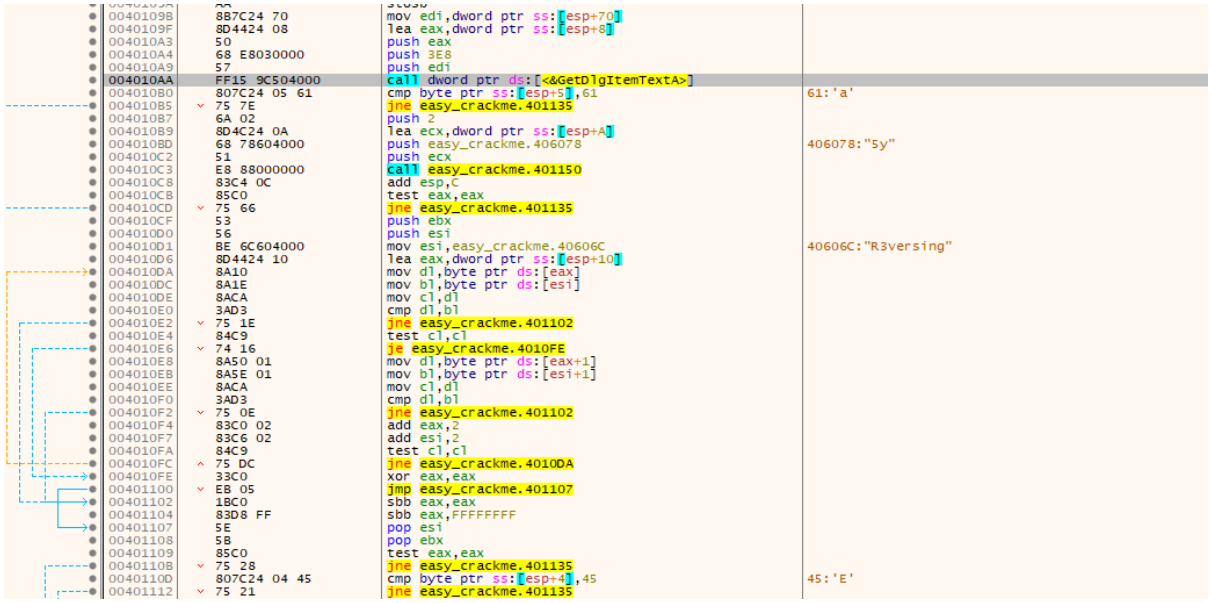


잘못된 입력 값의 결과로 출력된 메시지를 문자열 참조에서 검색 후 해당 문자열을 출력하는 코드에서 bp 설정

Bp 위치 근처에 “congratulation”이라는 문자열 출력 코드 확인

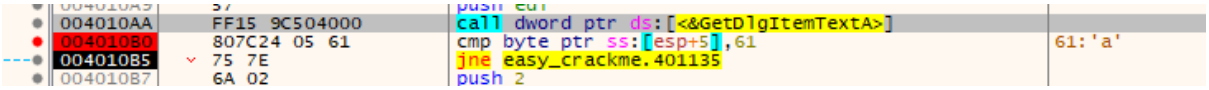


해당 위치 주변에서 입력한 문자열을 검사하는 루틴 발견

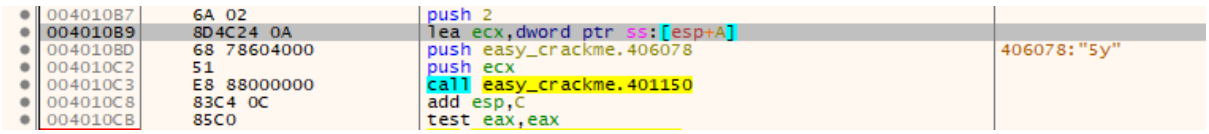


문자열 비교 루틴에 bp 설정 후 분석 재 실행

현재 esp+0x4 부터 입력한 문자열이 들어있으며 esp+0x5 의 값이 ASCII 값 61, 즉 'a' 인지 확인함 → 입력 값의 두번째 글자는 'a' 이어야함.



1번 루틴의 조건에 충족하는 문자열로 다시 실행 후 두번째 루틴 확인



현재 esp+0xA에 입력한 문자열의 3번째 문자부터 저장되어 있음. 이 문자열과 미리 저장되어있는 "5y" 와 비교연산

```
00401150   55          push ebp
00401151   8BEC        mov ebp,esp
00401153   57          push edi
00401154   56          push esi
00401155   53          push ebx
00401156   8B4D 10     mov ecx,dword ptr ss:[ebp+10]
00401159   E3 26       jecxz easy_crackme.401181
0040115B   8BD9        mov ebx,ecx                    ecx:"0123456789:;<=>"
0040115D   8B7D 08     mov edi,dword ptr ss:[ebp+8]
00401160   8BF7        mov esi,edi
00401162   33C0        xor eax,eax
00401164   F2:AE       repne scasb
00401166   F7D9        neg ecx                        ecx:"0123456789:;<=>"
00401168   03CB        add ecx,ebx                    ecx:"0123456789:;<=>"
0040116A   8BFE        mov edi,esi
0040116C   8B75 0C     mov esi,dword ptr ss:[ebp+C]
0040116F   F3:A6       repe cmpsb
00401171   8A46 FF     mov al,byte ptr ds:[esi-1]
00401174   33C9        xor ecx,ecx                    ecx:"0123456789:;<=>"
00401176   3A47 FF     cmp al,byte ptr ds:[edi-1]
00401179   77 04       ja easy_crackme.40117F
0040117B   74 04       je easy_crackme.401181
0040117D   49          dec ecx                        ecx:"0123456789:;<=>"
0040117E   49          dec ecx                        ecx:"0123456789:;<=>"
0040117F   F7D1        not ecx                        ecx:"0123456789:;<=>"
00401181   8BC1        mov eax,ecx                    ecx:"0123456789:;<=>"
00401183   5B          pop ebx
00401184   5E          pop esi
00401185   5F          pop edi
00401186   C9          leave
00401187   C3          ret
```

```
0019F7D0  00000064
0019F7D4  0019F884
0019F7D8  004010B0  easy_crackme.004010B0로 반환 (출발: ???)
0019F7DC  004010C8  easy_crackme.004010C8로 반환 (출발: easy_
0019F7E0  0019F7F2  "0123456789:;<=>"
0019F7E4  00406078  "5y"
0019F7E8  00000002
0019F7EC  00000111
0019F7F0  31306161
```

따라서 3, 4 번째 문자 = "5y" 이므로 두번째 루틴도 충족하는 문자열 다시 입력.

```
004010D1   BE 6C604000     mov esi,easy_crackme.40606C    esi:"R3versing", 40606C:"R3versing"
004010D6   8D4424 10       lea eax,dword ptr ss:[esp+10]
004010DA   8A10            mov dl,byte ptr ds:[eax]
004010DC   8A1E            mov bl,byte ptr ds:[esi]       esi:"R3versing"
```

이후 3번째 검사 루틴에서 5번째 문자부터 미리 저장된 "R3versing" 과 비교

```
004010D1   BE 6C604000     mov esi,easy_crackme.40606C
004010D6   8D4424 10       lea eax,dword ptr ss:[esp+10]
004010DA   8A10            mov dl,byte ptr ds:[eax]
004010DC   8A1E            mov bl,byte ptr ds:[esi]
004010DE   8ACA            mov cl,dl
004010E0   3AD3            cmp dl,bl
004010E2   75 1E           jne easy_crackme.401102
```

```
EAX  0019F7F4  "0123456789"
EBX  00000052  'R'
ECX  00000030  '0'
EDX  00000030  '0'
EBP  0019F884
ESP  0019F7E4  "h\n*"
ESI  0040606C  "R3versing"
```

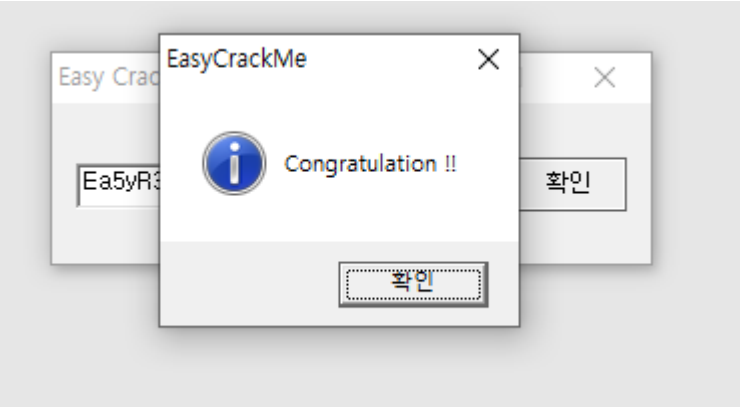
5번째 문자부터 "R3versing"과 한 문자라도 다르면 "Incorrect Password" 출력

문자열 입력값을 "aa5yR3versing"로 다시 시도

```
0040110B   75 28           jne easy_crackme.401135
0040110D   807C24 04 45    cmp byte ptr ss:[esp+4],45     45:'E'
00401112   75 21           jne easy_crackme.401135
```

마지막 검사 루틴에서 esp+0x4에 들어있는 입력 문자열의 첫번째 문자와 "E"를 비교

문자열 입력값 = "Ea5yR3versing" 으로 다시 시도



Flag = "Ea5yR3versing"