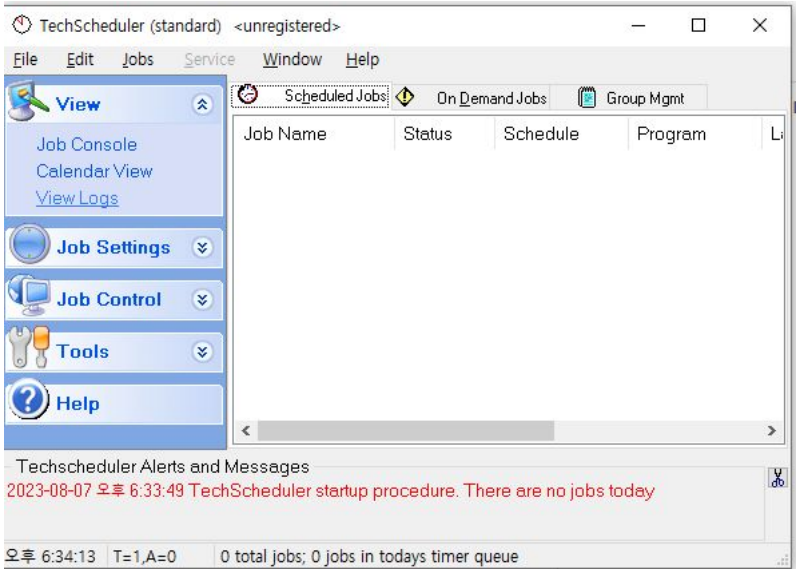
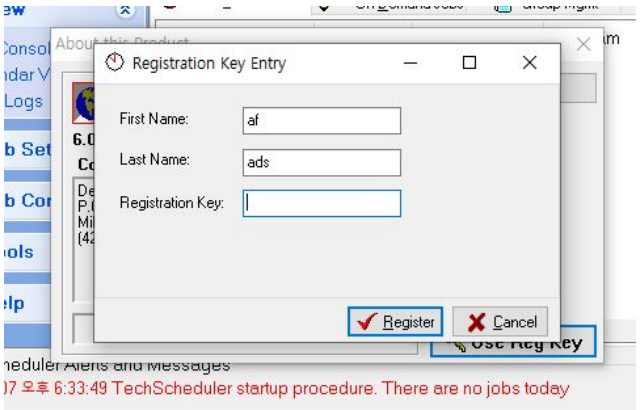
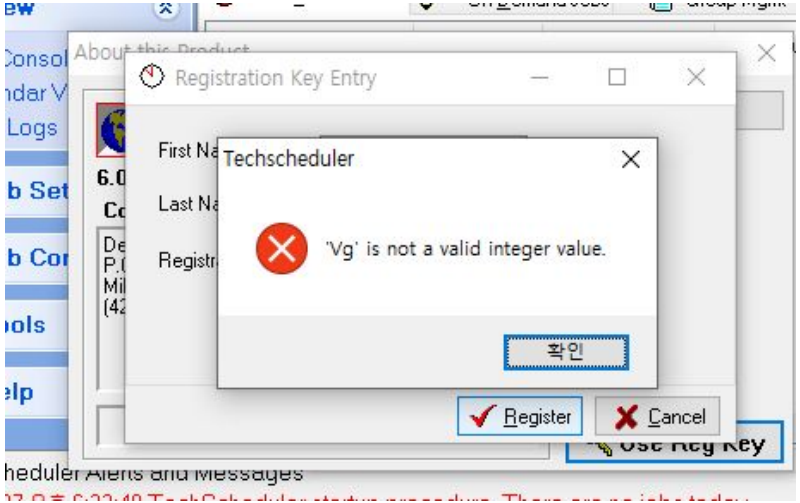




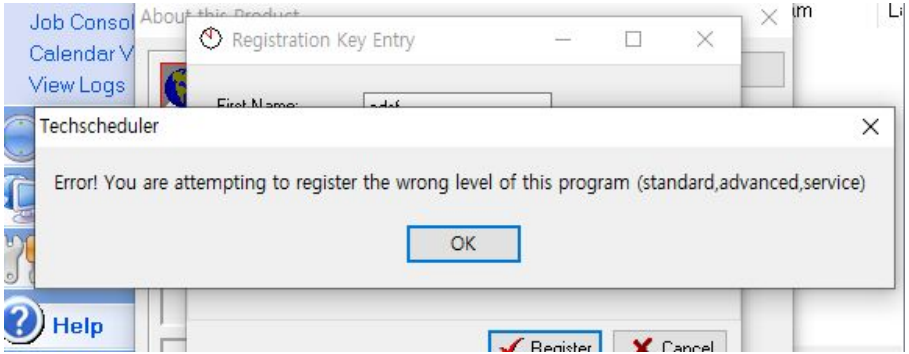
등록 키를 입력해주면, 오리지널 버전을 이용할 수 있다.

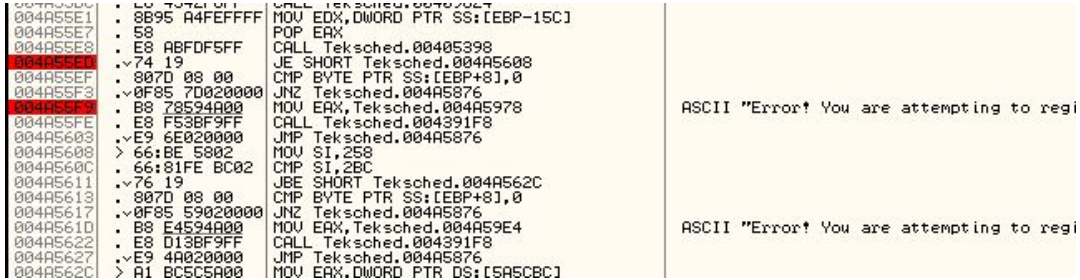
여느때와 다를거 없이 등록키를 입력해주는 것이 아닌 프로그램 자체를 우회하는 것이다.

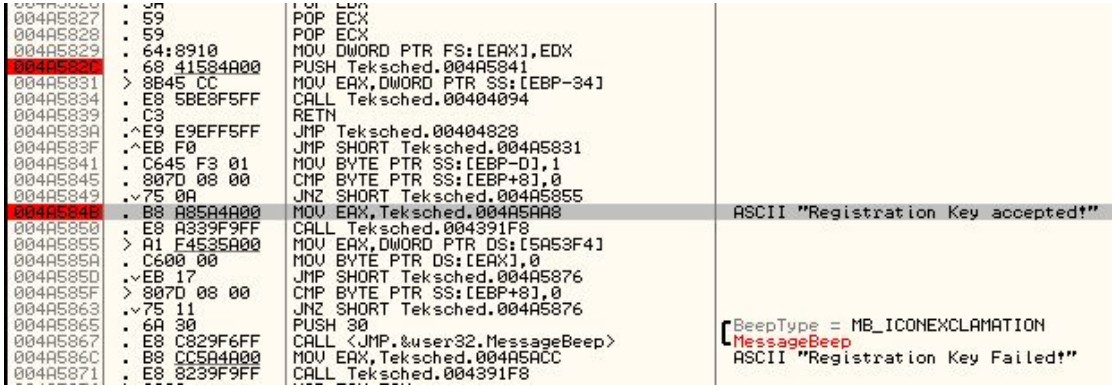


아무 키를 입력하고, 등록을 누르면 이와 같은 에러가 뜬다. 문자열 검색기능을 이용해 해당 부분으로 이동한다.

Vg라는 값을 입력하면, 아래와 같은 문구가 뜬다. 아래 문구로 검색해본다.





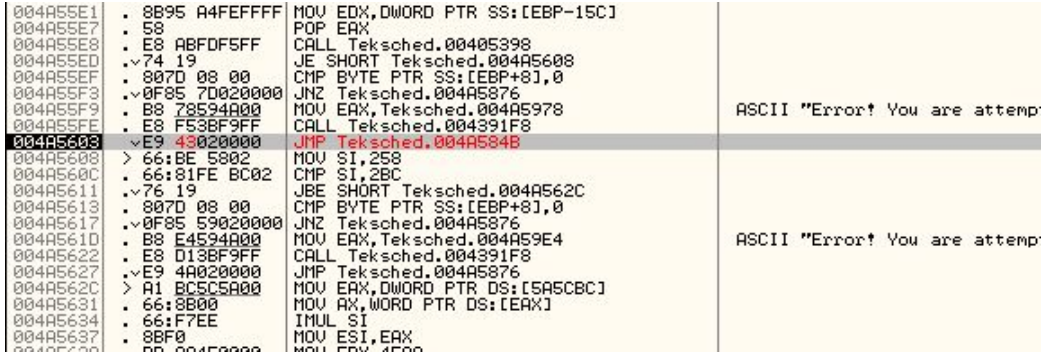


이동하면 이렇게 두개의 문자열이 보인다. 그리고 분기문에 따라 호출되는 문자열도 달라진다.

아래로 조금 더 살펴보면, Registeration Key accepted 문자열이 보인다. 그 주변 명령을 살펴본다.

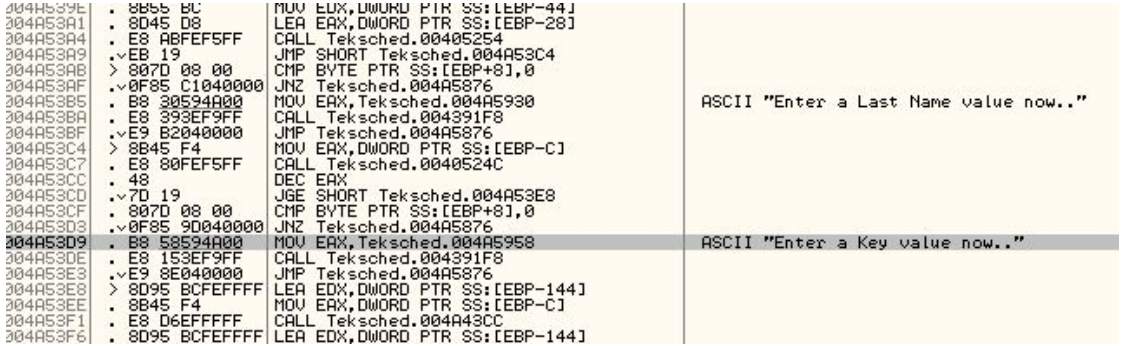
위에 비교문이 보이고 분기문도 보인다. 위에 또 살펴보니 해당 비교문에 점프하는 코드가 보인다.

4A592C를 살펴보면 4A5841을 PUSH하고 그다음 RETN 명령에 의해 unconditonal jump를 한다.



위에 JMP 4A4876 부분을 4A584B로 바꿔 등록 성공 메시지를 뜨게 할 수 있다. 하지만 이 방식은 비효율적이다. 매번 아무키로 등록을 해야 정식 버전을 사용할 수 있기 때문이다.

우회 목적은 프로그램을 키자마자 정식버전으로 사용할 수 있게 하기 위함이다.



4A592C주소에서 PUSH 4A5941 후 RETN함으로써 해당 주소로 넘어오게 된다.

분석하다 보면 다음과 같은 문자열을 볼 수 있다.

이 문자열은 사용자가 Last name 부분과 key를 입력하지 않았을 때 나타나는 부분이다.

분명 이 문자열을 호출하는 부분이 있을 것이다.

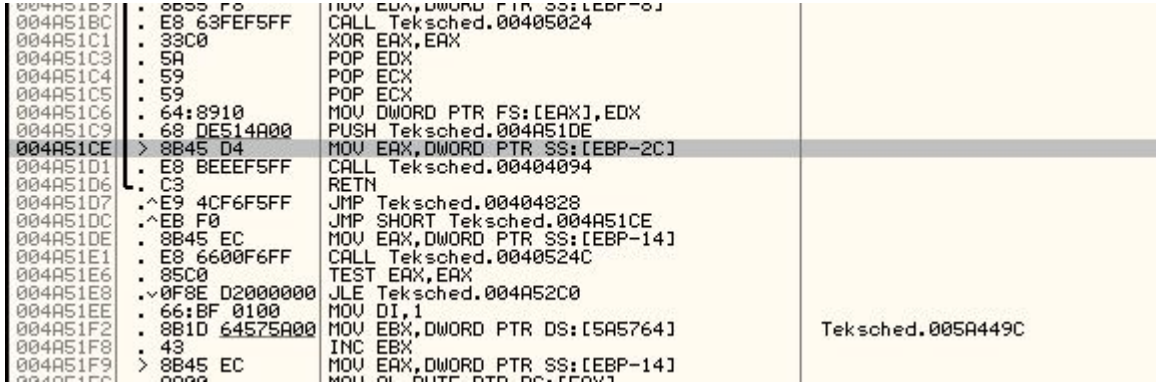
```
004A5396  . 8D45 BC       LEA EAX,DWORD PTR SS:[EBP-44]
004A5399  . E8 D6FDF5FF   CALL Teksched.00405174
004A539E  . 8B55 BC       MOV EDX,DWORD PTR SS:[EBP-44]
004A53A1  . 8D45 D8       LEA EAX,DWORD PTR SS:[EBP-28]
004A53A4  . E8 ABFEF5FF   CALL Teksched.00405254
004A53A9  .v EB 19        JMP SHORT Teksched.004A53C4
004A53AB  > 807D 08 00    CMP BYTE PTR SS:[EBP+8],0
004A53AF  .v 0F85 C1040000 JNZ Teksched.004A5876
004A53B5  . B8 30594A00   MOV EAX,Teksched.004A5930        ASCII "Enter a Last Name value now.."
004A53BA  . E8 393EF9FF   CALL Teksched.004391F8
004A53BF  .v E9 B2040000  JMP Teksched.004A5876
004A53C4  > 8B45 F4       MOV EAX,DWORD PTR SS:[EBP-C]
004A53C7  . E8 80FEF5FF   CALL Teksched.0040524C
004A53CC  . 48            DEC EAX
004A53CD  .v 7D 19        JGE SHORT Teksched.004A53E8
```

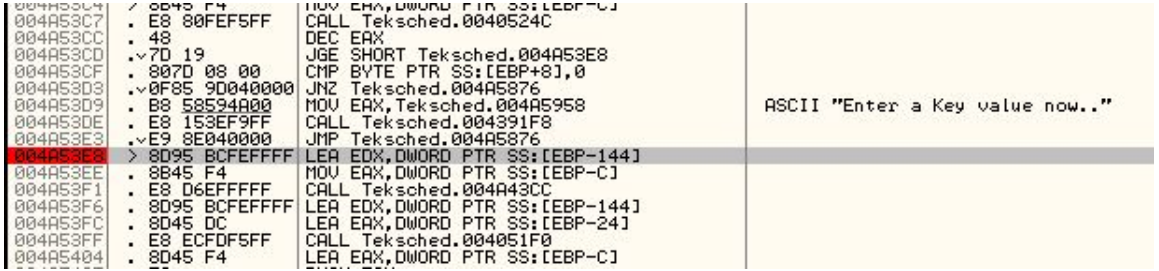
```
004A5408  . 8B45 DC
Jump from 004A52E3
```

```
004A52C0  > 807D 08 00    CMP BYTE PTR SS:[EBP+8],0
004A52C4  .v 0F85 AC050000 JNZ Teksched.004A5876
004A52CA  . B8 08594A00   MOV EAX,Teksched.004A5908        ASCII "Enter a First Name value now.."
004A52CF  . E8 243FF9FF   CALL Teksched.004391F8
004A52D4  .v E9 9D050000  JMP Teksched.004A5876
004A52D9  > 8B45 E8       MOV EAX,DWORD PTR SS:[EBP-18]
004A52DC  . E8 6BFFF5FF   CALL Teksched.0040524C
004A52E1  . 85C0          TEST EAX,EAX
004A52E3  .v 0F8E C2000000 JLE Teksched.004A53AB
004A52E9  . 66:BF 0100    MOV DI,1
004A52ED  . 8B1D 64575A00 MOV EBX,DWORD PTR DS:[5A5764]   Teksched.005A449C
004A52F3  . 43            INC EBX
004A52F4  > 8B45 E8       MOV EAX,DWORD PTR SS:[EBP-18]
```

이 부분을 살펴보면 4A52E3 부분에서 호출하는 것을 볼 수 있다.

해당 부분을 보면 First Name을 입력하지 않았을 시 나타나는 문자열이다.

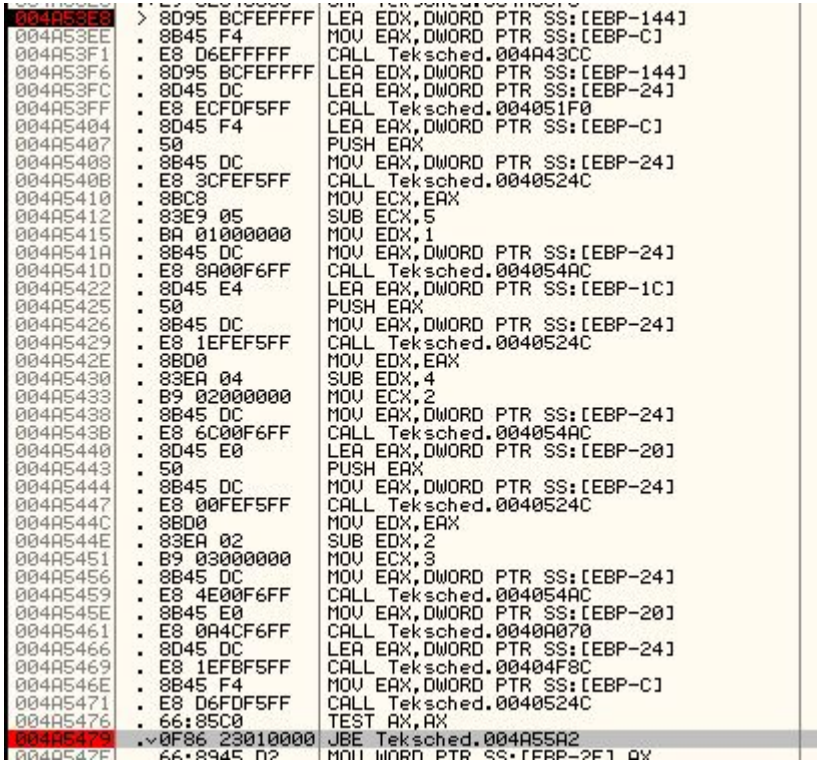




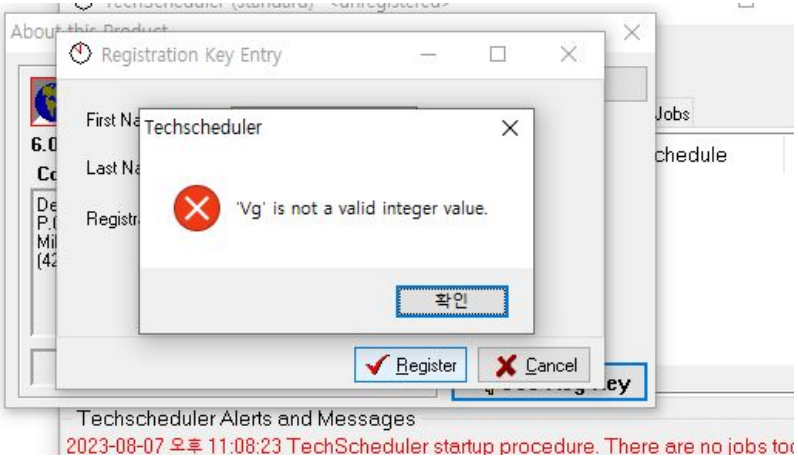
하나 하나 위로 올라가보면서 분기문을 호출하는 부분을 쭉 따라가보면 해당 부분이 나온다.

PUSH 4A51DE 을 하고, RETN 함으로써 해당 주소로 점프하는 코드임을 알 수 있다.

키가 입력되지 않았을경우 점프하는 부분 다음 주소에 bp를 걸고 살펴본다. 왜냐하면 해당 부분 이후에 분명 아까 키 등록에 실패하였을경우에 뜨는 문자열에 점프하는 부분이 있을것이다.



bp걸었던 부분 다음부터 다양한 함수를 호출하는데, 그 이후에 나타나는 분기문에 bp를 하나 더 걸고 분석해본다.



그다음 실행해보면 이런 유효한 정수값이 아니라는 문자열과 함께 오류가 뜬다.

아무래도 그 사이에 있는 함수 부분에서 오류를 내뱉는 것같아 하나 하나 따라가면서 분석한다.

```
778C4EA7 v74 0E          JE SHORT ntdll.778C4EB7
778C4EA9   8B0D A0699777 MOV ECX,DWORD PTR DS:[779769A0]
778C4EAF   FF15 E0919777 CALL DWORD PTR DS:[779791E0]      ntdll.RtlDebugPrintTimes
778C4EB5   FFE1          JMP ECX
778C4EB7   FC            CLD
778C4EB8   8B4C24 04     MOV ECX,DWORD PTR SS:[ESP+4]
778C4EBC   8B1C24        MOV EBX,DWORD PTR SS:[ESP]
778C4EBF   51            PUSH ECX
778C4EC0   53            PUSH EBX
778C4EC1   E8 F640FFFF   CALL ntdll.778B8FBC
778C4EC6   0AC0          OR AL,AL
```

하나 하나 따라가다가 어느 순간 ntdll로 진입하는걸 볼 수 있다.

'-'를 눌러 어디서 호출됐는지 보면 해당 부분에서 호출하는걸 알 수 있다.

```
004A544E  . 83EA 02      SUB EDX,2
004A5451  . B9 03000000  MOV ECX,3
004A5456  . 8B45 DC      MOV EAX,DWORD PTR SS:[EBP-24]
004A5459  . E8 4E00F6FF  CALL Teksched.004054AC
004A545E  . 8B45 E0      MOV EAX,DWORD PTR SS:[EBP-20]
004A5461  . E8 0A4CF6FF  CALL Teksched.0040A070
004A5466  . 8D45 DC      LEA EAX,DWORD PTR SS:[EBP-24]
004A5469  . E8 1EFBF5FF  CALL Teksched.00404F8C
004A546E  . 8B45 F4      MOV EAX,DWORD PTR SS:[EBP-C]
004A5471  . E8 D6FDF5FF  CALL Teksched.0040524C
004A5476  . 66:85C0      TEST AX,AX
004A5479  .v0F86 23010000 JBE Teksched.004A55A2
004A547F  . 66:8945 D2   MOV WORD PTR SS:[EBP-2E],AX
```

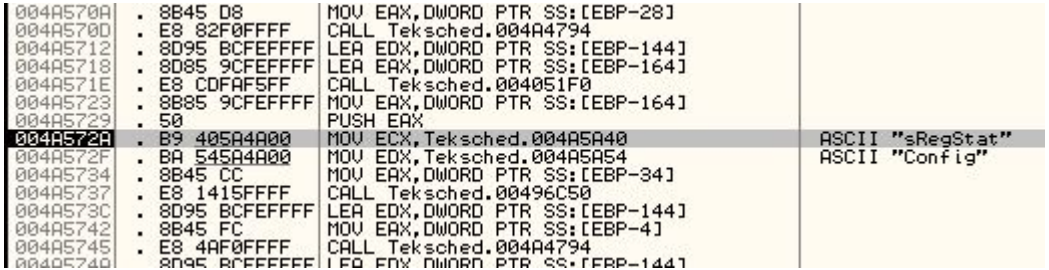
```
004A544E  . 83EA 02         SUB EDX,2
004A5451  . B9 03000000     MOV ECX,3
004A5456  . 8B45 DC         MOV EAX,DWORD PTR SS:[EBP-24]
004A5459  . E8 4E00F6FF     CALL Teksched.004054AC
004A545E  . 8B45 E0         MOV EAX,DWORD PTR SS:[EBP-20]
004A5461  . E8 0A4CF6FF     CALL Teksched.0040A070
004A5466  . 8D45 DC         LEA EAX,DWORD PTR SS:[EBP-24]
004A5469  . E8 1EFBF5FF     CALL Teksched.00404F8C
004A546E  . 8B45 F4         MOV EAX,DWORD PTR SS:[EBP-C]
004A5471  . E8 D6FDF5FF     CALL Teksched.0040524C
004A5476  . 66:85C0         TEST AX,AX
004A5479  .v0F86 23010000   JBE Teksched.004A55A2
004A547F  . 66:8945 D2      MOV WORD PTR SS:[EBP-2E],AX
```

```
004A544E  . 83EA 02         SUB EDX,2
004A5451  . B9 03000000     MOV ECX,3
004A5456  . 8B45 DC         MOV EAX,DWORD PTR SS:[EBP-24]
004A5459  . E8 4E00F6FF     CALL Teksched.004054AC
004A545E  . 8B45 E0         MOV EAX,DWORD PTR SS:[EBP-20]
004A5461    90              NOP
004A5462    90              NOP
004A5463    90              NOP
004A5464    90              NOP
004A5465    90              NOP
004A5466  . 8D45 DC         LEA EAX,DWORD PTR SS:[EBP-24]
004A5469  . E8 1EFBF5FF     CALL Teksched.00404F8C
004A546E  . 8B45 F4         MOV EAX,DWORD PTR SS:[EBP-C]
004A5471  . E8 D6FDF5FF     CALL Teksched.0040524C
004A5476  . 66:85C0         TEST AX,AX
004A5479  .v0F86 23010000   JBE Teksched.004A55A2
004A547F  . 66:8945 D2      MOV WORD PTR SS:[EBP-2E],AX
004A5483  . 66:BF 0100      MOV DI,1
004A5487  > 66:BE 0100      MOV SI,1
004A548B  . 8B1D 64575A00   MOV EBX,DWORD PTR DS:[5A5764]   Tek
```

분기 하는걸 막기 위해 해당 부분을 NOP으로 패치한다.

그리고 계속 실행한다.

하나하나 분석하다보면 분기문이 많이 나오는데, 에러나오는 분기문을 모두 안나오게 zero flag를 조절해가면서 분석했다.



그리고 내려가다 보면 해당 분기문이 나온는데, 4A585F 부분은 등록 실패 에러가 나오는 부분이므로, zero flag를 임의로 변경해 점프 못하도록 한다.

계속 분석해보면 아래와 같이 등록과 관련된 함수를 호출하는것처럼 보인다. PUSH EAX를 하고, 스택창을 봤더니 어떤 값이 저장되어있다.

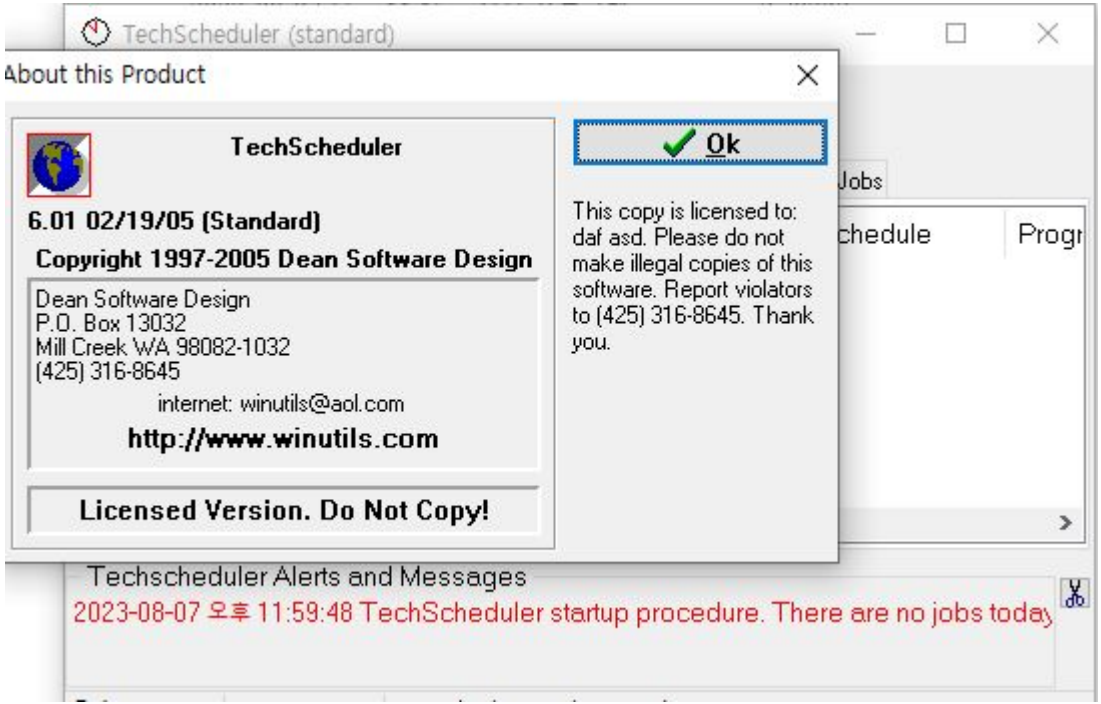
아마 해당 값이 레지스터 키 값인 걸로 보인다. 키를 등록하는것이 아닌 완전 우회하는것이 목적이기 때문에 안쓴다.

```
004A581F  . E8 2000FFFF   CALL Teksched.00495844
004A5824  . 33C0          XOR EAX,EAX
004A5826  . 5A            POP EDX
004A5827  . 59            POP ECX
004A5828  . 59            POP ECX
004A5829  . 64:8910       MOV DWORD PTR FS:[EAX],EDX
004A582C  . 68 41584A00   PUSH Teksched.004A5841
004A5831  > 8B45 CC       MOV EAX,DWORD PTR SS:[EBP-34]
004A5834  . E8 5BE8F5FF   CALL Teksched.00404094
004A5839  . C3            RETN
004A583A  .^E9 E9EFF5FF   JMP Teksched.00404828
004A583F  .^EB F0         JMP SHORT Teksched.004A5831
004A5841  . C645 F3 01    MOV BYTE PTR SS:[EBP-D],1
004A5845  . 807D 08 00    CMP BYTE PTR SS:[EBP+8],0
004A5849  .v75 0A         JNZ SHORT Teksched.004A5855
004A584B  . B8 A85A4A00   MOV EAX,Teksched.004A5AA8   ASCII "Registration Key accepted!"
004A5850  . E8 A339F9FF   CALL Teksched.004391F8
004A5855  > A1 F4535A00   MOV EAX,DWORD PTR DS:[5A53F4]
```

```
004A5834  . E8 5BE8F5FF   CALL Teksched.00404094
004A5839  . C3            RETN
004A583A  .^E9 E9EFF5FF   JMP Teksched.00404828
004A583F  .^EB F0         JMP SHORT Teksched.004A5831
004A5841  . C645 F3 01    MOV BYTE PTR SS:[EBP-D],1
004A5845  . 807D 08 00    CMP BYTE PTR SS:[EBP+8],0
004A5849  .v75 0A         JNZ SHORT Teksched.004A5855
004A584B  . B8 A85A4A00   MOV EAX,Teksched.004A5AA8   ASCII "Registration Key accepted!"
004A5850  . E8 A339F9FF   CALL Teksched.004391F8
004A5855  > A1 F4535A00   MOV EAX,DWORD PTR DS:[5A53F4]
004A585A  . C600 00       MOV BYTE PTR DS:[EAX],0
004A585D  .vEB 17         JMP SHORT Teksched.004A5876
004A585F  > 807D 08 00    CMP BYTE PTR SS:[EBP+8],0
```

내려가다 보면 아까와 같이 등록 성공/실패 메시지를 호출하는 부분으로 이동하는 코드를 볼 수 있다.

여태까지 패치한 결과를 보아 아까는 오류메시지로 분기했지만 이번에는 등록 성공 메시지로 분기하는걸 볼 수 있다.



패치한 프로그램을 살펴보면 실행하자마자 바로 원본 버전으로 실행하는걸 볼 수 있다.