

12.Guiding a program by multiple patching

2012년 1월 29일 일요일

오후 7:48

Hello everybody.

모두들 안녕.

Welcome to this Part 12 in my series about reversing for newbies/beginners.

나의 초보자 reversing series Part 12에 온 것을 환영해.

This "saga" is intended for complete starters in reversing, also for those without any programming experience at all.

이 "saga"는 완벽히 reversing 초보자를 맞춰서 만들어졌다. 또한 어떠한 programming 경험이 없어도 된다.

Lena151 (2006)

Set your screen resolution to 1152*864 and press F11 to see the movie full screen !!!

Again, I have made this movie interactive.

You screen 해상도를 1152*864로 설정해 그리고 full screen으로 movie를 보기 위해 F11를 눌러

So, if you are a fast reader and you want to continue to the next screen, just click here on this invisible hotspot. You don't see it, but it IS there on text screens.

그래서, 네가 이것을 빨리 읽고 다음 screen을 보고 싶다면, 보이는 hotspot 여기를 눌러. 보고 싶지 않을 때는 여기에 두지마.

Then the movie will skip the text and continue with the next screen.

Movie는 text와 다음 screen을 skip 할 수 있다.

If something is not clear or goes too fast, you can always use the control buttons and the slider below on this screen.

무언가 명확하지 않거나 빨리 넘기고자 할 때, 항상 control button과 이 screen 밑에 있는 slider 바를 사용해.

He, try it out and click on the hotspot to skip this text and to go to the next screen now!!!

도전해봐. 그리고 이 text와 다음 screen을 보기 위해 hotspot을 click 해.

1. Abstract

In this Part 12, we will reverse a "real" application to learn something about multiple patching a program to guide it to success.

이 Part 12에서, 우리는 multiple patching에 대하여 guide하고 success 배우기 위하여 "real" application을 reverse 할 것이다.

This is because indeed, the best practice is found in real applications.

Real application에서 가장 좋은 연습을 찾았다.

In previous parts, we have only talked about easy and simple patching.

이전 parts에서, 우리는 쉽고 간단한 patching을 이야기 했다.

I hope you are ready for some "a little more difficult" patching.

네가 "a little more difficult" patching에 대하여 준비가 됐다고 희망한다.

For better comprehension and if you are a newbie, I advise you to first see the previous part in this series before seeing this movie.

좀 더 좋게 이해하고 네가 초보자라면, 너에게 이 movie를 보기 전에 이 series의 이전 part를 먼저 보라고 조언을 한다.

The goal of tutorial is to teach you something about a program's behaviour.

이 tutorial의 목표는 너에게 이 program's behaviour를 가르치는 것이다.

In my search not to harm authors, I found an old version of Techscheduler (version 6.01) which is no longer available for download.

나의 연구는 제작자에게 해가 되지 않는다. 나는 Techscheduler의 old version을(version 6.01) 찾았다. 그것은 더 이상 download되지 않는다.

However, because it could be misused, I only included the main executable (not the install exe !) for your research in the shown techniques.

그러나, 우리에게 잘못 알려준다. 나는 오직 너의 연구를 위해 이 보여주는 technique에서 main executable만 포함했다.(install exe는 없다)

This makes the program useless for "wrong" purposes. Talking a look in the specialized media, I also found this application to be "cracked" already.

나는 program을 "wrong" 목적으로 사용하지 않는다. 특화된 media를 가져, 나는 이미 이 application의 "cracked" version을 찾았다.

Here, this application is only chosen because it is ideal for this tutorial in reversing and it is targeted for educational purposes only.

여기, 이 application은 오직 선택됐다. 왜냐하면 이 reversing tutorial에 이상적이고 target은 오직 교육적인 목적이 있다.

I hope you will exploit your newly acquired knowledge in a positive way. In this matter, I also want to refer to Part 1.

네가 얻은 새로운 지식을 긍정적인 방향으로 이용하길 희망한다. 문제가 있다면, Part1를 참고하기 바란다.

2. Tools and Target

이것도 똑같음

The tools for today are : Ollydebug and... your brain.

오늘의 tools은 : Ollydebug와 너의 두뇌다.

The first can be obtained for free at

먼저 무료로 얻을 수 있다.

<http://www.ollydbg.de>

Again, the brain is your responsibility ;)

다시 한 번 말하지만, 너의 두뇌는 너의 책임감이다 ;)

The target is a program called Techscheduler 6.01, I have included the main-exe-only in this package for research.

Target program 은 Techscheduler 6.01 로 불린다. 나는 오직 너의 연구를 위해 이 package 에 main-exe 만 포함했다.

3. Behaviour of the program

Because you know meanwhile about the importance of decent study of your target and because you've seen how to do it in my previous Parts in this series,

너의 target 의 적절한 공부는 중요하다. 그리고 너는 이 series 이전 part 에서 어떻게 하는지 봐왔다.

I will only show you the things we will need to take care of while searching for the patches.

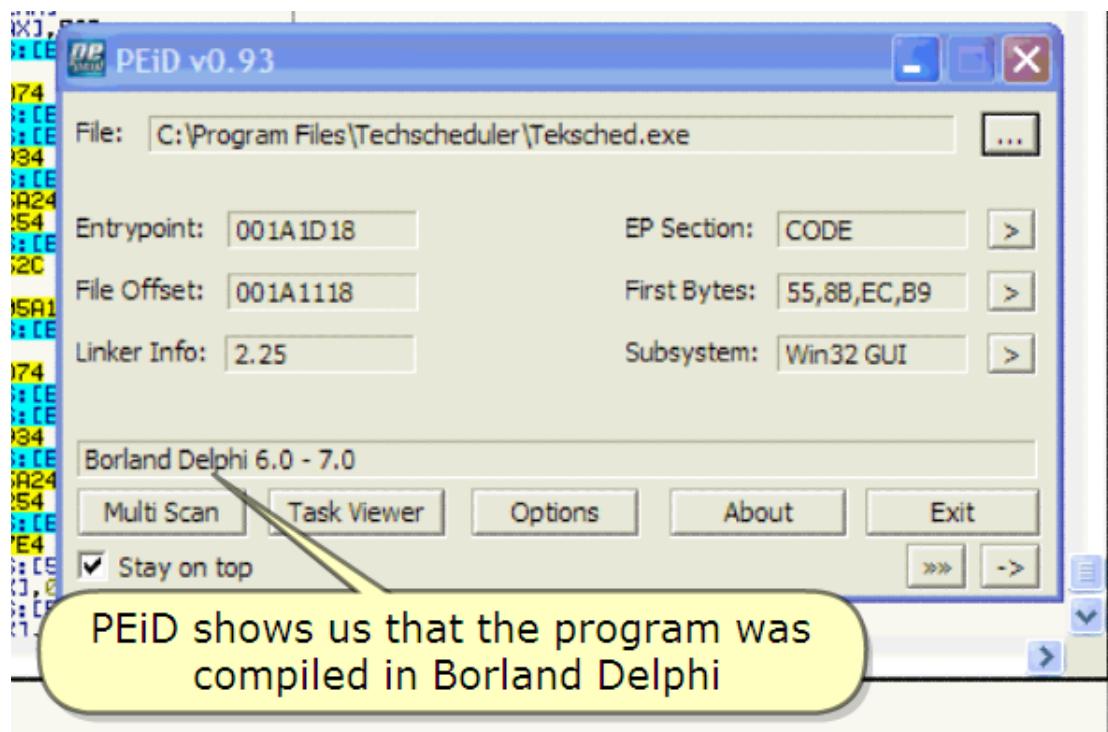
오직 너에게 Patche 를 위해 찾는 동안 우리는 돌봐 줄 것들을 필요한 것들을 보여주겠다.

Please do some preliminary study of the target on your own.

자신의 목표 중 일부 예비 연구를 수행해.

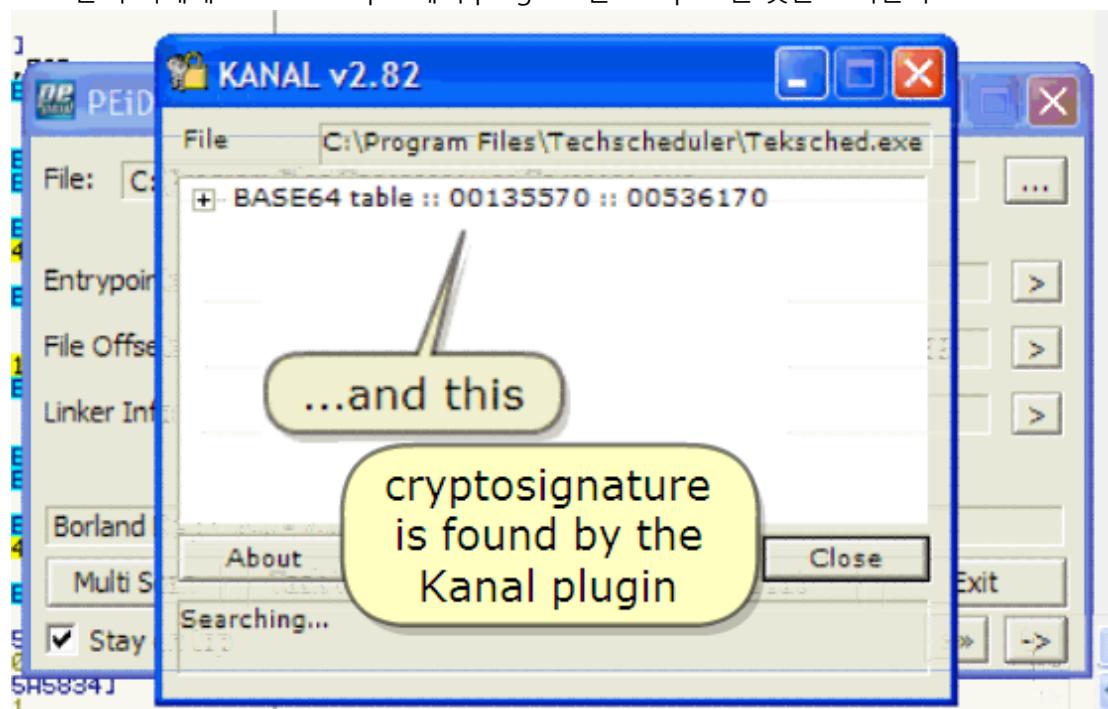
As you can see, I have already opened the target program in PEiD.

네가 볼 때, 나는 PEiD 에 이미 target program 을 열어놨다.



PEiD shows us that the program was compiled in Borland Delphi

PEiD 는 우리에게 Borland Delphi 에서 program 은 compile 된 것을 보여준다.



...and this cryptosignature is found by the Kanal plugin

그리고 Kanal plugin 에 의하여 cryptosignature 를 찾았다.

We can close PEiD now.

이제 우리는 PEiD 를 닫을 수 있다.

C CPU - main thread, module Teksched

```

005A1D18    55      PUSH EBP
005A1D19    8BEC    MOU EBP, ESP
005A1D1B    .E8 4F000000 MOU ECX, 4F
005A1D20    > 6A 00    PUSH 0
005A1D22    .E8 00    MOU ECX
005A1D24    .E8 00    MOU ECX
005A1D25    ^ 75 F9    JE SHORT Teksched.005A1D28
005A1D27    .53      PUSH ECX
005A1D28    .56      POP ECX
005A1D29    .57      POP ECX
005A1D2A    .E8 98155A00 CALL Teksched.00400934
005A1D2B    .E8 3856E6FF MOU ECX, ECX
005A1D2C    .33C0    XOR ECX, ECX
005A1D34    .55      PUSH EBP
005A1D36    .68 75245A00 MOV EDX, DWORD PTR FS:[ERAX], EBP
005A1D37    .68 00      MOU EDX, DWORD PTR SS:[EBP-18]
005A1D3C    .64:FF30    XOR EDX, EDX
005A1D3F    .64:FF20    LEA EDX, DWORD PTR SS:[EBP-18]
005A1D40    .64:FF20    XOR EDX, EDX
005A1D42    .64:FF20    LEA EDX, DWORD PTR SS:[EBP-14]
005A1D44    .64:FF20    XOR EDX, EDX
005A1D47    .68 2010E6FF CALL Teksched.00400D74
005A1D52    .6845 E8    MOV EDX, DWORD PTR SS:[EBP-18]
005A1D54    .6845 EC    LEA EDX, DWORD PTR SS:[EBP-14]
005A1D55    .6845 EC    MOU EDX, Teksched.005A248C
005A1D56    .68 8C245A00 CALL Teksched.00400934
005A1D57    .6845 EC    LEA EDX, DWORD PTR SS:[EBP-14]
005A1D58    .68 00      MOU EDX, DWORD PTR SS:[EBP-14]
005A1D59    .6845 EC    CALL Teksched.00400524
005A1D64    .6845 EC    LEA EDX, DWORD PTR SS:[EBP-14]
005A1D67    .68 00      MOU EDX, DWORD PTR SS:[EBP-14]
005A1D6C    .6840    CALL Teksched.00400934
005A1D6E    .74 2A    TEST AL, AL
005A1D70    .6855 E0    JE SHORT Teksched.005A1D9A
005A1D73    .33C0    LEA EDX, DWORD PTR SS:[EBP-20]
005A1D74    .68 A00FE6FF XOR EDX, EDX
005A1D77    .6845 E0    CALL Teksched.00400D74
005A1D79    .6855 E4    MOV EDX, DWORD PTR SS:[EBP-20]
005A1D80    .6855 E4    LEA EDX, DWORD PTR SS:[EBP-16]
005A1D83    .6845 E4    CALL Teksched.00400934
005A1D85    .6845 E4    LEA EDX, DWORD PTR SS:[EBP-16]
005A1D88    .68 8C245A00 MOU EDX, Teksched.005A248C
005A1D8D    .68 C234E6FF CALL Teksched.00400934
005A1D92    .6845 E4    MOU EDX, DWORD PTR SS:[EBP-16]
005A1D95    .68 4A8AE6FF CALL Teksched.004007E4
005A1D9A    > A1 74055A00 MOU EDX, DWORD PTR DS:[SASC74]
005A1D9F    .C600 00    MOU BYTE PTR DS:[ERAX], 0
005A1DA2    .A1 345SSA00 MOU EDX, DWORD PTR DS:[SAS5834]
005A1DA7    .FAAA A1    MOU RVTF PTR DS:[ERAX].1

```

The program has also been opened in Olly already
This is EP (Entry Point)

The program has also been opened in Olly already

이미 Olly에서 program을 열어놨다.

This is EP(Entry Point)

이것은 EP 다.(Entry Point)

번역 주) 프로그램은 실행 될 때, main 이나 WinMain, 혹은 DLLMain에서 시작하는데, 이 위치가 RVA로 저장되어 있다. 바로 이 위치를 EP(Entry Point)라고 한다.

RVA : Relative Virtual Address의 약자로서 Offset과 같은 개념이라고 봐도 무방하다.

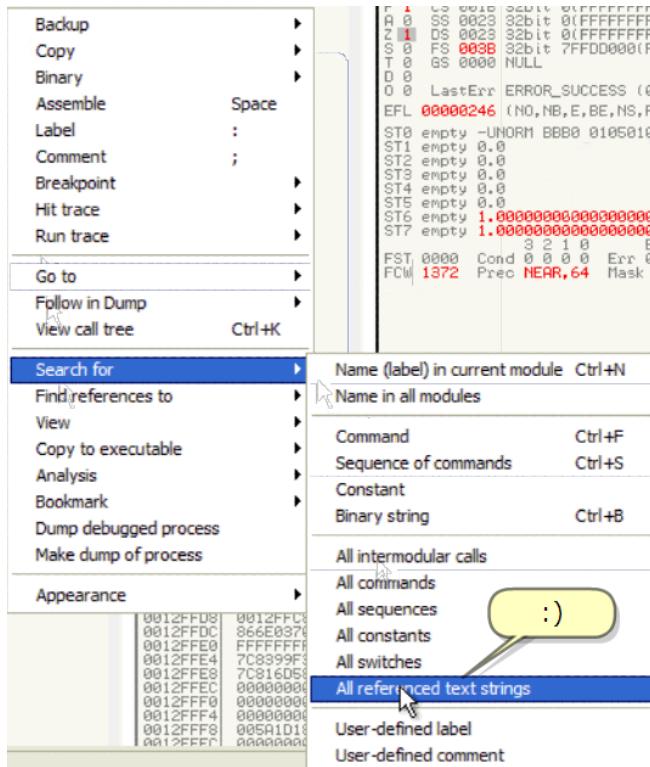
VA : Virtual Address의 약자로써, 실제 물리적 메모리 주소에 매핑한 가상 메모리 주소를 말한다.

INFO : In programs, there are some strings that are very often used for registration schemes that also can give good clues on how to proceed further. Just follow ...

Program에서는 그곳에 약간의 strings이 있다. Strings은 등록 scheme를 위해 매우 자주 사용된다.

Registration scheme는 우리에게 미래에 어떻게 진행할지 좋은 단서를 준다. 이제 따라가보자.

4. Finding the patches



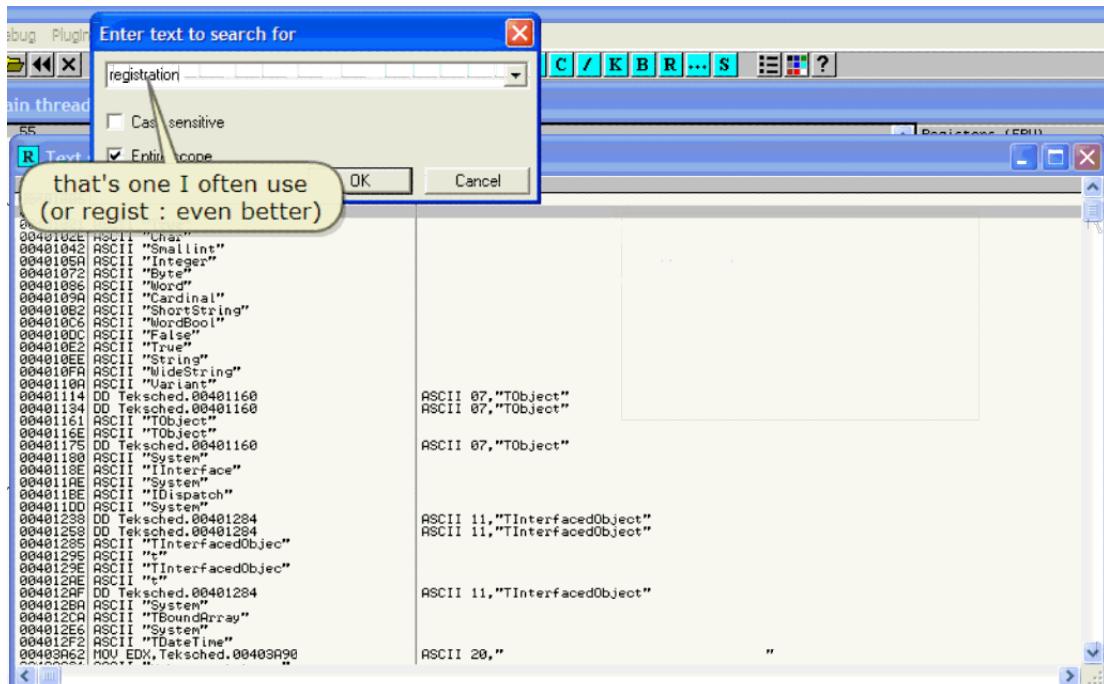
:)

R Text strings referenced in Teksched:CODE

Address	Disassembly	Text string
005A1014	ASCII "0",0	
005A1020	ASCII "sFreq0",0	
005A1029	ASCII "1900/01/01"	
005A103C	ASCII "SGlobalDateExclu"	
005A1045	ASCII "de",0	
005A1052	ASCII "sGlobalDates",0	
005A1060	ASCII "GlobalExclude",0	
005A1064	ASCII "Excel",0	
005A1074	ASCII "Debug", GLOBAL_d"	
005A1084	ASCII "date exclusion fs"	
005A1084	ASCII "on todays list..."	
005A10C4	ASCII 0	
005A10D0	ASCII "Debug, JOB SPEC"	
005A10E0	ASCII "TICK date exclus"	
005A10F0	ASCII "tion from todays "	
005A1100	ASCII "list",0	
005A1110	ASCII "sJobName",0	
005A1124	ASCII "lCustomTimeType",0	
005A1444	ASCII "",0	
005A1D18	PUSH EBP	(Initial CPU selection)
005A1D54	MOV EDX, Teksched_005A248C	ASCII "ts0000.bat"
005A1D5F	MOV EDX, Teksched_005A248C	ASCII "ts0000.bat"
005A1D6F	MOV EDX, Teksched_005A248C	ASCII "APJ"
005A1D80	MOU EDX, Teksched_005A2480	ASCII "Techscheduler Service"
005A1D89	MOU EDX, Teksched_005A2400	ASCII "Techscheduler"
005A1E2F	MOU ERX, Teksched_005A24E8	ASCII "APPID"
005A1E66	MOU ERX, Teksched_005A2504	ASCII "SILENT"
005A1EE4	MOU ERX, Teksched_005A2514	ASCII "DEBUG"
005A1F2C	MOU EDX, Teksched_005A2524	ASCII "teksched.inl"
005A1F64	MOU EDX, Teksched_005A2524	ASCII "teksched.inl"
005A1F70	MOU EDX, Teksched_005A2524	ASCII "ts0000.inl"
005A1F86	MOU EDX, Teksched_005A253C	ASCII "05","Admin"
005A1FCF	MOU ECX, Teksched_005A2544	ASCII 07,"bluseXML"
005A1FD4	MOU EDX, Teksched_005A253C	ASCII "05","Admin"
005A2019	MOU EDX, Teksched_005A2524	ASCII "teksched.inl"
005A2047	MOU ECX, Teksched_005A254C	ASCII 08,"\$InPath"
005A204C	MOU EDX, Teksched_005A253C	ASCII "05","Admin"
005A2098	MOU EDX, Teksched_005A2560	ASCII "teksched.xml"

Scroll up

번역 주)string 검색할 때는 제일 처음부터 검색하라고 했다. 기억이 나지 않는다면 이전 강의를 살펴보세요.



That's one I often use(or regist : even better)

이것은 자주 사용된다.(또는 regist : 이것보다 좋다)

Address	Disassembly	Text string
00444066	MOV ECX,Teksched.004A5110	ASCII "UserL"
00444072	MOV EDX,Teksched.004A50F0	ASCII "Config"
00444078	ASCII "Software\Dean So"	
00444084	ASCII "Teksched"	
00444090	ASCII "rules",0	
0044409C	ASCII "RegStar",0	
004440A8	ASCII "Config",0	
004440B4	ASCII "sUserF",0	
004440C0	ASCII "sUserL",0	
004440C6	MOV ECX,Teksched.004A58D8	ASCII "Software\Dean Software\\TechScheduler"
004440D2	MOV EDX,Teksched.004A58D8	ASCII "Enter a First Name value now."
004440D8	MOV EDX,Teksched.004A5238	ASCII "Enter a Last Name value now."
004440E4	MOV EDX,Teksched.004A5258	ASCII "Enter a Key value now.."
004440F9	MOV EDX,Teksched.004A5978	ASCII "Error You are attempting to register the wrong level of this program (st:)
00444105	MOV EDX,Teksched.004A59E4	ASCII "Error You are attempting to register the wrong version of this program"
00444111	MOV EDX,Teksched.004A5834	ASCII "SJU"
00444117	MOV EDX,Teksched.004A58D8	ASCII "Software\Dean Software\\TechScheduler"
00444167	MOV ECX,Teksched.004A5440	ASCII "sRegStat",0
00444173	MOV EDX,Teksched.004A5454	ASCII "Config"
00444179	MOV EDX,Teksched.004A5464	ASCII "Config"
00444185	MOV EDX,Teksched.004A5474	ASCII "Config",0
00444191	MOV EDX,Teksched.004A5854	ASCII "sUserL",0
00444197	MOV EDX,Teksched.004A5874	ASCII "Config"
004441A3	MOV EDX,Teksched.004A5854	ASCII "Config"
004441AB	MOV ECX,Teksched.004A5884	ASCII "sRegLevel"
004441C1	MOV EDX,Teksched.004A5854	ASCII "Config"
004441C7	MOV ECX,Teksched.004A5898	ASCII "sRegDef"
004441D3	MOV EDX,Teksched.004A5854	ASCII "Config",0
004441D9	MOV EDX,Teksched.004A5898	ASCII "sRegfacedObject",0
004441E5	MOV EDX,Teksched.004A58C0	ASCII "Registration Key accepted!"
004441E9	ASCII "Software\Dean So"	ASCII "Registration Key Failed!"
004441F5	ASCII "Teksched"	
004441FA	ASCII "rules",0	
004441FB	ASCII "Enter",0	
004441FC	ASCII "meva",0	
004441FD	ASCII "Enter",0	
004441FE	ASCII "e val",0	
004441FF	ASCII "Enter",0	

Mmmmm, we land in the right spot
:)

Mmmmm, we land in the right spot :)

음, 우리는 좋은 지점에 도착했다 :)

R Text strings referenced in Teksched:CODE

Address	Disassembly	Text string
004A406E	MOV ECX,Teksched.004A5110	ASCII "sUserL"
004A4C73	MOV EDX,Teksched.004A50F0	ASCII "Config"
004A50AC	ASCII "Software\Dean So"	
004A50B0	ASCII "Software\TechSched"	
004A50D0	ASCII "#RegStat",0	
004A50F0	ASCII "Config",0	
004A5100	ASCII "sUser",0	
004A5110	ASCII "sUserL",0	
004A5189	MOV ECX,Teksched.004A5808	ASCII "Software\Dean Software\TechScheduler"
004A52CA	MOV EDX,Teksched.004A5908	ASCII "Enter a First Name value now.."
004A52B5	MOV EDX,Teksched.004A5930	ASCII "Enter a Last Name value now.."
004A52C0	MOV EDX,Teksched.004A5958	ASCII "Enter a Key value now.."
004A55F9	MOV EDX,Teksched.004A5808	ASCII "Error! You are attempting to register the wrong level of this program (st: 004A561D)
004A561D	MOV EDX,Teksched.004A5808	ASCII "Error! You are attempting to register the wrong version of this program"
004A5687	MOV EDX,Teksched.004A5808	ASCII "GJ!"
004A5728	MOV ECX,Teksched.004A5840	ASCII "#RegStat",0
004A572F	MOV EDX,Teksched.004A5844	ASCII "Config"
004A5762	MOV ECX,Teksched.004A5864	ASCII "sUserF"
004A5805	MOV ECX,Teksched.004A5864	ASCII "Config",0
004A5820	MOV ECX,Teksched.004A5864	ASCII "sRegDef"
004A582F	MOV EDX,Teksched.004A5864	ASCII "Config\perfacedObject"
004A5845	MOV EDX,Teksched.004A5848	ASCII "Registration Key accepted!"
004A5850	MOV EDX,Teksched.004A58CC	ASCII "Registration Key Failed!"
004A5859	ASCII "Software\Dean So"	
004A5868	ASCII "Software\TechSched"	
004A5878	ASCII "User",0	
004A5908	ASCII "Enter a First Na"	
004A5918	ASCII "me value now..",0	
004A5930	ASCII "Enter a Last Na"	
004A5940	ASCII "@ value now..",0	
004A5958	ASCII "Enter a Key valu"	
004A5968	ASCII "me now..",0	

All important stuff

모두 중요한 재료

R Text strings referenced in Teksched:CODE

Address	Disassembly	Text string
004A406E	MOV ECX,Teksched.004A5110	ASCII "sUserL"
004A4C73	MOV EDX,Teksched.004A50F0	ASCII "Config"
004A50AC	ASCII "Software\Dean So"	
004A50B0	ASCII "Software\TechSched"	
004A50D0	ASCII "#RegStat",0	
004A50F0	ASCII "Config",0	
004A5100	ASCII "sUser",0	
004A5110	ASCII "sUserL",0	
004A5189	MOV ECX,Teksched.004A5808	ASCII "Software\Dean Software\TechScheduler"
004A52CA	MOV EDX,Teksched.004A5908	ASCII "Enter a First Name value now.."
004A52B5	MOV EDX,Teksched.004A5930	ASCII "Enter a Last Name value now.."
004A52C0	MOV EDX,Teksched.004A5958	ASCII "Enter a Key value now.."
004A55F9	MOV EDX,Teksched.004A5787	ASCII "Error! You are attempting to register the wrong level of this program (st: 004A561D)
004A561D	MOV EDX,Teksched.004A5808	ASCII "Error! You are attempting to register the wrong version of this program"
004A5687	MOV EDX,Teksched.004A5808	ASCII "GJ!"
004A5728	MOV ECX,Teksched.004A5808	ASCII "#RegStat",0
004A572F	MOV EDX,Teksched.004A5844	ASCII "Config"
004A5762	MOV ECX,Teksched.004A5864	ASCII "sUserF"
004A5805	MOV ECX,Teksched.004A5864	ASCII "Config",0
004A5820	MOV ECX,Teksched.004A5864	ASCII "sRegDef"
004A582F	MOV EDX,Teksched.004A5864	ASCII "Config\perfacedObject"
004A5845	MOV EDX,Teksched.004A5848	ASCII "Registration Key accepted!"
004A5850	MOV EDX,Teksched.004A58CC	ASCII "Registration Key Failed!"
004A5859	ASCII "Software\Dean So"	
004A5868	ASCII "Software\TechSched"	
004A5878	ASCII "User",0	
004A5908	ASCII "Enter a First Na"	
004A5918	ASCII "me value now..",0	
004A5930	ASCII "Enter a Last Na"	
004A5940	ASCII "@ value now..",0	
004A5958	ASCII "Enter a Key valu"	
004A5968	ASCII "me now..",0	

All important stuff

All important stuff

모두 중요한 재료

Address	Disassembly	Text string
0044406E	MOU ECX, Teksched.004A5110	ASCII "sUserL"
00444C73	MOU EDX, Teksched.004A50F0	ASCII "Config"
004450AC	ASCII "Software\Dean So"	
004450B0	ASCII "Software\TechScheduler"	
004450B4	ASCII "EnterFirstD	
004450D0	ASCII "EnterLastD	
004450F0	ASCII "Config", 0	
00445100	ASCII "msUserF", 0	
00445110	ASCII "sUserL", 0	
00445189	MOU ECX, Teksched.004A50D8	ASCII "Software\Dean Software\TechScheduler"
004452C8	MOU EDX, Teksched.004A50B8	ASCII "Enter a First Name value now.."
004453B5	MOU EDX, Teksched.004A5030	ASCII "Enter a Last Name value now.."
004453D9	MOU EDX, Teksched.004A5058	ASCII "Enter a Key value now.."
004453E3	MOU EDX, Teksched.004A5078	ASCII "Error! You are attempting to register the wrong level of this program (st: 00445610)"
00445610	MOU EDX, Teksched.004A50E4	ASCII "Error! You are attempting to register the wrong version of this program"
00445687	MOU EDX, Teksched.004A5034	ASCII "SJJ"
004456E2	MOU ECX, Teksched.004A5008	ASCII "Software\Dean Software\TechScheduler"
00445728	MOU ECX, Teksched.004A5040	ASCII "sRegStat", 0
0044572F	MOU EDX, Teksched.004A5054	ASCII "sRegDef", 0
00445730	MOU EDX, Teksched.004A5064	ASCII "Config", 0
00445734	MOU EDX, Teksched.004A5064	ASCII "sUserF", 0
00445738	MOU EDX, Teksched.004A5064	ASCII "Config", 0
0044573C	MOU EDX, Teksched.004A5064	ASCII "sUserL", 0
0044573E	MOU EDX, Teksched.004A5064	ASCII "sRegLevel", 0
0044573F	MOU EDX, Teksched.004A5064	ASCII "Config", 0
00445800	MOU ECX, Teksched.004A5098	ASCII "sRegDef", 0
00445804	MOU EDX, Teksched.004A5054	ASCII "Config", 0
00445808	MOU EDX, Teksched.004A5054	ASCII "Config", 0
0044580F	MOU EDX, Teksched.004A5054	ASCII "Config", 0
00445848	MOU EDX, Teksched.004A50A8	ASCII "Registration Key accepted!"
00445852	MOU EDX, Teksched.004A50C0	ASCII "Registration Key Failed!"
00445856	MOU ECX, Teksched.004A50C0	ASCII "Software\Dean So"
00445858	MOU ECX, Teksched.004A50C0	ASCII "Software\TechScheduler"
00445859	MOU ECX, Teksched.004A50C0	ASCII "sUser", 0
0044585B	MOU ECX, Teksched.004A50C0	ASCII "Enter a First Name value now..", 0
0044585D	MOU ECX, Teksched.004A50C0	ASCII "Enter a Last Name value now..", 0
0044585F	MOU ECX, Teksched.004A50C0	ASCII "Enter a Key value now..", 0
00445861	MOU ECX, Teksched.004A50C0	ASCII "Enter now..", 0

Let's go there

여기로 가자.

CPU - main thread, module Teksched	
00445003	. 8B85 8CFEFFF MOU EAX, DWORD PTR SS:[EBP-174]
00445009	. 50 PUSH EDX
0044500A	. B9 985B4A00 MOU ECX, Teksched.004A5A98
0044500F	. B8 545B4A00 MOU EDX, Teksched.004A5A54
00445014	. 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445017	. E8 3414FFFF CALL Teksched.004A96C50
0044501C	. 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
0044501F	. E8 2000FFFF CALL Teksched.004A50A4
00445024	. 33C0 XOR EAX, EAX
00445026	. 5A POP EDX
00445027	. 59 POP ECX
0044502C	. 59 POP ECX
0044502D	. 64:8910 MOU DWORD PTR FS:[EAX], EDX
0044502E	. 68 41584A00 PUSH Teksched.004A50A4
00445031	> 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445034	. E8 5BE8F5FF CALL Teksched.004A6094
00445039	. 58 RET
0044503A	. ^ E9 E9EFF5FF JMP Teksched.004A04828
0044503F	. ^ EB F0 JMP SHORT Teksched.004A50301
00445041	. C645 F3 01 MOU BYTE PTR SS:[EBP-01], 1
00445045	. 8070 00 00 CMP BYTE PTR SS:[EBP+01], 0
00445049	. > 75 0A JNC SHORT Teksched.004A5055
0044504B	. B8 885A4A00 MOU EAX, Teksched.004A5A08
00445050	. E8 A339F9FF CALL Teksched.004A391F8
00445053	> A1 F435A00 MOU EAX, DWORD PTR DS:[5A5F4]
00445056	. C600 00 MOU BYTE PTR DS:[EAX], 0
0044505D	. > EB 17 JMP SHORT Teksched.004A5074
0044505F	. 8070 00
00445063	. > 75 11 JNC SHORT Teksched.004A5074
00445065	. 6A 30 XOR EAX, EAX
00445067	. E8 C829F0FF CALL Teksched.004A50A4
0044506C	. B8 CC504A00 MOU EAX, Teksched.004A5A08
00445071	. E8 8239F0FF CALL Teksched.004A391F8
00445076	. 33C0 XOR EAX, EAX
00445078	. 5A POP EDX
00445079	. 59 POP ECX
0044507C	. 59 POP ECX
0044507D	. 64:8910 MOU DWORD PTR FS:[EAX], EDX
0044507E	. 68 41584A00 PUSH Teksched.004A50A4
00445085	. E8 8CFFEF7 MOU EDX, DWORD PTR SS:[EBP-174]
00445089	. E8 8C000000 MOU EDX, 0C
0044508D	. E8 1DF7FF FF CALL Teksched.004A04FB0
00445093	. 8D45 BC LEA EAX, DWORD PTR SS:[EBP-441]
00445099	. B8 04000000 MOU EDX, 4
0044509B	. F8 1AF7FF FF CALL Teksched.004A04FB0

004450A8=Teksched.004A50A8 (ASCII "Registration Key accepted!")

EAX=00000000

Looks good, doesn't it ???

좋아, 그렇지 ???

C CPU - main thread, module Teksched

```

00445803 . 8B85 8CFFFF MOU EAX, DWORD PTR SS:[EBP-174]
00445809 . 50 PUSH EAX
0044580A . B9 98584A00 MOU ECX, Teksched.00445A98
0044580F . BA E4584A00 MOU EDX, Teksched.00445A94
00445814 . 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445817 . E8 3414FFFF CALL Teksched.00445C58
0044581C . 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
0044581F . E8 2000FFFF CALL Teksched.00445844
00445824 . 33C0 XOR EAX, EAX
00445826 . 5A POP EDX
00445827 . 59 POP ECX
00445828 . 59 POP ECX
00445829 . 64:8910 MOU DWORD PTR FS:[EAX], EDX
0044582C . 68 41584A00 PUSH Teksched.00445841
00445830 . > 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445834 . E8 5BE8FF FF CALL Teksched.004404094
00445839 . C3 RETN
0044583D . ^ E9 E9EFF5FF JMP Tekshed.004404828
0044583E . ^ EB F0 JMP SHORT Tekshed.00445831
00445841 . C645 F3 01 MOU BYTE PTR SS:[EBP-0], 1
00445845 . 807D 00 00 CMP BYTE PTR SS:[EBP-8], 0
00445849 . ✓ 75 OA JNZ SHORT Tekshed.00445855
0044584B . BB A8584A00 MOU EAX, Teksched.00445A98
00445850 . E8 A339F9FF CALL Teksched.0044391F8
00445853 . > A1 F4535A00 MOU EAX, DWORD PTR DS:[5A53F4]
00445856 . C600 00 MOU BYTE PTR DS:[EAX], 0
00445857 . ^ EB 17 JMP SHORT Tekshed.00445876
00445858 . 807D 00 00 CMP BYTE PTR SS:[EBP+8], 0
00445859 . ✓ 75 11 JNZ SHORT Tekshed.00445876
00445865 . 64:8910 MOU DWORD PTR FS:[EAX], EDX
00445867 . 68 C2584A00 PUSH Teksched.004458C2
00445869 . > 8D85 8CFFFF LEA EAX, DWORD PTR SS:[EBP-174]
00445870 . BA 0C000000 MOU EDX, 0C
0044587E . E8 10F7FF FF CALL Teksched.004404FB0
00445882 . 8D45 BC LEA EAX, DWORD PTR SS:[EBP-44]
00445892 . BA 04000000 MOU EDX, 4
0044589R . FR 10F7FFFF CALL Teksched.004404FRA

```

00445A88=Teksched.00445A88 (ASCII "Registration Key accepted!")
EXX=00000000

ASCII "Registration Key accepted!"

So, we have to make sure we get here !!!

BeepType = MB_ICONEXCLAMATION
MessageBeep
ASCII "Registration Key Failed!"

kernel32!7C816D4F
kernel32!7C816D4F
kernel32!7C816D4F
ntdll!KiFastSystemCallRet

So, we have to make sure we get here !!!

그래서, 우리는 여기를 얻을 수 있게 만들어야 한다. !!!

C CPU - main thread, module Teksched

```

00445803 . 8B85 8CFFFF MOU EAX, DWORD PTR SS:[EBP-174]
00445809 . 50 PUSH EAX
0044580A . B9 98584A00 MOU ECX, Teksched.00445A98
0044580F . BA E4584A00 MOU EDX, Teksched.00445A94
00445814 . 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445817 . E8 3414FFFF CALL Teksched.00445C58
0044581C . 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
0044581F . E8 2000FFFF CALL Teksched.00445844
00445824 . 33C0 XOR EAX, EAX
00445826 . 5A POP EDX
00445827 . 59 POP ECX
00445828 . 59 POP ECX
00445829 . 64:8910 MOU DWORD PTR FS:[EAX], EDX
0044582C . 68 41584A00 PUSH Teksched.00445841
00445830 . > 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445834 . E8 5BE8FF FF CALL Teksched.004404094
00445839 . C3 RETN
0044583D . ^ E9 E9EFF5FF JMP Tekshed.004404828
0044583E . ^ EB F0 JMP SHORT Tekshed.00445831
00445841 . C645 F3 01 MOU BYTE PTR SS:[EBP-0], 1
00445845 . 807D 00 00 CMP BYTE PTR SS:[EBP-8], 0
00445849 . ✓ 75 OA JNZ SHORT Tekshed.00445855
0044584B . BB A8584A00 MOU EAX, Teksched.00445A98
00445850 . E8 A339F9FF CALL Teksched.0044391F8
00445853 . > A1 F4535A00 MOU EAX, DWORD PTR DS:[5A53F4]
00445856 . C600 00 MOU BYTE PTR DS:[EAX], 0
00445857 . ^ EB 17 JMP SHORT Tekshed.00445876
00445858 . 807D 00 00 CMP BYTE PTR SS:[EBP+8], 0
00445859 . ✓ 75 11 JNZ SHORT Tekshed.00445876
00445865 . 64:8910 MOU DWORD PTR FS:[EAX], EDX
00445867 . 68 C2584A00 PUSH Teksched.004458C2
00445869 . > 8D85 8CFFFF LEA EAX, DWORD PTR SS:[EBP-174]
00445870 . BA 0C000000 MOU EDX, 0C
0044587E . E8 10F7FF FF CALL Teksched.004404FB0
00445882 . 8D45 BC LEA EAX, DWORD PTR SS:[EBP-44]
00445892 . BA 04000000 MOU EDX, 4
0044589R . FR 10F7FFFF CALL Teksched.004404FRA

```

00445A88=Teksched.00445A88 (ASCII "Registration Key accepted!")
EXX=00000000

ASCII "Registration Key accepted!"

And not here ... LOL

BeepType = MB_ICONEXCLAMATION
MessageBeep
ASCII "Registration Key Failed!"

kernel32!7C816D4F
kernel32!7C816D4F
kernel32!7C816D4F
ntdll!KiFastSystemCallRet

And not here ... LOL

여기는 오지 않아야 한다.

C CPU - main thread, module Teksched

```

00445803 . 8B85 8CFEFFFF MOU EAX, DWORD PTR SS:[EBP-174]
00445809 . 50 PUSH EAX
0044580A . B9 98544A00 MOU ECX, Teksched.004A5A98
0044580F . BA E4564A00 MOU EDX, Teksched.004A5A4
00445814 . 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445817 . E8 3414FF FF CALL Teksched.004A5C50
0044581C . 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445824 . E8 2000FFFF CALL Teksched.00495844
00445826 . 33C0 XOR EAX, EAX
00445826
00445827 Now, remember that any jump past here is
00445828 99% of the chances a wrong jump
00445829
00445831
00445834 . E8 5BE8F5FF JMP Teksched.004A4094
00445839 . C3 RETN
0044583F ^ EB F0 JMP Teksched.00404828
00445841 . C645 CC SHORT Teksched.004A5831
00445845 . S07D 00 00 MOU BYTE PTR SS:[EBP-01], 0
00445849 . C645 CC CMP BYTE PTR SS:[EBP-01], 0
00445852 . 75 0A JZ SHORT Teksched.004A5855
00445853 . B8 00000000 MOU EAX, Tekshed.004A5A98
00445855 . E8 A33F9FF CALL Teksched.004391F8
00445856 > A1 F4533A00 MOU EDX, PTR DS:[5A53F4]
00445858 C600
00445859 > EB 00 POP ECX
0044585F > EB 00 POP ECX
00445863 > EB 00 POP ECX
00445865 > EB 00 POP ECX
00445867 . E8 8239F9FF CALL Teksched.004A4094
00445868 . 33C0 XOR EAX, EAX
00445876 . 5A PUSH EDX
00445879 . 59 POP ECX
0044587A . 59 POP ECX
0044587B . 64:8910 MOU DWORD PTR FS:[EAX], EDX
0044587E . 68 C2584A00 PUSH Teksched.004A58C2
00445883 . 80D8 8CFEFFFF LEA EAX, DWORD PTR SS:[EBP-174]
00445889 . BA 0C000000 MOU EDX, 0C
0044588E . E8 1DF7F5FF CALL Teksched.00404FB0
00445893 . 8045 BC LEA EAX, DWORD PTR SS:[EBP-44]
00445896 . BA 04000000 MOU EDX, 4
0044589R . FA 1AF7FFFF CALL Teksched.004A40F8A

```

32.7C816D4F
32.7C816D4F
32.7C816D4F
FastSystemCallRet

ASCII "Registration Key accepted!"

BeepType = MB_ICONEXCLAMATION
MessageBeep
ASCII "Registration Key Failed!"

kernel32.7C816D4F
kernel32.7C816D4F
kernel32.7C816D4F
ntdll.KiFastSystemCallRet

00445AA8=Teksched.004A5A8 (ASCII "Registration Key accepted!")
EAX=00000000

Now, remember that any jump past here is 99% of the chances a wrong jump

So, note down the VA 4A584B for future reference.

이제, 기억해. 여기를 지나가는 모든 jump 는 99% 잘못된 jump 다.

그래서, 미래의 참조를 위해 VA 4A584B 를 적어놓자.

C CPU - main thread, module Teksched

```

00445803 . 8B85 8CFEFFFF MOU EAX, DWORD PTR SS:[EBP-174]
00445809 . 50 PUSH EAX
0044580A . B9 98544A00 MOU ECX, Teksched.004A5A98
0044580F . BA E4564A00 MOU EDX, Teksched.004A5A4
00445814 . 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445817 . E8 3414FF FF CALL Teksched.004A5C50
0044581C . 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
00445824 . E8 00000000 CALL Teksched.00495844
00445826 . 33C0 XOR EAX, EAX
00445826
00445827 Here is
00445828 a return !
00445829
00445831
00445834 . E8 5BE8F5FF JMP Teksched.004A4094
00445839 . C3 RETN
00445841 . ^ EB F0 JMP Teksched.00404828
00445845 . C645 CC SHORT Teksched.004A5831
00445849 . S07D 00 00 MOU BYTE PTR SS:[EBP-01], 0
00445852 . C645 CC CMP BYTE PTR SS:[EBP-01], 0
00445853 . 75 0A JZ SHORT Teksched.004A5855
00445854 . B8 A885A00 MOU EAX, Tekshed.004A5A98
00445855 . E8 A33F9FF CALL Teksched.004391F8
00445856 > A1 F4533A00 MOU EDX, PTR DS:[5A53F4]
00445858 C600
00445859 > EB 00 POP ECX
0044585F > EB 00 POP ECX
00445863 > EB 00 POP ECX
00445865 > EB 00 POP ECX
00445867 . E8 8239F9FF CALL Teksched.004A4094
00445868 . 33C0 XOR EAX, EAX
00445876 . 5A PUSH EDX
00445879 . 59 POP ECX
0044587A . 59 POP ECX
0044587B . 64:8910 MOU DWORD PTR FS:[EAX], EDX
0044587E . 68 C2584A00 PUSH Teksched.004A58C2
00445883 . 80D8 8CFEFFFF LEA EAX, DWORD PTR SS:[EBP-174]
00445889 . BA 0C000000 MOU EDX, 0C
0044588E . E8 1DF7F5FF CALL Teksched.00404FB0
00445893 . 8045 BC LEA EAX, DWORD PTR SS:[EBP-44]
00445896 . BA 04000000 MOU EDX, 4
0044589R . FA 1AF7FFFF CALL Teksched.004A40F8A

```

Followed by jumps up again

Right. Now, how do we get here ????

and we see that all arriving jumps go past this code !!!

And we want the code to run this goodboy

32.7C816D4F
32.7C816D4F
32.7C816D4F
FastSystemCallRet

ASCII "Registration Key accepted!"

BeepType = MB_ICONEXCLAMATION
MessageBeep
ASCII "Registration Key Failed!"

kernel32.7C816D4F
kernel32.7C816D4F
kernel32.7C816D4F
ntdll.KiFastSystemCallRet

00445AA8=Teksched.004A5A8 (ASCII "Registration Key accepted!")
EAX=00000000

Right. Now, how do we get here ????

좋아. 이제, 여기를 어떻게 얻겠어 ????

Here is a return !

이곳은 return!

Followed by jumps up again

다시 Jump 에 의해 역행한다.

And we want the code to run this goodboy

이 goodboy 로 실행하기 위한 code 를 원한다.

And we see that all arriving jumps go past this code !!!

그리고 모든 도착하는 jumps 는 이 code 를 지나가는 걸 봐.

2012.02.04 03:00~

Mmmm, no jumps to this piece of code ...

음, 이 code 조각으로 jump 하는 것은 없다.

However, there are other ways of arriving here.

그러나, 여기에 도착하는 다른 방법이 많다.

A call for example could do the trick or a jump[register+something] would make it hard for Olly
to trace how we got here

Call 을 trick 하거나 Olly 가 여기에 어떻게 오는지 추적을 대비해 jump[register+something]을
어렵게 만든다.

Another important remark is the following possibility

또 다른 중요한 발언은 따라갈 가능성이다.

A push followed by a return == a (unconditional) jump

Return 에 뒤따라 PUSH 를 한다 == a (unconditional) jump

Anyway, see how we can search for it

게다가, 이것을 어떻게 찾는지 보자.

C CPU - main thread, module Teksched

```

004455203 . 8B85 8CFEFFFI MOU EAX, DWORD PTR SS:[EBP-174]
004455204 . 58 PUSH EAX
004455205 . B8 88544000 MOU ECX, Teksched.004A5A98
004455206 . BA 54544000 MOU EDX, Teksched.004A5B54
004455207 . EB45 CC MOU ERA, DWORD PTR SS:[EBP-34]
004455208 . E8 3414FFFF CALL Teksched.00495C58
004455209 . B845 CC MOU EAX, DWORD PTR SS:[EBP-34]
004455210 . E8 2000FFFF CALL Teksched.00495B44
004455211 . 33C0 XOR EAX, EAX
004455212 . C3 RETN
004455213 . E8 4155440000 PUSH Teksched.004A5841
004455214 > 0B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
004455215 . E8 SBEE8H FF CALL Teksched.004A04094
004455216 . C3 RETN
004455217 . ^ E9 E9EF5FF JMP Teksched.004A4828
004455218 . ^ EB F0 JMP SHORT Teksched.004A5B31
004455219 . C645 F3 B1 MOU BYTE PTR SS:[EBP-1], 0
004455220 . 2070 08 00 CMP BYTE PTR SS:[EBP-1], 0
004455221 . ^ EB 17 JMP SHORT Teksched.004A5855
004455222 . 207D 08 00 MOU BYTE PTR SS:[EBP-1], 0
004455223 . ^ EB 75 11 JMP SHORT Teksched.004A5855
004455224 . 6A 30 PUSH 30
004455225 . E8 C829F6FF CALL Teksched.004A5849
004455226 . B8 CC544000 MOU EAX, Teksched.004A5ACC
004455227 . E8 8239F9FF CALL Teksched.004391F8
004455228 . 33C0 XOR EAX, EAX
004455229 . SA POP EDX
004455230 . 59 POP ECX
004455231 . 59 POP ECX
004455232 . 64:8910 MOV DWORD PTR FS:[EAX], EDX
004455233 . 68 C2534000 PUSH Teksched.004A58C2
004455234 . 8085 8CFEFFFI LEA EAX, DWORD PTR SS:[EBP-174]
004455235 . BA 0C000000 MOU EDX, 0C
004455236 . E8 1DF7FFFB CALL Teksched.00404FB0
004455237 . 8045 BC LEA EAX, DWORD PTR SS:[EBP-44]
004455238 . BA 04000000 MOU EDX, 4
004455239 . FR 1AF7FFFF CALL Teksched.004A4FRA

```

004455238=Teksched.004A5A88 (ASCII "Registration Key accepted!")
EAX=00000000

Click the beginning of the code

Code 의 시작을 click 해

Rightclick

C CPU - main thread, module Teksched

```

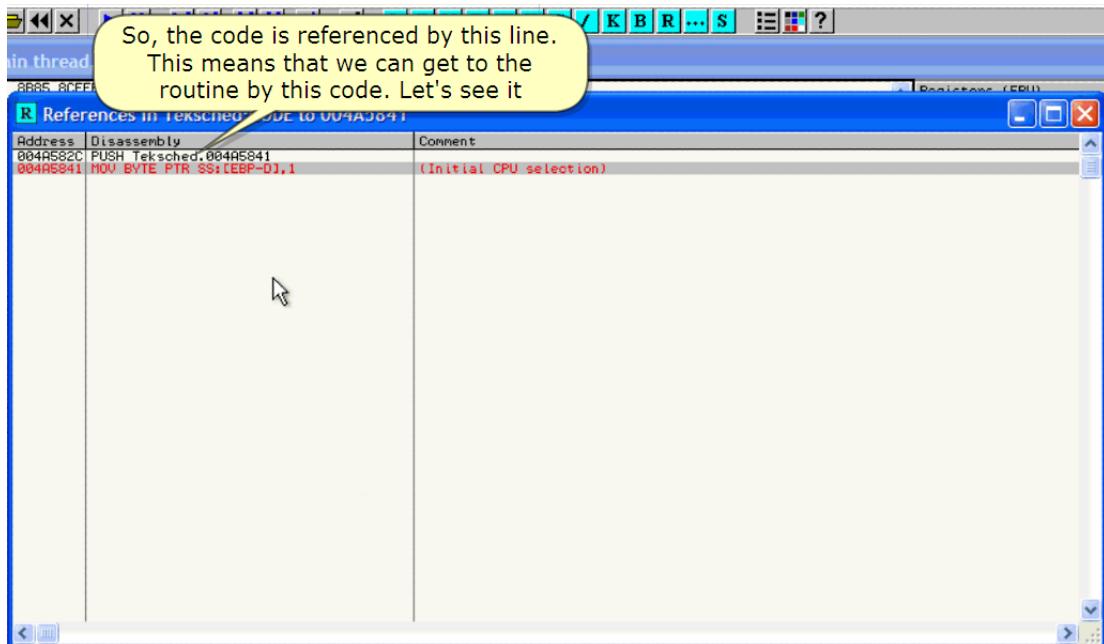
004455203 . 8B85 8CFEFFFI MOU EAX, DWORD PTR SS:[EBP-174]
004455204 . 58 PUSH EAX
004455205 . B8 88544000 MOU ECX, Teksched.004A5A98
004455206 . BA 54544000 MOU EDX, Teksched.004A5B54
004455207 . EB45 CC MOU ERA, DWORD PTR SS:[EBP-34]
004455208 . E8 3414FFFF CALL Teksched.00495C58
004455209 . B845 CC MOU EAX, DWORD PTR SS:[EBP-34]
004455210 . E8 2000FFFF CALL Teksched.00495B44
004455211 . 33C0 XOR EAX, EAX
004455212 . C3 RETN
004455213 . E8 4155440000 PUSH Teksched.004A5841
004455214 > 0B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
004455215 . E8 SBEE8H FF CALL Teksched.004A04094
004455216 . C3 RETN
004455217 . ^ E9 E9EF5FF JMP Teksched.004A4828
004455218 . ^ EB F0 JMP SHORT Teksched.004A5B31
004455219 . C645 F3 B1 MOU BYTE PTR SS:[EBP-1], 0
004455220 . 2070 08 00 CMP BYTE PTR SS:[EBP-1], 0
004455221 . ^ EB 17 JMP SHORT Teksched.004A5855
004455222 . 207D 08 00 MOU BYTE PTR SS:[EBP-1], 0
004455223 . ^ EB 75 11 JMP SHORT Teksched.004A5855
004455224 . 6A 30 PUSH 30
004455225 . E8 C829F6FF CALL Teksched.004A5849
004455226 . B8 CC544000 MOU EAX, Teksched.004A5ACC
004455227 . E8 8239F9FF CALL Teksched.004391F8
004455228 . 33C0 XOR EAX, EAX
004455229 . SA POP EDX
004455230 . 59 POP ECX
004455231 . 59 POP ECX
004455232 . 64:8910 MOV DWORD PTR FS:[EAX], EDX
004455233 . 68 C2534000 PUSH Teksched.004A58C2
004455234 . 8085 8CFEFFFI LEA EAX, DWORD PTR SS:[EBP-174]
004455235 . BA 0C000000 MOU EDX, 0C
004455236 . E8 1DF7FFFB CALL Teksched.00404FB0
004455237 . 8045 BC LEA EAX, DWORD PTR SS:[EBP-44]
004455238 . BA 04000000 MOU EDX, 4
004455239 . FR 1AF7FFFF CALL Teksched.004A4FRA

```

Stack: SS:[0012FFE3]=FF

Address	Hex dump
005A3300	00 00 00 00 00 00 00 00
005A3310	00 00 00 00 00 00 00 00
005A3320	00 00 00 00 32 13 80 C0
005A3330	00 8D 49 00 00 8D 49 00
005A3340	00 8D 49 00 00 8D 49 00
005A3350	00 CB CC C9 C9 07 CF C8
005A3360	00 DE DF E0 E1 E3 00 E4
005A3370	00 00 00 00 01 00 00 00
005A3380	00 00 00 00 00 00 00 00
005A3390	72 00 88 C0 S2 75 6E 74

:(



So, the code is referenced by this line.

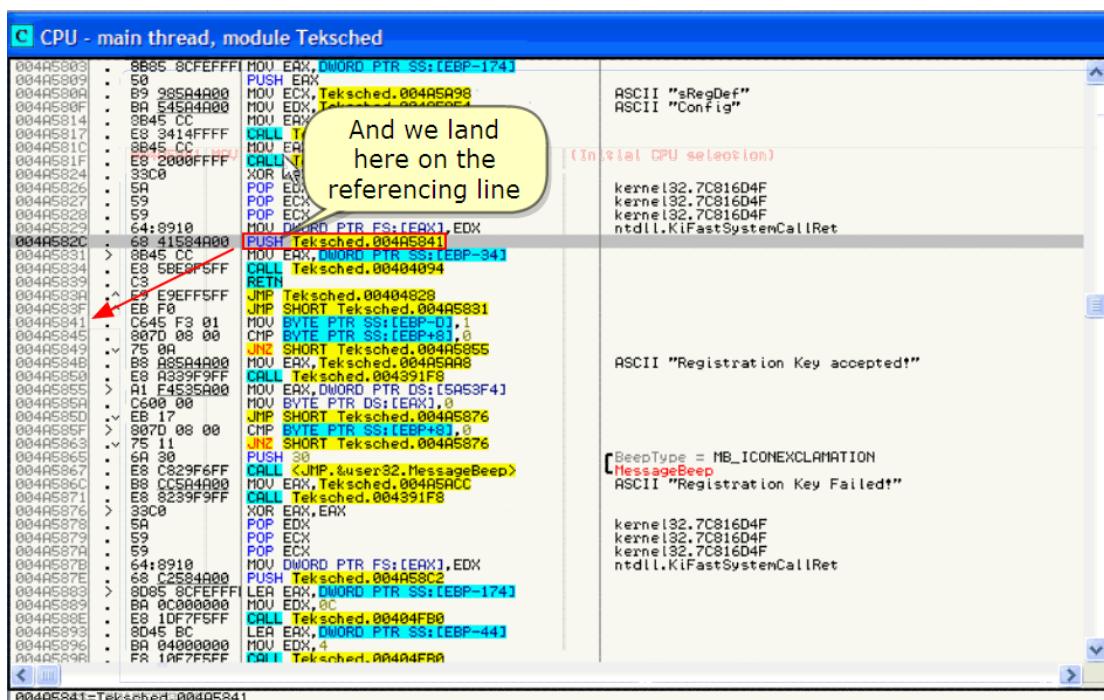
그래서, code 는 이 line 에 의해 참조됐다.

This means that we can get to the routine by this code.

우리는 이 code 에 의해 routine 으로 갈 수 있다.

Let's see it

이제 보자



And we land here on the referencing line

참조하고 있는 line 에 도착했다.

See once more what I mean :

내가 말하는 게 무엇인지 한 번 더 보자.

Indeed: PUSH 4A5841

Followed by a RETURN equals JMP 004A5841

RETURN 에 의해서 JMP 004A5841 과 같아진 것을 따라간다.

번역 주)PUSH 004A5841 후에 Return 을 하면 Stack 에 004A5841 이 들어가고 return 에 의해

004A5841 이 EIP 에 들어가게 되므로 JMP 004A5841 과 같다는 말이다.

PUSH + return ---> JMP 004A5841

Which will bring us to the line we want to go

우리가 원하는 line 으로 우리를 데려간다.

INFO :

Remember this for the future.

미래를 위해 이것을 기억해.

If you want to search for the end of a routine and you find a RETURN then see if there are no
PUSH xxxx

In front that are irrelevant for other purposes

네가 routine 의 끝을 찾기 원한다면 그리고 RETURN 을 찾아 봐. 그 후 만약 PUSH xxxx 가 없다면 이
앞에 있는 것은 다른 목적들과 상관이 없다.

Ok, now that this is clear ...

Ok, 이제 이것은 명확하다 ...

.... put BP as visual aid to where we want to arrive

```
C CPU - main thread, module Teksched
004A5803 : 8B85 8CFFFFF MOU EAX, DWORD PTR SS:[EBP-174]
004A5809 : 50 PUSH EAX
004A5824 : 33C0 XOR ECX, ECX
004A5826 : 59 POP ECX
004A5827 : 59 POP ECX
004A5828 : 59 POP ECX
004A5829 : 64 8910 MOU DWORD PTR FS:[EAX], EDX
004A582C : 61 41584A00 PUSH Tekshed.004A5841
004A5831 : > 8B45 CC MOU EAX, DWORD PTR SS:[EBP-34]
004A5834 : C8 5BE8F5FF CALL Tekshed.00404094
004A5839 : C3 RETN
004A583A : E9 E9EFF5FF JMP Tekshed.00404828
004A583F : EB F0 SHORT Tekshed.004A5831
004A5841 : C645 03 01 MOU BTTE PTR SS:[EBP-0], 1
004A5845 : 807D 00 00 CMP BTTE PTR SS:[EBP+0], 0
004A5848 : 75 04 JNC SHORT Tekshed.004A5855
004A584B : B8 A8584A00 MOU EAX, Tekshed.004A5848
004A5850 : E8 A339F9FF CALL Tekshed.00404094
004A5853 : > B1 F4553A00 MOU EAX, DWORD PTR DS:[EBP+0]
004A5854 : C7 00 00 MOU EAX, DS:[EBX]
004A5856 : EB 17 JMP SHORT Tekshed.004A5876
004A585F : 807D 00 00 CMP BTTE PTR SS:[EBP+0], 0
004A5863 : 75 11 JNC SHORT Tekshed.004A5876
004A5865 : E9 30 PUSH 30
004A5867 : B8 C829F6FF CALL <JNP.%user32.MessageBeep>
004A586C : B8 CC584A00 MOU EAX, Tekshed.004A586C
004A5871 : E8 8239F9FF CALL Tekshed.004391F8
004A5876 : > 33C0 XOR ECX, ECX
004A5878 : 59 POP ECX
004A5879 : 59 POP ECX
004A587A : 59 POP ECX
004A587B : 64:8910 MOU DWORD PTR FS:[EAX], EDX
004A587E : 68 C2584A00 PUSH Tekshed.004A58C2
004A5883 : > 8085 8CFFFFF LEA EAX, DWORD PTR SS:[EBP-174]
004A5889 : B8 0C000000 MOU EDX, 0C
004A588E : E8 1DF7F5FF CALL Tekshed.00404FB0
004A5893 : 8045 BC LEA EAX, DWORD PTR SS:[EBP-44]
004A5896 : B8 04000000 MOU EDX, 4
004A589F : FF 10F7FFFF CALL Tekshed.00404F80

Stack SS:[0012FFE3]=FF
```

.... Put BP as visual aid to where we want to arrive

우리가 도착하기를 원하는 곳에 시작화를 위해 BP 를 넣어라.

CPU - main thread, module Teksched

004AS803	• 8B05 8CFFEFFI MOU EDX, DWORD PTR SS:[EBP-174]	
004AS809	• 50 PUSH EDX	
004AS809	• B9 985A4000 MOU ECX, Teksched.004AS809	
004AS80F	• B8 00 MOV AL, 0	
004AS814	• B8 00 MOV AL, 0	
004AS817	• B8 00 MOV AL, 0	
004AS81C	• B8 00 MOV AL, 0	
004AS81F	• B8 00 MOV AL, 0	
004AS824	• 33C0 XOR EAX, EAX	
004AS825	• 5A POP ECX	
004AS827	• 59 POP ECX	
004AS828	• 59 POP ECX	
004AS829	• 64:8910 MOU DWORD PTR FS:[EAX], EDX	
004AS829	> 58 41584000 PUSH Teksched.004AS841	
004AS831	> 8645 CC MOU EAX, DWORD PTR SS:[EBP-341]	
004AS834	> E8 5BE8FSFF CALL Teksched.004AS8094	
004AS839	> C3 RETN	
004AS839	> E9 E9EFFSFF JMP Teksched.004AS828	
004AS839	> EB F0 JMP SHORT Teksched.004AS831	
004AS839	> C645 F3 01 MOU BYTE PTR SS:[EBP-0], 1	
004AS845	> 807D 08 00 CMP BYTE PTR SS:[EBP+8], 0	
004AS845	> 75 0A JNZ Teksched.004AS855	
004AS848	> B8 A85A4000 MOU EAX, Teksched.004AS848	
004AS850	> E8 B339F9FF CALL Teksched.004AS91FB	
004AS853	> R1 F4535000 MOU EAX, DWORD PTR DS:[ES:R1]	
004AS853	> C600 00 MOV BYTE PTR DS:[EAX], 0	
004AS853	> EB 17 JMP SHORT Teksched.004AS876	
004AS853	> 807D 08 00 CMP BYTE PTR SS:[EBP+8], 0	
004AS853	> 75 11 JNZ Teksched.004AS876	
004AS853	> B8 30 MOU EAX, 30	
004AS853	> E8 C829FF FF CALL [Kmp\&user32.MessageBoxBeep]	
004AS853	> 80CC4000 MOU EAX, Teksched.004AS848	
004AS871	> E8 B23979FF CALL Teksched.004AS91FB	
004AS871	> 33C0 XOR EAX, EAX	
004AS873	> 5A POP EDX	
004AS873	> 59 POP ECX	
004AS873	> 59 POP ECX	
004AS878	• 64:8910 MOU DWORD PTR FS:[EAX], EDX	
004AS878	• 68 C2584000 PUSH Teksched.004AS802	
004AS883	• 8B05 8CFFEFFI LEA EAX, DWORD PTR SS:[EBP-174]	
004AS889	• BA 0C000000 MOU EDX, 0C	
004AS88E	• E8 1DF7FSFF CALL Teksched.004AS808	
004AS893	• 8045 BC LEA EAX, DWORD PTR SS:[EBP-441]	
004AS896	• BA 04000000 MOU EDX, 4	
004AS896	• F8 1MF7FSFF CALL Teksched.004AS808	

Allways remember : take an overview first ...

(Initial CPU selection)

kernel32!7C816D4F
kernel32!7C816D4F
kernel32!7C816D4F
ntdll!KIFastSystemCall!Ret

ASCII "Registration Key accepted!"

BeepType = MB_ICONEXCLAMATION
MessageBeep
ASCII "Registration Key Failed!"

kernel32!7C816D4F
kernel32!7C816D4F
kernel32!7C816D4F
ntdll!KIFastSystemCall!Ret

004AS841=Teksched.004AS841

Yep, here too

예, 여기도

Always remember : take an overview first ...

항상 기억해 : 개요를 먼저 살펴봐...

So, scroll down to see what happens next

그래서, 다음에 어떤 일이 발생하는지 보기 위해 scroll 내려.

Mmmm, nothing important it seems

Probably the registering of the program if the key was accepted

Scroll up

음, 지금 보는 것은 중요하지 않다.

아마 key 가 받아 들여진다면 program 은 등록된다.

Scroll 올려.

And let's see what happens before all this

이 모든 것 다음에 어떤 일이 발생하는지 봄.

Delphi code can also be recognized by calls and calls and calls :)

Delphi code 는 call 과 call 과 call 에 의해 알려졌다.

C CPU - main thread, module Tekshed

```
004A55C8 • 8B45 D8 MOV ERX, DWORD PTR SS:[EBP-28]
004A55CB • E8 DCFEFSFF CALL Tekshed.004054AC
004A55D0 • 8B85 A0FEFFF1 MOV ERX, DWORD PTR SS:[EBP-160]
004A55D6 • 8D95 R4FEFFF1 LEA EDX, DWORD PTR SS:[EBP-15C]
004A55D9 • E8 4342F5FF CALL Tekshed.00405924
004A55E1 • 8B95 R4FEFFF1 MOV EDX, DWORD PTR SS:[EBP-15C]
004A55E7 • 5B POP ECX
004A55E8 • E8 A0FDF5FF CALL Tekshed.00405398
004A55E9 • 74 19 SHORT Tekshed.00405608
004A55F0 • 807D 08 08 CMP BYTE PTR DS:[EBP+8], 0
004A55F3 • >F855 7D200000 JNZ Tekshed.00405876
004A55F4 • B8 79E5940000 MOU ERX, Tekshed.004A5978
004A55F6 • E8 F5309FF9 CALL Tekshed.0040391F8
004A5603 • E9 6E020000 JMP Tekshed.004A5876
004A5608 • 74 19 SHORT Tekshed.004A5620
004A5611 • 807D 08 08 CMP BYTE PTR SS:[EBP+8], 0
004A5613 • >F855 59200000 JNZ Tekshed.00405876
004A5617 • B8 E4594000 MOU ERX, Tekshed.004A59E4
004A5622 • E8 D13BF9FF CALL Tekshed.0040391F8
004A5627 • >E9 4A020000 JMP Tekshed.00405876
004A562D • A1 BC5CS5000 MOU ERX, DWORD PTR DS:[5ASCBC]
004A5631 • 66:8800 MOV AX, WORD PTR DS:[ERX]
004A5634 • 66:F7EE IMUL SI
004A5637 • B8F0 MOV ES1, EA8
004A5639 • B8 89E40000 MOV EBX, 4E89
004A563E • 8D95 BCFFFFF1 LEA EDX, DWORD PTR SS:[EBP-144]
004A5644 • 0FB7C6 MOUZX EA8, SI
004A5647 • E8 9CE2F5FF CALL Tekshed.004058E8
004A564C • 8D95 BCFFFFF1 LEA EDX, DWORD PTR SS:[EBP-144]
004A5652 • 8D45 EC LEA ERX, DWORD PTR SS:[EBP-14]
004A5655 • E9 96FBF5FF CALL Tekshed.004051F0
004A5659 • 8D4D E8 LEA ECX, DWORD PTR SS:[EBP-18]
004A565D • 66:BE 0600 MOU DX, 6
004A5661 • 9BC3 MOV ERX, EBX
004A5663 • E8 0B84FFFF CALL Tekshed.00409DA70
004A5668 • 8D45 EC LEA EDX, DWORD PTR SS:[EBP-14]
004A566B • 8B85 E8 MOU EDX, DWORD PTR SS:[EBP-18]
004A5671 • E8 4342F5FF CALL Tekshed.00405254
004A5674 • 8B84 EC MOU ERX, DWORD PTR SS:[EBP-14]
004A5676 • 8B85 DC MOU EDX, DWORD PTR SS:[EBP-24]
004A5679 • E8 1AFDF5FF CALL Tekshed.00405398
004A567E • 0F85 60100000 JPC Tekshed.004A580F
004A5684 • AH85 NR LFA FAX, DWORD PTR SS:[EBP-28]
```

kernel32.7C0816D4F

ASCII "Error! You are attempting to register the wrong module."

Aha ! This will need our attention in a second. But let's first find the start of this registration routine

Aha! This will need our attention in a second. But let's first find the start of this registration routine

아하! 이것은 두번째로 우리의 주의를 필요로 할 것이다. 먼저 이 registration routine 의 시작부분을 찾아보자.

More BadBoys

Badboy 가 많다.

C CPU - main thread, module Teksched

Take a quick note of the VA's so we know that jumps that pass this code are probably good jumps

Take a quick note of the VA's so we know that jumps that pass this code are probably good jumps

빠르게 VA's 를 적어. 그래서 그 jump 는 이 code 를 피해야 good jump 로 간다.

C CPU - main thread, module Teksched

```

004A5380 . 8B45 E8 MOU EAX, DWORD PTR SS:[EBP-18]
004A5383 . 8B45 FEFFF CALL Teksched.004A524C
004A5386 . 8B55 E8 MOU EDX, DWORD PTR SS:[EBP-18]
004A5389 . 8B45 02 FF MOU AL,BYTE PTR DS:[EDX+EAX-1]
004A5392 . E8 A0DAF5FF CALL Teksched.004A528C
004A5394 . 8B08 MOU EDX, ERX
004A5396 . 8B45 BC LEA EAX, DWORD PTR SS:[EBP-44]
004A5399 . E8 D6FDFF CALL Teksched.004A5174
004A539E . 8B55 BC MOU EDX, DWORD PTR SS:[EBP-44]
004A53A1 . 8B45 D8 LEA EAX, DWORD PTR SS:[EBP-28]
004A53A4 . E8 ABFEFF CALL Teksched.004A5254
004A53A9 . EB 19 JMP SHORT Teksched.004A53C4
004A53B0 . 807D 08 00 CMP BTYE PTR SS:[EBP+8], 0
004A53B3 . 8F85 C1040000 JNZ Teksched.004A5876
004A53B5 . B8 30534000 MOU EAX, Teksched.004A5930
004A53B8 . E8 393EF9FF CALL Teksched.004A591F8
004A53C4 . E9 B2040000 JMP Teksched.004A5876
004A53C7 . 8B45 F4 MOU EAX, DWORD PTR SS:[EBP-0]
004A53C9 . E8 80FEFF CALL Teksched.004A524C
004A53CD . 48 DEC EAX
004A53CF . 7D 19 JNE SHORT Teksched.004A53E8
004A53D0 . 807D 08 00 CMP BTYE PTR SS:[EBP+8], 0
004A53D3 . 8F85 90040000 JNZ Teksched.004A5876
004A53D6 . B8 58524000 MOU EAX, Teksched.004A5958
004A53DE . E8 153EF9FF CALL Teksched.004A591F8
004A53E3 . E9 8E040000 JMP Teksched.004A5876
004A53E8 . 8095 BCFFFF LEA EDX, DWORD PTR SS:[EBP-144]
004A53E9 . E8 4445 F4 CALL Teksched.004A524C
004A53F1 . E8 D6EFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
004A53F6 . 8095 BCFFFF MOU EAX, DWORD PTR SS:[EBP-144]
004A53FC . 8045 DC LEA EAX, DWORD PTR SS:[EBP-24]
004A53F7 . E8 ECDF5FF CALL Teksched.004A51F8
004A5404 . 8045 F4 LEA EAX, DWORD PTR SS:[EBP-0]
004A5407 . 50 PUSH EAX
004A5408 . 8B45 DC MOU EAX, DWORD PTR SS:[EBP-24]
004A540B . E8 3CFFFF CALL Teksched.004A524C
004A5410 . 8BEC8 MOU ECX, EAX
004A5412 . 83E9 05 SUB ECX, 5
004A5415 . BA 01000000 MOU EDX, 1
004A5418 . 8B45 DC MOU EAX, DWORD PTR SS:[EBP-24]
004A541D . E8 8A00F6FF CALL Teksched.004A584C
004A5422 . 8045 E4 LEA EAX, DWORD PTR SS:[EBP-1C]
004A5425 . 8045 00 PUSH EAX
004A5426 . 8B45 F4 MOU EAX, DWORD PTR SS:[EBP-24]
004A5429 . FA 1FFFFFFF CALL Teksched.004A524C

```

004A5341=Teksched.004A5841

Aha, does this ring a bell ???

bingo?

This is the verification

"Is something filled in the messagebox?"(or did the user forget the input)

이것은 검증하는 것이다.

"messagebox에 무언가 채워졌어?"(아니면 user는 input을 잊었다)

ASCII "Enter a Last Name value now..."

kernel32.7C8399F3

Aha, does this ring a bell ???

ASCII "Enter a Key value now..."

kernel32.7C8399F3

This is the verification

"Is something filled in the messagebox?"
(or did the user forget the input)

C CPU - main thread, module Teksched

```

004A5358 . 83C8 02 ADD EBX, 2
004A535B . 66:83FF 1B CMP DI, 1B
004A535F . ^ 75 CA JNC SHORT Teksched.004A5328
004A5361 . 8B45 E8 MOU ERX, DWORD PTR SS:[EBP-18]
004A5364 . 8040 MOU RL,BYTE PTR DS:[ERX]
004A5366 . E8 01DAF5FF CALL Teksched.004A528C
004A5368 . 8B08 MOU EDX, EAX
004A5370 . 8045 C0 LEA EAX, DWORD PTR SS:[EBP-40]
004A5375 . E8 FFFDF5FF CALL Teksched.004A5174
004A5378 . 8045 D8 MOU EDX, DWORD PTR SS:[EBP-28]
004A537B . E8 D4FEFF CALL Teksched.004A5254
004A5380 . 8B45 E8 MOU EAX, DWORD PTR SS:[EBP-18]
004A5383 . E8 C4FEFF CALL Teksched.004A524C
004A5386 . 8B55 E8 MOU EDX, DWORD PTR SS:[EBP-18]
004A538B . 804482 FF MOU AL,BYTE PTR DS:[EDX+EAX-1]
004A538E . E8 A0DAF5FF CALL Teksched.004A528C
004A5394 . 8B08 MOU EDX, EAX
004A5396 . 8045 BC LEA EAX, DWORD PTR SS:[EBP-44]
004A5399 . E8 D6FDFF CALL Teksched.004A5174
004A539E . 8B55 BC MOU EDX, DWORD PTR SS:[EBP-44]
004A53A1 . 8B45 D8 LEA EAX, DWORD PTR SS:[EBP-28]
004A53A4 . E8 ABFEFF CALL Teksched.004A5254
004A53A9 . EB 19 JNP SHORT Teksched.004A53C4
004A53B0 . 807D 08 00 CMP BTYE PTR SS:[EBP+8], 0
004A53B3 . 8F85 C1040000 JNZ Teksched.004A5876
004A53B5 . B8 30534000 MOU EAX, Teksched.004A5930
004A53B8 . E8 393EF9FF CALL Teksched.004A591F8
004A53C4 . E9 B2040000 JNP Teksched.004A5876
004A53C7 . 48 DEC EAX
004A53CF . 7D 19 JNE SHORT Teksched.004A53E8
004A53D0 . 807D 08 00 CMP BTYE PTR SS:[EBP+8], 0
004A53D3 . 8F85 90040000 JNZ Teksched.004A5876
004A53D6 . B8 58524000 MOU EAX, Teksched.004A5958
004A53DE . E8 153EF9FF CALL Teksched.004A591F8
004A53E3 . E9 8E040000 JNP Teksched.004A5876
004A53E8 . 8095 BCFFFF LEA EDX, DWORD PTR SS:[EBP-144]
004A53F1 . E8 D6EFFFFF CALL Teksched.004A43CC
004A53F6 . 8095 BCFFFF LEA EDX, DWORD PTR SS:[EBP-144]
004A53FC . 8045 DC LEA EAX, DWORD PTR SS:[EBP-24]
004A53F7 . E8 ECDF5FF CALL Teksched.004A51F8
004A53F9 . 1FA FAX.DWORD PTR SS:[EBP-144]

```

004A5341=Teksched.004A5841

And this too

ASCII "Enter a Last Name value now..."

kernel32.7C8399F3

ASCII "Enter a Key value now..."

kernel32.7C8399F3

And this too

0| 것 또한

C CPU - main thread, module Teksched

```
004A52A4 . E8 930BFSFF CALL Tekshed.00402E3C  
004A52B7 . 8B08 MOU EDX, EAX  
004A52B8 . 8045 C4 LEA EAX, DWORD PTR SS:[EBP-30]  
004A52B9 . E8 21FEFF5Ff CALL Tekshed.00405174  
004A52B9 . 8055 C4 MOU EDX, DWORD PTR SS:[EBP-30]  
004A52B9 . 8045 D8 LEA EAX, DWORD PTR SS:[EBP-28]  
004A52B9 . E8 36FFF5FF CALL Tekshed.00405254  
004A52B9 > 807D 00 00 JMP SHORT Tekshed.004A52D9  
004A52C0 . 8085 AC050000 JNZ Tekshed.004A5878  
004A52C4 . 80F85 C2000000 MOU EAX, Tekshed.004A5898  
004A52CA . 8810 64575A00 MOU EBX, Tekshed.004A5908  
004A52CF . E8 243FF9FF CALL Tekshed.004391F8  
004A52D4 . E9 90050000 JMP Tekshed.004A5876  
004A52D9 > 8B45 E8 MOU EAX, DWORD PTR SS:[EBP-18]  
004A52D9 . E8 6BFFFF5Ff CALL Tekshed.0040524C  
004A52E1 . 89C0 TEST EAX, EAX  
004A52E3 . 80F85 C2000000 JLE Tekshed.004A5398  
004A52E9 . 66:BF 0100 MOU DI, 1  
004A52ED . 8B10 64575A00 MOU EBX, DWORD PTR DS:[5A5764]  
004A52F3 . 43 INC EBX  
004A52F4 > 8B45 E8 MOU EAX, DWORD PTR SS:[EBP-18]  
004A52F7 . 8A00 MOU AL, BYTE PTR DS:[EAX]  
004A52F9 . E8 3ED0BF5Ff CALL Tekshed.00402E3C  
004A52F6 . 50 PUSH EAX  
004A52F2 . 8A03 MOU AL, BYTE PTR DS:[EBX]  
004A5301 . E8 360BFF5Ff CALL Tekshed.00402E3C  
004A5307 . SH POP EDX  
004A5308 . 3AD0 POP DL, AL  
004A5309 > 75 0B JNE SHORT Tekshed.004A5316  
004A5309 . 8B7C7 MOU EDX, EAX, DI  
004A5309 . 8B15 BC5C5A00 MOU EDX, DWORD PTR DS:[5A5CBC]  
004A5314 . 80102 ADD EDX, PTR DS:[EDX], EAX  
004A5316 . 47C0 INC EDI  
004A5317 . 89C3 02 ADD EBX, 2  
004A5319 . 66:83FF 1B CMP DI, 1B  
004A531E . 75 04 JNZ SHORT Tekshed.004A52F4  
004A5320 . 66:BF 0100 MOU DI, 1  
004A5324 . 8B10 64575A00 MOU EBX, DWORD PTR DS:[5A5764]  
004A5324 . 43 INC EBX  
004A5324 > 8B45 E8 MOU EAX, DWORD PTR SS:[EBP-18]  
004A5324 . E8 19FF5FFf CALL Tekshed.0040524C  
004A5323 . 8B55 E8 MOU EDX, DWORD PTR SS:[EBP-18]  
004A5323 . 8A4402 FF MOU AL, BYTE PTR DS:[EDX+EAX-1]  
004A5323 . E8 FDD0AF5Ff CALL Tekshed.00402E3C  
004A5323 . 5A PUSH FAX  
004A5323 = 004A5841=Teksched.004A5841
```

And this

C CPU - main thread, module Teksched

```
004A51C4 . 59 POP ECX  
004A51C5 . 59 POP ECX  
004A51C5 > 64:8910 MOU DWORD PTR FS:[EAX], EDV  
004A51C5 . 89 DE514A00 PUSH Tekshed.004A510E  
004A51CE . 8B45 D4 MOU EAX, _LOCAL_11  
004A51D1 . E8 0000 RETN  
004A51D6 . 50 CALL Tekshed.00404094  
004A51D7 . EB F0 JNP Tekshed.004A51CE  
004A51D8 . 8945 F0 RET  
004A51F1 > 8B45 EC MOU EBX, DWORD PTR DS:[5A5764]  
004A51F1 . 8A00 MOU EAX, DWORD PTR DS:[5A5764]  
004A51F1 . E8 390CF5FF PUSH Tekshed.00402E3C  
004A5203 . 50 MOU AL, BYTE PTR DS:[EBX]  
004A5204 . 8A03 POP EDX  
004A5204 . 5A MOU EDX, DWORD PTR DS:[5A5764]  
004A5204 . 3AD0 CMP DL, AL  
004A5204 > 75 0B JNE SHORT Tekshed.004A5218  
004A5204 . 8B7C7 MOU EDX, EAX, DI  
004A5204 . 8B15 BC5C5A00 MOU EDX, DWORD PTR DS:[5A5CBC]  
004A5219 . 89902 ADD EDX, PTR DS:[EDX], EAX  
004A5218 . 47C0 INC EDI  
004A521C . 89C3 02 ADD EBX, 2  
004A521F . 66:83FF 1B CMP DI, 1B  
004A5223 . 75 04 JNZ SHORT Tekshed.004A51F9  
004A5223 . 66:BF 0100 MOU DI, 1  
004A5223 . 8B10 64575A00 MOU EBX, DWORD PTR DS:[5A5764]  
004A5223 . 43 INC EBX  
004A5230 . 8B45 EC MOU EAX, DWORD PTR SS:[EBP-14]  
004A5230 . 8B55 EC CALL Tekshed.0040524C  
004A5230 . 8A4402 FF MOU EDX, DWORD PTR SS:[EBP-14]  
004A5230 . E8 F80BFF5Ff CALL Tekshed.00402E3C  
004A5244 . 50 PUSH EAX  
004A5245 . 8A03 MOU AL, BYTE PTR DS:[EBX]  
004A5245 . E8 F00BFF5Ff CALL Tekshed.00402E3C  
004A5245 . 5A POP EDX  
004A5245 . 80NA CMP NI, AI  
004A5245 = 004A5841=Teksched.004A5841
```

Mmmmm, finally :)

음, 마지막으로 :)

We have found the beginning of the routine. (Well, at least far enough up in the code :))

우리는 routine 의 시작하는 부분을 찾았다. (좋아, code 에서 적어도 이 정도면 충분하다. :))

Again, see the PUSH XXXXXXXXXXXXXXXX 봐.

다시, PUSH XXXXXXXXXXXXXXXX 봐.

Followed by a return

Return 을 따라가.

So, we will jump here !!!

그래서, 우리는 여기로 jump 할 것이다!!!

Do you understand that effectively, this is NOT the beginning of the registration routine?

효과적으로 이해했어? 이것은 registration routine 의 시작이 아니다.

But I'm assuming that I'm far enough up in the code because I've passed the "Is something entered?" code already.

그러나 그것은 이 code 에서 충분하다고 생각한다. 왜냐하면 나는 이미 "무언가로 들어갔어?" code 를 지나쳤다.

Scroll down again

Scroll 다시 밑으로 내려.

...and place a BP after all the code

... 이 code 뒤에 BP 를 설치 해.

To verify "Is first name entered ?"

검증하기 "첫번째 이름 입력됐어?"

To verify "Is last name entered ?"

검증하기 "마지막 이름 입력됐어?"

To verify "Is key value entered ?"

검증하기 "key value 입력됐어?"

C CPU - main thread, module Teksched

```
004A53A4 . E8 ABFEF5FF CALL Teksched.00405254
004A53A9 .> EB 19 JNP SHORT Teksched.004A53C4
004A53AB .> 807D 08 00 CMP BTTE PTR SS:[EBP+8], 0
004A53A9 .> 0F85 C104000 JNZ Teksched.004A5876
004A53B5 .> B8 30524000 MOU EAX, Teksched.004A5900
004A53B8 .> E8 393EF9F CALL Teksched.004A391F8
004A53B9 .> E9 B2040000 JMP Teksched.004A5876
004A53C4 .> 8045 F4 MOU EAX, DWORD PTR SS:[EBP-0]
004A53C7 .> E8 80FEF5FF CALL Teksched.00405240
004A53C9 .> 40 19 DEE EAX
004A53C9 .> 0F85 00000000 SHORT Teksched.004A53E8
004A53C9 .> 807D 08 00 CMP BTTE PTR SS:[EBP+8], 0
004A53D0 .> 0F85 90040000 JNZ Teksched.004A5876
004A53D9 .> B8 58524000 MOU EAX, Teksched.004A5958
004A53D9 .> E8 153EF9F CALL Teksched.004A391F8
004A53E3 .> E9 8E040000 JMP Teksched.004A5876
004A53E8 .> 8045 00000000 LEA EDX, DWORD PTR SS:[EBP-144]
004A53E8 .> 8045 04 MOU EAX, DWORD PTR SS:[EBP-C]
004A53F1 .> E8 D0EFFFFF CALL Teksched.004A430C
004A53F6 .> 8095 BCFFFFFF LER EDX, DWORD PTR SS:[EBP-144]
004A53FC .> 8045 DC LER EAX, DWORD PTR SS:[EBP-241]
004A5404 .> E8 EC0DF5FF CALL Teksched.004A430C
004A5404 .> 8045 F4 LER EAX, DWORD PTR SS:[EBP-241]
004A5407 .> 50 PUSH EBP
004A5407 .> 8045 DC MOU EDI, EBP
004A5408 .> E8 EC0EF5FF CALL Teksched.004A430C
004A5408 .> 8045 00 MOU ECX, EBP
004A5410 .> E8 80C8 SUB ECX, 8
004A5412 .> 8045 05 MOU EDI, ECX
004A5415 .> B8 01000000 MOU EDI, 1
004A541A .> 8045 DC MOU EAX, ECX
004A541D .> E8 8A00F6FF CALL Teksched.004A430C
004A5422 .> 8045 E4 LER EAX, DWORD PTR SS:[EBP-241]
004A5425 .> 50 PUSH EBP
004A5426 .> 8045 DC MOU EAX, ECX
004A5429 .> E8 1EFFEF5FF CALL Teksched.004A430C
004A5429 .> 8000 00 MOU EDX, EAX
004A5430 .> 8045 04 SUB EDX, 4
004A5433 .> B9 02000000 MOU ECX, 2
004A5433 .> 8045 DC MOU EAX, DWORD PTR SS:[EBP-241]
004A5438 .> E8 6C00F6FF CALL Teksched.004054AC
004A5440 .> 8045 E0 LER EAX, DWORD PTR SS:[EBP-201]
004A5443 .> 50 PUSH EAX
004A5444 .> 8045 DC MOU EAX, DWORD PTR SS:[EBP-241]
004A5447 .> E8 80FEF5FF CALL Teksched.00405240
004A5447 .> 8000 00 MNH FNX.FAX
```

Stack address=0012FEAC
EDX=7C900B94 (ntdll.KiFastSystemCall!Ret)
Jump from 004A53C0

Let me resume what we've done so far.

내가 지금까지 한 일을 재개해

Looking in the textstrings, I found a string that definitely belongs to the registration scheme.

Textstrings 에서 봐. 나는 분명히 registration scheme 이 위치해 있는 string 을 찾았다.

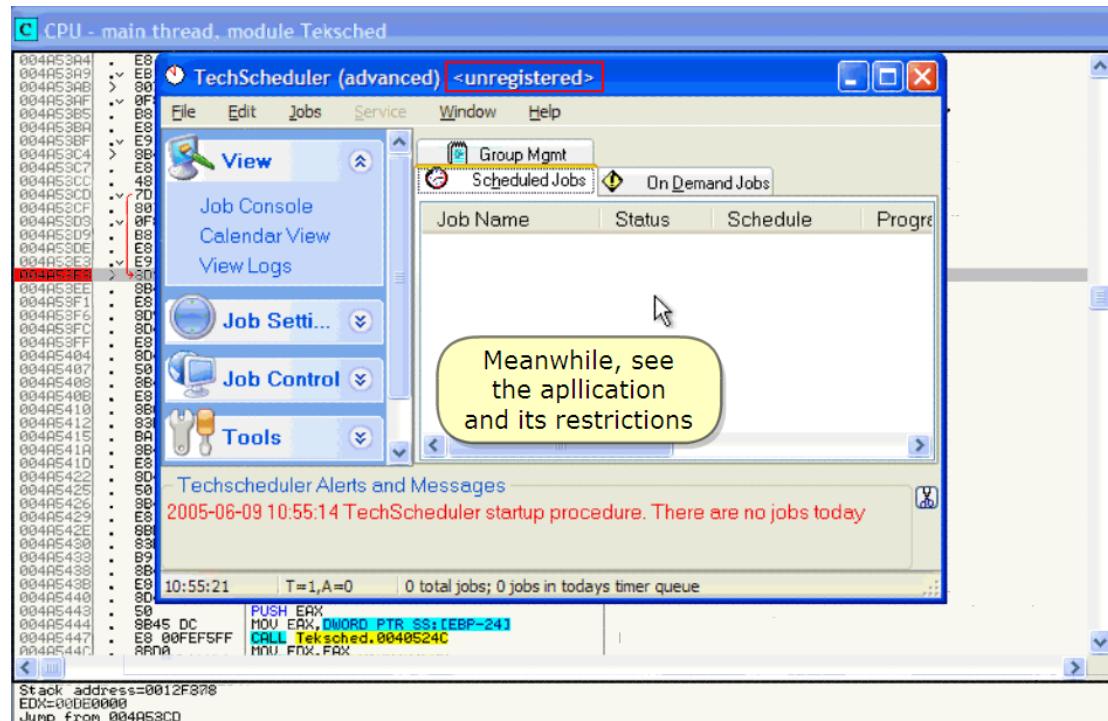
I've set a BP after the "useless code" and now, let's run the registration to see if Olly breaks here

...

나는 "쓸모없는 code" 후에 BP 를 set 했다. Olly 가 여기에서 멈춰 있을 때 보기 위해 Registration 을 실행 해.

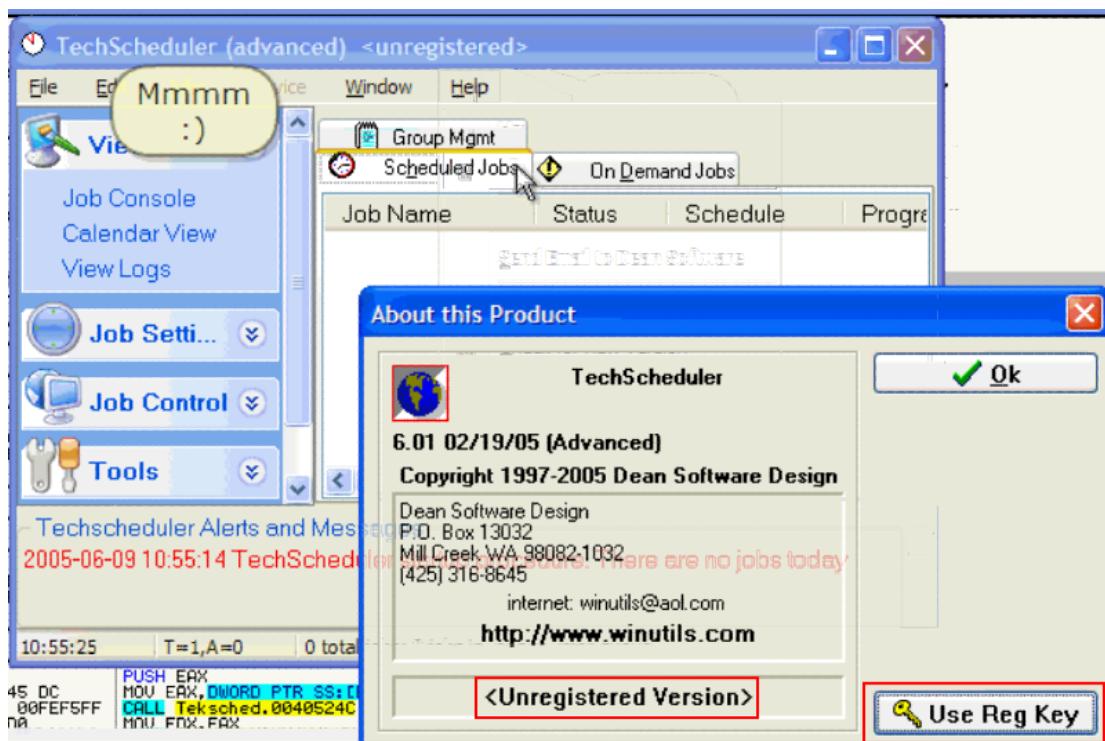
And run

실행



Meanwhile, see the application and its restrictions

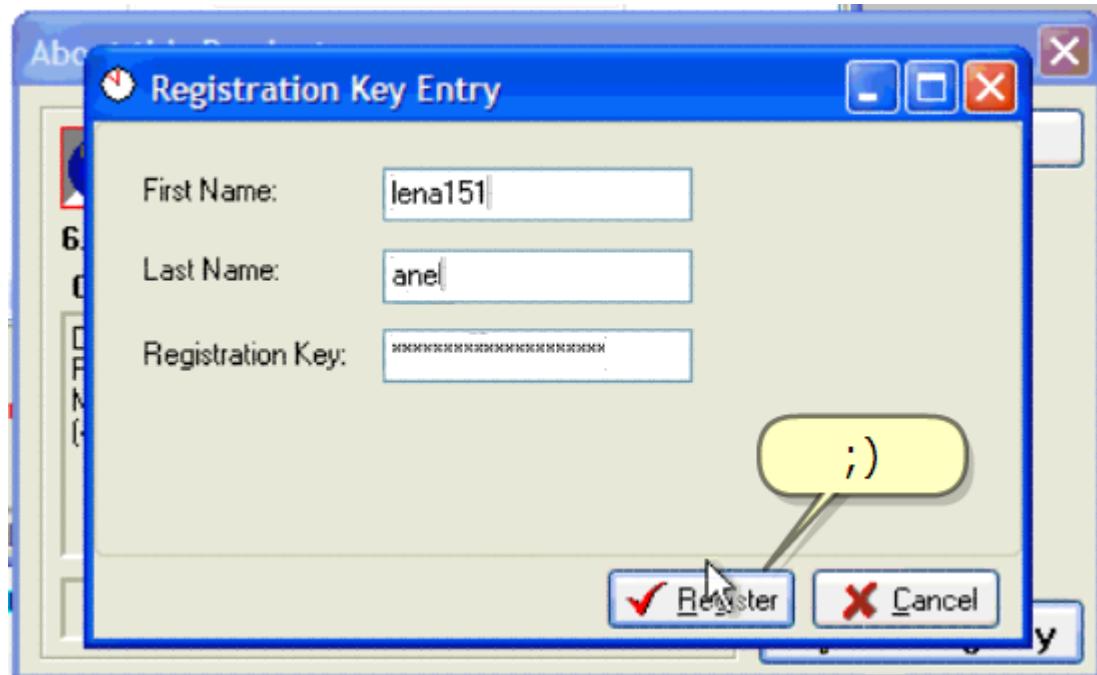
Application 을 봐. 그리고 그것은 제한적이다.



Mmmm :)

:)

;)



That's also me, but backwards ;)(anel)

그것도 나지만, 거꾸로

번역 주)lena -> anel

;))

CPU - main thread, module Tekshed

```
004453C7 E8 80FEFSFF CALL Tekshed.0040524C
004453D0 ✓ 48 8D 19 MOU EDX, DQWORD PTR SS:[EBP-19]
004453D3 8B70 08 00 CMP BYTE PTR SS:[EBP-8], 0
004453D6 8B95 00404000 JNE Tekshed.00405574
004453D9 B8 58594000 MOU EAX, Tekshed.00405958
004453E0 E8 153EF5F9 MOU EAX, Tekshed.004391F0
004453E3 E9 8E840000 JNE Tekshed.00405875
004453E9 × 3D95 BCFFFFFF LEA EDX, DQWORD PTR SS:[EBP-144]
004453F0 8B45 F4 MOU EAX, DQWORD PTR SS:[EBP-C]
004453F1 E8 D6FFFFF9 CALL Tekshed.004043CC
004453F4 8B95 BCFFFFFF LEA EDX, DQWORD PTR SS:[EBP-144]
004453F5 8D45 DC LEA EDX, DQWORD PTR SS:[EBP-24]
004453F8 E8 ECFDF5FF CALL Tekshed.004051F0
00445401 8B45 F4 LEA EAX, DQWORD PTR SS:[EBP-C]
00445402 8B45 0C MOU EAX, DQWORD PTR SS:[EBP-24]
00445403 E8 00000000 CALL Tekshed.0040524C
00445410 E8 00000000 EAX
00445411 E8 00000000 DQWORD PTR SS:[EBP-24]
00445412 8B45 E4 LEA EAX, DQWORD PTR SS:[EBP-1C]
00445413 E8 00000000 CALL Tekshed.004054AC
00445414 8B45 DC PUSH EAX
00445415 E8 00000000 MOU EAX, DQWORD PTR SS:[EBP-24]
00445416 8B95 1DCFEFF5F9 CALL Tekshed.0040524C
00445419 8B90 00 MOU EDX, EAX
00445420 89E9 04 SUB EDX, 4
00445421 8B90 02000000 MOU ECX, 2
00445422 8B45 DC MOU EAX, DQWORD PTR SS:[EBP-24]
00445423 8B 6C00F6FF CALL Tekshed.004054AC
00445424 8B45 E0 LEA EAX, DQWORD PTR SS:[EBP-20]
00445425 5B PUSH EAX
00445426 8B45 DC MOU EAX, DQWORD PTR SS:[EBP-24]
00445427 8B 00FEFF5FF CALL Tekshed.0040524C
00445428 8B90 00 MOU EDX, EAX
00445429 89E9 02 SUB EDX, 2
00445430 8B 03000000 MOU ECX, 3
00445431 8B45 DC MOU EAX, DQWORD PTR SS:[EBP-24]
00445432 E8 4E00F6FF CALL Tekshed.004054AC
00445433 8B45 E0 MOU EAX, DQWORD PTR SS:[EBP-20]
00445434 E8 0AA4CF6FF CALL Tekshed.00404070
00445435 8B45 DCFBFF MOU EAX, DQWORD PTR SS:[EBP-24]
00445436 FF 1FFRFF5FF PFI Tekshed.004044F0A0

Fine : we land in the BP
```

Fine : we land in the BP

좋아 : BP 에 도착했다.

Remove BP

BP 제거.

Scroll down

Scroll 내려.

CPU - main

004A5400
004A5410
004A5412
004A5413

Stack address=0012E8F8
EDC=00DEB298
Jump from 004A53CD

```
004A541H    BB45 DC    MOV EAX, DIWORD PTR SS:[EBP-24]
004A541D    EB 8A00F6FF  CALL Teksched.0040854AC
004A5422    9945 E4    LEA EAX, DIWORD PTR SS:[EBP-10]
004A5425    50          PUSH EAX
004A5429    8945 10    MOU EAX, DIWORD PTR SS:[EBP-24]
004A542E    00FEFFSF  CALL Teksched.00408524C
004A5430    BB04          MOU EDX, EAX
004A5431    83E4 04    SUB EDX, 4
004A5432    99 02000000  MOU ECX, 2
004A5433    8945 DC    MOU EAX, DIWORD PTR SS:[EBP-24]
004A5436    EB 6C00F6FF  CALL Teksched.0040854AC
004A5439    8D45 E0    LEA EAX, DIWORD PTR SS:[EBP-20]
004A5440    50          PUSH EAX
004A5443    8945 DC    MOU EAX, DIWORD PTR SS:[EBP-24]
004A5446    E8 00FEFFSF  CALL Teksched.00408524C
004A544C    BB04          MOU EDX, EAX
004A544E    83E4 02    SUB EDX, 2
004A5451    B9 03000000  MOU ECX, 3
004A5456    8945 DC    MOU EAX, DIWORD PTR SS:[EBP-24]
004A5459    EB 8E00F6FF  CALL Teksched.0040854AC
004A545E    8945 E0    MOU EAX, DIWORD PTR SS:[EBP-20]
004A5461    EB 0A40CFEF  CALL Teksched.00408078
004A5466    8D45 DC    LEA EAX, DIWORD PTR SS:[EBP-24]
004A5469    E8 1EEFB5FF  CALL Teksched.004084F8C
004A546E    8945 F4    MOU EAX, DIWORD PTR SS:[EBP-24]
004A5471    E8 D6FD05FF  CALL Teksched.00408524C
004A5476    66185C0    TEST AX, AX
004A5479    > 0F86 23010000 JBE Teksched.0040A55A2
004A547F    6618945 D2    MOU WORD PTR SS:[EBP-2E], AX
004A5483    > 66BF 0100  MOU DI, 1
004A5487    > 66BE 0100  MOU SI, 1
004A548B    8B10 64575A8B MOU EBX, DIWORD PTR DS:[5A5764]
004A5491    43          INC EBX
004A5492    0FB7C7    MOVZX EAX, DI
004A5495    8B55 F4    MOV EDX, DIWORD PTR SS:[EBP-C]
004A5498    8A4402 FF    MOV AL, BYTE PTR DS:[EAX+EBX-1]
004A549C    EB 98D095FF  CALL Teksched.004082E3C
004A54A1    50          PUSH EAX
004A54A2    8A03          MOU AL, BYTE PTR DS:[EBX]
004A54A4    FB 9A09F5FF  CALL Teksched.0040A55A1

And place BP here to test if we can skip all these calls
```

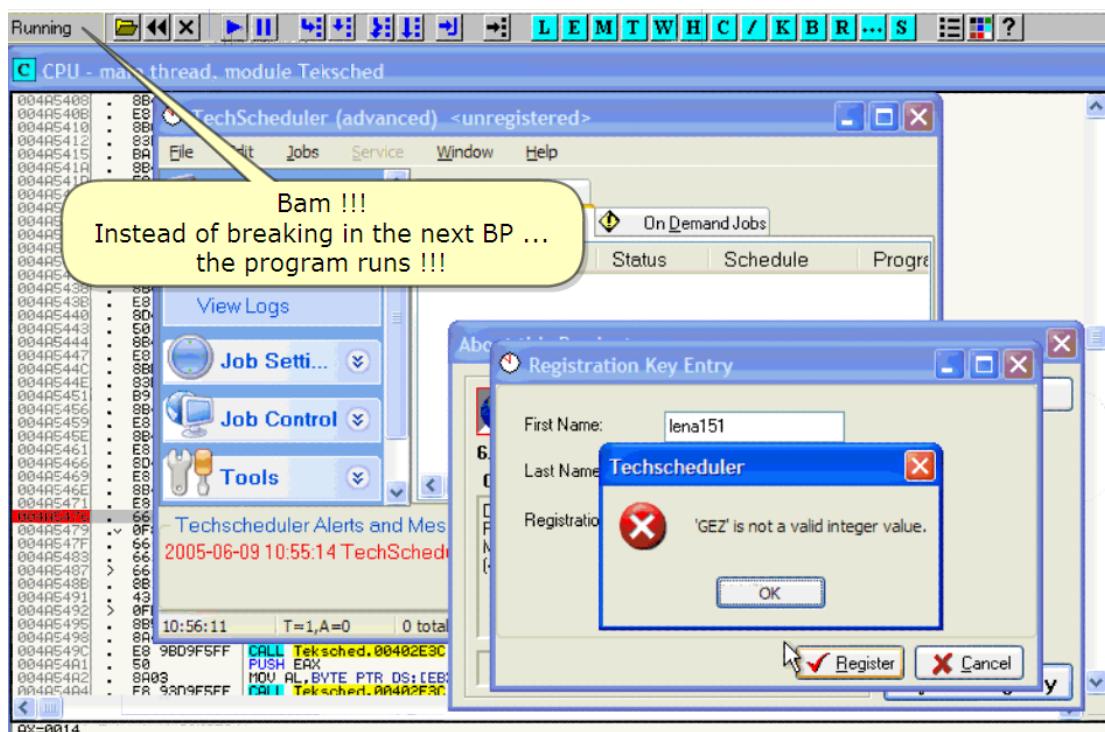
And place BP here to test if we can skip all these calls

그 calls 을 skip 할 것인지 test 하기 위해 여기에 BP 설치 해.

```
00445403 . 8B45 DC MOU EAX,DWORD PTR SS:[EBP-24] 00445408 . E8 3CFFEF5FF CALL Tekshed.0040524C 00445410 . 8BC8 E8 MOU ECX,ECX 00445412 . 83C8 05 SUB ECX,5 00445415 . BA 010000 0044541A . 8B45 DC MOU EAX,DWORD PTR SS:[EBP-24] 0044541D . E8 8A00FF CALL Tekshed.0040524C 00445422 . 8045 E4 LEA ECX,[EBP-10] 00445425 . 50 PUSH ECX 00445426 . 8B45 DC MOU EAX,DWORD PTR SS:[EBP-24] 00445429 . E8 1EFFEF5FF CALL Tekshed.0040524C 0044542E . 8B0D E8 MOU EDX,EDX 00445430 . 83C8 04 SUB EDX,4 00445433 . 8B45 DC MOU EAX,DWORD PTR SS:[EBP-24] 00445436 . E8 82000000 CALL Tekshed.0040524C 00445439 . 8B45 DC MOU EAX,DWORD PTR SS:[EBP-24] 0044543B . E8 8C00F6FF CALL Tekshed.0040524C 00445440 . 8045 E8 LEA EAX,DWORD PTR SS:[EBP-20] 00445443 . 50 PUSH ECX 00445444 . 8B45 DC MOU EAX,DWORD PTR SS:[EBP-24] 00445447 . E8 00FEF5FF CALL Tekshed.0040524C 0044544C . 8B0D E8 MOU EDX,EDX 0044544E . 83C8 02 SUB EDX,2 00445451 . 8B45 DC MOU EAX,DWORD PTR SS:[EBP-24] 00445456 . E8 4E00F6FF CALL Tekshed.0040524C 00445459 . 8B45 E8 MOU EAX,DWORD PTR SS:[EBP-20] 00445461 . E8 0A4CF6FF CALL Tekshed.0040524C 00445466 . 8D45 DC LEA EAX,DWORD PTR SS:[EBP-24] 00445469 . E8 1EFFBF5FF CALL Tekshed.00404F8C 0044546E . 8B45 F4 MOV EAX,DWORD PTR SS:[EBP-C] 00445471 . E8 D6FDFF5F CALL Tekshed.0040524C 00445472 . 66:85C0 TEST AX,AX 00445479 . 66:0F8E 23A10000 JBE Tekshed.004A55A2 0044547F . 66:8945 D2 MOU WORD PTR SS:[EBP-2E],AX 00445483 . 66:BF 0100 MOU DI,1 00445487 . > 66:BE 0100 MOU SI,1 0044548E . 83D1 64575A0 INC EBX 00445491 . > 66:0F8E 23A10000 MOUZX EBX,DI 00445492 . 66:85C7 MOU EDX,DWORD PTR SS:[EBP-C] 00445495 . 8B55 F4 MOU AL,BYTE PTR DS:[EDX+EAH-1] 00445498 . E8 9B09F5FF CALL Tekshed.00402E3C 0044549C . 50 PUSH ECX 00445491 . 8A03 MOU AL,BYTE PTR DS:[EBX] 00445492 . 8A03 MOU AL,BYTE PTR DS:[EBX] 00445494 . FA 8A09FFFF CALL Tekshed.00402E3C
```

And press F9

F9 를 눌러.



Bam !!!

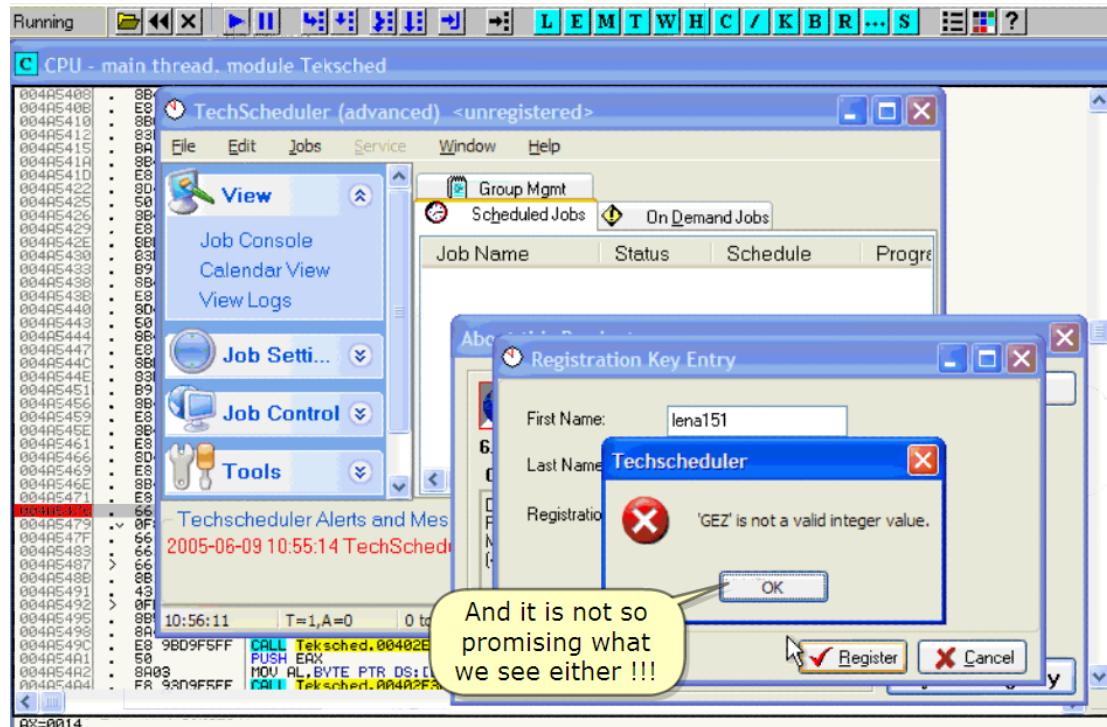
오 !!!

Instead of breaking in the next BP ...

다음 BP 대신에 멈췄다.

The program runs !!!

Program 실행중 !!!



And it is not so promising what we see either !!!

우리가 보는 것 또한 유망하지 않다.

INFO :

This is suspicious for the program detecting some mistake somewhere.

이것은 어딘가에 몇 가지 실수를 감지하는 program 이 의심된다.

Always be suspicious for code integrity checks but it is not in this case.

항상 code 의 진실성 check 가 의심된다. 이 case 에서는 그렇지 않다.

You can check that by removing the BP's and verifying.

너는 BP 를 삭제하는 것과 검증하는 것을 check 할 수 있다.

You get the same error message.

너는 똑같은 error message 를 얻는다.

In this case, it's the program's bizarre way of telling us we used a bad serial.

이 경우, 이것은 program's 은 우리에게 전하는 기이한 방법이다. 우리는 bad serial 을 사용했다.

BTW, programs can also check for software BP's !!!!

By the way, programs 은 software BP's 를 위해 check 할 수 있다. !!!!

We will see all this more in detail in later parts in this series.

우리는 이 series 의 나중 part 에서 상세하게 이 모든 것을 좀 더 볼 수 있다.

Let's find out where the program has detected the mistake

Program 이 어디에서 잘못 발견했는지 찾아보자.

Return to Olly

Olly 로 돌아가자.

(click in Olly)

(Olly 에서 click 해)

Scroll up ...

Scroll 올려 ...

C CPU - main thread, module Teksched

```
004453CD 7D 19 JGE SHORT Teksched.004A53E8
004453CF 807D 08 00 CMP BTTE PTR SS:[EBP+8],0
004453D3 8F85 90040000 JNC Teksched.004A5376
004453D9 B8 58594000 MOU EAX,Teksched.004A5958
004453DE E8 153EF9FF CALL Teksched.004A391F8
004453E3 8E095 BCFFFFF LEA EDX,0MORD PTR SS:[EBP-144]
004453E8 > E9 8E040000 JMF Teksched.004A5376
004453E9 8095 F3 MOV EAX,0MORD PTR SS:[EBP-C]
004453E1 E8 D6EFFFFF CALL Teksched.004A4A3CC
004453F1 8095 BCFFFFF LEA EDX,0MORD PTR SS:[EBP-144]
004453F6 8095 DC MOV EAX,0MORD PTR SS:[EBP-24]
004453FC E8 1CFDF5FF CALL Teksched.004A51F0
00445404 8045 F4
00445407 50
00445408 8B45 DC
0044540B E8 3CFFEF5FF CALL Teksched.004A5240
00445410 8BC8 EAX MOU ECX,EAX
00445411 SUB ECX,5
00445415 89E9 05 MOU EDX,5
00445416 B8 31000000 MOU EAX,0MORD PTR SS:[EBP-24]
0044541A 8B45 DC LEA EAX,0MORD PTR SS:[EBP-24]
0044541D E8 8000F6FF CALL Teksched.004A540C
00445422 8045 E4 LEA EAX,0MORD PTR SS:[EBP-10]
00445425 PUSH EAX
00445426 50
00445429 8B45 DC MOU EAX,0MORD PTR SS:[EBP-24]
0044542D E8 1EFEF5FF CALL Teksched.004A5240
0044542E 8BD0 MOU EDX,EAX
00445430 83EA 04 SUB EDX,4
00445433 B9 02000000 MOU ECX,2
00445438 8B45 DC MOU EAX,0MORD PTR SS:[EBP-24]
0044543B E8 6C00F6FF CALL Teksched.004A540C
00445440 8045 E0 LEA EAX,0MORD PTR SS:[EBP-20]
00445443 50
00445444 8B45 DC MOU EAX,0MORD PTR SS:[EBP-24]
00445447 E8 00FEF5FF CALL Teksched.004A5240
0044544C 8BD0 MOU EDX,EAX
0044544E 83EA 02 SUB EDX,2
00445451 B9 03000000 MOU ECX,3
00445456 8B45 DC MOU EAX,0MORD PTR SS:[EBP-24]
00445459 E8 4E00F6FF CALL Teksched.004A540C
0044545E 8B45 E8 MOU EAX,0MORD PTR SS:[EBP-20]
00445461 E8 0A4CF6FF CALL Teksched.004A6070
00445466 8045 DC LEA EAX,0MORD PTR SS:[EBP-24]
00445469 E8 1E05F5FF CALL Teksched.004A540C
0044546E 8B45 F4 MOV EAX,0MORD PTR SS:[EBP-0]
00445471 FA DAF0FFFF CALL Teksched.004A5240
```

AX=0014

ASCII "Enter a Key value now.." user32.77D1885A

ntdll.RtRaiseException

user32.77D1885A

And place the former BP back

옛날 BP 에 설치 해.

번역 주)어디에서 잘못됐는지 찾기 위해 예전에 BP 를 설치했다 삭제한 곳에 다시 BP 를 설치합니다.

그리고 어디에서 잘못됐는지 trace 합니다.

If we run the registration again, Olly will break in the BP again

다시 registration 을 실행한다면, Olly 는 다시 BP 에서 멈출 것이다.

:)

And step F8 to see what is going on where ...

어떻게 되는지 보기 위해 F8 을 눌러.

C CPU - main thread, module ntdll

```

7C90EAF0 0B1C24 MOU EBX, DWORD PTR SS:[ESP]           Tekshed.00408774
7C90EBF3 51 PUSH ECX
7C90EBF5 53 PUSH EBX
7C90EBF5 E8 C78C0200 CALL ntdll.7C90EB0A
7C90EBFC v 74 0C OR AL, AL
7C90EBFF 59 JS SHORT ntdll.7C90EB0A
7C90EB02 50 C0 POP ECX
7C90EB02 51 EB 00 PUSH 0
7C90EB03 6A 00 PUSH ECX
7C90EB03 E8 11EBFFFF CALL ntdll.ZwContinue
7C90EB03 v EB 0B JMP SHORT ntdll.7C90EB15
7C90EB04 5B POP EBX
7C90EB0B 59 PUSH ECX
7C90EB0C 6A 00 PUSH 0
7C90EB0D 51 PUSH ECX
7C90EB0D 53 PUSH EBX
7C90EB10 E8 30F7FFFF CALL ntdll.ZwRaiseException
7C90EB15 89C4 EC ADD ESP, -14
7C90EB18 898424 MOU DWORD PTR SS:[ESP], EAX
7C90EB1B C74424 04 0100 MOU DWORD PTR SS:[ESP+4], 0
7C90EB23 895C24 08 MOU DWORD PTR SS:[ESP+8], EBX
7C90EB27 C74424 10 0000 MOU DWORD PTR SS:[ESP+10], 0
7C90EB2F 54 PUSH ESP
7C90EB30 E8 77000000 CALL ntdll.RtlRaiseException
7C90EB30 C2 0000 RETN 0
7C90EB31 90 NOP
7C90EB32 90 NOP
7C90EB33 90 NOP
7C90EB34 90 NOP
7C90EB35 90 NOP
7C90EB36 90 NOP
7C90EB37 90 NOP
7C90EB38 90 NOP
7C90EB39 90 NOP
7C90EB3A 90 NOP
7C90EB3B 90 NOP
7C90EB3C 90 NOP
7C90EB3D 55 PUSH EBP
7C90EB3E 8BEC MOU EBP, ESP
7C90EB40 89EC 50 SUB ESP, 50
7C90EB43 894424 0C MOU DWORD PTR SS:[ESP+C], EAX
7C90EB43 64R1 18000000 MOU EAX, DWORD PTR FS:[18]
7C90EB47 8B80 A4010000 MOU EAX, DWORD PTR DS:[EAX+A4]
7C90EB53 898424 MOU DWORD PTR SS:[ESP], EAX
7C90EB56 C74424 04 0000 MOU DWORD PTR SS:[ESP+4], 0
7C90EB56 C74424 08 0000 MOU DWORD PTR SS:[ESP+8], 0
7C90EB66 C74424 10 0000 MOU DWORD PTR SS:[ESP+10], 0
7C90EB66 54 PUSH ESP
7C90EB6F E8 38000000 CALL ntdll.RtlRaiseException
7C90EB74 8B8424 MOU EAX, DWORD PTR SS:[ESP]
7C90EB77 RRF5 HNU FSP, FRP

```

Aha, this is not good

Aha, this is not good

아하, 이것은 좋지 않은데 ...

Right, press <-> to see what caused this

좋아, 눌러 <-> 이것이 어디에서 발생했는지 보기 위해

번역 주)마이너스(-): 자기가 code 를 실행하고 있는데 갑자기 다른 곳으로 왔다면 -를 눌러. 왔던 곳으로 돌아갈 수 있다.

INFO :

Pressing the minus sign on your keyboard lets you return on your steps(without effectively undoing them of course)

너의 keyboard 의 Minus sign 을 눌러. 네가 진행해 왔던 곳으로 돌아가게 할거야(효과적으로 그것들을 바로 잡지는 않는다)

And we land back on the call that caused this

발생했던 Call로 돌아올 것이다.

... only a few lines before the second BP I wanted to break on :(

내가 원할 때 멈출 수 있는 2 번째 BP 전에 오직 몇 line 이 있다.

C CPU - main thread, module Tekshed

00445461 . E8 0A4C6FF CALL Tekshed.00404070 00445466 . 8045 DC LEA EAX,DWORD PTR SS:[EBP-24] 00445469 . E8 1E8BF5FF CALL Tekshed.00404F8C 0044546E . BB45 F4 MOU EAX,DWORD PTR SS:[EBP-C] 00445471 . E8 D6FD5FF CALL Tekshed.00405240 00445476 . 66:85C0 TEST AX,AX 00445479 > . 0F86 2301000 MOU WORD PTR SS:[EBP-24] 0044547F . 66:8945 D2 MOU DI,DI 00445483 . 66:EF 0100 MOU SI,SI 00445487 > . 66:BE 0100 MOU EBX,DWORD PTR DS:[EBP-1] 0044548B . BB1D 64575A0 INC EBX 00445491 . 43 00445495 . 0FB7C7 MOUZX EAX,DI 00445498 . BB55 F4 MOU EDX,DWORD PTR SS:[EBP-C] 0044549B . 804402 FF FF MOU AL,BYTE PTR DS:[EDX+EBX-1] 004454A1 . E8 9BD9FF FF CALL Tekshed. 004454A2 . 56 PUSH EBP 004454A3 . 50 MOU PL,BYTE PTR DS:[EBP-1] 004454A4 . 5A POP EDX 004454A9 . 5A00 CMP DL,AL 004454AC . 0FB7C7 JNC Tekshed.004A5589 004454B2 . 83E8 01 MOUZX EAX,DI 004454B5 . 85C0 AND EAX,1 004454B8 . 75 66 TEST EAX,EAX 004454B9 > . 75 66 JNE SHORT Tekshed.004A5522 004454BC . 0D95 BCFFEFFI LEA EDX,DWORD PTR SS:[EBP-144] 004454C2 . 0FB7C6 MOUZX EAX,SI 004454C5 . E8 1EE4F5FF CALL Tekshed.004038E8 004454CA . 0D95 BCFFEFFI LEA EDX,DWORD PTR SS:[EBP-144] 004454D0 . 0D45 EC LEA EAX,DWORD PTR SS:[EBP-14] 004454D3 . E8 18FDFF5F CALL Tekshed.004051F0 004454D8 . 66:83FE 09 CMP SI,9 004454D9 > . 76 24 JNE SHORT Tekshed.004A5502 004454E0 . 0D85 B8EFFFFI LEA EAX,DWORD PTR SS:[EBP-148] 004454E4 . 6B55 EC MOU EDX,DWORD PTR SS:[EBP-14] 004454E7 . 8A52 01 MOU AL,BYTE PTR DS:[EDX+1] 004454F1 . E8 05CF5FF CALL Tekshed.00405747 004454F5 . 8995 B8EFFFFI MOU EDX,DWORD PTR SS:[EBP-148] 004454F8 . 0D45 EC LEA EAX,DWORD PTR SS:[EBP-24] 004454F9 > . E8 57FDFF5F CALL Tekshed.00405589 00445502 . E8 87000000 JMP Tekshed.00405589 00445503 . 0D85 B4FEFFF LEA EAX,DWORD PTR SS:[EBP-14C] 00445508 . 8B55 EC MOU EDX,DWORD PTR SS:[EBP-14] 0044550F . AA13 MNH NI .RVTF PTR DS:[FFN1]	Tekshed.0040445A Mmm, probably another verification of the key 9C 74 Tekshed.0040445A ntdll.RtlRaiseException 00120DFFC Tekshed.0040445A ntdll.RtlRaiseException Tekshed.00404FD6 Tekshed.00404FD6
--	--

Mmmm, probably another verification of the key

Scroll up to see better

음, 아마 다른 key 검증이 있을 것이다.

좀 더 보기 좋게 Scroll 올려.

Mmmm,

Let's just NOP the call and return to see what happens.

음,

Call 을 NOP 하고 무슨 일이 일어나는지 보기 위해 return 해.

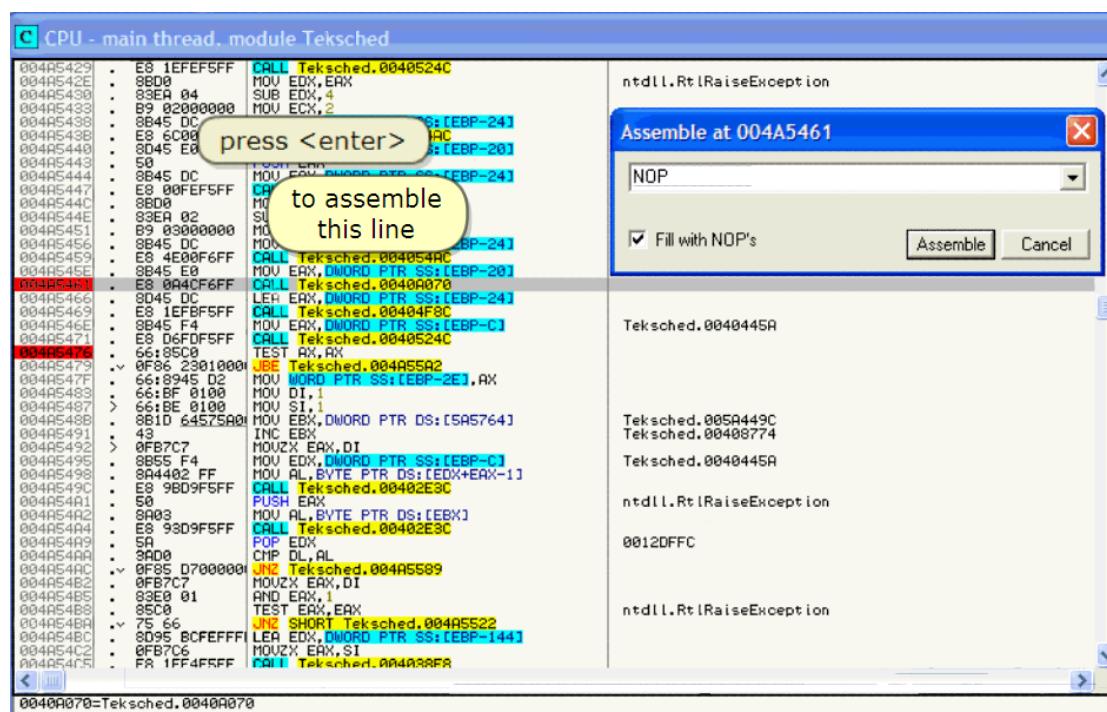
INFO :

In later parts, we will dig deeper into this kind of problems.

나중 part 에서, 우리는 이 종류의 문제에 깊이 파고들 것입니다.

This means to step IN the call that caused this of course :)

물론 이것이 발생했던 Call 로 들어가자.



Press <enter>

To assemble this line

Enter 를 눌러.

이 line 을 assemble 하기 위해

And remove the BP in the changed code

바뀐 code 에서 BP 를 삭제 해.

And set it just above

이 위를 바로 set 해.

Right

좋아

INFO :

This is only to avoid having to re-enable the BP's at restart(see previous parts)

이것은 restart 할 때 오직 BP 를 재사용을 회피할 때 사용한다.(이전 part 를 봄)

번역 주)restart 해서 BP 가 쓸모 없을 때 사용한다.

Run because the program is paused

실행 해. 왜냐하면 program 이 멈춰있다.

Yes, we were already in the call that causes this

예, 우리는 이미 발생한 call에 있다.

CPU - main thread, module Tekshed

```
004A53E8 > 8D95 BCFFFFFF LEA EDX, [DWORD PTR SS:[EBP-144]]  
004A53E8 BB45 F4 MOU ECX, [DWORD PTR SS:[EBP-C]]  
004A53F1 EB D6EFFFFF CALL Tekshed.004A43CC  
004A53F6 BB95 BCFFFFFF LEA EDX, [DWORD PTR SS:[EBP-144]]  
004A53F6 BB45 DC LEA ECX, [DWORD PTR SS:[EBP-24]]  
004A53F9 EB ECFDF5FF CALL Tekshed.004051FB  
004A5404 BB45 00 MOU ECX, [DWORD PTR SS:[EBP-C]]  
004A5407 BB45 00 PUSH ECX  
004A540B BB45 00 MOU ECX, [DWORD PTR SS:[EBP-24]]  
004A540B CALL Tekshed.0040524C  
004A5410 EB 00 00 00 00 LEA EDX, [DWORD PTR SS:[EBP-144]]  
004A5412 BB45 00 MOU ECX, ECX  
004A5415 B3 E9 05 SUB EDX, ECX  
004A5415 B8 61900000 MOU EDX, 1  
004A5418 BB45 DC MOU ECX, [DWORD PTR SS:[EBP-24]]  
004A541D EB 8A00F6FF CALL Tekshed.004054AC  
004A5422 BB45 E4 LEA ECX, [DWORD PTR SS:[EBP-1C]]  
004A5425 B9 50 PUSH ECX  
004A5426 BB45 DC MOU ECX, [DWORD PTR SS:[EBP-24]]  
004A5429 EB 1EFFEF5F CALL Tekshed.0040524C  
004A542E BB00 MOU EDX, ECX  
004A5430 B3 E9 04 SUB EDX, 4  
004A5433 B9 02000000 MOU ECX, 2  
004A5433 BB45 DC MOU ECX, [DWORD PTR SS:[EBP-24]]  
004A5438 EB 6C00F6FF CALL Tekshed.004054AC  
004A5440 BB45 E0 LEA ECX, [DWORD PTR SS:[EBP-20]]  
004A5443 B9 50 PUSH ECX  
004A5444 BB45 DC MOU ECX, [DWORD PTR SS:[EBP-24]]  
004A5447 EB 00FEF5FF CALL Tekshed.0040524C  
004A544C BB00 MOU EDX, ECX  
004A544E B3 E9 02 SUB EDX, 2  
004A5451 B9 03000000 MOU ECX, 3  
004A5456 BB45 DC MOU ECX, [DWORD PTR SS:[EBP-24]]  
004A5459 EB 4E00F6FF CALL Tekshed.004054AC  
004A545E BB45 E0 MOU ECX, [DWORD PTR SS:[EBP-20]]  
004A5461 B9 00 NOP  
004A5462 B9 00 NOP  
004A5463 B9 00 NOP  
004A5464 B9 00 NOP  
004A5465 B9 00 NOP  
004A5466 B9 00 NOP  
004A5467 B9 00 NOP  
004A5468 BB45 DC LEA EDX, [DWORD PTR SS:[EBP-24]]  
004A5469 EB 1EFFEF5F CALL Tekshed.00404F80  
004A546E BB45 F4 MOU ECX, [DWORD PTR SS:[EBP-C]]  
004A5471 EB D6FDFF5F CALL Tekshed.0040524C  
004A5476 B6 95C0 TEST RX, RX  
004A5479 AFAA 23A1AAA JFA Tekshed.A04AE5A2  
Stack address=0012E3F8 (0E87B)  
EDX=00000F48Z (0x0) L.Rt!RaiseException()  
Jump from 004A53CD
```

1

Ok. Now let's step F8 to see if there are no errors by NOP'ing the call

좋아. 이제 NOP'ing 된 call에 의해 아무 error가 없는지 보기 위해 F8을 눌러.

C CPU - main thread, module Teksched

```

004453EE1: 8B45 F4 MOV EAX, DWORD PTR SS:[EBP-0]
004453F11: E8 06FFFFFF CALL Teksched.004A49CC
004453F21: 8D95 BCFFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
004453F31: 8D45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004453F41: E8 EC0DF5FF CALL Teksched.004051F0
004454041: 8D45 F4 MOV EAX, DWORD PTR SS:[EBP-0]
004454071: 50 PUSH EAX
004454081: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004454091: E8 3C0EF5FF CALL Teksched.0040524C
004454101: 8BC8 SUB ECX, 5
004454121: 83E9 05 MOV EDX, 1
004454151: BA 01000000 MOV EAX, DWORD PTR SS:[EBP-24]
004454161: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
0044541D1: E8 8A00F6FF CALL Teksched.004054AC
004454221: 8D45 E4 LEA EAX, DWORD PTR SS:[EBP-10]
004454251: 50 PUSH EAX
004454261: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004454291: E8 1EFEF5FF CALL Teksched.0040524C
0044542E1: 8BD8 MOV EDX, EAX
004454301: 83E9 04 SUB EDX, 4
004454331: B9 02000000 MOV EAX, DWORD PTR SS:[EBP-24]
004454381: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
0044543B1: E8 6C00F6FF CALL Teksched.004054AC
0044543C1: 8D45 E0 LEA EAX, DWORD PTR SS:[EBP-20]
0044543D1: 8B45 E8 PUSH EAX
004454411: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004454471: 50 NOP
0044544C1: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
0044544E1: 8B45 E8 MOV EAX, DWORD PTR SS:[EBP-20]
004454511: B9 00000000 NOP
004454561: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004454591: E8 4E00F6FF CALL Teksched.004054AC
0044545E1: 8B45 E0 MOV EAX, DWORD PTR SS:[EBP-20]
004454611: 90 NOP
004454621: 90 NOP
004454631: 90 NOP
004454641: 90 NOP
004454651: 90 NOP
004454661: 8D45 DC LEA EAX, DWORD PTR SS:[EBP-24]
004454691: E8 1EFEF5FF CALL Teksched.00404F8C
0044546E1: 8B45 F4 MOV EAX, DWORD PTR SS:[EBP-0]
004454711: E8 D60DF5FF CALL Teksched.0040524C
004454761: 66:95C0 TEST AX, AX
004454791: 0F8E 23010000 JBE Teksched.004A55A2
0044547F1: 6A:R945 N2 HLT INT PTR SS:[EBP-2F1], AX

```

So, we passed the NOPed call ...

So, we passed the NOPed call ...

그래서, NOP 된 call 을 피한다.

Mmmm, ok, scroll up

음, ok scroll 올려.

C CPU - main thread, module Teksched

```

004A53BF1: E9 B2040000 JMP Teksched.004A5876
004A53C41: 8B45 F4 MOV EAX, DWORD PTR SS:[EBP-0]
004A53C51: 8B45 F4 JMP Teksched.004A5876
004A53C61: E8 06FFFFFF CALL Teksched.004A430C
004A53C71: 8D95 BCFFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
004A53C81: 8D45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004A53C91: E8 EC0DF5FF CALL Teksched.004051F0
004A53C91: 8D45 F4 MOV EAX, DWORD PTR SS:[EBP-0]
004A54041: 50 PUSH EAX
004A54081: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004A54091: E8 3C0EF5FF CALL Teksched.0040524C
004A54101: 8BC8 SUB ECX, 5
004A54121: 83E9 05 MOV EDX, 1
004A54151: BA 01000000 MOV EAX, DWORD PTR SS:[EBP-24]
004A54161: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004A541D1: E8 8A00F6FF CALL Teksched.004054AC
004A54221: 8D45 E4 LEA EAX, DWORD PTR SS:[EBP-10]
004A54251: 50 PUSH EAX
004A54291: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004A54321: E8 1EFEF5FF CALL Teksched.0040524C
004A54331: 8BD8 MOV EDX, EAX
004A54361: 83E9 04 SUB EDX, 2
004A54381: B9 02000000 MOV EAX, DWORD PTR SS:[EBP-24]
004A543B1: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004A543C1: E8 6C00F6FF CALL Teksched.004054AC
004A543D1: 8D45 E0 PUSH EAX
004A54411: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004A54441: E8 00FEF5FF CALL Teksched.0040524C
004A544C1: 8BD8 MOV EDX, EAX
004A544E1: 83E9 02 SUB EDX, 2
004A54511: B9 03000000 MOV EAX, DWORD PTR SS:[EBP-24]
004A54561: 8B45 DC MOV EAX, DWORD PTR SS:[EBP-24]
004A54591: E8 4E00F6FF CALL Teksched.004054AC
004A54611: 8B45 E8 MOV EAX, DWORD PTR SS:[EBP-20]
004A54611: 90 NOP

```

First, remove this BP, it's no longer needed(I hope :))

First, remove this BP, it's no longer needed(I hope :))

먼저, 이 BP 를 삭제 해, 더 이상 필요하지 않다.(나는 희망한다 :))

Continue F8

F8 눌러.

C CPU - main thread, module Teksched

```
004A5556 . E8 19FCFF CALL Teksched.00405174
004A5558 . 8B95 B8FEFFF MOV EDX, DWORD PTR SS:[EBP-150]
004A5561 . 8045 DC LEA EAX, DWORD PTR SS:[EBP-24]
004A5564 . E8 EBFCF5FF CALL Teksched.00405254
004A5569 > EB 1E JMP SHORT Teksched.004A5589
004A5571 . 8085 ACFEFFF LEA EAX, DWORD PTR SS:[EBP-154]
004A5574 . 8B55 EC MOU EDX, DWORD PTR SS:[EBP-143]
004A5576 . 8B95 10 CMP EAX, DWORD PTR DS:[EDX]
004A5578 . E8 F9 10 CALL Teksched.00405174
004A5580 . 8045 DC CMP EAX, DWORD PTR SS:[EBP-154]
004A5581 . E8 CBFCF5FF CALL Teksched.00405254
004A5584 > 46 INC ESI
004A5589 . 83C3 02 ADD EBX, 2
004A558A . 8B45 E4 CMP SI, 1B
004A558D . 8B85 88FE 1B MOU EAX, DWORD PTR SS:[EBP-150]
004A5591 > 0F85 FFBEFFF JNZ Teksched.004A5492
```

Jump is taken
004A5492=Teksched.004A5492

Teksched.00440002

0012E548

ASCII "Error! You are attempting to register the wro

C CPU - main thread, module Teksched

```
004A5556 . E8 19FCFF CALL Teksched.00405174
004A5558 . 8B95 B8FEFFF MOV EDX, DWORD PTR SS:[EBP-150]
004A5561 . 8045 DC LEA EAX, DWORD PTR SS:[EBP-24]
```

Mmm, it seems we're running in circles like a squirrel in a wheel ...

```
004A5578 . 8B95 ACFEFFF MOV EDX, DWORD PTR SS:[EBP-154]
004A5581 . 8045 DC LEA EAX, DWORD PTR SS:[EBP-24]
004A5584 . E8 CBFCF5FF CALL Teksched.00405254
004A5589 > 46 INC ESI
004A558A . 83C3 02 ADD EBX, 2
004A558B . 8B45 E4 CMP SI, 1B
004A5591 > 0F85 FFBEFFF JNZ Teksched.004A5492
```

BTW, I tried it out and we come back here several times, so, find the jump out of the loop and BP or put BP after loop if there is no jump out of it ...

```
004A5597 . 47 INC EDI
004A5598 . 66:FF40 D2 DEC WORD PTR SS:[EBP-2E]
004A559C > 0F85 ESFEFFF JNZ Teksched.004A5487
004A559D . 8095 A8FEFFF LEA EAX, DWORD PTR SS:[EBP-158]
004A559E . E8 7442F6FF CALL Teksched.00409824
004A559F . 8B85 A8FEFFF MOU EAX, DWORD PTR SS:[EBP-158]
004A55B0 . 50 PUSH EAX
004A55B1 . 8085 A0FEFFF LEA EAX, DWORD PTR SS:[EBP-160]
004A55B2 . 50 PUSH EAX
004A55B3 . B9 02000000 MOU ECX, 2
004A55C1 . 8045 D8 MOU EDX, 1
004A55C8 . 8B45 D8 MOU EAX, DWORD PTR SS:[EBP-28]
004A55CB . E8 DCFEFFF CALL Teksched.0040854AC
004A55D0 . 8B85 A8FEFFF MOU EAX, DWORD PTR SS:[EBP-160]
004A55D6 . 8D95 A4FEFFF LEA EDX, DWORD PTR SS:[EBP-150]
004A55DC . E8 4342F6FF CALL Teksched.00409824
004A55E1 . 8B95 A4FEFFF MOU EDX, DWORD PTR SS:[EBP-150]
004A55E7 . 58 POP EAX
004A55E8 . E8 ABFDFF CALL Teksched.00405398
004A55E9 > 74 19 JE SHORT Teksched.004A5608
004A55F3 . 8070 00 00 CMP BTTE PTR SS:[EBP+8], 0
004A55F9 > 0F85 7D020000 JNZ Teksched.004A5876
004A55F9 . B8 78534000 MOU EAX, Teksched.004A5978
004A55F6 . E8 F53BF9FF CALL Teksched.004391F8
004A5603 > E9 6E020000 JMP Teksched.004A5876
004A5608 . 66:81FE BC02 MOU SI, 258
004A560C . 66:81FE BC02 CMP SI, 2BC
004A5611 > 7F 19 JRF SHRT Teksched.004A5620
```

Jump is taken
004A5492=Teksched.004A5492

0012E548

ASCII "Error! You are attempting to register the wro

Mmm, it seems we're running in circles like a squirrel in a wheel ...

음, 우리가 circle에서 실행되는 것으로 보인다. 마치 다행쥐가 챗바퀴 도는 것처럼.

BTW, I tried it out and we come back here several times,

By the way, 우리는 이것에 도전 했다. 그리고 우리는 많은 시간 동안 여기로 돌아왔다.

so, find the jump out of the loop and BP or put BP after loop if there is no jump out of it ...

그래서, loop 를 탈출하는 jump 를 찾았다. 그리고 만약에 이것을 벗어나는 jump 가 없다면 loop 후에 BP 를 설치했다.

Ok, BP and run ...

Ok, BP 설치하고 실행 해 ...

C CPU - main thread, module Teksched

```

004A5571 . 8B55 EC MOU EDX, DWORD PTR SS:[EBP-14]
004A5574 . 8A12 MOV DL,BYTE PTR DS:[EDX]
004A5576 . E8 F9FBFF CALL Teksched.00405174
004A5578 . 8B95 ACFFEFFF MOU EDX, DWORD PTR SS:[EBP-14]
004A5581 . 8D45 DC LEA EAX,DWORD PTR SS:[EBP-14]
004A5584 . E8 CBFCFF CALL Teksched.00405174
004A5589 > 46 INC ESI
004A558C . 89C3 02 ADD EBX,2
004A558E . 66:83FE 1B OR SI,1B
004A5591 . 8B85 FBFFFF CALL Teksched.004A5492
004A5592 . 47 1C INC EDX
004A5598 . 66:FF40 D2 DEC WORD PTR SS:[EBP-2E]
004A55A2 > 8095 A6FFFFFF LEA EDX,DWORD PTR SS:[EBP-14]
004A55A8 . 8B45 E4 MOU EAX,DWORD PTR SS:[EBP-14]
004A55B0 . E8 7442F6FF CALL Teksched.00409824
004A55B3 . 8B85 R8FFFF MOU EAX,DWORD PTR SS:[EBP-14]
004A55B7 . 50 PUSH EAX
004A55B8 . 8085 A0FFFFFF LEA EAX,DWORD PTR SS:[EBP-160]
004A55B9 . 50 PUSH EAX
004A55B9 . B9 02000000 MOU ECX,2
004A55C3 . BA 01000000 MOV EDX,1
004A55C8 . 8B45 D8 MOU EAX,DWORD PTR SS:[EBP-28]
004A55CB . E8 DCFFEF5F CALL Teksched.004054AC
004A55D0 . 8B85 A0FFFFFF MOU EAX,DWORD PTR SS:[EBP-160]
004A55D3 . 8095 A4FFFFFF LEA EDX,DWORD PTR SS:[EBP-15C]
004A55DC . E8 4342F6FF CALL Teksched.00409824
004A55E1 . 8B95 A4FFFFFF MOU EDX,DWORD PTR SS:[EBP-15C]
004A55E7 . 58 POP EAX
004A55E8 . E8 ABFDFF CALL Teksched.00405398
004A55ED .> 74 19 JE SHORT Teksched.004A5608
004A55ED . 807D 00 00 CMP BYTE PTR SS:[EBP+1],0
004A55F3 > 0F85 70020000 JNC Teksched.004A5876
004A55F6 . 8B 50000000 MOU EAX,Teksched.004A5876
004A5603 > 50 6E000000 CALL Teksched.004391F8
004A5608 .> 50 BE 5002 JNP Teksched.004A5876
004A560C . 66:81FE BC02 CMP SI,256
004A5611 .> 76 19 JBE SHORT Teksched.004A5620
004A5613 . 807D 00 00 CMP BYTE PTR SS:[EBP+1],0
004A5617 > 0F85 59020000 JNC Teksched.004A5876
004A561D . 8B E4534000 MOU EAX,Teksched.004A59E4
004A5622 . E8 D13BF9FF CALL Teksched.004391F8
004A5627 > E9 4A020000 JNP Teksched.004A5876
004A562F > B1 F05F5AAA MUW FAX,DWORD PTR DS:[5A55C0]

```

Jump is taken
004A5492=Teksched.004A5492

this one ;)

Notice both JNZ jumping above if executed.
This one and

0012E548

ASCII "Error! You are attempting to register the wro

ASCII "Error! You are attempting to register the wro

Notice both JNZ jumping above if executed.

This one and this one ;)

실행될 경우, JNZ 가 모두 위에 있습니다.

이것 하나와 이것 하나다 ;)

C CPU - main thread, module Teksched

```

00445571 . 6B55 EC MOU EDX, DWORD PTR SS:[EBP-14]
00445574 . 8A12 MOU DL, BYTE PTR DS:[EDX]
00445576 . E8 F9FBF5FF CALL Teksched.00405174
00445578 . 00445581
00445580 . 00445584
00445581 . 00445589
00445582 . 0044558A
00445583 . 0044558D
00445591 . ^ 8F85 FBFEFFF JNZ Teksched.004A5492
00445597 . 47 INC EDI
00445598 . 66:FF40 D2 DEC WORD PTR SS:[EBP-2E]
0044559C . ^ 0F85 ESFFFF JNZ Teksched.004A5487
0044559D > 90445592 00445592 LEA EDX, DWORD PTR SS:[EBP-158]
004455A8 . 47 INC EDI
004455B0 . 66:FF42 F6FF CALL Teksched.00409824
004455B2 . 8B85 A8FFFF MOU EAX, DWORD PTR SS:[EBP-158]
004455B6 . 50 PUSH EAX
004455B7 . 8F85 A0FFFF MOU EAX, DWORD PTR SS:[EBP-160]
004455B8 . 50 PUSH EAX
004455B9 . 8B45 D8 MOU EDX, 2
004455CB . 8B45 DC MOU EDX, 1
004455C2 . 8B85 00000000 MOU EDX, 0
004455C8 . 004455C8 DCFEEFF9 CALL Teksched.0040854AC
004455D0 . 8B85 A8FFFF MOU EAX, DWORD PTR SS:[EBP-160]
004455D1 . 004455D1 004455D1 R095 A4FEFFF LEA EDX, DWORD PTR SS:[EBP-15C]
004455D2 . 8F85 4342F6FF CALL Teksched.00409824
004455E1 . 8B95 A4FFFF MOU EDX, DWORD PTR SS:[EBP-15C]
004455E2 . 58 POP EAX
004455E8 . 8B ABFDFF5F CALL Teksched.00405398
004455ED . ^ 74 19 JE SHORT Teksched.004A5608
004455EF . 007D 00 00 CMP BTTE PTR SS:[EBP+8], 0
004455F3 . ^ 0F85 7002000 JNC Teksched.004A5876
004455F9 . 8B 78534000 MOU EAX, Teksched.004A5978
004455F0 . 8B85 F53BF9FF CALL Teksched.004391F8
004455D0 . 8B95 BE5802 MOU SI, 258
004455D0 . 66:81FE BC02 CMP SI, 280
004455D1 . ^ 76 19 JE SHORT Teksched.004A5620
004455D3 . 007D 00 00 CMP BTTE PTR SS:[EBP+8], 0
004455D7 . ^ 0F85 5902000 MOU EAX, Teksched.004A59E4
004455D8 . 8B E4534000 MOU EAX, Teksched.004391F8
004455D9 . 8B D13BF9FF CALL Teksched.004391F8
004455D9 . ^ 8B A4620000 JNC Teksched.004A5876
004455D9 > A1 R0575AAA MUU FAX, DWORD PTR DS:[EBP+8]

```

Stack address=0012E3E4
EDX=00000024
Jump from 004A5479

So, I'm setting a BP after both looping JNZ to land after them

그래서, 나는 2 가지 JNZ looping 이 끝나는 곳에 BP 를 설치한다.

INFO :

It is very well possible that the right serial is calculated in a loop like this.

이것은 매우 잘 될 가능성이 있다. 옳은 serial 은 loop 에서 이것처럼 계산됐다.

I'm not interested in it today, I've not even verified it :)

오늘은 이것에 흥미가 없다, 나는 이것조차 검증하지 않았다 :)

C CPU - main thread, module Teksched

```

00445571 . 6B55 EC MOU EDX, DWORD PTR SS:[EBP-14]
00445574 . 8A12 MOU DL, BYTE PTR DS:[EDX]
00445576 . E8 F9FBF5FF CALL Teksched.00405174
00445578 . 00445581
00445580 . 00445584
00445581 . 00445589
00445582 . 0044558A
00445583 . 0044558D
00445591 . ^ 8F85 FBFEFFF JNZ Teksched.004A5492
00445597 . 47 INC EDI
00445598 . 66:FF40 D2 DEC WORD PTR SS:[EBP-2E]
0044559C . ^ 0F85 ESFFFF JNZ Teksched.004A5487
0044559D > 90445592 00445592 LEA EDX, DWORD PTR SS:[EBP-158]
004455A8 . 47 INC EDI
004455B0 . 66:FF42 F6FF CALL Teksched.00409824
004455B2 . 8B85 A8FFFF MOU EAX, DWORD PTR SS:[EBP-158]
004455B6 . 50 PUSH EAX
004455B7 . 8F85 A0FFFF MOU EAX, DWORD PTR SS:[EBP-160]
004455B8 . 50 PUSH EAX
004455B9 . 8B45 D8 MOU EDX, 2
004455CB . 8B45 DC MOU EDX, 1
004455C2 . 8B85 00000000 MOU EDX, 0
004455C8 . 004455C8 DCFEEFF9 CALL Teksched.0040854AC
004455D0 . 8B85 A8FFFF MOU EAX, DWORD PTR SS:[EBP-160]
004455D1 . 004455D1 004455D1 R095 A4FEFFF LEA EDX, DWORD PTR SS:[EBP-15C]
004455D2 . 8F85 4342F6FF CALL Teksched.00409824
004455E1 . 8B95 A4FFFF MOU EDX, DWORD PTR SS:[EBP-15C]
004455E2 . 58 POP EAX
004455E8 . 8B ABFDFF5F CALL Teksched.00405398
004455ED . ^ 74 19 JE SHORT Teksched.004A5608
004455EF . 007D 00 00 CMP BTTE PTR SS:[EBP+8], 0
004455F3 . ^ 0F85 7002000 JNC Teksched.004A5876
004455F9 . 8B 78534000 MOU EAX, Teksched.004A5978
004455F0 . 8B85 F53BF9FF CALL Teksched.004391F8
004455D0 . 8B95 BE5802 MOU SI, 258
004455D0 . 66:81FE BC02 CMP SI, 280
004455D1 . ^ 76 19 JE SHORT Teksched.004A5620
004455D3 . 007D 00 00 CMP BTTE PTR SS:[EBP+8], 0
004455D7 . ^ 0F85 5902000 MOU EAX, Teksched.004A59E4
004455D8 . 8B E4534000 MOU EAX, Teksched.004391F8
004455D9 . 8B D13BF9FF CALL Teksched.004391F8
004455D9 . ^ 8B A4620000 JNC Teksched.004A5876
004455D9 > A1 R0575AAA MUU FAX, DWORD PTR DS:[EBP+8]

```

Stack address=0012E3E4
EDX=00000046
Jump from 004A5479

i)

번역 주) 2 가지 loop 가(역행하는) 끝난 후에 BP 를 설치하고 F9 를 눌러서 실행했더니 정상적으로 넘어왔다. 이렇게 해서 다행히 셋바퀴 도는 것을 한 방에 끝낼 수 있다.

Now that the loops are history, continue stepping F8 again

이제 loops 는 지나갔다. 다시 F8 로 계속 진행하자.

Now it gets interesting

이제 이것에 흥미가 있다.

Mmmm, but first, let's place our BP here

음, 먼저 우리의 BP 를 이곳에 설치하자.

번역 주)Error message 가 있으니까 이것을 잘 회피하면 goodboy 에 갈 수 있으니까 흥미롭습니다!

Because so far, nothing interesting has happened after the NOPed call

지금까지, NOPed call 후에 흥미가 없었다.

And if a re-run should be necessary ...

재시작 하는 것이 필요하다면 ...

It's always interesting to remember that one can run till the next BP

항상 다음 BP 까지 실행할 수 있다는 것을 기억하는 것이 흥미롭다.

Ok. We won't need the other BPs any longer.

Scroll up ...

Ok. 우리는 더 이상 다른 BPs가 필요하지 않다.

Scroll 올려 ...

C CPU - main thread, module Tekshed

0040544C	8B00	MOV EDX, EAX
0040544E	83EA 02	SUB EDX, 2
00405451	B9 00000000	MOV ECX, 0
00405456	9845 DC	MOV ERX, DWORD PTR SS:[EBP-24]
00405459		
00405465		and remove the old ones
00405466	90	NOP
00405467	90	NOP
00405468	90	NOP
00405469	90	NOP
0040546A	8045 DC	LEA ERX, DWORD PTR SS:[EBP-24]
0040546B	E8 1EFBF5FF	CALL Tekshed.00404F8C
0040546C	8B45 F4	MOV ERX, DWORD PTR SS:[EBP-C]
0040546D	E8 D6DF5FF	CALL Tekshed.0040524C
0040546E	66:85C0	TEST AX, AX
0040546F	0F86 23010000	JBE Tekshed.004055A2
00405470	66:8945	MOV WORD PTR SS:[EBP-2E], AX
00405471	66:BF 0100	MOV DI, 1
00405472	> 66:BE 0100	MOV SI, 1
00405473	881D 64575B01	MOV EBX, DWORD PTR DS:[5A5764]
00405474	43	INC EBX
00405475	> 0FB7C7	MOUZX EAX, DI
00405476	8B55 F4	MOV EDX, DWORD PTR SS:[EBP-C]
00405477	8A4482 FF	MOV AL, BYTE PTR DS:[EDX+EAX-1]
00405478	E8 9B09F5FF	CALL Tekshed.00402E30
00405479	50	PUSH EAX
0040547A	9A03	MOU AL, BYTE PTR DS:[EBX]
0040547B	E8 93D9F5FF	CALL Tekshed.00402E30
0040547C	5A	POP EDX
0040547D	9A00	CMP DL, AL
0040547E	> 0FB85 D7000001	JNZ Tekshed.004A5589
0040547F	0FB7C7	MOUZX EAX, DI
00405480	83E0 01	AND EAX, 1
00405481	85C0	TEST EAX, EAX
00405482	> 75 66	JNZ SHORT Tekshed.004A5522
00405483	0B95 BCFFFF	LEA EDX, DWORD PTR SS:[EBP-144]
00405484	0FB7C6	MOUZX EAX, SI
00405485	E8 1EE4F5FF	CALL Tekshed.004058E8
00405486	0B95 BCFFFF	LEA EDX, DWORD PTR SS:[EBP-144]
00405487	8045 EC	LEA ERX, DWORD PTR SS:[EBP-14]
00405488	E8 18FDF5FF	CALL Tekshed.004051F0
00405489	66:83FE 09	CMP SI, 9
0040548A	> 76 24	JBE SHORT Tekshed.004A5502
0040548B	AN45 RAFFFF	LEA FAX, WORD PTR SS:[EBP-148]

And remove the old ones

오래된 것들을 삭제 해.

Continue F8

F8 눌러.

C CPU - main thread, module Tekshed

004A55D0	88B5 0AFFFFFF	MOV EAX,DWORD PTR SS:[EBP-160]
004A55D6	89D5 A4FFFFFF	LEA EDX,DWORD PTR SS:[EBP-15C]
004A55D8	E8 4342F6FF	CALL Tekshed.00409824
004A55E1	89D5 A4FFFFFF	MOV EDX,DWORD PTR SS:[EBP-15C]
004A55E8	F0F9 EDX	
004A55E9	E8 AB0DF5FF	CALL Tekshed.00405398
004A55F0	74 00 JE	SHORI Tekshed.00405608
004A55F1	80D0 00 00	CMP BYTE PTR SS:[EBP+8],0
004A55F2	0F85 59020000	JNC Tekshed.00405876
004A55F3	B8 78E94000	MOV EAX,Tekshed.004A5978
004A55F4	F33BF9FF	CALL Tekshed.004391F8
004A55F5	E9 6E020000	JMP Tekshed.00405876
004A55F6	66:8B 50	
004A55F7	66:181FE	
004A55F8	76 19	
004A55F9	80D0 00 00	LMP BYTE PTR SS:[EBP+8],0
004A55FA	0F85 59020000	JNC Tekshed.00405876
004A55FB	B8 E4594000	MOV EAX,Tekshed.004A59E4
004A55FC	E8 D13BF9FF	CALL Tekshed.004391F8
004A55FD	E9 4A020000	JMP Tekshed.00405876
004A55FE	> A1 BC5CS9000	MOV ECX,DWORD PTR DS:[5A5BC0]
004A55FF	66:8B0000	MOV AX,WORD PTR DS:[EAX]
004A5600	66:F7EE	IMUL SI
004A5601	8BF0	MOV ESI,EAX
004A5602	B8 894E0000	MOV EBX,4EB8
004A5603	89D5 BCFEFFFF	LEA EDX,DWORD PTR SS:[EBP-144]
004A5604	0FB7C6	MULZX EXX,SI
004A5605	E8 9CE2FFFF	CALL Tekshed.004038E8
004A5606	89D5 BCFEFFFF	LEA EDX,DWORD PTR SS:[EBP-144]
004A5607	8045 EC	LEA ECX,DWORD PTR SS:[EBP-14]
004A5608	E8 96FBFF5F	CALL Tekshed.004051F8
004A5609	8040 E8	LEA ECX,DWORD PTR SS:[EBP-10]
004A560A	66:BA 0600	MOV ECX,0600
004A560B	8B 00	MOV ECX,ECX
004A560C	E8 9824FFFF	CALL Tekshed.00490078
004A560D	8045 EC	LEA ECX,DWORD PTR SS:[EBP-14]
004A560E	88E5 E9	MOU EDX,DWORD PTR SS:[EBP-18]
004A560F	E8 E1FBFF5F	CALL Tekshed.00408254
004A5610	8045 EC	MOU EAX,DWORD PTR SS:[EBP-14]
004A5611	88E5 DC	MOU EDX,DWORD PTR SS:[EBP-24]
004A5612	E8 1AFDF6FF	CALL Tekshed.00405398
004A5613	> 0F85 DB010000	JNC Tekshed.0040585F
004A5614	8045 D8	LEA EAX,DWORD PTR SS:[EBP-28]
004A5615	B4 35A40000	MOU EDX,Tekshed.004A5934
004A5616	F8 C3RFRFFF	CALL Tekshed.00405254

Aha, but this

0012E548

ASCII "Error! You are attempting to register the wro

ASCII "Error! You are attempting to register the wro

Tekshed.005A44D1

ASCII "GJJ"

Aha, but this

야한 그러나 이건으

Not being executed

실행되지 않는다

C CPU - main thread, module Teksched

```

004455D0  . 8B85 A0FFFFF1 MOU EAX, DWORD PTR SS:[EBP-160]
004455D6  . 8095 A4FEFFF1 LEA EDX, DWORD PTR SS:[EBP-15C]
004455D0  . E8 4342F6FF CALL Teksched.00409824
004455E1  . 8B95 A4FEFFF1 MOU EDX, DWORD PTR SS:[EBP-15C]
004455E7  . 58 POP EAX
004455E8  . E8 ABFD05FF CALL Teksched.00405398
004455E9  . 74 19 JE SHORT Teksched.00405608
004455F3  . 0F85 70020000 JNP PTR SS:[EBP+1], 0
004455F9  . 8B85 59020000 MOU EAX, Teksched.00405978
004455FE  . E8 F53BF9FF CALL Teksched.004391F8
00445603  . E9 6E020000 JIF Teksched.00405876
00445608  . 58 MOU SI, 258
0044560C  . 00445611
00445613  . 00445617
00445622  . 00445620
00445620  . > E9 45920000 MOU EAX, Teksched.004059E4
00445622  . E8 D136F9FF CALL Teksched.004391F8
00445624  . 58 MOU EDX, 004455D2
0044562C  . 00445630
00445631  . 00445633
00445634  . 66:7EE IMUL SI
00445637  . 00445639
00445639  . BBF0 MOU ESI, EAX
0044563E  . BB 894E0000 MOU EBX, 4E99
00445644  . 0FB7C6 MOUZX EAX, SI
00445647  . E8 9CE2F5FF CALL Teksched.004038E8
0044564C  . 8095 BCFFEFF1 LEA EDX, DWORD PTR SS:[EBP-144]
00445652  . 8045 EC LER EAX, DWORD PTR SS:[EBP-14]
00445655  . E8 96FBF5FF CALL Teksched.004051F0
0044565A  . 804D E8 LER ECX, DWORD PTR SS:[EBP-18]
0044565D  . 66:BA 0600 MOU DX, 6
00445661  . 00445663
00445663  . 00445668
00445668  . 8045 EC LER EAX, DWORD PTR SS:[EBP-14]
0044566B  . 8B55 E8 MOU EDX, DWORD PTR SS:[EBP-18]
0044566E  . E8 E1FBFF5F CALL Teksched.00405254
00445673  . 8B45 EC MOU EAX, DWORD PTR SS:[EBP-14]
00445676  . 8B55 DC MOU EDX, DWORD PTR SS:[EBP-24]
00445679  . 0044567E
0044567E  . > E8 1AFDF5FF CALL Teksched.00405398
00445680  . 0F85 DB010000 JNP Teksched.00405876
00445684  . 8045 D8 LER EAX, DWORD PTR SS:[EBP-28]
00445687  . BB 345A4000 MOU EDX, Teksched.00405A34
0044568F  . FR FFFFFFFF CALL Teksched.00405254

```

Mmm, that's not good, because ...

... else we will probably land in a badboy message

우리는 아마 badboy message 에 도착할 거야.

C CPU - main thread, module Teksched

```

004455D0  . 8B85 A0FFFFF1 MOU EAX, DWORD PTR SS:[EBP-160]
004455D6  . 8095 A4FEFFF1 LEA EDX, DWORD PTR SS:[EBP-15C]
004455D0  . E8 4342F6FF CALL Teksched.00409824
004455E1  . 8B95 A4FEFFF1 MOU EDX, DWORD PTR SS:[EBP-15C]
004455E7  . 58 POP EAX
004455E8  . E8 ABFD05FF CALL Teksched.00405398
004455E9  . 74 19 JE SHORT Teksched.00405608
004455F3  . 0F85 70020000 JNP PTR SS:[EBP+1], 0
004455F9  . 8B85 59020000 MOU EAX, Teksched.00405978
004455FE  . E8 F53BF9FF CALL Teksched.004391F8
00445603  . E9 6E020000 JIF Teksched.00405876
00445608  . 58 MOU SI, 258
0044560C  . 00445611
00445613  . 00445617
00445622  . 00445620
00445620  . > E9 45920000 MOU EAX, Teksched.004059E4
00445622  . E8 D136F9FF CALL Teksched.004391F8
00445624  . 58 MOU EDX, 004455D2
0044562C  . 00445630
00445631  . 00445633
00445634  . 66:7EE IMUL SI
00445637  . 00445639
00445639  . BBF0 MOU ESI, EAX
0044563E  . BB 894E0000 MOU EBX, 4E99
00445644  . 0FB7C6 MOUZX EAX, SI
00445647  . E8 9CE2F5FF CALL Teksched.004038E8
0044564C  . 8095 BCFFEFF1 LEA EDX, DWORD PTR SS:[EBP-144]
00445652  . 8045 EC LER EAX, DWORD PTR SS:[EBP-14]
00445655  . E8 96FBF5FF CALL Teksched.004051F0
0044565A  . 804D E8 LER ECX, DWORD PTR SS:[EBP-18]
0044565D  . 66:BA 0600 MOU DX, 6
00445661  . 00445663
00445663  . 00445668
00445668  . 8045 EC LER EAX, DWORD PTR SS:[EBP-14]
0044566B  . 8B55 E8 MOU EDX, DWORD PTR SS:[EBP-18]
0044566E  . E8 E1FBFF5F CALL Teksched.00405254
00445673  . 8B45 EC MOU EAX, DWORD PTR SS:[EBP-14]
00445676  . 8B55 DC MOU EDX, DWORD PTR SS:[EBP-24]
00445679  . 0044567E
0044567E  . > E8 1AFDF5FF CALL Teksched.00405398
00445680  . 0F85 DB010000 JNP Teksched.00405876
00445684  . 8045 D8 LER EAX, DWORD PTR SS:[EBP-28]
00445687  . BB 345A4000 MOU EDX, Teksched.00405A34
0044568F  . FR FFFFFFFF CALL Teksched.00405254

```

Right, let's help a little :)

우리는 아마 badboy message 에 도착할 거야.

Right, let's help a little :)

좋아, 여기에서 도움을 얻자 :)

C CPU - main thread, module Teksched

```

00445500 · 8B85 A0FEFFFF MOU EAX, DWORD PTR SS:[EBP-160]
00445506 · 8D95 A4FEFFFF LEA EDX, DWORD PTR SS:[EBP-1EC]
0044550C · E8 4342F6FF CALL Tekshed.00409824
004455E1 · 8B95 A4FEFFFF MOU EDX, DWORD PTR SS:[EBP-1EC]
004455E7 · 58
004455E8 · E8 ABFD5FF CALL Tekshed.00405398
004455E9 · 74 19 JE SHORT Tekshed.00405608
004455F3 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
004455F9 · BB 78534000 MOU EAX, Tekshed.00405978
004455FE · E8 F53BF9FF CALL Tekshed.004391F8
00445603 · > E9 6E020000 JIF Tekshed.004AS876
00445608 · 58
00445609 · 66:81FB BC02 MOU SI, 258
00445611 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
00445613 · 76 19 JBE SHORT Tekshed.004A5620
00445617 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
00445622 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
00445627 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
0044562C · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
00445634 · BB 894E0000 MOU EBX, 4E89
00445638 · 8D95 BCFFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
00445644 · 0FB7C6 MOU ECX, EAX, SI
00445647 · E8 9CE2FF5F CALL Tekshed.004038E8
0044564C · 8D95 BCFFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
00445652 · 8D45 EC LEA EAX, DWORD PTR SS:[EBP-144]
00445655 · E8 96FBFF5F CALL Tekshed.004051F0
0044565A · 8D4D E8 LEA ECX, DWORD PTR SS:[EBP-160]
0044565D · 66:BA 0600 MOU DX, 6
00445661 · 8BC3 MOV EAX, EBX
00445663 · E8 0B84FFFF CALL Tekshed.0049DA70
00445668 · 8D45 EC LEA EAX, DWORD PTR SS:[EBP-14]
0044566B · BB 55 E8 MOU EDX, DWORD PTR SS:[EBP-18]
0044566E · E8 E1FBFF5F CALL Tekshed.00405254
00445673 · 8B45 EC MOU EAX, DWORD PTR SS:[EBP-14]
00445676 · 8B55 DC MOU EDX, DWORD PTR DS:[EBP-24]
00445679 · E8 1AFDF5FF CALL Tekshed.00405398
0044567E · > 0F85 DB010000 JNC Tekshed.004AS85F
00445684 · 8D45 D8 LEA EAX, DWORD PTR DS:[EBP-28]
00445687 · BB 345B4000 MOU EDX, Tekshed.00405A34
004456A0 · FA C0F0FFF CALL Tekshed.00405A4401

```

Jump is taken
004A5608=Teksched.004A5608

But remember for later that this is only a temporary solution!
Continue stepping for now.

But remember for later that this is only a temporary solution!

그러나 나중을 위해 기억해. 이것은 오직 임시적인 해결책이야!

Continue stepping for now.

이제 계속하자.

C CPU - main thread, module Teksched

```

00445500 · 8B85 A0FEFFFF MOU EAX, DWORD PTR SS:[EBP-160]
00445506 · 8D95 A4FEFFFF LEA EDX, DWORD PTR SS:[EBP-1EC]
0044550C · E8 4342F6FF CALL Tekshed.00409824
004455E1 · 8B95 A4FEFFFF MOU EDX, DWORD PTR SS:[EBP-1EC]
004455E7 · 58
004455E8 · E8 ABFD5FF CALL Tekshed.00405398
004455E9 · 74 19 JE SHORT Tekshed.00405608
004455F3 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
004455F9 · BB 78534000 MOU EAX, Tekshed.00405978
004455FE · E8 F53BF9FF CALL Tekshed.004391F8
00445603 · > E9 6E020000 JIF Tekshed.004AS876
00445608 · 58
00445609 · 66:81FE BC02 MOU SI, 258
00445611 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
00445613 · 76 19 JBE SHORT Tekshed.004A5620
00445617 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
00445622 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
00445627 · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
0044562C · 0F85 70020000 CMP BTTE PTR SS:[EBP+1], 0
00445634 · BB 894E0000 MOU EBX, 4E89
00445638 · 8D95 BCFFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
00445644 · 0FB7C6 MOU ECX, EAX, SI
00445647 · E8 9CE2FF5F CALL Tekshed.004038E8
0044564C · 8D95 BCFFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
00445652 · 8D45 EC LEA EAX, DWORD PTR SS:[EBP-144]
00445655 · E8 96FBFF5F CALL Tekshed.004051F0
0044565A · 8D4D E8 LEA ECX, DWORD PTR SS:[EBP-160]
0044565D · 66:BA 0600 MOU DX, 6
00445661 · 8BC3 MOV EAX, EBX
00445663 · E8 0B84FFFF CALL Tekshed.0049DA70
00445668 · 8D45 EC LEA EAX, DWORD PTR SS:[EBP-14]
0044566B · BB 55 E8 MOU EDX, DWORD PTR SS:[EBP-18]
0044566E · E8 E1FBFF5F CALL Tekshed.00405254
00445673 · 8B45 EC MOU EAX, DWORD PTR SS:[EBP-14]
00445676 · 8B55 DC MOU EDX, DWORD PTR SS:[EBP-24]
00445679 · E8 1AFDF5FF CALL Tekshed.00405398
0044567E · > 0F85 DB010000 JNC Tekshed.004AS85F
00445684 · 8D45 D8 LEA EAX, DWORD PTR DS:[EBP-28]
00445687 · BB 345B4000 MOU EDX, Tekshed.00405A34
004456A0 · FA C0F0FFF CALL Tekshed.00405A4401

```

Jump is taken
004A562C=Teksched.004A562C

We jump passed the BadBoy.
That's good for this time.

We jump passed the BadBoy.

That's good for this time.

우리는 지나가 버린 badboy 를 jump 한다.

좋아.

Now, think with me ...

If this JBE doesn't get executed for one or another reason if we are registered

이제, 나랑 같이 생각해보자.

우리가 등록하기를 원할 때 만약에 한 가지 이유나 다른 이유로 이 JBE 가 실행되지 않는다면

...i.e. after some amount of time or whatever

꽤 많은 시간이 지나거나 무엇이든지

So, for extra security, let's ...

그래서, 추가적인 보호가 필요하다.

CPU - main thread, module Tekshed

004A562D > . 74 19 JE SHORT Tekshed.004A5608

BP here too, to remember later to JMP always here

004A5600 > . 80 7D SUB EDI,EDI

004A5601 > . 0F 85 JNC Tekshed.004A5676

004A5602 > . E9 60 JMP .

004A5603 > . 66:BE .

004A5608 > . 66:81FE BC02 CMP SS:[EBP],BC

004A5611 > . 75 19 JNE SHORT Tekshed.004A562C

004A5613 > . 0F 80 00 CMP BT,WORD PTR SS:[EBP+8],0

004A5617 > . 0F 84 00 MOU EDX, Tekshed.004A5676

004A5619 > . B8 F4494000 CALL Tekshed.004A5694

004A5622 > . ED D13BF9FF

004A5627 > . E9 400200A0 JMP Tekshed.004A5676

004A562D > . A1 BC5CSA00 MOU EDX, DWORD PTR DS:[5A5CBC]

004A5631 > . 66:8000 MOV AX,WORD PTR DS:[ERAX]

004A5634 > . 66:87EE IMUL SI,SI

004A5635 > . 66:8000 MOV EDI,ESI,EDI

004A5639 > . BB 894E0000 LEA EDX,DWORD PTR SS:[EBP-144]

004A563E > . 8D95 BCFFEF MOUZX EDX,SI

004A5644 > . 0F7C6 LEA ECX,DWORD PTR SS:[EBP-16]

004A5647 > . E8 9CE2FF5F CALL Tekshed.004A938E

004A5652 > . 8D95 BCFEFF LEA EDX,DWORD PTR SS:[EBP-144]

004A5652 > . 8045 EC LEA EAX,DWORD PTR SS:[EBP-14]

004A5655 > . E8 96FBFB5F CALL Tekshed.004A951F

004A5658 > . 804D E8 LEA ECX,DWORD PTR SS:[EBP-16]

004A5659 > . 66:8B 0600 MOU DX,DX

004A5661 > . 8EC3 MOU EAX,EBX

004A5663 > . E8 0884FFFF CALL Tekshed.0049DA70

004A5668 > . 8045 EC LEA EAX,DWORD PTR SS:[EBP-14]

004A5669 > . 8B55 E8 MOU EDX,DWORD PTR SS:[EBP-16]

004A566E > . E8 E1FBFB5F CALL Tekshed.004A9524

004A5672 > . 8B45 D8 MOU EDX,DWORD PTR SS:[EBP-24]

004A5676 > . 8B55 DC MOU EDX,DWORD PTR SS:[EBP-24]

004A5679 > . E8 1AFDF5FF CALL Tekshed.004A9539

004A567E > . 0F85 D81000000 JNC Tekshed.004A55F

004A5684 > . 8045 D8 LEA EAX,DWORD PTR SS:[EBP-28]

004A5687 > . BA 3450A000 MOU EDX,Tekshed.004D5A94

004A568C > . E8 C3FBFF5F CALL Tekshed.00495254

004A5691 > . A1 BC5CSA00 MOU EDX,DWORD PTR DS:[5A5CBC]

004A5696 > . 66:6930 BC02 IMUL SI,WORD PTR DS:[EAX],BC

004A569B > . 8D95 BCFFEF LEA EDX,DWORD PTR SS:[EBP-144]

004A56A1 > . 0F87C6 MOUZX EDX,SI

004A56A4 > . E8 3FE2FF5F CALL Tekshed.004A938E

004A56A9 > . A195 RFFFFFFF IFR Fnx,MOU EDX,DWORD PTR SS:[EBP-144]

DS:[00505EBC0]=005A6DF0
EDI=0000000000
JUNK From 004A5611

ASCII "Error! You are attempting to register the wrong ver

Tekshed.005A44D1

ASCII "GJJ"

BP here too, to remember later to JMP always here

항상 JMP 하기 위해 BP 를 여기에 설치 해.

Continue stepping F8

F8로 계속 해.

C CPU - main thread, module Teksched

```

00445617 .> 0F85 59020000 JNC Teksched.004A5876
0044561D . B8 E4594000 MOU EAX, Teksched.004A59E4
00445622 . E8 D13BF9FF CALL Teksched.004A59F8
00445627 .> E9 40020000 JMP Teksched.004A5878
0044562C . A1 BC5C5A00 MOU EAX, DWORD PTR DS:[5A5CBC]
00445631 . 66:8B00 MOV AX, WORD PTR DS:[EAX]
00445634 . 66:F7EE IMUL SI
00445637 . 66:8B00 MOV ESI, EAX
0044563B . B8 894E0000 MOU EAX, EB9
0044563E . B8 894E0000 MOU EAX, EB9
00445644 . 0F93C0 BCFFFF CALL Teksched.004A5888
00445647 . B995 BCFFFF LEA EDX, DWORD PTR SS:[EBP-144]
0044564C . 0F93C0 BCFFFF LEA EDX, DWORD PTR SS:[EBP-144]
00445652 . 8045 EC LEA EAX, DWORD PTR SS:[EBP-14]
00445655 . E8 96FBF5FF CALL Teksched.004A51F0
00445658 . 8045 EC LEA EAX, DWORD PTR SS:[EBP-18]
0044565D . 8040 D8 MOU DX, 6
00445661 . B8 C3 MOU EAX, EBX
00445663 . E8 0884FFFF CALL Teksched.004A5A70
00445668 . 8045 EC LEA EAX, DWORD PTR SS:[EBP-14]
0044566B . 8045 EC LEA EAX, DWORD PTR SS:[EBP-18]
0044566E . E8 E1FBF5FF CALL Teksched.004A5254
00445672 . 8045 EC LEA EAX, DWORD PTR SS:[EBP-14]
00445673 . 8045 DC MOU EDX, DWORD PTR SS:[EBP-24]
00445678 . E8 1AFDF5FF CALL Teksched.004A5398
0044567E . 0F85 DB010000 JNC Teksched.004A585F
00445684 . 8045 D8 LEA EAX, DWORD PTR SS:[EBP-28]
00445687 . B9 34594000 MOU EDX, Teksched.004A5A34
0044568C . E8 C3FBF5FF CALL Teksched.004A5254
00445691 . A1 BC5C5A00 MOU EAX, DWORD PTR DS:[5A5CBC]
00445696 . 66:6938 BC02 IMUL SI, WORD PTR DS:[EAX], 28C
0044569B . 8095 BCFFFF LEA EDX, DWORD PTR SS:[EBP-144]
0044569C . 0F87C6 MOUZX EAX, SI
004456A4 . E8 3FE2FBFFF CALL Teksched.004A5888
004456A9 . B9 39FBF5FF LEA EDX, DWORD PTR SS:[EBP-144]
004456A2 . 8045 DC MOU EDX, DWORD PTR SS:[EBP-24]
004456A5 . E8 39FBF5FF CALL Teksched.004A51F0
004456A8 . 8045 D8 LEA EAX, DWORD PTR SS:[EBP-28]
004456B1 . 8045 DC MOU EDX, DWORD PTR SS:[EBP-24]
004456B4 . E8 92FBF5FF CALL Teksched.004A5254
004456C2 . A1 E8535A00 MOU EAX, DWORD PTR DS:[5A53E0]
004456C7 . 8000 MOU EAX, DWORD PTR DS:[EAX]
004456C9 . 50 PUSH EAX
004456C9 . A1 5C5B5A00 MOU EAX, DWORD PTR DS:[5A5B5C]
004456C9 . AAAA MOU AL, BYTE PTR DS:[EAX]

```

ASCII "Error! You are attempting to register the wrong ver..."

Teksched.004A5898

mmmm, remember that this is tooooooo far ?

Remember that I told you to note down the VA we didn't want to jump passed ...

Arg4 = 00000007

Jump is taken
0044585F=Teksched.004A585F

Aha, but here it gets serious ...

아하, 좀 더 진지하게 얻을 수 있다.

Mmmm, remember that this is tooooooo far ?

음, 이것이 너무 멀다는 것을 기억해?

Remember that I told you to note down the VA didn't want to jump passed ...

VA 를 적으라고 했던 것을 기억해. VA 는 jump 를 원하지 않는다.

C CPU - main thread, module Teksched

```

00445655 . E8 96FBF5FF CALL Teksched.004A51F0
00445655 . 8040 E8 LEA EAX, DWORD PTR SS:[EBP-18]
0044565D . 66:6900 MOV AX, 6
00445661 . 8000 MOU EAX, EBX
00445663 . E8 0884FFFF CALL Teksched.004A5A70
00445668 . 8045 EC LEA EDX, DWORD PTR SS:[EBP-14]
0044566B . 8045 EC LEA EAX, DWORD PTR SS:[EBP-18]
0044566E . E8 E1FBF5FF CALL Teksched.004A5254
00445673 . 8045 EC LEA EAX, DWORD PTR SS:[EBP-14]
00445676 . 8045 DC MOU EDX, DWORD PTR SS:[EBP-24]
00445679 . E8 1AFDF5FF CALL Teksched.004A5398
0044567E . 0F85 DB010000 JNC Teksched.004A585F
00445684 . 8045 D8 LEA EAX, DWORD PTR SS:[EBP-28]
00445687 . B9 34594000 MOU EDX, Teksched.004A5A34
0044568C . E8 C3FBF5FF CALL Teksched.004A5254
00445691 . A1 BC5C5A00 MOU EAX, DWORD PTR DS:[5A5CBC]
00445696 . 66:6938 BC02 IMUL SI, WORD PTR DS:[EAX], 28C
0044569B . 8095 BCFFFF LEA EDX, DWORD PTR SS:[EBP-144]
0044569C . 0F87C6 MOUZX EAX, SI
004456A4 . E8 3FE2FBFFF CALL Teksched.004A5888
004456A9 . B9 39FBF5FF LEA EDX, DWORD PTR SS:[EBP-144]
004456A2 . 8045 DC MOU EDX, DWORD PTR SS:[EBP-24]
004456A5 . E8 39FBF5FF CALL Teksched.004A51F0
004456A8 . 8045 D8 LEA EAX, DWORD PTR SS:[EBP-28]
004456B1 . 8045 DC MOU EDX, DWORD PTR SS:[EBP-24]
004456B4 . E8 92FBF5FF CALL Teksched.004A5254
004456C2 . A1 E8535A00 MOU EAX, DWORD PTR DS:[5A53E0]
004456C7 . 8000 MOU EAX, DWORD PTR DS:[EAX]
004456C9 . 50 PUSH EAX
004456C9 . A1 5C5B5A00 MOU EAX, DWORD PTR DS:[5A5B5C]
004456C9 . 8000 MOU AL, BYTE PTR DS:[EAX]
004456D1 . 50 PUSH EAX
004456D2 . A1 64545A00 MOU EAX, DWORD PTR DS:[5A5464]
004456D7 . 8000 MOU EAX, DWORD PTR DS:[EAX]
004456D9 . 50 PUSH EAX
004456D9 . A1 C8595A00 MOU EAX, DWORD PTR DS:[5A59C8]
004456D9 . 8000 MOU AL, BYTE PTR DS:[EAX]
004456E1 . 50 PUSH EAX
004456E3 . B8 0884A000 MOU EAX, Teksched.004A5808
004456E3 . B2 01 MOU DL, 1

```

ASCII "GJJ"

OK, help needed once again

Arg4 = 00000007

Arg3 = 00000007

Arg2 = 00000007

Arg1 = 00000007

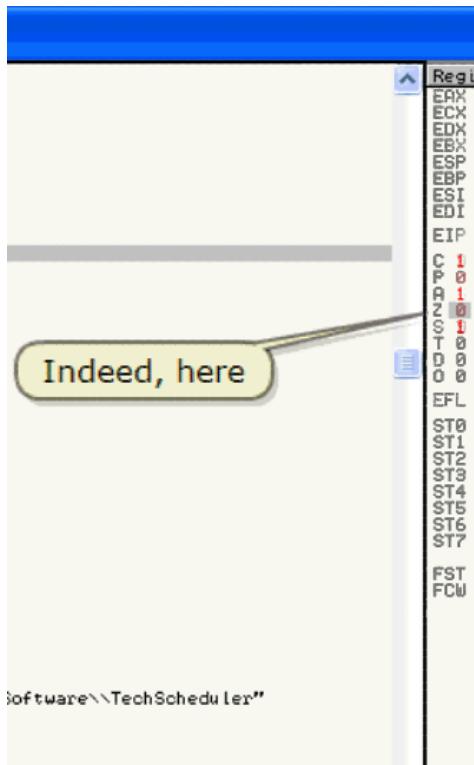
ASCII "Software\\Dean Software\\TechScheduler"

Teksched.004A5808

Jump is taken
0044585F=Teksched.004A585F

OK, help needed once again

Ok, 다시 한 번 도움을 얻었다.



Indeed, here

정말, 여기

;

Place RP to remember

기억하기 위해 BP 를 설치 해.

C CPU - main thread, module Tekshed

004A5609	59	PUSH EAX
004A560A	A1 3A595A00	MOU EDX, DWORD PTR DS:[5A59C8]
004A560B	8A00	MOU AL, BYTE PTR DS:[EAX]
004A5611	59	PUSH EAX
004A5622	B9 05594A00	MOU ECX, Tekshed.004A5908
004A5623	B2 01	MOU DL,1
004A5624	A1 88564900	MOU EDX, DWORD PTR DS:[495668]
004A5625	E8 1918FFFF	CALL Tekshed.0049678C
004A5626	8945 CC	MOV EDX, SS:[EBP-34], EAX
004A5627	33C0	XOR EDX, EAX
004A5628	55	PUSH EBX
004A5629	68 3A594A00	PUSH Tekshed.004A583A
004A562A	44:F9	PUSH DWORD PTR FS:[EAX]
004A562B	64:8924	MOU EDI, DWORD PTR FS:[EAX],ESP
004A562C	8D95 D9	LEA EDX, DWORD PTR SS:[EBP-144]
004A562D	8945 DC	MOV EDX, SS:[EBP-144], EBP-28
004A562E	E8 82F9FFFF	CALL Tekshed.004A4734
004A562F	8995 BCFFFFFF	LEA EDX, DWORD PTR SS:[EBP-144]
004A5630	8983 9CFEFFF1	LEA EDX, DWORD PTR SS:[EBP-164]
004A5631	E8 C0F5EF	CALL Tekshed.004B81F8
004A5632	8885 9CFEFFF1	MOU EDX, DWORD PTR SS:[EBP-164]
004A5633	59	PUSH EAX
004A5634	B9 40540400	MOU ECX, Tekshed.004A5A40
004A5635	B4 54540400	MOU EDX, Tekshed.004A5A54
004A5636	8845 CC	MOU ERX, DWORD PTR SS:[EBP-34]
004A5637	E8 1415FFFF	CALL Tekshed.00496C50
004A5638	8D95 BCFFFFFF	LEA EDX, DWORD PTR SS:[EBP-144]
004A5639	8945 FC	MOV EDX, SS:[EBP-144], EBP-40
004A563A	E8 40FFFF	CALL Tekshed.004A4734
004A563B	8995 BCFFFFFF1	LEA EDX, DWORD PTR SS:[EBP-144]
004A563C	8983 9CFEFFF1	LEA EDX, DWORD PTR SS:[EBP-163]
004A563D	E8 95F0F5FF	CALL Tekshed.004B81F8
004A563E	8885 9CFEFFF1	MOU EDX, DWORD PTR SS:[EBP-163]
004A563F	59	PUSH EAX
004A5640	B9 6459	MOU ECX, Tekshed.004A5A40
004A5641	B4 6451	MOU EDX, Tekshed.004A5A54
004A5642	8845 C1	MOU ERX, DWORD PTR SS:[EBP-34]
004A5643	E8 D1	LEA EDX, DWORD PTR SS:[EBP-144]
004A5644	8D95 B1	MOV EDX, SS:[EBP-144], EBP-40
004A5645	8845 F1	CALL Tekshed.00496C50
004A5646	E8 12F1	LEA EDX, DWORD PTR SS:[EBP-144]
004A5647	8D95 B0	MOV EDX, SS:[EBP-144], EBP-40
004A5648	8885 94	CALL Tekshed.004B81F8
004A5649	E8 5D9F5FF	CALL Tekshed.004B81F8
004A564A	ARRR 94FFFFFF HNU FAX. FINISH PTR SS:[EBP-163]	

Arg2 = 00E11C08 ASCII "00126B787C01736103191E621A070000"

Arg1 = 00E11C08 ASCII "00126B787C01736103191E621A070000" ASCII "Software\\Dean Software\\TechScheduler"

Tekshed.0049678C

ASCII "sRegStat"
ASCII "Config"

BTW, there seems to be a lot of interesting serial-like data around. We will see more about serial fishing later. I deliberately don't want to look into that already. See later !!!

BTW, there seems to be a lot of interesting serial-like data around. We will see more about serial fishing later. I deliberately don't want to look into that already. See later !!!

BTW, there seems to be a lot of interesting serial-like data around.

By the way, 그곳은 serial data 가 꽤 흥미롭게 있는 것이 보인다.

We will see more about serial fishing later.

우리는 serial 낚시에 대하여 나중에 좀 더 볼 수 있다.

I deliberately don't want to look into that already. See later !!!

나는 고의적으로 그것을 미리 보기 원하지 않는다. 나중에 보자 !!!

C CPU - main thread, module Tekshed

```
004A570C . 8B45 CC MOU EDX, [MDRD PTR SS:[EBP-34]]  
004A570F . E8 6C14FFFF CALL Tekshed.00496C50  
004A57E4 . 8D95 BCFFFFFF LEA EDX, [MDRD PTR SS:[EBP-144]]  
004A57E9 . 8B45 E0 MOU EDX, [MDRD PTR SS:[EBP-201]]  
004A57ED . E8 A2EFFFFF CALL Tekshed.004A4794  
004A57F2 . 8D95 BCFFFFFF LEA EDX, [MDRD PTR SS:[EBP-144]]  
004A57F8 . 2085 8CFFFFFF LEA EDX, [MDRD PTR SS:[EBP-174]]  
004A5803 . E8 E0F9FFFF CALL Tekshed.004951F0  
004A5809 . 8B45 CC MOU EDX, [MDRD PTR SS:[EBP-174]]  
004A580A . E9 98544000 MOU ECX, Tekshed.004A5A98  
004A580F . B8 64584000 MOU EDX, Tekshed.004A5A94  
004A5814 . 8B45 CC MOU EDX, [MDRD PTR SS:[EBP-34]]  
004A5817 . E8 3414FFFF CALL Tekshed.00496C50  
004A581C . 8B45 CC MOU EDX, [MDRD PTR SS:[EBP-34]]  
004A5821 . E8 2009FFFF CALL Tekshed.00495844  
004A5824 . 33C0 XOR EDX, EDX  
004A5826 . 5A POP EDX  
004A5827 . 59 POP ECX  
004A5828 . 59 POP ECX  
004A5829 . E4: S913 MOU EDX, [MDRD PTR FS:[TEAK], EDX]  
004A5830 . 8B45 CC PUSH Tekshed.004A5A91  
004A5831 > 8B45 CC MOU EDX, [MDRD PTR SS:[EBP-34]]  
004A5834 . E8 5BEE05FF CALL Tekshed.00404094  
004A5839 . C3 RETN  
004A583A . ^ E9 E9EFFFFF JMP Tekshed.00404828  
004A583B . ^ EB F0 SHORT Tekshed.004A5831  
004A583C . ^ E6 C45 F3 01 MOV BYTE PTR SS:[EBP-0], 1  
004A583D . 8070 00 00 CMP BYTE PTR SS:[EBP+1], 0  
004A583E . 75 0A JNE SHORT Tekshed.004A5855  
004A5848 . B8 H8544000 MOU EDX, Tekshed.004A5A98  
004A5850 . E8 B33939FF CALL Tekshed.004A5A98  
004A5851 . E8 B33939FF MOU EDX, [MDRD PTR SS:[5A5F41]]  
004A5854 . C600 00 MOV BYTE PTR DS:[TERMX], 0  
004A5855 . EB 17 JMP SHORT Tekshed.004A5876  
004A5856 . 897D 00 00 CMP BYTE PTR SS:[EBP+1], 0  
004A5857 . 75 11 JNE SHORT Tekshed.004A5876  
004A5865 . E8 30 POP ECX  
004A5867 . E8 C829F6FF CALL <JMP, user32.MessageBoxBeep>  
004A5870 . B8 CCC54000 MOU EDX, Tekshed.004A5ACC  
004A5871 . E8 B239FF9FF CALL Tekshed.004A391FB  
004A5876 . 33C0 XOR EDX, EDX  
004A5878 . 5A POP EDX  
004A5879 . 59 POP ECX  
004A587A . 59 POP ECX  
Aha, look here what happens  
Remember this ???  
ASCII "Registration Key accepted!"  
ASCII "Registration Key Failed!"  
BeepType AMB_ICONEXCLAMATION  
MessageBeep  
004A58A1-Tekshed.004A58A1
```

Aha, look here what happens Remember this ???

아하, 여기를 봐. 이것에 무슨 일이 발생했는지 기억해 ???

C CPU - main thread, module Tekshed

```
004A57DC · BB45 CC MOU EDX, DWORD PTR SS:[EBP-34] 
004A57DF · EB 3C14FFFF CALL Tekshed.00496C5B 
004A57E4 · BB95 8CFFFFFF MOU EDX, DWORD PTR SS:[EBP-144] 
004A57E9 · BB45 E9 MOU EDX, DWORD PTR SS:[EBP-20] 
004A57ED · EB 02EFFFFF CALL Tekshed.00494794 
004A57F2 · BB95 BCFFFFFF LER EDX, DWORD PTR SS:[EBP-144] 
004A57F8 · BB85 8CFFFFFF LEA EDX, DWORD PTR SS:[EBP-174] 
004A57FE · EB E0F95FF5 CALL Tekshed.004981F8 
004A5803 · BB85 8CFFFFFF MOU EDX, DWORD PTR SS:[EBP-174] 
004A5809 · 50 PUSH EDX 
004A580A · BB45 CC MOU ECX, Tekshed.004A5A98 
004A580F · BB45 CC MOU EDX, Tekshed.004A5A94 
004A5814 · BB45 CC MOU EDX, DWORD PTR SS:[EBP-34] 
004A5817 · EB 3414FFFF CALL Tekshed.00496C5B 
004A5821C · BB45 CC MOU EDX, DWORD PTR SS:[EBP-34] 
004A5821F · EB 2000FFFF CALL Tekshed.00495844 
004A5824 · 50 XOR ECX, ECX 
004A5827 · 59 POP ECX 
004A5828 · 59 POP ECX 
004A5829 · 59 POP ECX 
004A5829 > 54:3910 MOU DWORD PTR FS:[TEAKW], EDX 
004A5829 > 58:415541D0 PUSH Tekshed.004A5A94 
004A5831 > 5B:45 CC MOU EDX, DWORD PTR SS:[EBP-34] 
004A5834 > 59:5B02FF5F CALL Tekshed.00404094 
004A5839 > C3 RETN 
004A5839 > EB F0 JMP Tekshed.00404828 
004A5839 > C3 JMP SHORT Tekshed.004A5831 
004A5845 > 50:645 F3 01 JNP BYTE PTR SS:[EBP-0], 1 
004A5845 > 50:8070 00 00 CMP BYTE PTR SS:[EBP+0], 0 
004A5845 > 75:0A JNE SHORT Tekshed.004A5855 
004A5850 > 50:80554000 MOU EDX, DWORD PTR DS:[EBP+40H] 
004A5855 > 50:8352FF5F CALL Tekshed.004391F8 
004A5859 > A1 F4535A00 MOU EDX, DWORD PTR DS:[EBP+40H] 
004A585D > C600 00 JNP SHORT Tekshed.004A5860 
004A585D > EB 17 CMP BYTE PTR SS:[EBP+0], 17 
004A585D > 50:8070 00 00 JNE SHORT Tekshed.004A5876 
004A5863 > 75:11 PUSH 3B 
004A5863 > EB 30 CALL <JMP .user32.MessageBoxBeep> 
004A5867 > EB C829F6FF MOU EDX, Tekshed.004A5ACC 
004A5871 > EB CCS54000 MOU EDX, Tekshed.00494794 
004A5876 > EB 8239FF9F CALL Tekshed.004391F8 
004A5878 > 3C00 XOR ECX, ECX 
004A5878 > 50 POP EDX 
004A5879 > 59 POP ECX 
004A5879 > R9 PNP FNX 
```

Hupakee

PUSH + RET == JMP

BeepType = 1 ICONEXCLAMATION
MessageBeep
ASCII "Registration Key accepted!"

ASCII "Registration Key Failed!"

Stack SS:0012E5E1=00

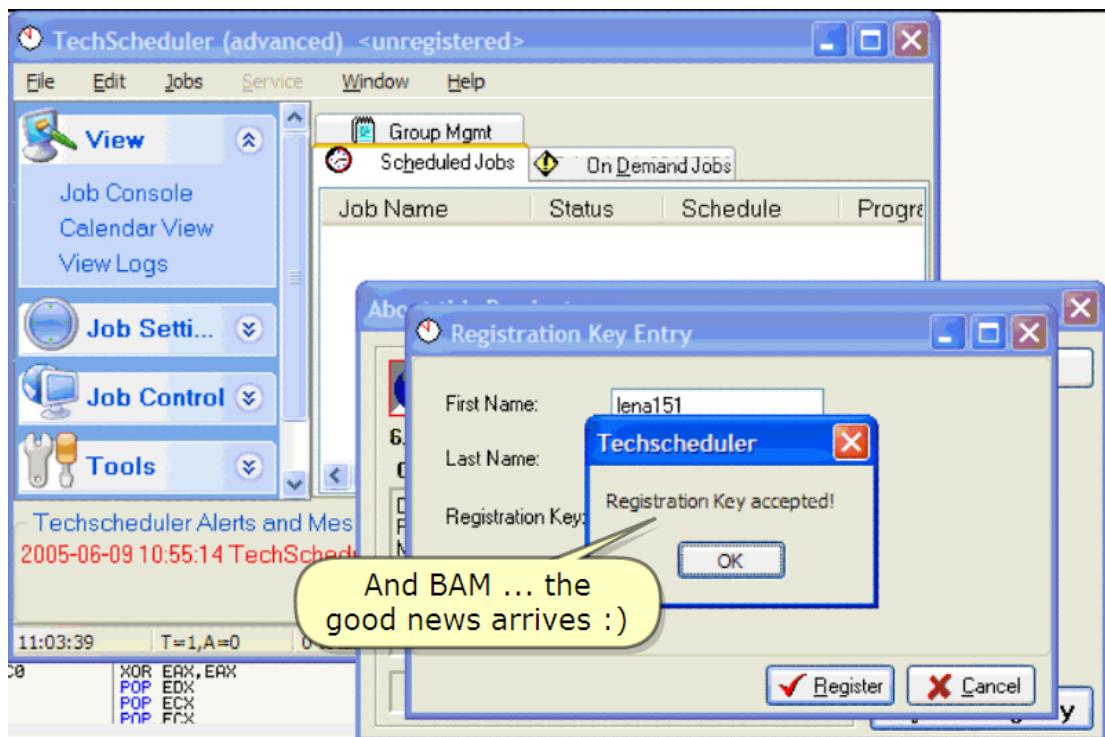
Stack SS:[0012E52F]=00

Hupakēe

C CPU - main thread, module Tekshed

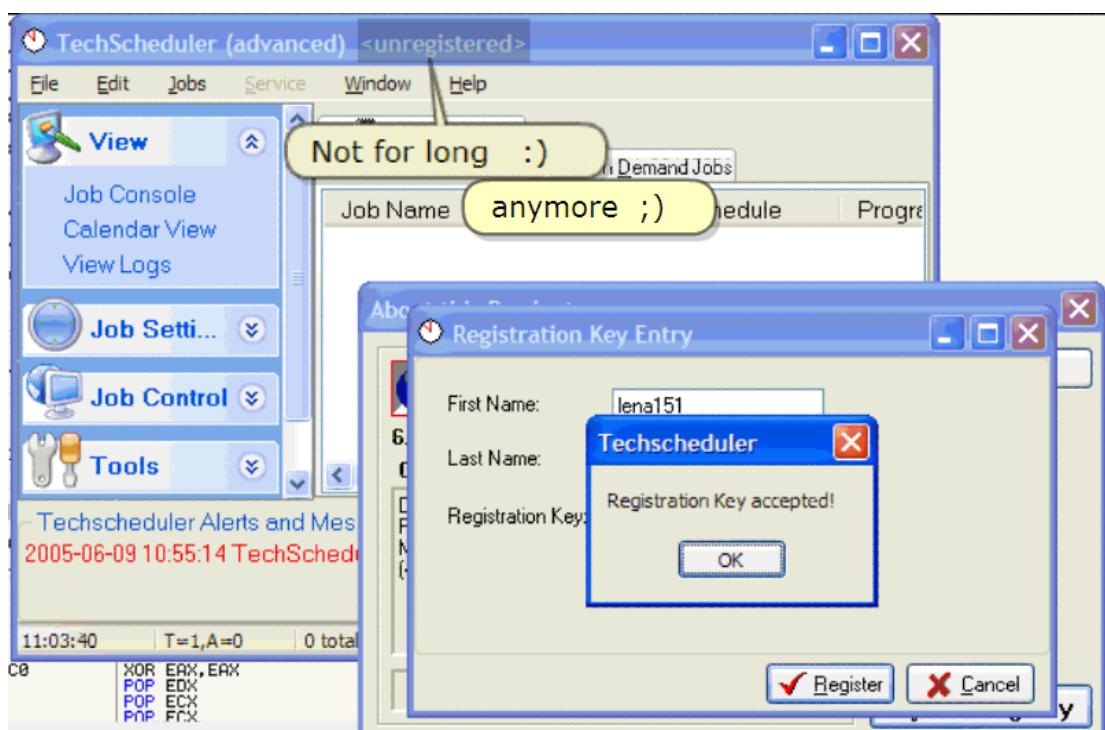
```
004457DC . EB45 CC MOV EAX, DWORD PTR SS:[EBP-34]
004457DF . E8 6C14FFFF CALL Tekshed.00446C50
004457E4 . B095 BCFFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
004457E9 . E8 00000000 XOR EDX, EDX
004457F0 . E8 B2E0FFFF CALL Tekshed.0044A794
004457F2 . B095 BCFFFFFF LEA EDX, DWORD PTR SS:[EBP-144]
004457F8 . B085 BCFFFFFF LEA EAX, DWORD PTR SS:[EBP-174]
004457FE . E8 EDF9FF5F CALL Tekshed.0044851F8
00445803 . B085 BCFFFFFF MOV EAX, DWORD PTR SS:[EBP-174]
00445809 . 50 PUSH EAX
0044580A . B9 985A4B00 MOV ECX, Tekshed.00445A98
0044580F . BA 645A4B00 MOV EDX, Tekshed.00445A94
00445814 . BB45 CC MOV EBX, DWORD PTR SS:[EBP-34]
00445817 . EB 3414FFFF CALL Tekshed.00446C50
0044581C . BB45 CC MOV EBX, DWORD PTR SS:[EBP-34]
0044581F . EB 2000FFFF CALL Tekshed.004495844
00445824 . E9 00000000 XOR EAX, EAX
00445826 . E9 00000000 POP EDX
00445827 . E9 00000000 POP ECX
00445828 . E9 00000000 POP ECX
00445829 . 54 3910 MOV DWORD PTR FS:[EAX], EDK
0044582C . B8 41534B00 PUSH Tekshed.0044A5841
00445831 > B845 CC MOV EAX, DWORD PTR SS:[EBP-34]
00445834 . E8 SBE8FF5F CALL Tekshed.004484094
00445839 . C3 RETN
0044583A . A9 E9EFF5FF JMP Tekshed.004404828
0044583B . EB F0 JMP SHORT Tekshed.00445A831
0044583C . D545 F3 01 MOV BYTE PTR SS:[EBP+1], 0
0044583D . 5570 00 00 JNC SHORT Tekshed.00440555
0044583E . 75 80 JNC SHORT Tekshed.00440555
00445840 . E8 3B5A4B00 MOV EAX, Tekshed.00445A948
00445850 > A1 F4535A00 CALL Tekshed.0044391F8
00445855 . C600 00 MOV EAX, DWORD PTR DS:[5A5F34]
00445858 . EB 17 JHR SHORT Tekshed.0044A5876
0044585F . 5070 00 00 CMP BYTE PTR SS:[EBP+3], 0
00445863 . 75 11 JNC SHORT Tekshed.0044A5876
00445865 . 6A 30 PUSH 30
00445867 . E8 C829F6FF CALL <JMP.>user32.MessageBoxBeep>
0044586C . B8 C55A4B00 MOV EBX, Tekshed.004484094
00445871 . E8 3B29FF5F CALL Tekshed.0044391F8
00445878 > B8C8 XOR EAX, EAX
00445879 . 50 POP EDX
0044587A . 59 POP ECX
0044587B . R9 PNP FFXX

004458A8-Tekshed.00445A88 (ASCII "Registration Key accepted")
EBX=00000000
```



And BAM ... the good news arrives :)

꽝... 좋은 소식이 도착했다 :)



Not for long :)

Anymore ;)

영원히 없다. :)

더 이상 ;)

5. Patching and testing

Scroll up

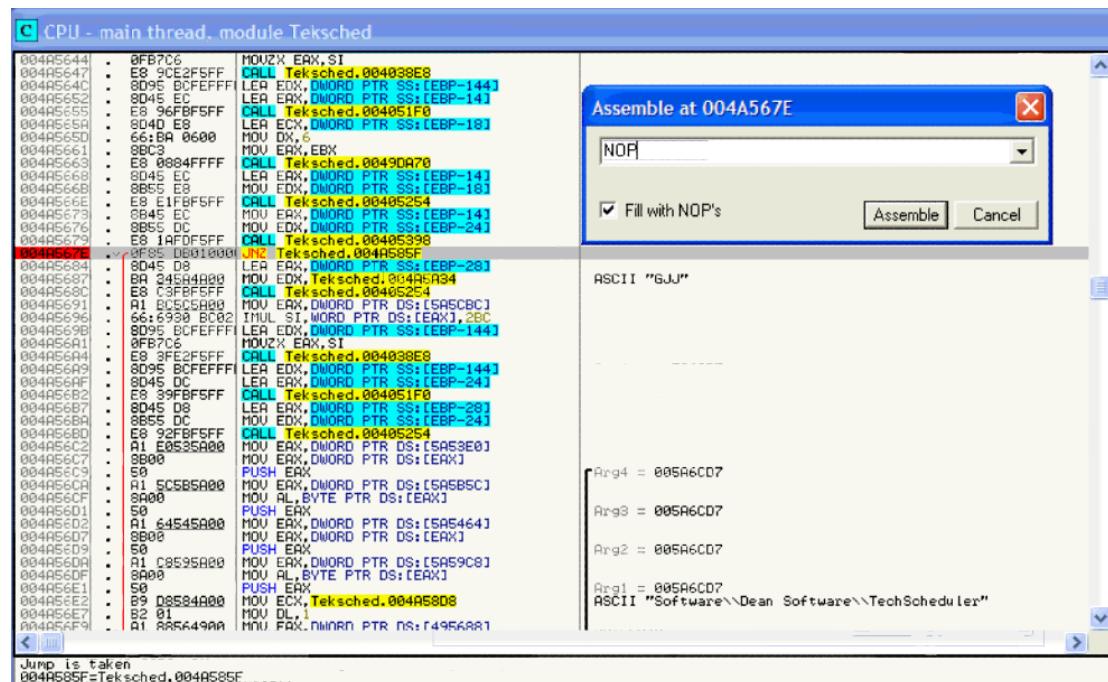
Scroll 올려.

And make the necessary changes permanent

영원히 변화가 필요한 것을 만들어.

And change as follows ...

따라와. 변경하자.



i)

Scroll up

Scroll 올려

C CPU - main thread, module Teksched

00445S09	> 46	INC ESI	Teksched.00489818
00445S0A	. 89C3 02	ADD EBX, 2	
00445S0B	. 66:89FE 1B	CDQ	
00445S0C	^ 895 FBFEBFFF	JNZ Teksched.004A5492	
00445S0D	^ 47	DEC EDI	
00445S0E	. 66:FF4D D2	DEC WORD PTR SS:[EBP-2E]	
00445S0F	^ 8F85 ESFFFFFF	JNZ Teksched.004A5487	
00445S10	> 8095 A8FFFFFF	LEA EDX, DWORD PTR SS:[EBP-1581]	
00445S11	BB45 E4	MOU ERX, DWORD PTR SS:[EBP-1C1]	
00445S12	E8 7442F6FF	CALL Teksched.00409824	
00445S13	BB85 A8FFFFFF	MOU ERX, DWORD PTR SS:[EBP-1581]	
00445S14	. 50	PUSH ERX	
00445S15	0D85 A8FFFFFF	LEA ERX, DWORD PTR SS:[EBP-160]	
00445S16	. 50	PUSH ERX	
00445S17	B9 02000000	MOU ECX, 2	
00445S18	EB 01000000	MOV EDX, ECX	
00445S19	BB45 D8	MOU ERX, DWORD PTR SS:[EBP-28]	
00445S20	. 66:0CFFCFFF	CALL Teksched.004854C4	
00445S21	BB85 A8FFFFFF	MOU ERX, DWORD PTR SS:[EBP-1581]	
00445S22	8095 A8FFFFFF	LEA EDX, DWORD PTR SS:[EBP-1581]	
00445S23	E8 4342F6FF	CALL Teksched.00409824	
00445S24	BB95 A8FFFFFF	MOU EDX, DWORD PTR SS:[EBP-150]	
00445S25	. 50	POP ERX	
00445S26	E8 ABFDFFEF	JHL Teksched.004A5398	
00445S27	EB 19	JMP SHORT Teksched.004A5608	
00445S28	. 5070 00 00	CMP BYTE PTR SS:[EBP+91], 0	
00445S29	^ 8F85 7D020000	JNZ Teksched.004A5876	
00445S30	BB 78554000	MOU ERX, Teksched.004A5978	
00445S31	E8 F53BFF9F	CALL Teksched.004391F8	
00445S32	. 5060 000000	JMP Teksched.004A5876	
00445S33	> 66:BE 5802	MOU SI, 258	
00445S34	. 581BE BC02	CDQ SI, 25C	
00445S35	. ^ 19	JMP SHORT Teksched.004A562C	
00445S36	. 5070 00 00	CMP BYTE PTR SS:[EBP+91], 0	
00445S37	^ 8F85 59020000	JNZ Teksched.004A5876	
00445S38	BB E3540000	MOU ERX, Teksched.004A59E4	
00445S39	E8 D138E9FF	CALL Teksched.004391F8	
00445S40	. ^ E9 A0020000	JMP Teksched.004A5876	
00445S41	> A1 BC5C5A00	MOU ERX, DWORD PTR DS:[5A5CBC]	
00445S42	. 66:8000	MOV AX, WORD PTR DS:[ERAX]	
00445S43	. 66:F7EE	IMUL SI	
00445S44	BBF0	MOV ESI, EAX	
00445S45	BB 894E0000	MOU EBX, 4E89	
00445S46	. BB95 FBFEBFFF	LEA EDX, DWORD PTR SS:[EBP-144]	
00445S47	A8F77C	MHU7 FXR, ST	

Stack: SS:[0012E80C] - RP

ASCII "Error! You are attempting to register the wrong lev

ASCII "Error! You are attempting to register the wrong ver

Teksched.005A6CD7

Teksched.005A6CD7

ntdll.7C96E0F0

ntdll.7C96D8A2

C CPU - main thread, module Teksched

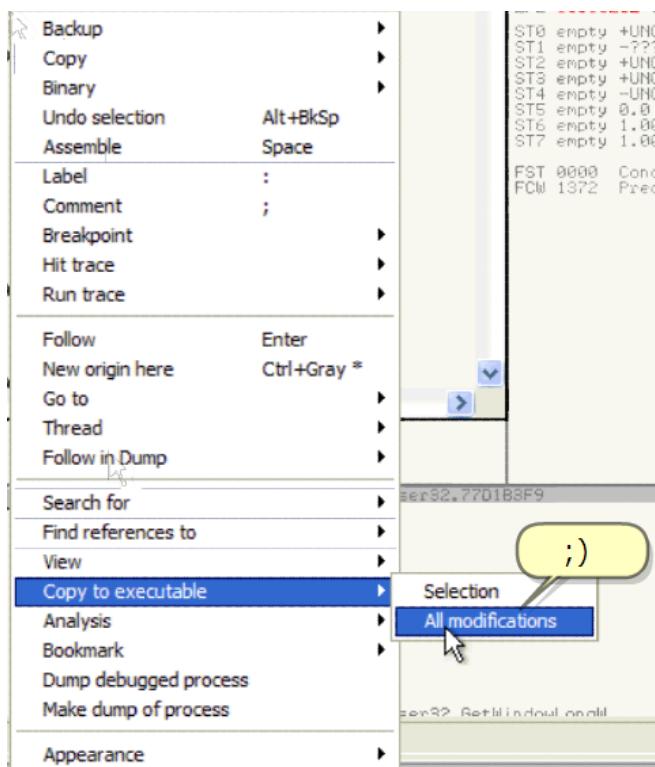
```

004A5589 > 46      INC ESI
004A558A . 83C3 02 ADD EBX,2
004A558B . 66189FE 1B CMP SI,1B
004A5591 .^ 0F85 FBFFFF JNC Teksched.004A5492
004A5597 . 47      INC EDI
004A5598 . 66FF4D D2 DEC WORD PTR SS:[EBP-2E]
004A559C .^ 0F85 E5FFFF JNC Teksched.004A5487
004A55A2 > 0D95 A8FFFF LEA EDX, DWORD PTR SS:[EBP-160]
004A55A8 . 8845 E4 MOV EBX, DWORD PTR SS:[EBP-1C]
004A55B0 . E9 7442F6FF CALL Teksched.004A89824
004A55B4 . 0B85 A0FFFF MOV EDX,DWORD PTR SS:[EBP-160]
004A55B8 . 50      PUSH EAX
004A55B9 . 0D95 A0FFFF MOV EBX,DWORD PTR SS:[EBP-160]
004A55B0 . 50      PUSH EBX
004A55B1 . 89 02000000 MOUL ECX,2
004A55C2 . 80      TR SS:[EBP-28]
004A55C8 . 040549C
004A55D0 . 0        TR SS:[EBP-160]
004A55D4 . 0        TR SS:[EBP-160]
004A55E1 . 0        TR SS:[EBP-160]
004A55E3 . 58      ADD ECX,2
004A55E9 . E8 AP 0F5FF CALL Teksched.004A5603
004A55F0 . 0        JNP SHORT Teksched.004A5603
004A55F1 . 80 08 00 CNE BTNE PTR SS:[EBP-8],0
004A55F3 .^ 0F85 7D020000 JNC Teksched.004A5876
004A55F5 . B8 7B594000 MOU EBX,Teksched.004A5978
004A55F6 . E8 F536F9FF CALL Teksched.004A391F8
004A55F7 . E9 6E020000 JMP Teksched.004A5876
004A55F8 > 66:BE 5802 MOU SI,2BC
004A55F9 . 56181FE BC02 CMP SI,2BC
004A5611 . EB 19 JNP SHORT Teksched.004A562C
004A5613 . 807D 08 00 CNE BTNE PTR SS:[EBP-8],0
004A5617 .^ 0F85 7D020000 JNC Teksched.004A5876
004A5620 . B8 E4594000 MOU EBX,Teksched.004A391F8
004A5622 . E8 D136F9FF CALL Teksched.004A391F8
004A5627 > E9 4A020000 JNP Teksched.004A5876
004A562C . A1 BC5C5A00 MOU EBX,DWORD PTR DS:[5A5CBC]
004A5631 . 66:8B00 MOU EBX,WORD PTR DS:[EBX]
004A5634 . 66:F7EE INUL SI
004A5637 . BBF0 MOV ESI,EAX
004A5639 . BB 894E0000 MOV EBX,4E89
004A563E . SD95 BCFFFFI LEA EDX,DWORD PTR SS:[EBP-144]
004A5644 . 0F87C0 MOUZX FAX,ST

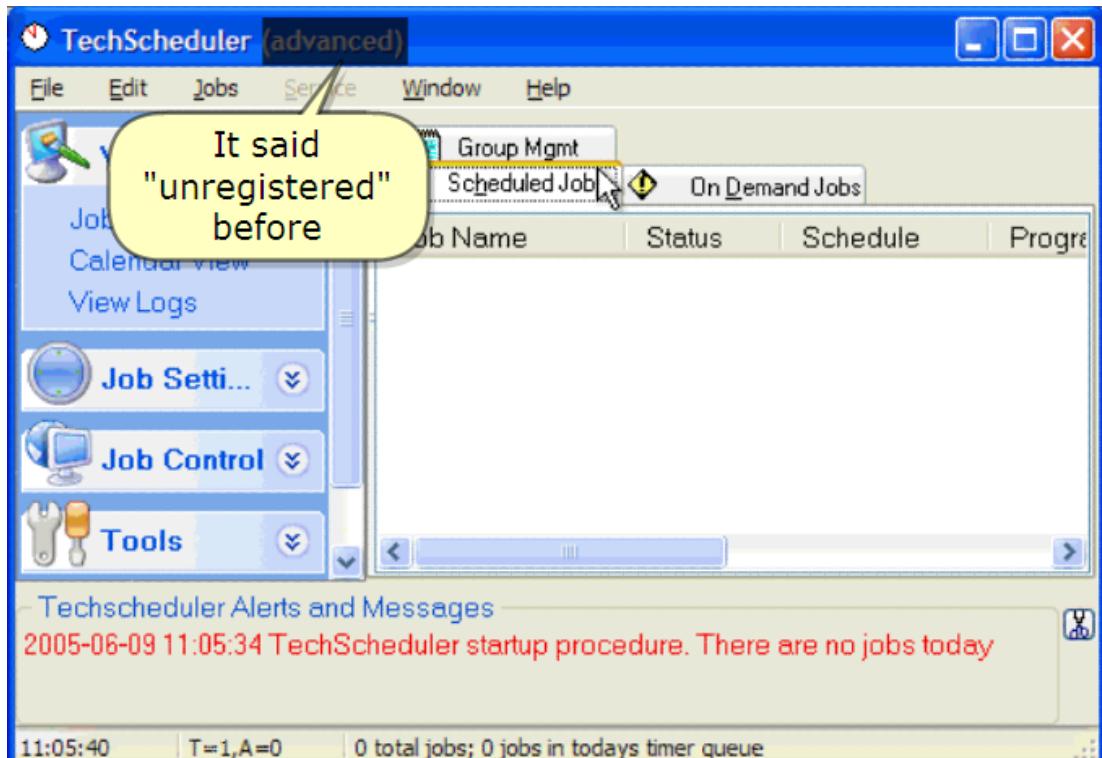
```

;) Scroll up for the last change

마지막 변경을 위해 Scroll 올려.

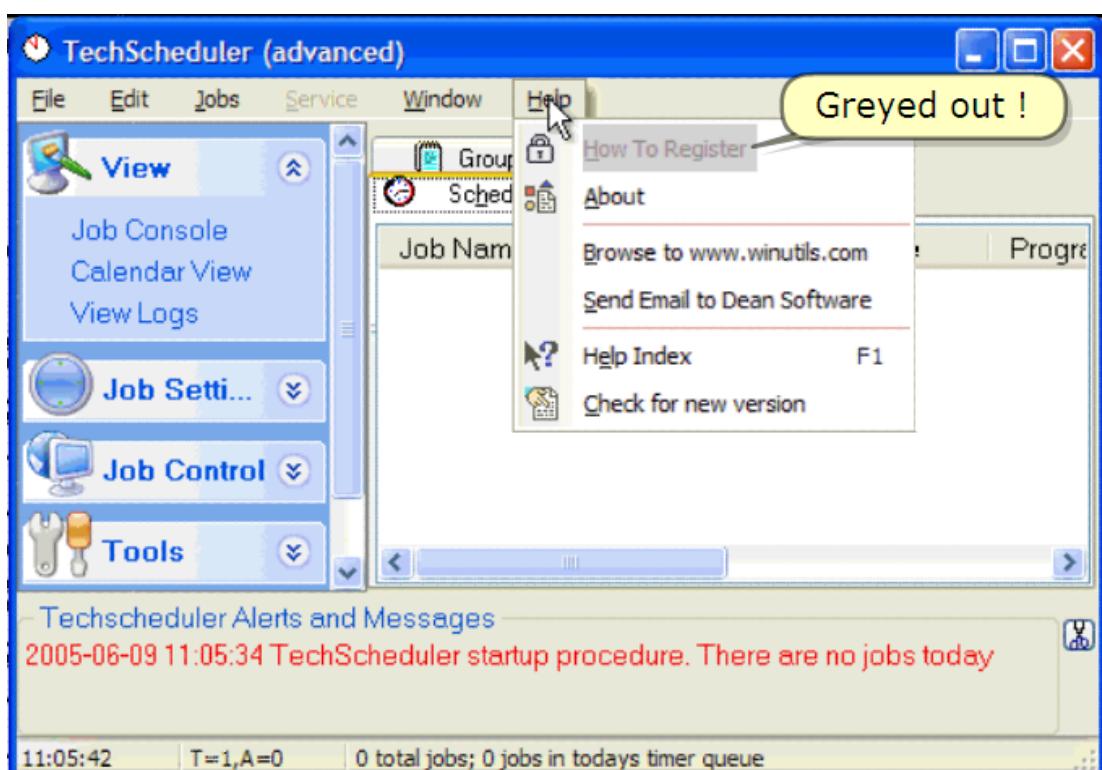


;)
();
();



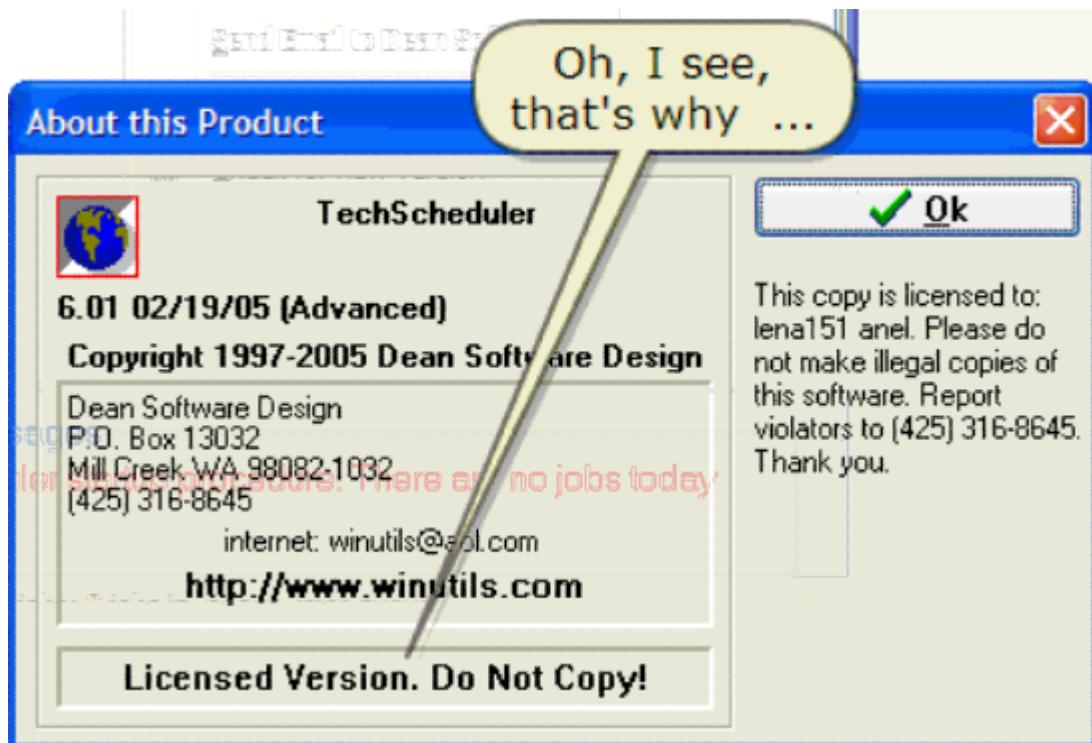
It said "unregistered" before

전에는 "unregistered" 라고 말했다.



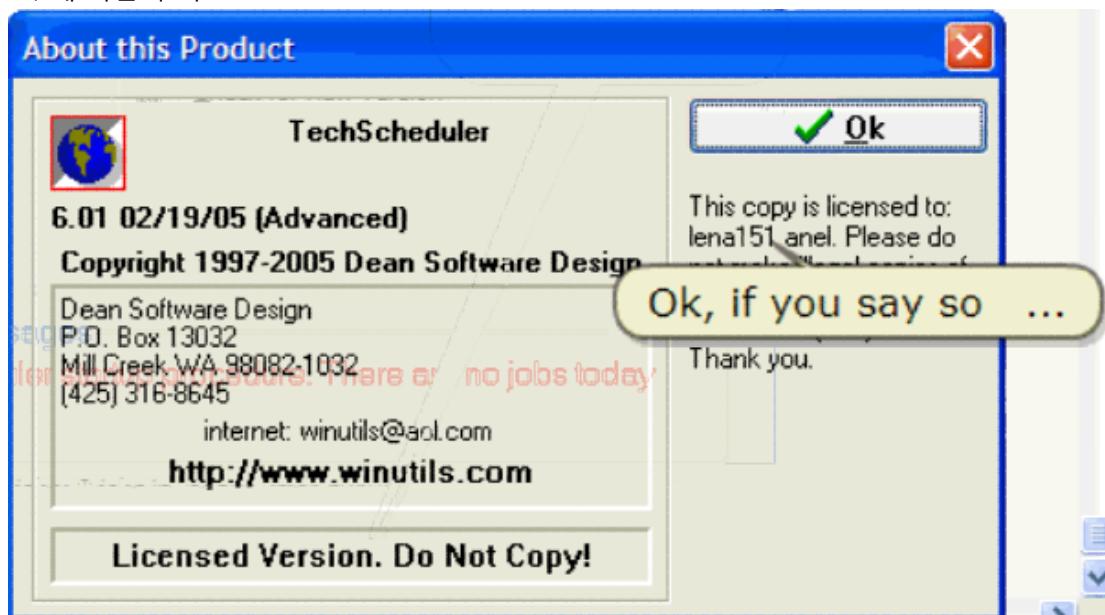
Greyed out !

회색으로 변했어 !



Oh, I see, that's why ...

오, 왜 하는지 봐.



Ok, if you say so ...

Ok, 네 id 기억하고 있어...

Ok. All seems fine.

Ok. 모두 괜찮아 보인다.

INFO :

Only guiding the program by patching the reg scheme to the goodboys doesn't often also effectively register the program.

오직 program 을 reg scheme patching 에 의해 Goodboy 로 guiding 한다. 효과적인 program 등록은 자주 필요하지 않다.

It did in this case however. Remember though that there are often also doublechecks afterwards or at a restart.

이번 경우는 해냈다. 그러나 그곳에 자주 doublecheck 가 후에 있는지 또는 restart 할 때가 있는지 기억해.

There may even be a separate registration scheme that is run at startup.

나눠진 registration scheme 가 있다. Startup 할 때 있다.

We will discuss these cases in later Parts in this series. Stay tuned.

우리는 이 series 의 나중 Parts 에서 이 경우를 의논할 수 있다. 계속 지켜봐.

6. Conclusion

In this part 12, the primary goal was to show multiple patching to guide a program into "registered".

이번 Part 12 에서, 주요한 목표는 등록하기 위해 program 을 다양한 patch 로 guide 하는 것을 보여주는 것이었다.

I hope you understood everything fine and I also hope someone somewhere learned something from this. See me back in part 13 ;)

네가 모든 것을 좋게 이해했기를 희망한다. 또한 누구든지 어디서든지 이것에서 무언가를 배웠으면 좋겠다. Part 13 에서 보자.

The other parts are available at

다른 parts 는 사용 가능하다.

<http://tinyurl.com/27dzdn> (tuts4you)

<http://tinyurl.com/r89zq> (SnD Filez)

<http://tinyurl.com/l6srv> (fixdown)

Regards to all and especially to you for taking the time to look at this tutorial.

Lena151 (2006, updated 2007)

모두에게 안부를 전하고 특별히 이 tutorial 에 시간을 투자해준 너에게 감사한다.