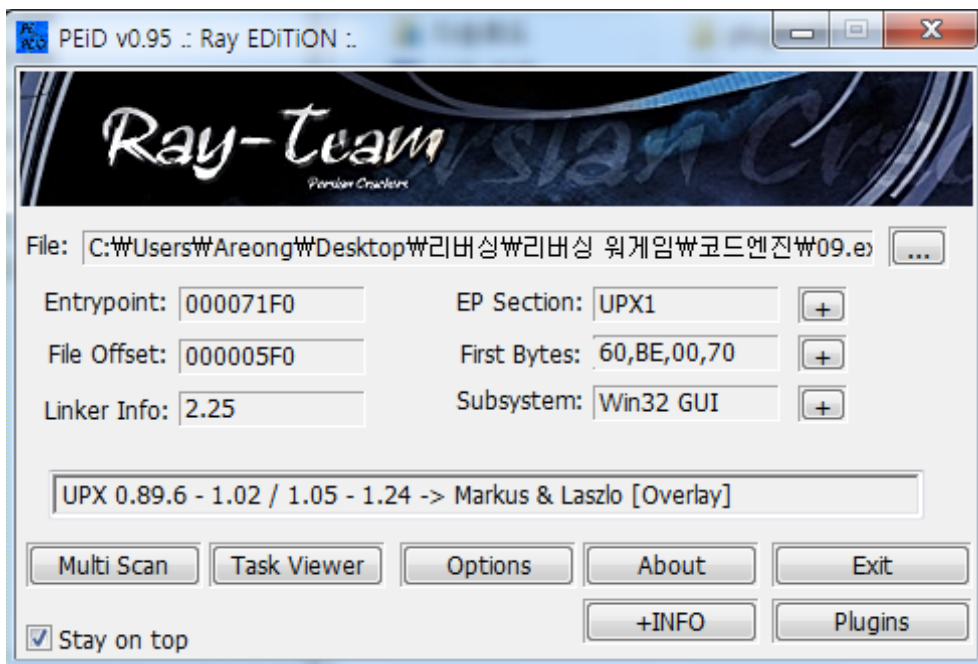


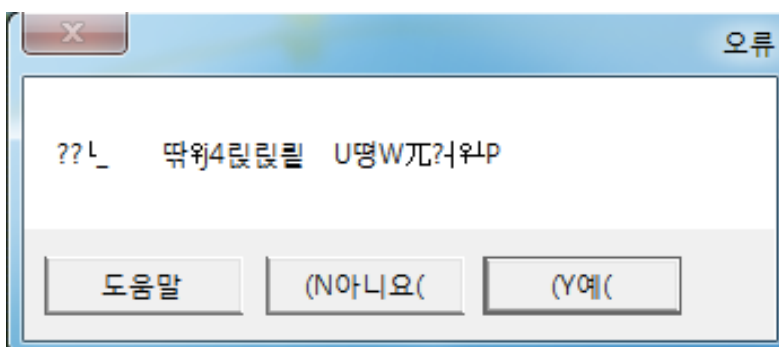
9번 문제랑 같은 문제입니다.

프로그램 실행 화면입니다. keyfile을 찾을 수 없다고 나옵니다.



해당 프로그램은 UPX로 패킹되어 있음을 알 수 있습니다.

언패킹을 하고 진행하도록 하겠습니다.



언패킹 후, 프로그램을 실행해보면 글자가 깨져있는 것을 알 수 있습니다.

OllyDbg - 09.exe

File View Debug Plugins Options Window Help

CPU - main thread, module 09

00401000	90	NOP
00401001	90	NOP
00401002	90	NOP
00401003	90	NOP
00401004	90	NOP
00401005	90	NOP
00401006	90	NOP
00401007	90	NOP
00401008	90	NOP
00401009	90	NOP
0040100A	90	NOP
0040100B	90	NOP
0040100C	6A 00	PUSH 0
0040100E	E8 8C000000	CALL <JMP, &USER32.MessageBoxA>
00401013	6A 00	PUSH 0
00401015	68 80000000	PUSH 80
0040101A	6A 03	PUSH 3
0040101C	6A 00	PUSH 0

hOwner = NULL  
 hWndTemplateFile = NULL  
 Attributes = NORMAL  
 Mode = OPEN\_EXISTING  
 pSecurity = NULL

이유가 무엇인지 올리디버거로 살펴해보도록 하겠습니다.

MessageBox 함수를 호출하는 부분에 인자값 부분이 NOP으로 채워져 있습니다.

인자값이 NOP라서 메세지 박스의 글자가 깨지는 것을 알 수 있습니다.

CPU - main thread, module 09

00407354	57	PUSH EDI
00407355	FFD5	CALL EBP
00407357	8D87 1F020000	LEA EAX, DWORD PTR DS:[EDI+21F]
0040735D	8020 7F	AND BYTE PTR DS:[EAX], 7F
00407360	8060 28 7F	AND BYTE PTR DS:[EAX+28], 7F
00407364	58	POP EAX
00407365	50	PUSH EAX
00407366	54	PUSH ESP
00407367	50	PUSH EAX
00407368	53	PUSH EBX
00407369	57	PUSH EDI
0040736A	FFD5	CALL EBP
0040736C	58	POP EAX
0040736D	61	POPAD
0040736E	6A 00	PUSH 0
00407370	68 00204000	PUSH 09.00402000
00407375	68 12204000	PUSH 09.00402012
0040737A	8D4424 80	LEA EAX, DWORD PTR SS:[ESP+80]
0040737E	75 FA	JNZ SHORT 09.0040737E
00407384	83EC 80	SUB ESP, -80
00407387	E9 809CFFFF	JMP 09.0040100C
0040738C	00	DB 00

ASCII "abex' 3rd crackme"  
 ASCII "Click OK to check for the keyfile."

언패킹을 하지 않은 코드입니다.

잘 보시면 POPAD뒤에 MessageBox의 인자값으로 보이는 문자열들이 있습니다.

00407367	50	PUSH EAX			EIP 00000000
00407368	53	PUSH EBX			EIP 00407387 09.00407387
00407369	57	PUSH EDI			C 1 ES 0023 32bit 0(FFFFFFFF)
0040736A	FFD5	CALL EBIP			P 0 CS 001B 32bit 0(FFFFFFFF)
0040736C	58	POP EAX			A 0 SS 0023 32bit 0(FFFFFFFF)
0040736D	B1	POPAD			Z 0 DS 0023 32bit 0(FFFFFFFF)
0040736E	6A 00	PUSH 0			S 0 FS 003B 32bit 7FFDF000(FFF)
00407370	68 00204000	PUSH 09.00402000	ASCII "abex' 3rd crackme"		T 0 GS 0000 NULL
00407375	68 12204000	PUSH 09.00402012	ASCII "Click OK to check for the keyfile."		D 0
00407376	5B 4424 00	LEN EAX, EBX+1, 0, 0, 0, 0, 0, 0			D 0 LastErr ERROR_SUCCESS (00000000)
0040737E	6A 00	PUSH 0			EFL 00000203 (NO,B,NE,BE,NS,PO,GE,G)
00407380	39C4	CMP ESP, EAX			ST0 empty 0.0
00407382	75 FA	JNZ SHORT 09.0040737E			ST1 empty 0.0
00407384	83EC 80	SUB ESP, -80			ST2 empty 0.0
00407387	E9 809CFFFF	JMP 09.0040100C			ST3 empty 0.0
0040738C	00	DB 00			ST4 empty 0.0
0040738D	00	DB 00			ST5 empty 0.0
0040738E	00	DB 00			ST6 empty 1.00000000000000000000
0040738F	00	DB 00			ST7 empty 1.1071487177940905030
0040100C=09.0040100C					
Address	Hex dump	ASCII			
00408000	00 00 00 00 DC 36 E5 37	....??			
00408008	00 00 00 00 00 00 01 00	.....f.			
00408010	10 00 00 00 18 00 00 80	+...T...			
00408018	00 00 00 00 DC 36 E5 37	....??			
00408020	00 00 00 00 00 00 00 00	.....			
0012FF80	00402012	ASCII "Click OK to check for the keyfile."			
0012FF84	00402000	ASCII "abex' 3rd crackme"			
0012FF88	00000000				
0012FF8C	75F33C45	RETURN to kernel32.75F33C45			
0012FF90	7FFD4000				
0012FF94	7FFD4000				

그 문자열들이 JMP를 통해 원래 소스코드 흐름으로 돌아가기 전에 스택에 저장되어 있습니다.

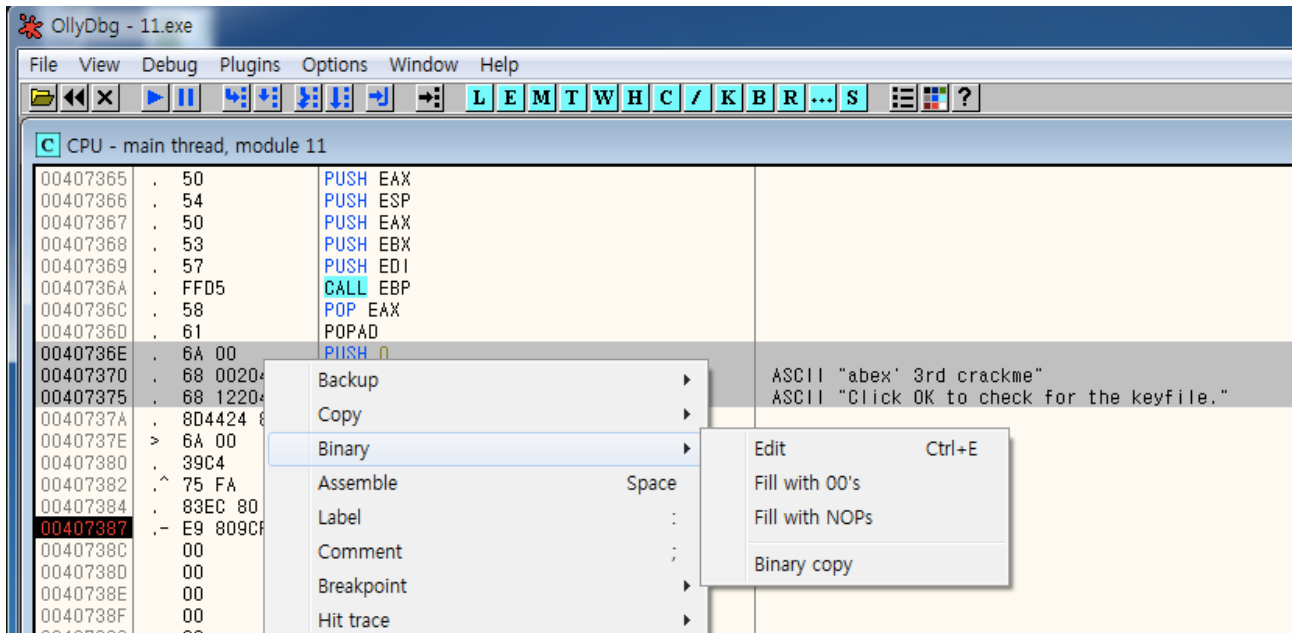
OlllyDbg - 09.exe					
File View Debug Plugins Options Window Help					
CPU - main thread, module 09					
0040100C	6A 00	PUSH 0			Registers (FPU)
0040100E	E8 8C000000	CALL 09.0040109F	JMP to USER32.MessageBoxA		EAX 0012FF00
00401013	6A 00	PUSH 0			ECX 00000000
00401015	68 80000000	PUSH 80			EDX 004071F0 09.<ModuleEntryPoint>
0040101A	6A 03	PUSH 3			EBX 7FFD4000
0040101C	6A 00	PUSH 0			ESP 0012FF70
0040101E	6A 00	PUSH 0			EBP 0012FF94
00401020	68 00000080	PUSH 80000000			ESI 00000000
00401025	68 B9204000	PUSH 09.00402089	ASCII "abex.i2c"		EDI 00000000
0040102A	E8 5E000000	CALL 09.0040108D			EIP 0040100E 09.0040100E
0040102F	A3 CA204000	MOV DWORD PTR DS:[4020CA],EAX			C 1 ES 0023 32bit 0(FFFFFFFF)
00401034	83F8 FF	CMP EAX, -1			P 0 CS 001B 32bit 0(FFFFFFFF)
00401037	74 3C	JB SHORT 09.00401075			A 0 SS 0023 32bit 0(FFFFFFFF)
00401039	6A 00	PUSH 0			Z 0 DS 0023 32bit 0(FFFFFFFF)
0040103B	FF35 CA204000	PUSH DWORD PTR DS:[4020CA]			S 0 FS 003B 32bit 7FFDF000(FFF)
00401041	E8 40000000	CALL 09.00401093	JMP to kernel32.GetFileSize		T 0 GS 0000 NULL
00401046	83F8 12	CMP EAX, 12			D 0
00401049	75 15	JNZ SHORT 09.00401060			D 0 LastErr ERROR_SUCCESS (00000000)
0040104B	6A 00	PUSH 0			EFL 00000203 (NO,B,NE,BE,NS,PO,GE,G)
0040104D	68 35204000	PUSH 09.00402035	ASCII "Well done!"		ST0 empty 0.0
00401052	68 40204000	PUSH 09.00402040	ASCII "Yep, keyfile found!"		ST1 empty 0.0
00401057	6A 00	PUSH 0			ST2 empty 0.0
00401059	E8 41000000	CALL 09.0040109F	JMP to USER32.MessageBoxA		ST3 empty 0.0
0040105E	EB 28	JMP SHORT 09.00401088			ST4 empty 0.0
00401060	6A 00	PUSH 0			ST5 empty 0.0
00401062	68 79204000	PUSH 09.00402079	ASCII "Error"		ST6 empty 1.00000000000000000000
00401067	68 7F204000	PUSH 09.0040207F	ASCII "The found file is not a valid keyfile!"		ST7 empty 1.1071487177940905030
0040109F=09.0040109F					
Address	Hex dump	ASCII			
00408000	00 00 00 00 DC 36 E5 37	....??			
00408008	00 00 00 00 00 00 01 00	.....f.			
00408010	10 00 00 00 18 00 00 80	+...T...			
00408018	00 00 00 00 DC 36 E5 37	....??			
00408020	00 00 00 00 00 00 00 00	.....			
0012FF7C	00000000	owner = NULL			
0012FF80	00402012	Text = "Click OK to check for the keyfile."			
0012FF84	00402000	Title = "abex' 3rd crackme"			
0012FF88	00000000	Style = MB_OK MB_APPLMODAL			
0012FF8C	75F33C45	RETURN to kernel32.75F33C45			
0012FF90	7FFD4000				

JMP한 후 바로 MessageBox를 호출하기 됩니다.

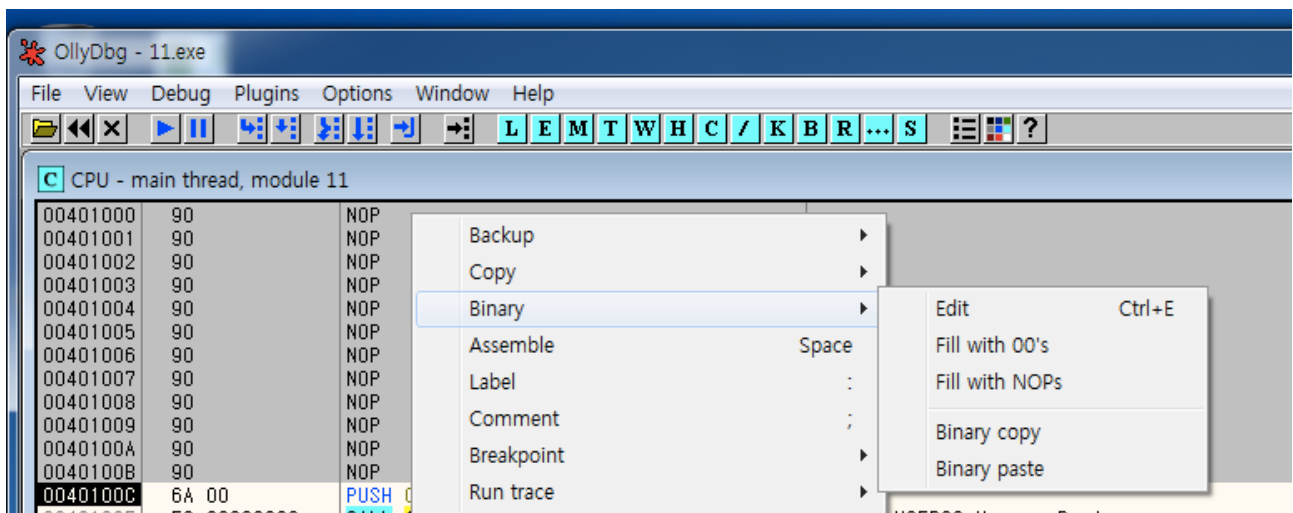
이처럼 특정 소스코드 라인을 숨겨놓는 방법이 Stolen byte라는 일종의 안티 디버깅 기법입니다.

원래의 흐름으로 돌아오기 전에 중요한 함수의 인자값을 스택에 저장하고 그 후 사용하게 됩니다.

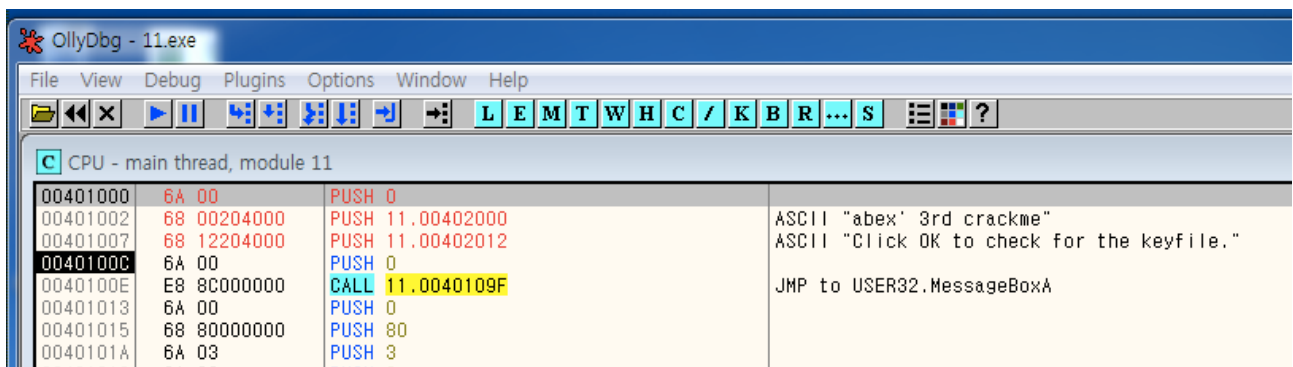
이런 상태에서 툴이나 일반적인 방법으로 언패킹을 하게 되면 인자값을 가져오지 못하게 되는 경우가 발생합니다.



이 값들을 복사해서 원래의 코드 흐름위에 붙혀 넣으면 됩니다.



해당 코드의 크기가 12바이트이기 때문에 12바이트를 선택한 후 붙혀넣기를 합니다.



그러면 이렇게 코드가 옮겨진 것을 볼 수 있고 이대로 저장한 다음 언패킹을 진행하면 됩니다.