

# 리버스 엔지니어링(2)

발표자: 양유진

링크: <https://youtu.be/oiuS7dfvP5o>

# 실습 환경 조성

1. lena reversing 다운로드 >> 공유폴더에 복사 >> 압축풀기

[https://github.com/re4lf10w/lena\\_reversing](https://github.com/re4lf10w/lena_reversing)

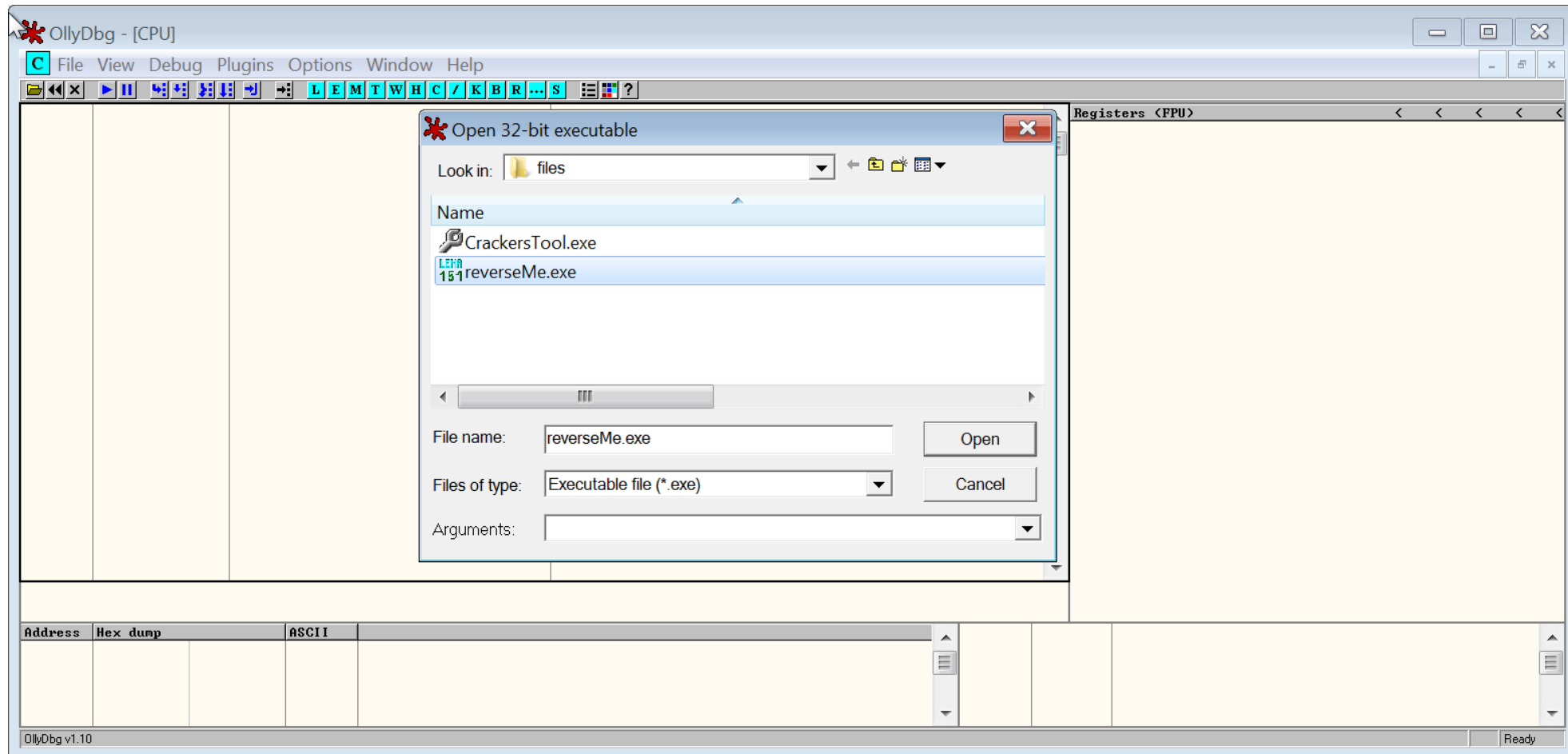
2. lena reversing tutorial 2에서 Keyfile.dat 파일 삭제

lena\_reversing-master\snd-reversingwithlena-tutorials\snd-reversingwithlena-tutorial01.tutorial\files

# 1. OllyDbg로 실습 파일 열기

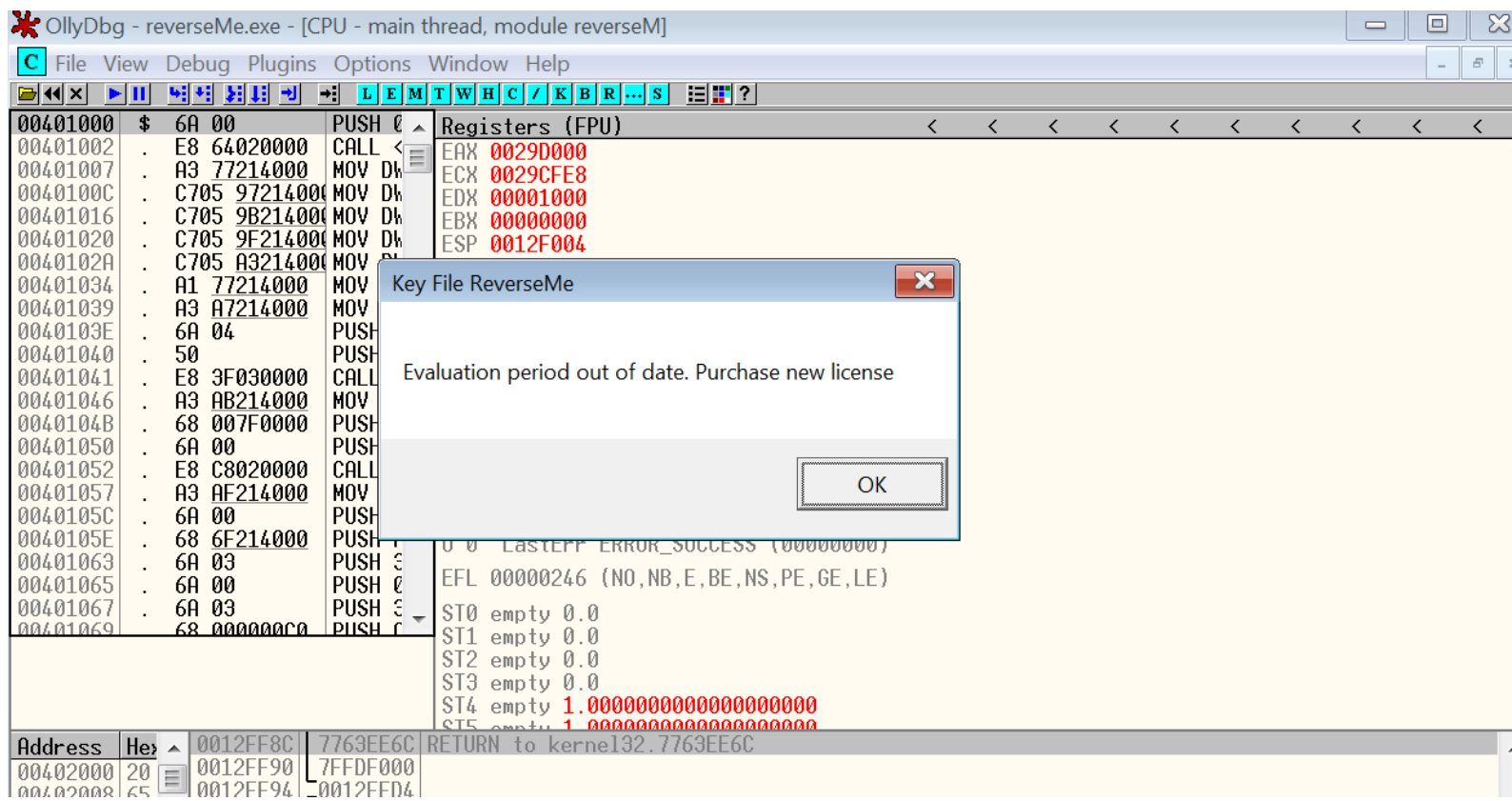
File >> Open >> reverseMe.exe

lena\_reversing-master\snd-reversingwithlena-tutorials\snd-reversingwithlena-tutorial02.tutorial\files

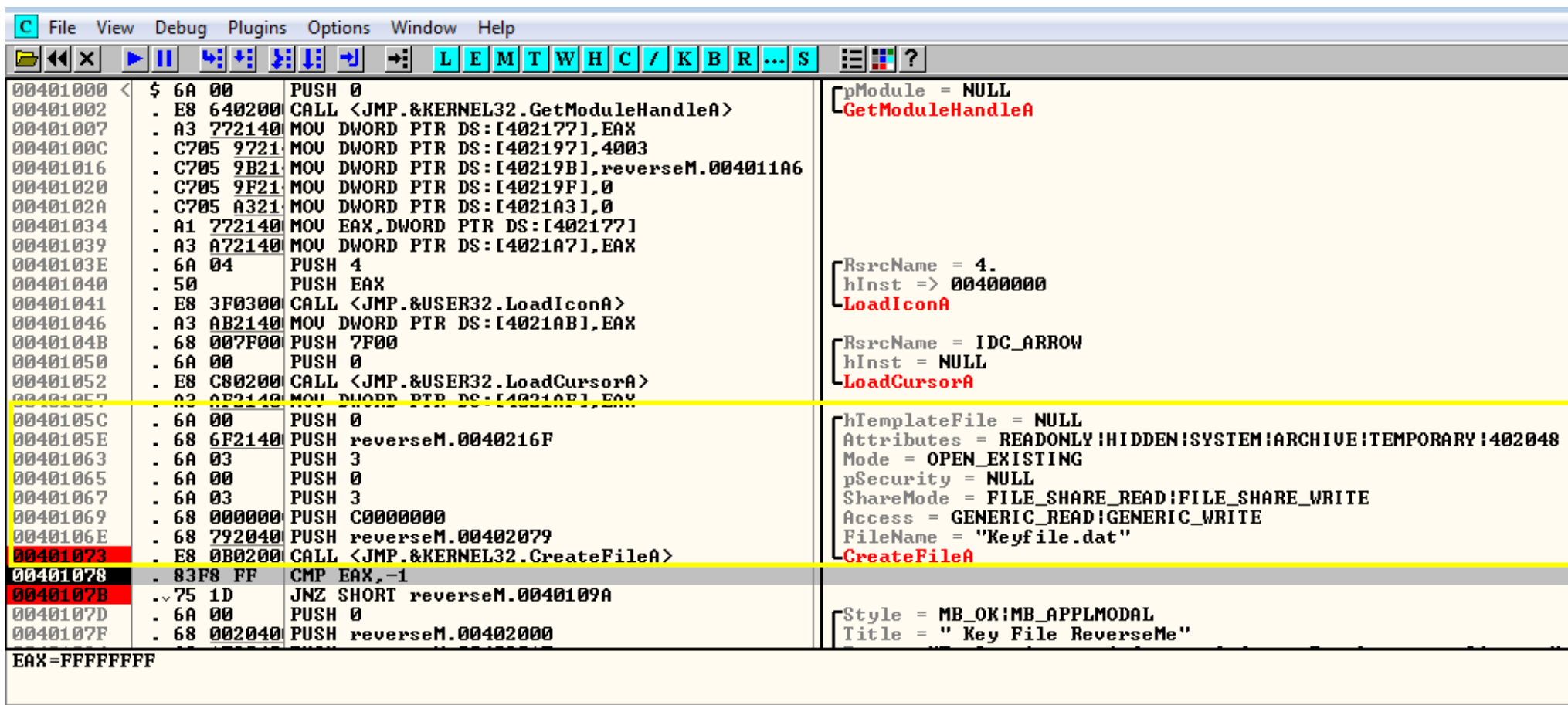


## 2. 리버싱 전 파일 실행

F9 또는 실행버튼(▶) 눌러 실행



### 3. 시도 (1) CreateFileA



Address	Disassembly	Comment
00401000	PUSH 0	
00401002	CALL <JMP.&KERNEL32.GetModuleHandleA>	[pModule = NULL GetModuleHandleA
00401007	MOV DWORD PTR DS:[402177],EAX	
0040100C	MOV DWORD PTR DS:[402197],4003	
00401016	MOV DWORD PTR DS:[40219B],reverseM.004011A6	
00401020	MOV DWORD PTR DS:[40219F],0	
0040102A	MOV DWORD PTR DS:[4021A3],0	
00401034	MOV EAX,DWORD PTR DS:[402177]	
00401039	MOV DWORD PTR DS:[4021A7],EAX	
0040103E	PUSH 4	[RsrcName = 4. hInst => 00400000
00401040	PUSH EAX	LoadIconA
00401041	CALL <JMP.&USER32.LoadIconA>	
00401046	MOV DWORD PTR DS:[4021AB],EAX	
0040104B	PUSH 7F00	[RsrcName = IDC_ARROW hInst = NULL
00401050	PUSH 0	LoadCursorA
00401052	CALL <JMP.&USER32.LoadCursorA>	
00401057	MOV DWORD PTR DS:[4021AF],EAX	
0040105C	PUSH 0	[hTemplateFile = NULL Attributes = READONLY HIDDEN SYSTEM ARCHIVE TEMPORARY 402048
0040105E	PUSH reverseM.0040216F	Mode = OPEN_EXISTING
00401063	PUSH 3	pSecurity = NULL
00401065	PUSH 0	ShareMode = FILE_SHARE_READ FILE_SHARE_WRITE
00401067	PUSH 3	Access = GENERIC_READ GENERIC_WRITE
00401069	PUSH C0000000	FileName = "Keyfile.dat"
0040106E	PUSH reverseM.00402079	CreateFileA
00401073	CALL <JMP.&KERNEL32.CreateFileA>	
00401078	CMP EAX,-1	
0040107B	JNZ SHORT reverseM.0040109A	
0040107D	PUSH 0	[Style = MB_OK MB_APPLMODAL Title = "Key File ReverseMe"
0040107F	PUSH reverseM.00402000	

EAX=FFFFFFFF

### 3. 시도 (1) CreateFileA

```
hTemplateFile = NULL
Attributes = READONLY|HIDDEN|SYSTEM|ARCHIVE|TEMPORARY|402048
Mode = OPEN_EXISTING
pSecurity = NULL
ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE
Access = GENERIC_READ|GENERIC_WRITE
FileName = "Keyfile.dat"
CreateFileA
```

hTemplateFile	기존 파일과 동일한 특성을 가지는 새 파일 생성시 사용되는 전달인자 // 기존파일 열 때는 NULL
Attributes	파일 속성//읽기전용, 숨김파일, 운영체제 독점적, 기록O, 캐시 메모리에 저장
Mode	파일 존재 여부에 따른 수행 작업 // OPEN_EXISTING: 파일이 존재할 경우에만 열기
pSecurity	파일 보안 속성 // NULL이 기본값
ShareMode	공유모드 // 다른 프로세스가 FILE_SHARE_READ: 파일 읽기 허용, FILE_SHARE_WRITE: 파일 쓰기 허용
Access	파일 권한 // GENERIC_READ: 읽기 가능, GENERIC_WRITE: 쓰기 가능
FileName	파일 이름

### 3. 시도 (1) CreateFileA

00401002	. E8 640200 CALL <JMP.&KERNEL32.GetModuleHandleA>	GetModuleHandleA	EAX FFFFFFFF
00401007	. A3 772140 MOV DWORD PTR DS:[402177],EAX		ECX 77676860 ntdll.77676860
0040100C	. C705 9721 MOV DWORD PTR DS:[402197],4003		EDX 001F0174
00401016	. C705 9B21 MOV DWORD PTR DS:[40219B],reverseM.004011A6		EBX 7FFDF000
00401020	. C705 9F21 MOV DWORD PTR DS:[40219F],0		ESP 0012FF8C
0040102A	. C705 A321 MOV DWORD PTR DS:[4021A3],0		EBP 0012FF94
00401034	. A1 772140 MOV EAX,DWORD PTR DS:[402177]		ESI 00000000
00401039	. A3 A72140 MOV DWORD PTR DS:[4021A7],EAX		EDI 00000000
0040103E	. 6A 04 PUSH 4	[RsrcName = 4.	EIP 00401078 reverseM.00401078
00401040	. 50 PUSH EAX	hInst => 00400000	C 0 ES 0023 32bit 0<FFFFFFFF>
00401041	. E8 3F0300 CALL <JMP.&USER32.LoadIconA>	LoadIconA	P 1 CS 001B 32bit 0<FFFFFFFF>
00401046	. A3 AB2140 MOV DWORD PTR DS:[4021AB],EAX		O 0 SS 0023 32bit 0<FFFFFFFF>
0040104B	. 68 007F00 PUSH 7F00	[RsrcName = IDC_ARROW	Z 1 DS 0023 32bit 0<FFFFFFFF>
00401050	. 6A 00 PUSH 0	hInst = NULL	S 0 FS 003B 32bit 7FFDE000<FFF>
00401052	. E8 C80200 CALL <JMP.&USER32.LoadCursorA>	LoadCursorA	T 0 GS 0000 NULL
00401057	. A3 AF2140 MOV DWORD PTR DS:[4021AF],EAX		D 0
0040105C	. 6A 00 PUSH 0	hTemplateFile = NULL	C 0 LastErr ERROR_FILE_NOT_FOUND <00000002>
0040105E	. 68 6F2140 PUSH reverseM.0040216F	Attributes = READONLY HIDDEN SYSTEM ARCHIVE TEMPORARY 402048	EFL 00000246 <NO,NB,E,BE,NS,PE,GE,LE>
00401063	. 6A 03 PUSH 3	Mode = OPEN_EXISTING	SI0 empty 0.0
00401065	. 6A 00 PUSH 0	pSecurity = NULL	SI1 empty 0.0
00401067	. 6A 03 PUSH 3	ShareMode = FILE_SHARE_READ FILE_SHARE_WRITE	SI2 empty 0.0
00401069	. 68 000000 PUSH C0000000	Access = GENERIC_READ GENERIC_WRITE	SI3 empty 0.0
0040106E	. 68 792040 PUSH reverseM.00402079	FileName = "Keyfile.dat"	SI4 empty 0.0
00401073	. E8 0B0200 CALL <JMP.&KERNEL32.CreateFileA>	CreateFileA	SI5 empty 0.0
00401078	. 83F8 FF CMP EAX,-1		SI6 empty 0.0
0040107B	. 75 1D JNZ SHORT reverseM.0040109A		SI7 empty 0.0
0040107D	. 6A 00 PUSH 0	Style = MB_OK MB_APPLMODAL	
0040107F	. 68 002040 PUSH reverseM.00402000	Title = " Key File ReverseMe"	
EAX=FFFFFFFF			3 2 1 0 ESPUOZDI
			FSI 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 <GT>
			FCW 027F Prec NEAR.53 Mask 1 1 1 1 1 1

EAX: 함수의 반환값 전달하는 역할 (FFFFFFFF = -1)

값이 같으면 Z 플래그는 1이 됨.

CMP 결과는 Z 플래그에 영향을 미침.

\*CMP 연산 방법: (비교조건1)-(비교조건2)

### 3. 시도 (1) CreateFileA

00401041	E8 3F0300	CALL <JMP.&USER32.LoadIconA>	LoadIconA	C 0 ES 0023 32bit 0<FFFFFFFF>
00401046	A3 AB2140	MOV DWORD PTR DS:[4021AB1],EAX		P 1 CS 001B 32bit 0<FFFFFFFF>
0040104B	68 007F00	PUSH 7F00	RsrcName = IDC_ARROW	A 0 SS 0023 32bit 0<FFFFFFFF>
00401050	6A 00	PUSH 0	hInst = NULL	Z 1 DS 0023 32bit 0<FFFFFFFF>
00401052	E8 C80200	CALL <JMP.&USER32.LoadCursorA>	LoadCursorA	S 0 FS 003B 32bit 7FFDE000<FFF>
00401057	A3 AF2140	MOV DWORD PTR DS:[4021AF1],EAX		T 0 GS 0000 NULL
0040105C	6A 00	PUSH 0	hTemplateFile = NULL	D 0
0040105E	68 6F2140	PUSH reverseM.0040216F	Attributes = READONLY HIDDEN SYSTEM ARCHIVE TEMPORARY 402048	O 0 LastErr ERROR_FILE_NOT_FOUND <00000002>
00401063	6A 03	PUSH 3	Mode = OPEN_EXISTING	EFL 00000246 <NO,NB,E,BE,NS,PE,GE,LE>
00401065	6A 00	PUSH 0	pSecurity = NULL	ST0 empty 0.0
00401067	6A 03	PUSH 3	ShareMode = FILE_SHARE_READ FILE_SHARE_WRITE	ST1 empty 0.0
00401069	68 000000	PUSH C0000000	Access = GENERIC_READ GENERIC_WRITE	ST2 empty 0.0
0040106E	68 792040	PUSH reverseM.00402079	FileName = "Keyfile.dat"	ST3 empty 0.0
00401073	E8 0B0200	CALL <JMP.&KERNEL32.CreateFileA>	CreateFileA	ST4 empty 0.0
00401078	83F8 FF	CMP EAX,-1		ST5 empty 0.0
0040107B	75 1D	JNZ SHORT reverseM.0040109A		ST6 empty 0.0
0040107D	6A 00	PUSH 0	Style = MB_OK MB_APPLMODAL	ST7 empty 0.0
0040107F	68 002040	PUSH reverseM.00402000	Title = "Key File ReverseMe"	

Jump is NOT taken  
0040109A=reverseM.0040109A

Z=1 JUMP 수행하지 않음.

Registers <FPU>

EAX	001FD000
ECX	001FCFE8
EDX	00001000
EBX	00000000
ESP	0012F004
EBP	0012F058
ESI	7FFDE000
EDI	0012F0EC
EIP	776671B4 ntdll.KiFastSystemCallRet

JNZ(Jump Not Zero): Z=0일 때, JUMP 수행

00401020	C705 9F21	MOV DWORD PTR DS:[40219F1],0		
0040102A	C705 A321	MOV DWORD PTR DS:[4021A31],0		
00401034	A1 722140	MOV EAX,DWORD PTR DS:[4021771]		
00401039	A3 A72140	MOV DWORD PTR DS:[4021A71],EAX		
0040103E	6A 04	PUSH 4	RsrcName = 4.	
00401040	50	PUSH EAX	hInst => 00400000	
00401041	E8 3F0300	CALL <JMP.&USER32.LoadIconA>	LoadIconA	
00401046	A3 AB2140	MOV DWORD PTR DS:[4021AB1],EAX		
0040104B	68 007F00	PUSH 7F00	RsrcName = IDC_ARROW	
00401050	6A 00	PUSH 0	hInst = NULL	
00401052	E8 C80200	CALL <JMP.&USER32.LoadCursorA>	LoadCursorA	
00401057	A3 AF2140	MOV DWORD PTR DS:[4021AF1],EAX		
0040105C	6A 00	PUSH 0	hTemplateFile = NULL	
0040105E	68 6F2140	PUSH reverseM.0040216F	Attributes = READONLY HIDDEN SYSTEM ARCHIVE TEMPORARY 402048	
00401063	6A 03	PUSH 3	Mode = OPEN_EXISTING	
00401065	6A 00	PUSH 0	pSecurity = NULL	
00401067	6A 03	PUSH 3	ShareMode = FILE_SHARE_READ FILE_SHARE_WRITE	
00401069	68 000000	PUSH C0000000	Access = GENERIC_READ GENERIC_WRITE	
0040106E	68 792040	PUSH reverseM.00402079	FileName = "Keyfile.dat"	
00401073	E8 0B0200	CALL <JMP.&KERNEL32.CreateFileA>	CreateFileA	
00401078	83F8 FF	CMP EAX,-1		
0040107B	75 1D	JNZ SHORT reverseM.0040109A		
0040107D	6A 00	PUSH 0	Style = MB_OK MB_APPLMODAL	
0040107F	68 002040	PUSH reverseM.00402000	Title = "Key File ReverseMe"	
00401084	68 172040	PUSH reverseM.00402017	Text = "Evaluation period out of date. Purchase new license"	
00401089	6A 00	PUSH 0	hOwner = NULL	
0040108B	E8 D70200	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA	
00401090	E8 240200	CALL <JMP.&KERNEL32.ExitProcess>	ExitProcess	
00401095	EB 830100	JMP reverseM.00401210		

00401367=<JMP.&USER32.MessageBoxA>

Registers <FPU>

EAX	001FD000
ECX	001FCFE8
EDX	00001000
EBX	00000000
ESP	0012F004
EBP	0012F058
ESI	7FFDE000
EDI	0012F0EC
EIP	776671B4 ntdll.KiFastSystemCallRet

Key File ReverseMe

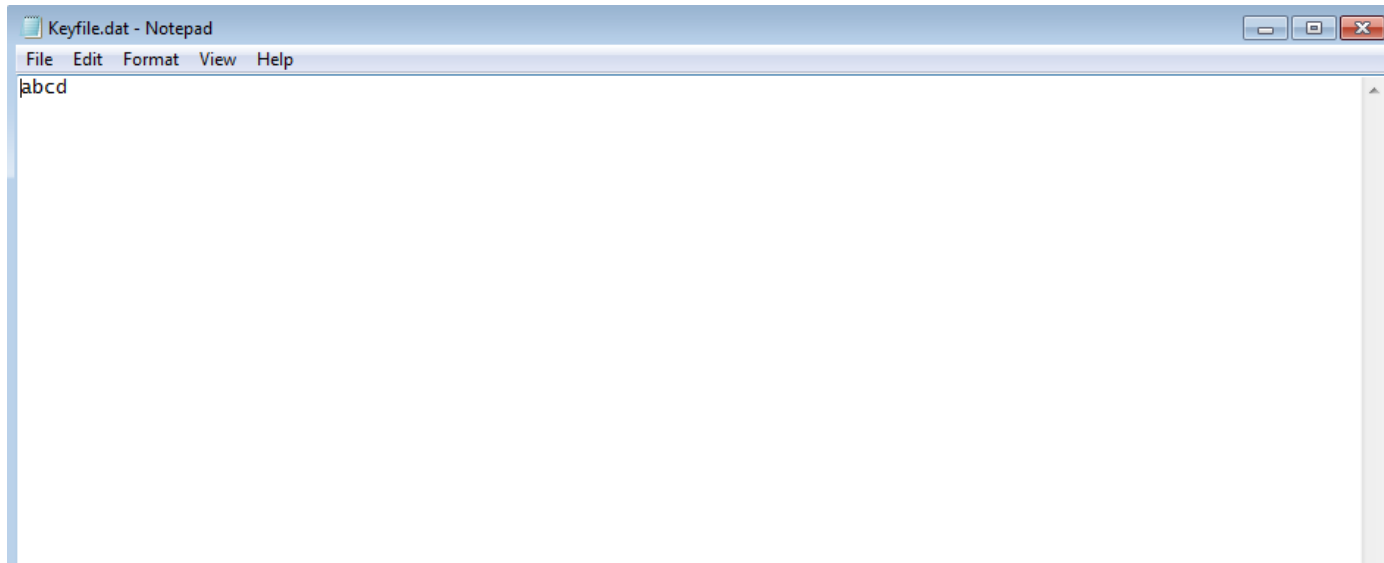
Evaluation period out of date. Purchase new license

OK



### 3. 시도 (1)의 솔루션 - Keyfile.dat 파일 생성

Keyfile.dat 파일이 없어서 CreateFileA 함수에서 -1 을 반환함  
→ 같은 폴더 내에 메모장 이용하여 파일 생성



### 3. 시도 (2) ReadFile

EAX(0000006C)와 -1은 다름

<pre> 00401000 &lt; \$ 6A 00 PUSH 0 00401002 . E8 640200 CALL &lt;JMP.&amp;KERNEL32.GetModuleHandleA&gt; 00401007 . A3 772140 MOV DWORD PTR DS:[402177],EAX 0040100C . C705 9721 MOV DWORD PTR DS:[402197],4003 00401016 . C705 9B21 MOV DWORD PTR DS:[40219B],reverseM.004011A6 00401020 . C705 9F21 MOV DWORD PTR DS:[40219F],0 0040102A . C705 A321 MOV DWORD PTR DS:[4021A3],0 00401034 . A1 772140 MOV EAX,DWORD PTR DS:[402177] 00401039 . A3 A72140 MOV DWORD PTR DS:[4021A7],EAX 0040103E . 6A 04 PUSH 4 00401040 . 50 PUSH EAX 00401041 . E8 3F0300 CALL &lt;JMP.&amp;USER32.LoadIconA&gt; 00401046 . A3 AB2140 MOV DWORD PTR DS:[4021AB],EAX 0040104B . 68 007F00 PUSH 7F00 00401050 . 6A 00 PUSH 0 00401052 . E8 C80200 CALL &lt;JMP.&amp;USER32.LoadCursorA&gt; 00401057 . A3 AF2140 MOV DWORD PTR DS:[4021AF],EAX 0040105C . 6A 00 PUSH 0 0040105E . 68 6F2140 PUSH reverseM.0040216F 00401063 . 6A 03 PUSH 3 00401065 . 6A 00 PUSH 0 00401067 . 6A 03 PUSH 3 00401069 . 68 000000 PUSH C0000000 0040106E . 68 792040 PUSH reverseM.00402079 00401073 . E8 0B0200 CALL &lt;JMP.&amp;KERNEL32.CreateFileA&gt; 00401078 . 83F8 FF CMP EAX,-1 0040107B . 75 1D JNZ SHORT reverseM.0040109A 0040107D . 6A 00 PUSH 0 0040107F . 68 002040 PUSH reverseM.00402000 </pre>	<pre> pModule = NULL GetModuleHandleA  RsrcName = 4. hInst =&gt; 00400000 LoadIconA  RsrcName = IDC_ARROW hInst = NULL LoadCursorA  hTemplateFile = NULL Attributes = READONLY HIDDEN SYSTEM ARCHIVE TEMPORARY 402048 Mode = OPEN_EXISTING pSecurity = NULL ShareMode = FILE_SHARE_READ FILE_SHARE_WRITE Access = GENERIC_READ GENERIC_WRITE FileName = "Keyfile.dat" CreateFileA  Style = MB_OK MB_APPLMODAL Title = " Key File ReverseMe" </pre>	<pre> Registers (FPU) EAX 0000006C ECX 77676860 ntdll.77676860 EDX 001B0174 EBX 7FFD5000 ESP 0012FF8C EBP 0012FF94 ESI 00000000 EDI 00000000 EIP 0040107B reverseM.0040107B C 1 ES 0023 32bit 0&lt;FFFFFFFF&gt; P 0 CS 001B 32bit 0&lt;FFFFFFFF&gt; S 4 SS 0023 32bit 0&lt;FFFFFFFF&gt; Z 0 DS 0023 32bit 0&lt;FFFFFFFF&gt; S 0 FS 003B 32bit 7FFDF000&lt;FFF&gt; T 0 GS 0000 NULL D 0 O 0 LastErr ERROR_SUCCESS &lt;00000000&gt; EFL 00000213 &lt;NO,B,NE,BE,NS,PO,GE,G&gt; ST0 empty 0.0 ST1 empty 0.0 ST2 empty 0.0 ST3 empty 0.0 ST4 empty 0.0 ST5 empty 0.0 ST6 empty 0.0 ST7 empty 0.0 3 2 1 0 ESPUOZDI FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 &lt;GT&gt; FCW 027F Prec NEAR.53 Mask 1 1 1 1 1 1 </pre>
---	--	--

Jump is taken  
0040107B=reverseM.0040109A

Z=0 → 0040109A로 JUMP 수행

### 3. 시도 (2) ReadFile

00401073	. E8 0B0200	CALL <JMP.&KERNEL32.CreateFileA>	CreateFileA
00401078	. 83F8 FF	CMP EAX,-1	
0040107B	. 75 1D	JNZ SHORT reverseM.0040109A	
0040107D	. 6A 00	PUSH 0	
0040107F	. 68 002040	PUSH reverseM.00402000	
00401084	. 68 172040	PUSH reverseM.00402017	
00401089	. 6A 00	PUSH 0	
0040108B	. E8 D70200	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
00401090	. E8 240200	CALL <JMP.&KERNEL32.ExitProcess>	ExitProcess
00401095	. E9 830100	JMP reverseM.0040121D	
0040109A	> 6A 00	PUSH 0	pOverlapped = NULL
0040109C	. 68 732140	PUSH reverseM.00402173	pBytesRead = reverseM.00402173
004010A1	. 6A 46	PUSH 46	BytesToRead = 46 (70.)
004010A3	. 68 1A2140	PUSH reverseM.0040211A	Buffer = reverseM.0040211A
004010A8	. 50	PUSH EAX	hFile
004010A9	. E8 2F0200	CALL <JMP.&KERNEL32.ReadFile>	ReadFile
004010AE	. 85C0	TEST EAX,EAX	
004010B0	. 75 02	JNZ SHORT reverseM.004010B4	
004010B2	. EB 43	JMP SHORT reverseM.004010F7	

pOverlapped    비동기 입출력 위한 OVERLAPPED 구조체 포인터 //NULL은 비동기 입출력 사용X

pBytesRead    입력데이터<BytesToRead일 경우, 읽어들인 바이트 수 저장하는 공간 주소

BytesToRead    한 번에 읽어오는 바이트 수

Buffer    버퍼(읽은 데이터 저장할 공간) 포인터

hFile    파일의 handle

### 3. 시도 (2) ReadFile

00401073	. E8 0B0200	CALL <JMP.&KERNEL32.CreateFileA>	CreateFileA
00401078	. 83F8 FF	CMP EAX,-1	
0040107B	. 75 1D	JNZ SHORT reverseM.0040109A	
0040107D	. 6A 00	PUSH 0	
0040107F	. 68 002040	PUSH reverseM.00402000	Style = MB_OK!MB_APPLMODAL
00401084	. 68 172040	PUSH reverseM.00402017	Title = " Key File ReverseMe"
00401089	. 6A 00	PUSH 0	Text = "Evaluation period out of date. Purchase new license"
0040108B	. E8 D70200	CALL <JMP.&USER32.MessageBoxA>	hOwner = NULL
00401090	. E8 240200	CALL <JMP.&KERNEL32.ExitProcess>	MessageBoxA
00401095	. E9 830100	JMP reverseM.0040121D	ExitProcess
0040109A	> 6A 00	PUSH 0	pOverlapped = NULL
0040109C	. 68 732140	PUSH reverseM.00402173	pBytesRead = reverseM.00402173
004010A1	. 6A 46	PUSH 46	BytesToRead = 46 (70.)
004010A3	. 68 1A2140	PUSH reverseM.0040211A	Buffer = reverseM.0040211A
004010A8	. 50	PUSH EAX	hFile
004010A9	. E8 2F0200	CALL <JMP.&KERNEL32.ReadFile>	ReadFile
004010AE	. 85C0	TEST EAX,EAX	
004010B0	. 75 02	JNZ SHORT reverseM.004010B4	
004010B2	. EB 43	JMP SHORT reverseM.004010F7	

4bytes

Address	Hex dump	ASCII
00402173	04 00 00 00 00 00 40 00	.....@.
0040217B	00 00 00 00 00 00 00 00	.....
00402183	00 00 00 00 00 00 00 00	.....
0040218B	00 00 00 00 00 00 00 00	.....
00402193	00 00 00 00 03 40 00 00	....♥@..
0040219B	A6 11 40 00 00 00 00 00	æ♥@.....
004021A3	00 00 00 00 00 00 40 00	.....@.
004021AB	87 01 28 00 03 00 01 00	ç@(.♥.@.
004021B3	00 00 00 00 00 00 00 00	.....

Address	Hex dump	ASCII
0040211A	61 62 63 64 00 00 00 00	abcd....
00402122	00 00 00 00 00 00 00 00	.....
0040212A	00 00 00 00 00 00 00 00	.....
00402132	00 00 00 00 00 00 00 00	.....
0040213A	00 00 00 00 00 00 00 00	.....

### 3. 시도 (2) ReadFile

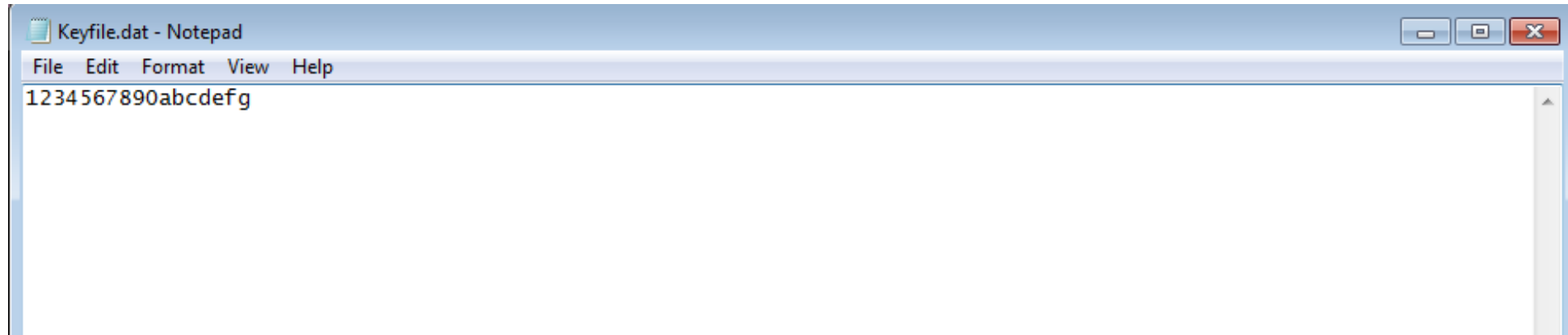
00401073	. E8 0B0200	CALL <JMP.&KERNEL32.CreateFileA>	<b>CreateFileA</b> Style = MB_OK MB_APPLMODAL Title = " Key File ReverseMe" Text = "Evaluation period out of date. Purchase new license" hOwner = NULL <b>MessageBoxA</b> <b>ExitProcess</b>	<b>Registers &lt;FPU&gt;</b> EAX 00000001 ECX 75D79D17 kernel32.75D79D17 EDI 776671B4 ntdll.KiFastSystemCallRe EBX 00000000 ESP 0012FF8C EBP 0012FF94 ESI 00000000 EDI 00000000 EIP 004010BF reverseM.004010BF C 1 ES 0023 32bit 0<FFFFFFFF> P 0 CS 001B 32bit 0<FFFFFFFF> A 0 SS 0023 32bit 0<FFFFFFFF> Z 0 DS 0023 32bit 0<FFFFFFFF> S 1 FS 003B 32bit 7FFDE000<FFF> T 0 GS 0000 NULL D 0 O 0 LastErr ERROR_SUCCESS <00000000> EFL 00000283 <NO,B,NE,BE,S,PO,L,LE> ST0 empty 0.0 ST1 empty 0.0 ST2 empty 0.0 ST3 empty 0.0 ST4 empty 0.0 ST5 empty 0.0 ST6 empty 0.0 ST7 empty 0.0 FST 0000 Cond 0 0 0 0 Err 0 0 0 0 FCW 027F Prec NEAR.53 Mask 1 1 1
00401078	. 83F8 FF	CMP EAX,-1		
0040107B	. 75 1D	JNZ SHORT reverseM.00401079A		
0040107D	. 6A 00	PUSH 0		
0040107F	. 68 00204000	PUSH reverseM.00402000		
00401084	. 68 17204000	PUSH reverseM.00402017		
00401089	. 6A 00	PUSH 0		
0040108B	. E8 D70200	CALL <JMP.&USER32.MessageBoxA>		
00401090	. E8 240200	CALL <JMP.&KERNEL32.ExitProcess>		
00401095	. E9 830100	JMP reverseM.0040121D		
0040109A	. 6A 00	PUSH 0	<b>ReadFile</b> pOverlapped = NULL pBytesRead = reverseM.00402173 BytesToRead = 46 <70.> Buffer = reverseM.0040211A hFile <b>ReadFile</b>	
0040109C	. 68 73214000	PUSH reverseM.00402173		
004010A1	. 6A 46	PUSH 46		
004010A3	. 68 1A214000	PUSH reverseM.0040211A		
004010A8	. 50	PUSH EAX		
004010A9	. E8 2F0200	CALL <JMP.&KERNEL32.ReadFile>		
004010AE	. 85C0	TEST EAX,EAX		
004010B0	. 75 02	JNZ SHORT reverseM.004010B4		
004010B2	. EB 43	JMP SHORT reverseM.004010F7		
004010B4	. 33DB	XOR EBX,EBX		
004010B6	. 33F6	XOR ESI,ESI		
004010B8	. 833D 7321	CMP DWORD PTR DS:[402173],10	<b>TEST: 두 operand를 AND 연산하는 명령어</b> <b>4&lt;16(x10) → JUMP 수행</b> <b>JL(Jump if A less than B): A&lt;B이면 JUMP</b> <b>*CMP와 함께 쓰임.</b>	
004010BF	. 7C 36	JL SHORT reverseM.004010F7		
004010C1	. 8A83 1A21	MOV AL,BYTE PTR DS:[EBX+40211A]		
004010C7	. 3C 00	CMP AL,0		
004010C9	. 74 08	JE SHORT reverseM.004010D3		
004010CB	. 3C 47	CMP AL,47		
004010CD	. 75 01	JNZ SHORT reverseM.004010D0		
004010CF	. 46	INC ESI		
Jump is taken 004010F7=reverseM.004010F7				

004010F7	. 6A 00	PUSH 0	<b>Style = MB_OK MB_APPLMODAL</b> <b>Title = " Key File ReverseMe"</b> <b>Text = "Keyfile is not valid. Sorry."</b> <b>hOwner = NULL</b> <b>MessageBoxA</b> <b>ExitProcess</b>
004010F9	. 68 00204000	PUSH reverseM.00402000	
004010FE	. 68 86204000	PUSH reverseM.00402086	
00401103	. 6A 00	PUSH 0	
00401105	. E8 5D020000	CALL <JMP.&USER32.MessageBoxA>	
0040110A	. E8 AA010000	CALL <JMP.&KERNEL32.ExitProcess>	

004010BF에서 JUMP 수행하지 않도록 해야함 → 입력 글자>=16

### 3. 시도 (2)의 솔루션 - Keyfile.dat 글자수 늘리기

16바이트 이상이 되도록 16글자 이상 입력함.



### 3. 시도 (3)

004010B8	833D 7321	CMP DWORD PTR DS:[402173],10		C 0 ES 0023 32bit 0<FFFFFFFF>
004010BF	7C 36	JL SHORT reverseM.004010F7		P 1 CS 001B 32bit 0<FFFFFFFF>
004010C1	8A83 1A21	MOV AL,BYTE PTR DS:[EBX+40211A]		A 0 SS 0023 32bit 0<FFFFFFFF>
004010C7	3C 00	CMP AL,0		Z 1 DS 0023 32bit 0<FFFFFFFF>
004010C9	74 08	JE SHORT reverseM.004010D3		S 0 FS 003B 32bit 7FFDF000<FFF>
004010CB	3C 47	CMP AL,47		T 0 GS 0000 NULL
004010CD	75 01	JNZ SHORT reverseM.004010D0		D 0
004010CF	46	INC ESI		0 0 LastErr ERROR_SUCCESS <00000000>
004010D0	43	INC EBX		EFL 00000246 <NO,NB,E,BE,NS,PE,GE,LE>
004010D1	EB EE	JMP SHORT reverseM.004010C1		ST0 empty 0.0
004010D3	83FE 08	CMP ESI,8		ST1 empty 0.0
004010D6	7C 1F	JL SHORT reverseM.004010F7		ST2 empty 0.0
004010D8	E9 280100	JMP reverseM.00401205		ST3 empty 0.0
004010DD	00	DB 00		ST4 empty 0.0
004010DE	00000000	DD 00000000		ST5 empty 0.0
004010E2	00	DB 00		ST6 empty 0.0
004010E3	00	DB 00		ST7 empty 0.0
004010E4	00	DB 00		
DS:[00402173]=00000011			17bytes	

0040109A	6A 00	PUSH 0	pOverlapped = NULL	Registers <FPU>
0040109C	68 732140	PUSH reverseM.00402173	pBytesRead = reverseM.00402173	
004010A1	6A 46	PUSH 46	BytesToRead = 46 <70.>	EAX 00000001
004010A3	68 1A2140	PUSH reverseM.0040211A	Buffer = reverseM.0040211A	ECX 75D79D17 kernel32.75D79D17
004010A8	50	PUSH EAX	hFile	EDX 776671B4 ntdll.KiFastSystemCallRet
004010A9	E8 2F0200	CALL <JMP.&KERNEL32.ReadFile>	ReadFile	EBX 00000000
004010AE	85C0	TEST EAX,EAX		ESP 0012FF8C
004010B0	75 02	JNZ SHORT reverseM.004010B4		EBP 0012FF94
004010B2	EB 43	JMP SHORT reverseM.004010F7		ESI 00000000
004010B4	33DB	XOR EBX,EBX		EDI 00000000
004010B6	33F6	XOR ESI,ESI		EIP 004010BF reverseM.004010BF
004010B8	833D 7321	CMP DWORD PTR DS:[402173],10		C 0 ES 0023 32bit 0<FFFFFFFF>
004010BF	7C 36	JL SHORT reverseM.004010F7		P 0 CS 001B 32bit 0<FFFFFFFF>
004010C1	8A83 1A21	MOV AL,BYTE PTR DS:[EBX+40211A]		A 0 SS 0023 32bit 0<FFFFFFFF>
004010C7	3C 00	CMP AL,0		Z 0 DS 0023 32bit 0<FFFFFFFF>
004010C9	74 08	JE SHORT reverseM.004010D3		S 0 FS 003B 32bit 7FFDF000<FFF>
004010CB	3C 47	CMP AL,47		T 0 GS 0000 NULL
004010CD	75 01	JNZ SHORT reverseM.004010D0		D 0
004010CF	46	INC ESI		0 0 LastErr ERROR_SUCCESS <00000000>
004010D0	43	INC EBX		EFL 00000202 <NO,NB,NE,A,NS,PO,GE,G>
004010D1	EB EE	JMP SHORT reverseM.004010C1		ST0 empty 0.0
004010D3	83FE 08	CMP ESI,8		ST1 empty 0.0
004010D6	7C 1F	JL SHORT reverseM.004010F7		ST2 empty 0.0
004010D8	E9 280100	JMP reverseM.00401205		ST3 empty 0.0
004010DD	00	DB 00		ST4 empty 0.0
004010DE	00000000	DD 00000000		ST5 empty 0.0
004010E2	00	DB 00		ST6 empty 0.0
004010E3	00	DB 00		ST7 empty 0.0
004010E4	00	DB 00		
Jump is NOT taken 004010F7=reverseM.004010F7				FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 <GT>
				FCW 027F Prec NEAR.53 Mask 1 1 1 1 1 1

### 3. 시도 (3)

문자 다 읽을 때까지 4010C1~4010D1 반복

다 읽고나면 4010D3으로 JUMP

```

004010C1 > 8A83 1A21 MOV AL, BYTE PTR DS:[402173],10
004010C7 > 3C 00 CMP AL,0
004010C9 > 74 08 JE SHORT reverseM.004010D3
004010CB > 3C 47 CMP AL,47
004010CD > 75 01 JNZ SHORT reverseM.004010D0
004010CF > 46 INC ESI
004010D0 > 43 INC EBX
004010D1 > EB EE JMP SHORT reverseM.004010C1
004010D3 > 83FE 08 CMP ESI,8
004010D6 > 7C 1F JL SHORT reverseM.004010F7
004010D8 > E9 280100 JMP reverseM.00401205
004010DD > 00 DB 00
004010DE > 00 DD 00000000
004010E2 > 00 DB 00
004010E3 > 00 DB 00
004010E4 > 00 DB 00
  
```

AL=31 ('1')

파일의 문자(40211A에 위치) 1바이트씩 가져와서 AL에 저장

AL=0 → 더 이상 읽을 문자가 없다. (다 읽으면 4010D3으로 JUMP)

JE(Jump Equal): CMP한 값이 같으면 JUMP 수행

0x47(71)은 G의 ASCII 코드 → CMP해서 같으면 JUMP하지 않음.

EBX+1하여 다음 문자 주소로 이동

```

0040109A > 6A 00 PUSH 0
0040109C > 68 732140 PUSH reverseM.00402173
004010A1 > 6A 46 PUSH 46
004010A3 > 68 1A2140 PUSH reverseM.0040211A
004010A8 > 50 PUSH EAX
004010A9 > E8 2F0200 CALL <JMP.&KERNEL32.ReadFile>
004010AE > 85C0 TEST EAX,EAX
004010B0 > 75 02 JNZ SHORT reverseM.004010B4
004010B2 > EB 43 JMP SHORT reverseM.004010F7
004010B4 > 33DB XOR EBX,EBX
004010B6 > 33F6 XOR ESI,ESI
004010B8 > 833D 7321 CMP DWORD PTR DS:[402173],10
004010BF > 7C 36 JL SHORT reverseM.004010F7
004010C1 > 8A83 1A21 MOV AL, BYTE PTR DS:[EBX+40211A]
004010C7 > 3C 00 CMP AL,0
004010C9 > 74 08 JE SHORT reverseM.004010D3
004010CB > 3C 47 CMP AL,47
004010CD > 75 01 JNZ SHORT reverseM.004010D0
004010CF > 46 INC ESI
004010D0 > 43 INC EBX
004010D1 > EB EE JMP SHORT reverseM.004010C1
004010D3 > 83FE 08 CMP ESI,8
004010D6 > 7C 1F JL SHORT reverseM.004010F7
004010D8 > E9 280100 JMP reverseM.00401205
004010DD > 00 DB 00
004010DE > 00 DD 00000000
004010E2 > 00 DB 00
004010E3 > 00 DB 00
004010E4 > 00 DB 00
  
```

```

pOverlapped = NULL
pBytesRead = reverseM.00402173
BytesToRead = 46 (<70.>)
Buffer = reverseM.0040211A
hFile
  ReadFile
  
```

ESI(0) ≠ 8 → Z플래그=0

ESI<8 → (S플래그=1)JUMP수행

Jump is taken  
004010F7=reverseM.004010F7

Registers (FPU)

```

EAX 00000000
ECX 75D79D17 kernel32.75D79D17
EDX 776671B4 ntdll.KiFastSystemCallRet
EBX 00000011
ESP 0012FF8C
EBP 0012FF94
ESI 00000000
EDI 00000000

EIP 004010D6 reverseM.004010D6

C 1 ES 0023 32bit 0<FFFFFFFF>
P 0 CS 001B 32bit 0<FFFFFFFF>
A 1 SS 0023 32bit 0<FFFFFFFF>
Z 0 DS 0023 32bit 0<FFFFFFFF>
S 1 FS 003B 32bit 7FFDF000<FFF>
I 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS <00000000>
EFL 00000293 <NO,B,NE,BE,S,PO,L,LE>

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0

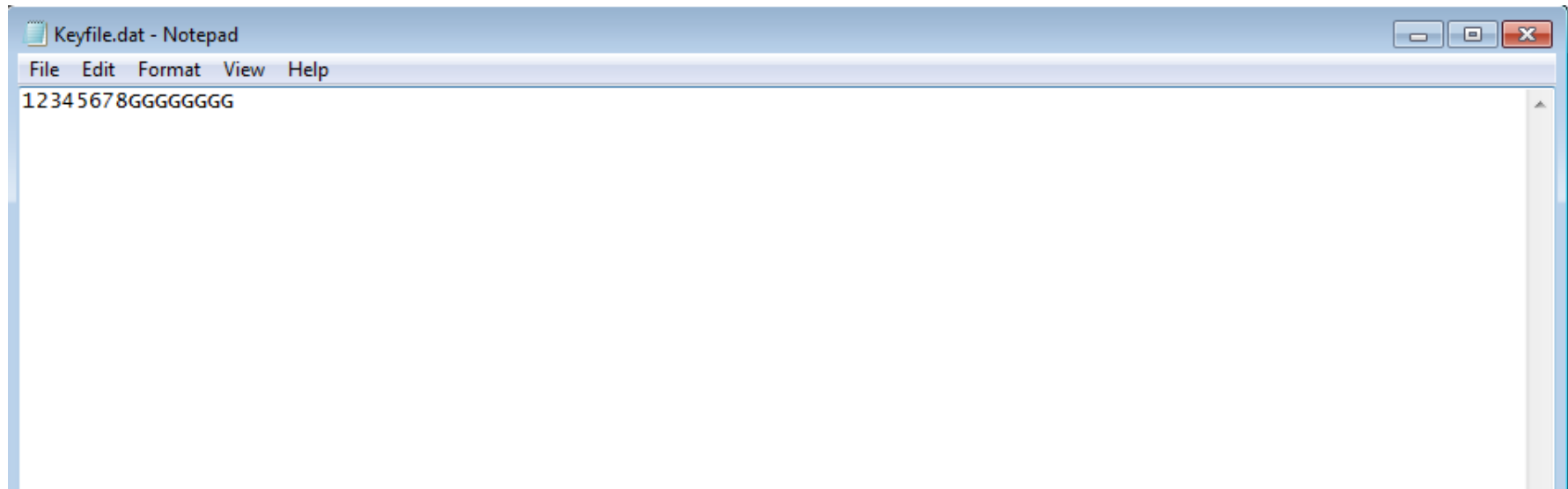
FST 0000 Cond 0 0 0 0 Err 0 0 0 0
FCW 027F Prec NEAR.53 Mask 1 1 1
  
```

004010D6에서 JUMP 수행하지 않도록 해야함 → ESI>=8 → G가 8개 이상 들어가야 함.



### 3. 시도 (3)의 솔루션 - Keyfile.dat에 G 8개 추가

16글자 이상에 G가 8번 이상 나와줘야함.



### 3. 시도 (4)

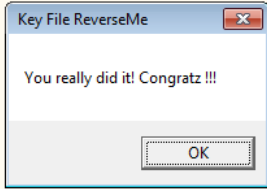
0040109A	> 6A 00	PUSH 0	<pre> pOverlapped = NULL pBytesRead = reverseM.00402173 BytesToRead = 46 &lt;70.&gt; Buffer = reverseM.0040211A hFile ReadFile </pre>	<b>Registers (FPU)</b> EAX 00000047 ECX 75D77D17 kernel32.75D77D17 EDX 776671B4 ntdll.KiFastSystemCallRet EBX 00000008 ESP 0012FF8C EBP 0012FF94 ESI 00000000 EDI 00000000 EIP 004010CD reverseM.004010CD C 0 ES 0023 32bit 0<FFFFFFFF> P 1 CS 001B 32bit 0<FFFFFFFF> A 0 SS 0023 32bit 0<FFFFFFFF> <b>Z 1</b> DS 0023 32bit 0<FFFFFFFF> S 0 FS 003B 32bit 7FFDF000<FFF> T 0 GS 0000 NULL D 0 O 0 LastErr ERROR_SUCCESS <00000000> EFL 00000246 <NO,NB,E,BE,NS,PE,GE,LE> ST0 empty 0.0 ST1 empty 0.0 ST2 empty 0.0 ST3 empty 0.0 ST4 empty 0.0 ST5 empty 0.0 ST6 empty 0.0 ST7 empty 0.0 FST 0000 Cond 0 0 0 0 Err 0 0 0 0 FCW 027F Prec NEAR.53 Mask 1 1 1
0040109C	. 68 732140	PUSH reverseM.00402173		
004010A1	. 6A 46	PUSH 46		
004010A3	. 68 1A2140	PUSH reverseM.0040211A		
004010A8	. 50	PUSH EAX		
004010A9	. E8 2F0200	CALL <JMP.&KERNEL32.ReadFile>		
004010AE	. 85C0	TEST EAX,EAX		
004010B0	. 75 02	JNZ SHORT reverseM.004010B4		
004010B2	. EB 43	JMP SHORT reverseM.004010F7		
004010B4	> 33DB	XOR EBX,EBX		
004010B6	. 33F6	XOR ESI,ESI		
004010B8	. 833D 7321	CMP DWORD PTR DS:[402173],10		
004010BF	. 7C 36	JL SHORT reverseM.004010F7		
004010C1	> 8A83 1A21	MOV AL,BYTE PTR DS:[EBX+40211A]		
004010C7	. 3C 00	CMP AL,0		
004010C9	. 74 08	JE SHORT reverseM.004010D3		
004010CB	. 3C 47	CMP AL,47		
004010CD	. 75 01	JNZ SHORT reverseM.004010D0		
004010CF	. 46	INC ESI ESI+1		
004010D0	> 43	INC EBX		
004010D1	. EB EE	JMP SHORT reverseM.004010C1		
004010D3	> 83FE 08	CMP ESI,8		
004010D6	. 7C 1F	JL SHORT reverseM.004010F7		
004010D8	. E9 280100	JMP reverseM.00401205		
004010DD	00	DB 00		
004010DE	. 00000000	DD 00000000		
004010E2	00	DB 00		
004010E3	00	DB 00		
004010E4	00	DB 00		

Jump is NOT taken  
004010D0=reverseM.004010D0

### 3. 시도 (4)

0040109A	> 6A 00	PUSH 0	<pre>pOverlapped = NULL pBytesRead = reverseM.00402173 BytesToRead = 46 (70.) Buffer = reverseM.0040211A hFile ReadFile</pre>	<b>Registers (FPU)</b> EAX 00000000 ECX 75D79D17 kernel32.75D79D17 EDX 776671B4 ntdll.KiFastSystemCallRet EBX 00000010 ESP 0012FF8C ESI 00000008 EDI 00000000 EIP 004010D6 reverseM.004010D6 C 0 ES 0023 32bit 0<FFFFFFFF> P 1 CS 001B 32bit 0<FFFFFFFF> A 0 SS 0023 32bit 0<FFFFFFFF> Z 1 DS 0023 32bit 0<FFFFFFFF> <b>S 0</b> FS 003B 32bit 7FFDF000<FFF> T 0 GS 0000 NULL D 0 O 0 LastErr ERROR_SUCCESS (00000000) EFL 00000246 <NO,NB,E,BE,NS,PE,GE,LE> ST0 empty 0.0 ST1 empty 0.0 ST2 empty 0.0 ST3 empty 0.0 ST4 empty 0.0 ST5 empty 0.0 ST6 empty 0.0 ST7 empty 0.0 3 2 1 0 ESPUO; FSI 0000 Cond 0 0 0 0 Err 0 0 0 0 FCW 027F Prec NEAR.53 Mask 1 1 1
0040109C	. 68 732140	PUSH reverseM.00402173		
004010A1	. 6A 46	PUSH 46		
004010A3	. 68 1A2140	PUSH reverseM.0040211A		
004010A8	. 50	PUSH EAX		
004010A9	. E8 2F0200	CALL <JMP.&KERNEL32.ReadFile>		
004010AE	. 85C0	TEST EAX,EAX		
004010B0	. 75 02	JNZ SHORT reverseM.004010B4		
004010B2	. EB 43	JMP SHORT reverseM.004010F7		
004010B4	> 33DB	XOR EBX,EBX		
004010B6	. 33F6	XOR ESI,ESI		
004010B8	. 833D 7321	CMP DWORD PTR DS:[402173],10		
004010BF	. 7C 36	JL SHORT reverseM.004010F7		
004010C1	> 8A83 1A21	MOV AL,BYTE PTR DS:[EBX+40211A]		
004010C7	. 3C 00	CMP AL,0		
004010C9	. 74 08	JE SHORT reverseM.004010D3		
004010CB	. 3C 47	CMP AL,47		
004010CD	. 75 01	JNZ SHORT reverseM.004010D0		
004010CF	. 46	INC ESI		
004010D0	> 43	INC EBX		
004010D1	. EB EE	JMP SHORT reverseM.004010C1		
004010D3	> 83FE 08	CMP ESI,8		
004010D6	. 7C 1F	JL SHORT reverseM.004010F7	ESI(8)=8 → S플래그=0, Z플래그=1	
004010D8	. E9 280100	JMP reverseM.00401205	조건과 관계없이 401205로 JUMP	
004010DD	. 00	DB 00		
004010DE	. 00000000	DD 00000000		
004010E2	. 00	DB 00		
004010E3	. 00	DB 00		
004010E4	. 00	DB 00		
Jump is NOT taken 004010F7=reverseM.004010F7				

00401205	> 6A 00	PUSH 0	<pre>Style = MB_OK MB_APPLMODAL Title = "Key File ReverseMe" Text = "You really did it! Congratz !!!" hOwner = NULL MessageBox ExitProcess</pre>
00401207	. 68 002040	PUSH reverseM.00402000	
0040120C	. 68 DE2040	PUSH reverseM.004020DE	
00401211	. 6A 00	PUSH 0	
00401213	. E8 4F0100	CALL <JMP.&USER32.MessageBoxA>	
00401218	. E8 9C0000	CALL <JMP.&KERNEL32.ExitProcess>	
0040121D	> C3	RETN	
0040121F	. 00	DB 00	
00401220	. 00	DB 00	
00401221	. 00	DB 00	
00401222	. 00	DB 00	
00401223	. 00	DB 00	
00401224	. 00	DB 00	
00401225	. 00	DB 00	
00401226	. 00	DB 00	
00401227	. 00	DB 00	
00401228	. 00	DB 00	
00401229	. 00	DB 00	
0040122A	. 00	DB 00	
0040122B	. 00	DB 00	
0040122C	. 00	DB 00	
0040122D	. 00	DB 00	
0040122E	. 00	DB 00	
0040122F	. 00	DB 00	
00401230	. 00	DB 00	



## 4. 결론 (성공하기 위한 조건)

- Keyfile.dat파일은 라이선스 파일이다.
- Keyfile.dat 파일이 존재해야 함.
- 글자가 16글자 이상이어야 함.
- 글자에 대문자 G가 8번 이상 나와야 함.

감사합니다