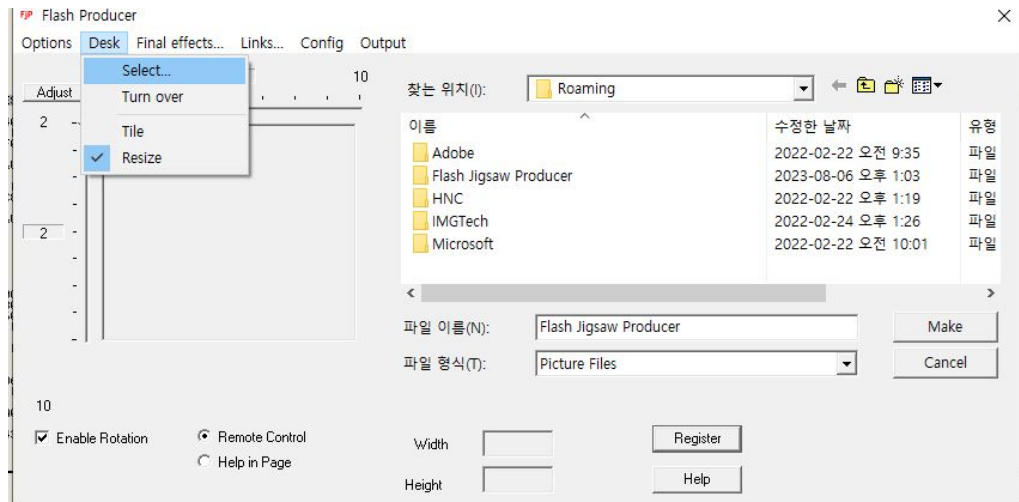


이번 튜토리얼의 목적은 **pane windows**를 이용한 중급 패치이다.

이전 프로그램과 다르지 않게 **register**하는건데, 레지스터 키를 직접 찾아 등록하지 않고, **register**하는게 이번 튜토리얼의 목적이다.

이 그림은 **register** 실패 시 보이는 오류이다.



desk -> select 버튼을 누르면
등록한 사람만 사용할 수 있다고,
나온다. 이번 패치를 통해 이
기능을 사용할 수 있게 하는것이
목적이다.

프로그램 실행시 윈도우 바에 원래 **unregister**라는 문자열이 있는데, 이 프로그램에서는 보이지 않아 있다고 가정하에 분석을 진행한다.

이전 분석과 다를바 없이 문자열 검색으로 **unregistered** 문자열을 호출하는 영역으로 이동한다.

그럼 다음과 같이 두개의 **SetWindowsText** API가 나온다.

bp가 걸린 분기문을 통해 보여지는 문자열이 다르다.

위에 보면 **AL** 레지스터에 **ESP+4** 주소에 있는 값을 집어넣는데, 이 부분의 값이 어디서 결정되는지 위에서 살펴보도록 한다.

004046E0	8A4424 04	MOV AL,BYTE PTR SS:[ESP+4]	
004046E4	84C0	TEST AL,AL	
004046E6	74 12	JE SHORT f.jproduct.004046FA	
004046E8	A1 440A4300	MOV EAX,DWORD PTR DS:[430A44]	
004046ED	68 F09C4200	PUSH f.jproduct.00429CF0	
004046F2	50	PUSH EAX	
004046F3	FF15 5C924200	CALL DWORD PTR DS:[&USER32.SetWindowTextA]	Text = "Flash Jigsaw Producer"
004046F9	C3	RETN	hWnd => NULL
004046FA	8B0D 440A4300	MOV ECX,DWORD PTR DS:[430A44]	
00404700	68 C89C4200	PUSH f.jproduct.00429CC8	
00404705	51	PUSH ECX	
00404706	FF15 5C924200	CALL DWORD PTR DS:[&USER32.SetWindowTextA]	Text = "Flash Jigsaw Producer (unregistered)"
0040470C	C3	RETN	hWnd => NULL
0040470D	90	NOP	SetWindowTextA
0040470E	90	NOP	
0040470F	90	NOP	

004046E0	\$ 8A4424 04	MOV AL,BYTE PTR SS:[ESP+4]	
004046E4	. 84C0	TEST AL,AL	
004046E8	^ 74 12	JE SHORT fjprouduc.004046FA	
004046ED	. A1 440A4300	MOV EAX,DWORD PTR DS:[430A44]	
004046F2	. 68 F09C4200	PUSH fjprouduc.00429CF0	Text hWnd
004046F3	. 50	PUSH EAX	Set!
004046F9	. FF15 5C924200	CALL DWORD PTR DS:[&USER32.SetWindowTe	
004046FA	. C3	RETN	
004046FA	> 8B0D 440A4300	MOV ECX,DWORD PTR DS:[430A44]	
00404700	. 68 C89C4200	PUSH fjprouduc.00429CC8	Text hWnd
00404705	. 51	PUSH ECX	Set!
00404706	. FF15 5C924200	CALL DWORD PTR DS:[&USER32.SetWindowTe	
0040470C	. C3	RETN	
0040470D	. 90	NOP	
0040470E	. 90	NOP	
0040470F	. 90	NOP	
00404710	\$ 83EC 28	SUB ESP,28	
00404713	. 56	PUSH ESI	
00404714	. 68 0C034300	PUSH fjprouduc.0043030C	
00404719	. 68 309D4200	PUSH fjprouduc.00429D30	ASC
0040471E	. E8 3DFDFFFF	CALL fjprouduc.00404460	
00404723	. 8B0D 88E14200	MOV ECX,DWORD PTR DS:[42E188]	
00404729	. 6A 0A	PUSH 0A	
0040472B	. 8D4424 24	LEA EAX,DWORD PTR SS:[ESP+24]	
0040472F	. 50	PUSH EAX	
00404730	. 51	PUSH ECX	
00404731	. E8 A0390200	CALL fjprouduc.004280D6	
00404736	. 8D5424 2C	LEA EDX,DWORD PTR SS:[ESP+2C]	
0040473A	. 52	PUSH EDX	
Local calls from 004047D3, 00404880			

위 MOV AL... 부분을 클릭해 보면
어디서 호출하는지 알 수 있다. 이
부분이 바로 pane windows다.

해당 주소로 이동하여 bp를 건 후 다시
실행한다.

004047C0	. 68 14704200	PUSH fjdproduc.00427014	
004047C2	. FF15 68904200	CALL DWORD PTR DS:[<&KERNEL32.GetPrivate	GetPrivate
004047C8	. 68 34034300	PUSH fjproduc.00430334	ASC]
004047D0	. E8 6EFEFFFF	CALL fjproduc.00404640	
004047D2	. 50	PUSH EAX	
004047D8	. E8 08FFFFFF	CALL fjproduc.004046E0	
004047DB	. 83C4 08	ADD ESP,8	
004047DB	. 5E	POP ESI	
004047DC	. 83C4 28	ADD ESP,28	
004047DF	. C3	RETN	
004047E0	. 8B4424 08	MOV EAX,DWORD PTR SS:[ESP+8]	

그럼 해당 주소에서 호출하는 걸 볼 수 있다. 그 전에 AL에 값을 집어넣는 **PUSH EAX** 함수가 보인다. 아마 그전 함수에 의해 AL 값이 결정되는걸로 보인다. **4047CD** 주소에 bp를 걸고 함수를 분석해본다.

안에 보면 내가 입력한 레지스터 값과 글자수를 비교한다. 여기 보니 총 4글자를 입력하라고 나온다.

00404640	. 8049 00	LEA ECX,DWORD PTR DS:[ECX]	
00404650	> 8A08	MOV CL, BYTE PTR DS:[EAX]	
00404652	. 40	INC EAX	
00404653	. 84C9	TEST CL, CL	
00404655	. ^75 F9	JNZ SHORT fjproduc.00404650	
00404657	. 2BC2	SUB EAX, EDX	
00404659	. 83F8 04	CMP EAX, 4	
0040465C	. ^73 07	JNB SHORT fjproduc.00404665	
0040465E	> 32C0	XOR AL, AL	
00404660	. 5E	POP ESI	
00404661	. 83C4 2C	ADD ESP, 2C	
00404664	. C3	RETN	
00404665	> 6A 03	PUSH 3	
00404667	. 68 C49C4200	PUSH fjproduc.00429CC4	AS
0040466C	. 56	PUSH ESI	
0040466D	. E8 6EAA0100	CALL fjproduc.0041F0E0	
00404672	. 83C4 0C	ADD ESP, 0C	
00404675	. 5E	POP ESI	

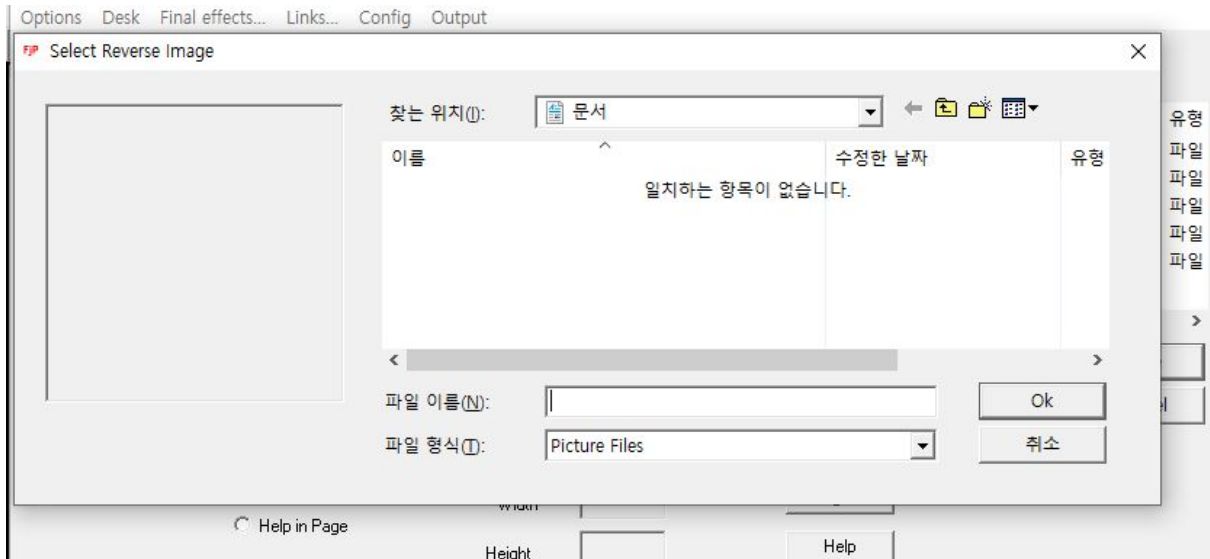
그리고 해당 주소에서 **XOR AL, AL**을 통해 값을 결정하는걸 알 수 있다.

이 명령을 **MOV AL, 1**로 바꿔 AL 값이 1이 되게 한다.

그럼 AL값이 1로 바뀐걸 볼 수 있다.

Registers (FPU)	
EAX	FFFFFF01
ECX	FFFFFFFF
EDX	00430335 ASCII "
EBX	00000001
ESP	001908C4
EBP	0019097C
ESI	00430334 ASCII "
EDI	00000110
EIP	00404660 fjprodu

그러면 아까 **TEST AL, AL**값이 1이 되고, **JE**도 성립되지 않아 **unregistered**문자열이 호출 안되는걸 볼 수 있다.



그리고 실행해보면 아까 못썸던
select기능이 잘 실행되고, 등록할 때
나왔던 오류 메시지도 안뜨게 된다.