

How to make nice search & replace patches...

Tools i used for this tutorial

- Ollydbg www.ollydbg.de
- dUP 2 www.diablo2oo2.cjb.net

Why search & replace?

The answer is simple: while usual offset patcher in most cases work only for a certain version of the target, a search & replace patch can work for all future versions if it's made good.

Some rules...

There are just some rules you need to follow to make good s&r patch. You also need good assembler and coding knowledge.

- the searchpattern is searching for a unique command sequence
- always use wildcards (??) for
 - **CALL's**
ASM Code : CALL ?? ?? ?? ??
Searchpattern : E8 ?? ?? ?? ??
 - **Memory Addresses**
ASM Code : CMP BYTE PTR [?? ?? ?? ??],1
Searchpattern : 80 3D ?? ?? ?? ?? 01
 - **Long Jumps**
ASM Code : JE LONG ?? ?? ?? ??
Searchpattern : OF 84 ?? ?? ?? ??

This are the most important examples for wildcard usage!

An Example...

0040B250	\$ 60	PUSHAD	Protect = PAGE_READWRITE AllocationType = MEM_COMMIT Size = 5555 (21845.) Address = NULL VirtualAlloc
0040B251	. 6A 04	PUSH 4	
0040B253	. 68 00100000	PUSH 1000	
0040B258	. 68 55550000	PUSH 5555	
0040B25D	. 6A 00	PUSH 0	
0040B25F	. E8 50340000	CALL <JMP.&kernel32.VirtualAlloc>	
0040B264	. BBF0	MOV ESI,EAX	
0040B266	. 6A 02	PUSH 2	
0040B268	. 6A FF	PUSH -1	
0040B26A	. 68 0C100000	PUSH 100C	
0040B26F	FF35 98AA4200	PUSH DWORD PTR DS:[42AA98]	
0040B275	E8 14330000	CALL <JMP.&user32.SendMessageA>	
0040B279	. 83F8 FF	CMP EAX,-1	
0040B27D	✓>74 43 JE SHORT 0040B2C2		we patch this to 74 00 (never jump)
0040B27F	. A3 18B24200	MOV DWORD PTR DS:[42B218],EAX	
0040B284	C705 14B24200 01000000	MOV DWORD PTR DS:[42B214],1	
0040B28E	C705 2C824200 55550000	MOV DWORD PTR DS:[42B22C],5555	
0040B298	C705 1CB24200 00000000	MOV DWORD PTR DS:[42B21C],0	
0040B2A2	. 56	PUSH ESI	
0040B2A3	. 8F05 28B24200	POP DWORD PTR DS:[42B228]	
0040B2A9	. 68 14B24200	PUSH 0042B214	
0040B2AE	. 6A 00	PUSH 0	
0040B2B0	. 68 05100000	PUSH 1005	
0040B2B5	FF35 98AA4200	PUSH DWORD PTR DS:[42AA98]	
0040B2BB	E8 CE320000	CALL <JMP.&user32.SendMessageA>	
0040B2C0	. EB 13	JMP SHORT 0040B2D5	
0040B2C2	> 68 55550000	PUSH 5555	Count = 5555 (21845.)

The bytes for the wildcards are marked red!

You want to patch the jump @0040B27D so that it never jumps

0040B27D	. /74 43	JE SHORT 0040B2C2
↓		
0040B27D	. /74 00	JE SHORT 0040B27F

We will search for this code sequence:

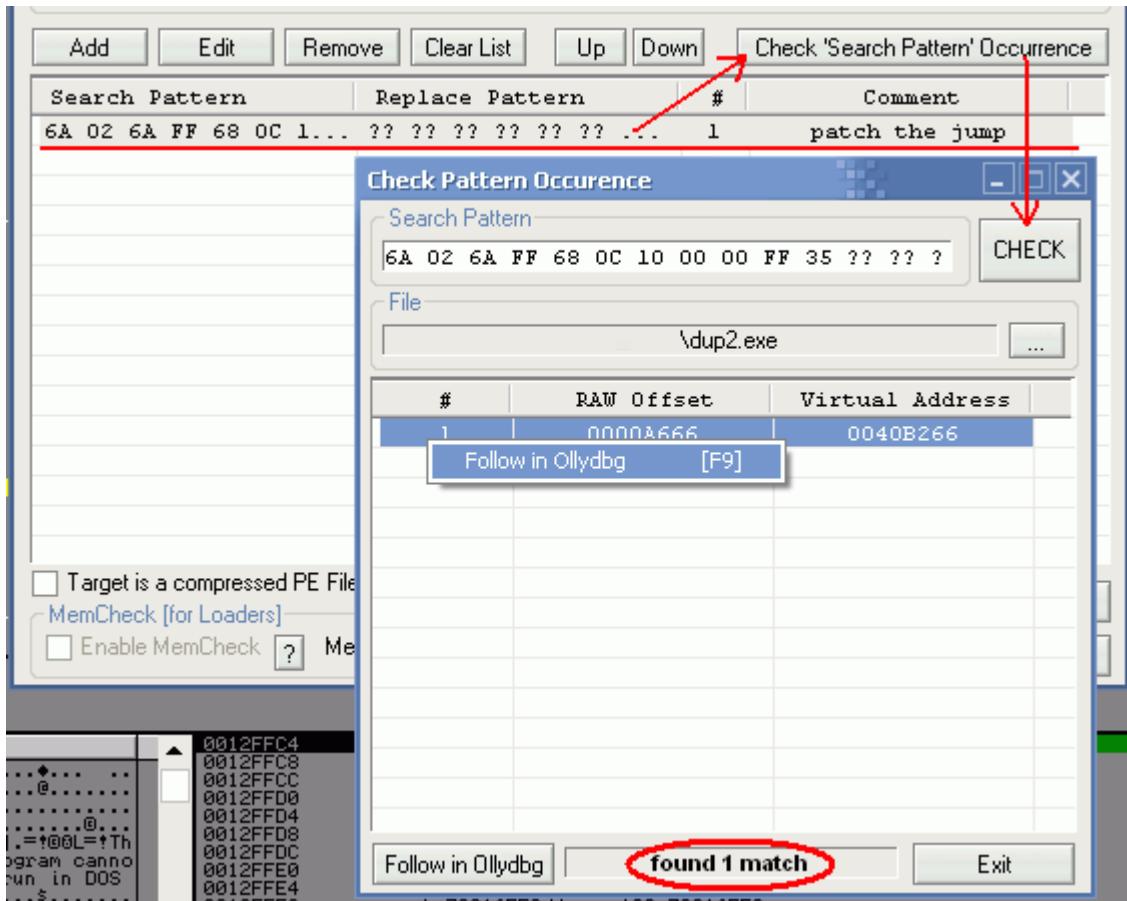
```
PUSH 2
PUSH -1
PUSH 100C
PUSH DWORD PTR DS:[ ?? ?? ?? ?? ]
CALL <?? ?? ?? ?? >
CMP EAX,-1
JE SHORT ???
MOV DWORD PTR DS:[ ?? ?? ?? ?? ?? ],EAX
MOV DWORD PTR DS:[ ?? ?? ?? ?? ?? ],1
MOV DWORD PTR DS:[ ?? ?? ?? ?? ?? ],55555
MOV DWORD PTR DS:[ ?? ?? ?? ?? ?? ],0
```

The searchpattern would look like this:

6A 02 6A FF 68 0C 10 00 00 FF 35 ?? ?? ?? ?? ?? E8 ?? ?? ?? ?? ?? 83 F8 FF 74 ?? A3 ?? ?? ??
?? ?? C7 05 ?? ?? ?? ?? 01 00 00 00 C7 05 ?? ?? ?? ?? ?? 55 55 00 00 C7 05 ?? ?? ?? ?? ??
00 00 00 00

And the replacepattern:

Now start dUP 2 and enter the search- and replacepattern. Then use the function "Check occurrence":



dUP 2 will check how often it can find our searchpattern. The result should be **one match** only! After the check you can use the function "Follow in Ollydbg" to be sure that it's the right pattern.

What to do if there is more than one match?

- Use a longer pattern
- Use less wildcards (but carefully!)

I hope it wasn't too hard to understand. ;D