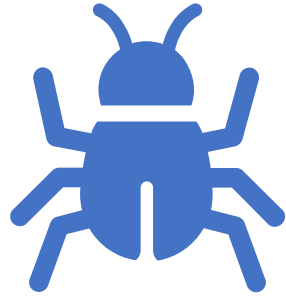




Hello World 문자열 패치

이지훈

목차

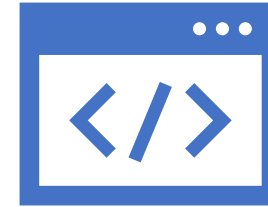


Olly Debugger

Last Week

Base Camp

Code Find



Hello World!

HelloWorld Patch

Next week

Last Week

```
1  #include "windows.h"
2  #include "tchar.h"
3
4  int _tmain(int argc, TCHAR* argv[])
5  {
6      MessageBox(NULL,
7                  L"Hello World!",
8                  L"www.reversecore.com",
9                  MB_OK);
10
11     return 0;
12 }
```



Last Week

1. CALL 00401000 명령어를 따라가니
 2. MessageBox API를 호출하는 코드를 찾았다
 3. 문자열을 비교해보면 일치하는 것을 확인
- 따라서 00401000 함수가 _tmain 함수이다.

00401138	:	FF35 80AF4000	PUSH DWORD PTR DS:[40AF80]	
0040113E	:	FF35 78AF4000	PUSH DWORD PTR DS:[40AF78]	
00401144	:	E8 B7FEFFFF	CALL HelloWor.00401000	
00401149	:	83C4 0C	ADD ESP,0C	
0040114C	:	8945 E0	MOV DWORD PTR SS:[EBP-20],EAX	
0040114F	:	837D F4	COMPP DWORD PTR SS:[EBP-1C],0	

00401000	:	6A 00	PUSH 0	[Style = MB_OK;MB_APPLMODAL
00401002	:	68 78924000	PUSH HelloWor.004092	Title = "www.reversecore.com"
00401007	:	68 A0924000	PUSH HelloWor.004092	Text = "Hello World!"
0040100C	:	6A 00	PUSH 0	hOwner = NULL
0040100E	:	FF15 E4804000	CALL DWORD PTR DS:[4080E4]	MessageBoxW
00401014	:	33C0	XOR EAX,EAX	
00401016	:	C3	RETN	

```
1  #include "windows.h"
2  #include "tchar.h"
3
4  int _tmain(int argc, TCHAR* argv[])
5  {
6      MessageBox(NULL,
7          L"Hello World!",
8          L"www.reversecore.com",
9          MB_OK);
10
11     return 0;
12 }
```





Base Camp

디버거를 재실행할 때마다 Entry Point 부터 시작하는 불편함이 있다.

큰 파일을 분석할 때는 중요한 부분을 쉽게 찾기 위해
캠프를 지정하는 것이 디버깅을 편리하게 만들어 준다.

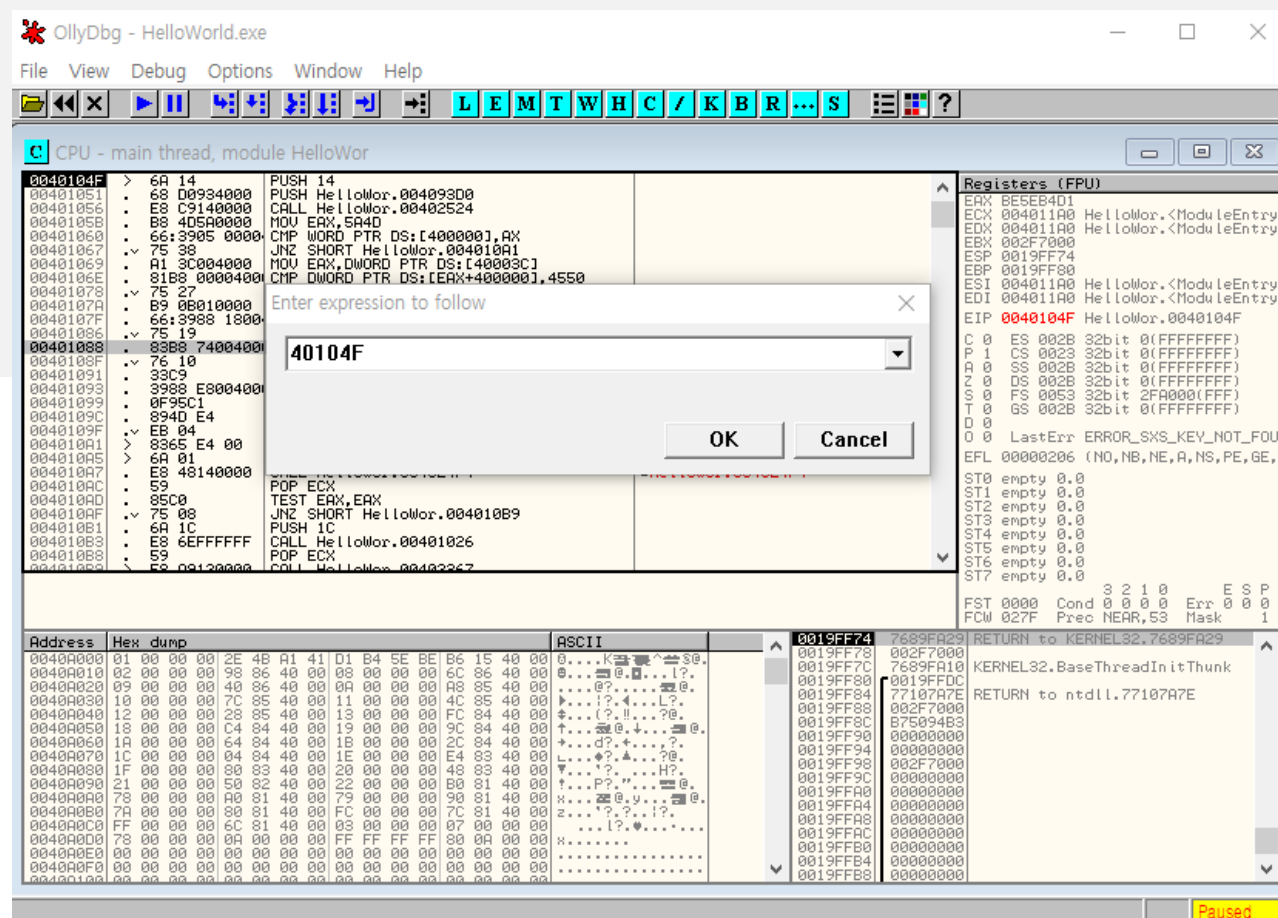


Base Camp

(1) Go to

Ctrl + G : 직접 베이스 캠프의 주소를 입력하여
Go to 명령으로 주소로 이동

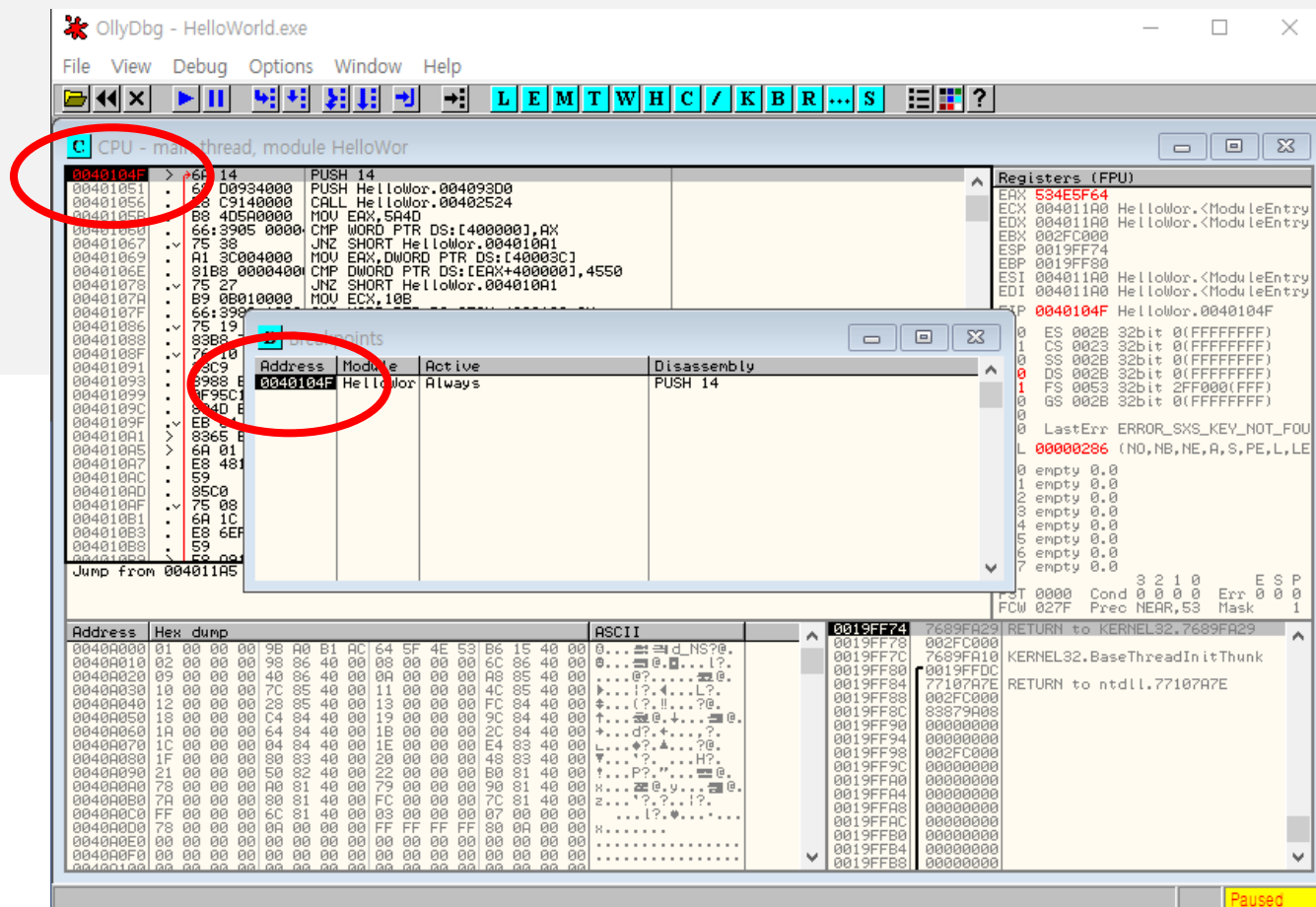
F4 : Execute till cursor 명령으로 커서까지
실행



Base Camp (2) BP

F2 : Break Point(BP) 를 설치하여 디버거가 실행 도중(F9) BP에서 멈추게 한다.

ALT + B : BP 목록을 확인할 수 있음
: BP를 더블클릭 -> 해당 주소로 이동

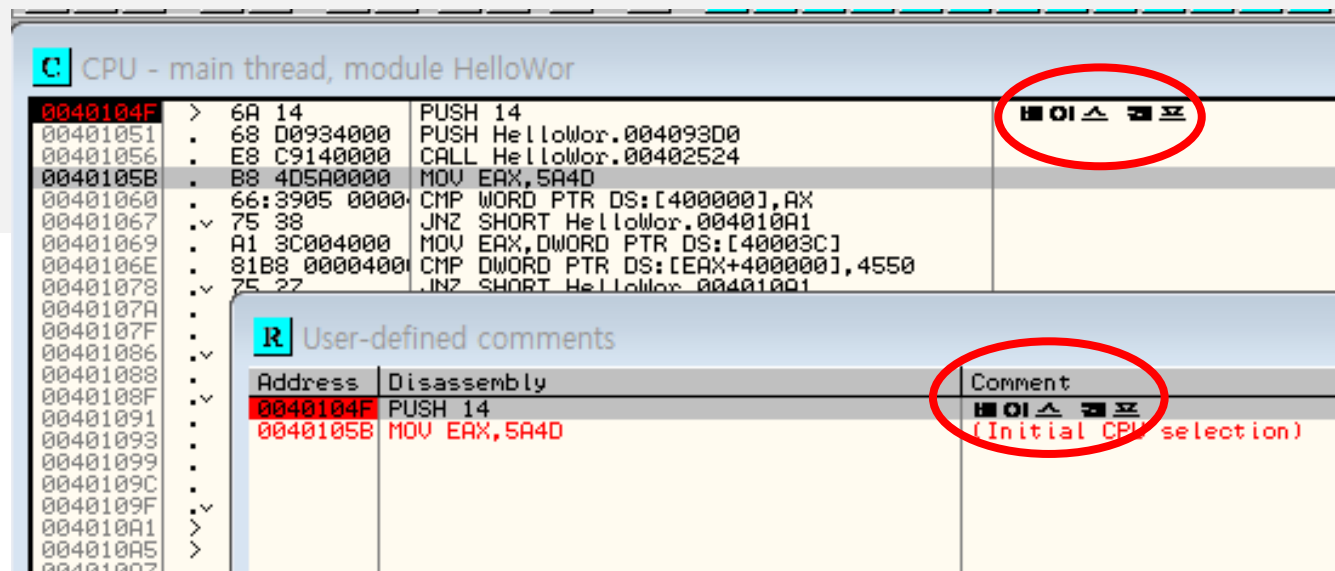


Base Camp

(3) Comment

[:] : 특정 주소에 주석을 달아 확인하는 방법
: 주석 더블 클릭 -> 해당 주소로 이동

커서 위치를 다른 곳으로 -> 오른쪽 클릭 후 ->
Search for -> User-defined comment



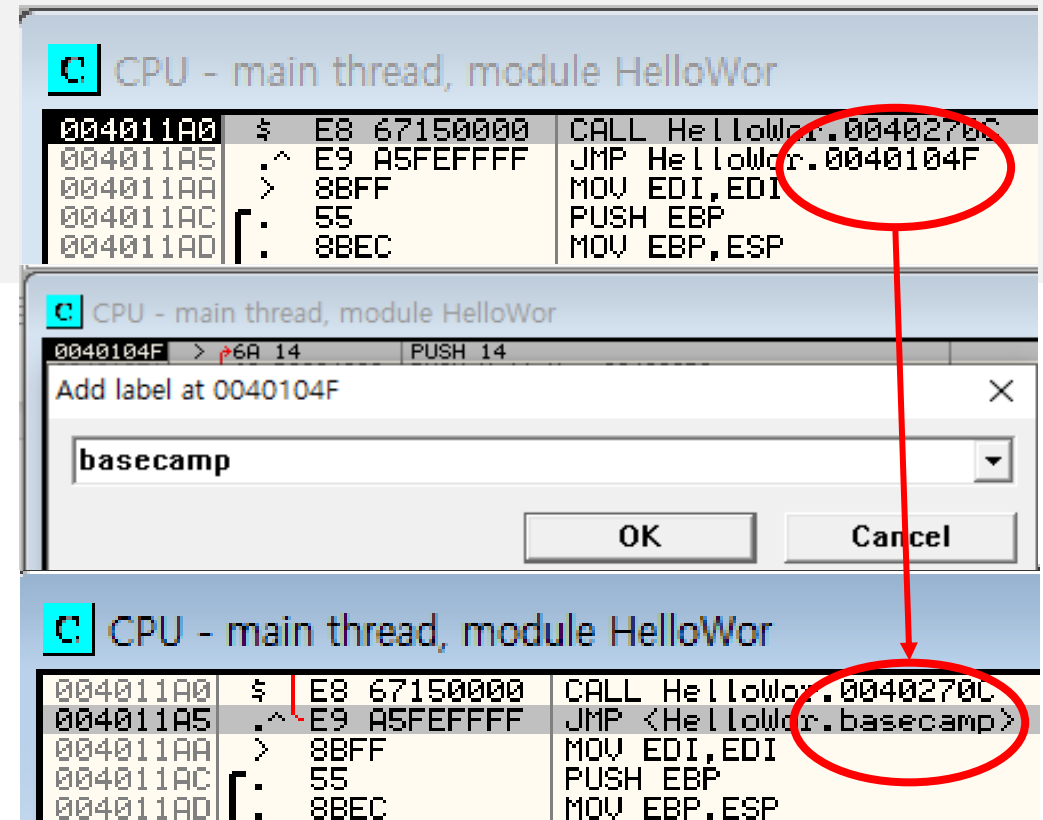
Base Camp

(4) Label

[:] : 원하는 주소에 특정 이름을 붙이는 방법

마우스 오른쪽 클릭 -> Search for -> User defined labels : label 확인 가능

: 더블클릭 -> 해당 주소로 이동





Code Find

1. 코드 실행 방법 : 코드의 크기가 작고 기능이 명확한 경우
2. 문자열 검색 방법 : 프로그램 내 특정 문자열을 사용한 부분을 확인할 경우
3. API 호출 코드에 BP : 코드에서 사용된 API를 예상할 수 있을 경우
4. API 코드에 직접 BP : Packer 와 Protector 등 실행 파일이 압축 또는 보호되어 DLL 코드에 직접 BP를 걸어야 할 경우



Hello World

문자열 검색

문자열 검색 방법 사용

오른쪽 클릭 -> Search for ->

All referenced text strings

_tmain 함수 시작 주소(00401000)에
BP 설정

R Text strings referenced in HelloWor.text		
Address	Disassembly	Text string
00401000	PUSH 0	(Initial CPU selection)
00401002	PUSH HelloWor.00409278	UNICODE "www.reversecore.com"
00401007	PUSH HelloWor.004092A0	UNICODE "Hello World!"
0040135E	PUSH HelloWor.00408154	UNICODE "mscoree.dll"
0040136D	PUSH HelloWor.00408144	ASCII "CoreInitProcess"
0040169D	PUSH HelloWor.00408710	ASCII "Runtime Error! Program: "
004016E5	PUSH HelloWor.004086F8	ASCII "<program name unknown>"
0040172A	PUSH HelloWor.004086F4	ASCII "..."
00401752	PUSH HelloWor.004086F0	ASCII "00"
0040179C	PUSH HelloWor.004086C8	ASCII "Microsoft Visual C++ Runtime"
004018BB	MOV ESI,HelloWor.0040B2C0	UNICODE "C:\Users\11211\Downloads\
00401F90	MOV ESI,HelloWor.004087B8	UNICODE "KERNEL32.DLL"
00401FAB	PUSH HelloWor.004087A8	ASCII "EncodePointer"
0040200B	MOV ESI,HelloWor.004087B8	UNICODE "KERNEL32.DLL"
00402026	PUSH HelloWor.004087D4	ASCII "DecodePointer"
004020CA	MOV ESI,HelloWor.004087B8	UNICODE "KERNEL32.DLL"
004020F8	PUSH HelloWor.004087A8	ASCII "EncodePointer"
0040210C	PUSH HelloWor.004087D4	ASCII "DecodePointer"
0040236B	MOV ESI,HelloWor.004087B8	UNICODE "KERNEL32.DLL"

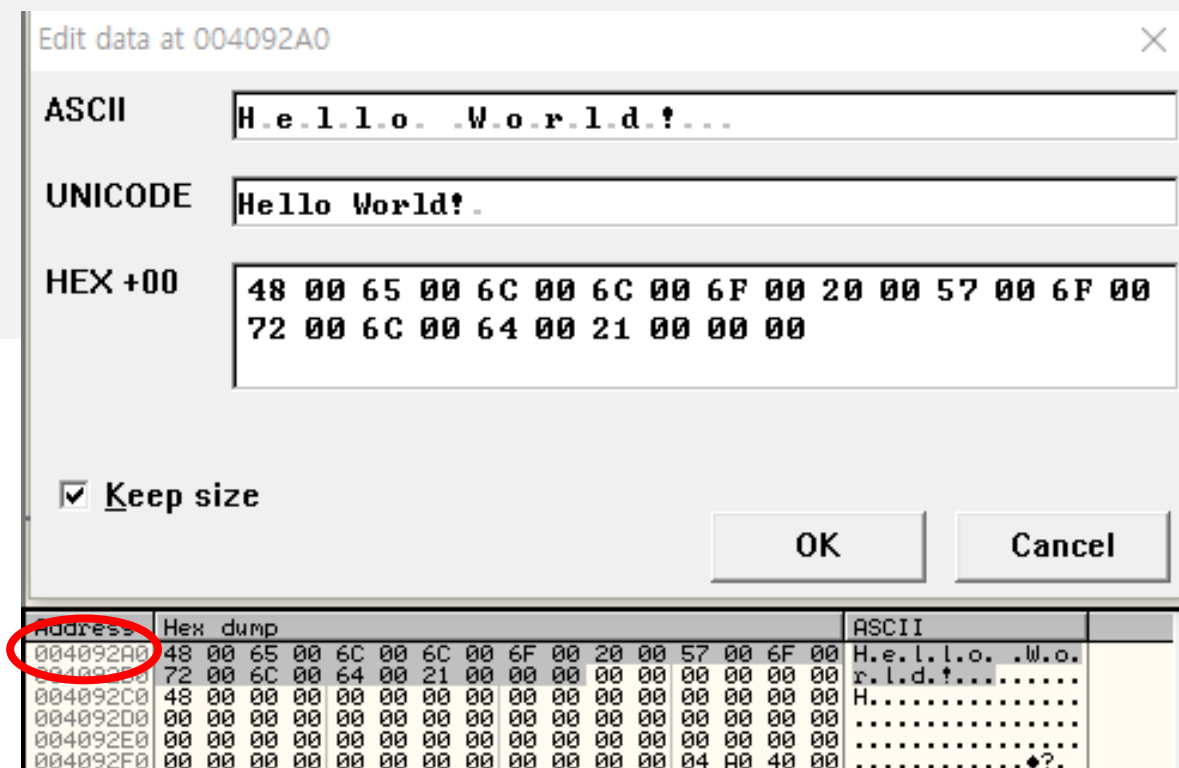


Hello World

문자열 검색

덤프 창에서 MessageBoxW 함수의 전달인자 주소
(004092A0)로 Go to 명령 사용

Ctrl + E : Edit dialog



Hello World

문자열 변경

유니코드 문자열은 2바이트 크기의 NULL 로 끝난다.

일반적으로 원본 문자열 뒤쪽에
의미 있는 데이터가 존재할 수 있기 때문에
항상 주의하여 문자열 변경을 해야 한다.

Edit data at 004092A0

ASCII I . L . o . v . e . S . C . P . . ? . . .

UNICODE I Love SCP ? .

HEX +1A 49 00 20 00 4C 00 6F 00 76 00 65 00 20 00 53 00
43 00 50 00 20 00 21 00 00 00

☒ Keep size


OK Cancel

Address	Hex dump	ASCII
004092A0	48 00 65 00 6C 00 6F 00 20 00 57 00 6F 00	H.e.l.l.o . .w.o.
004092B0	72 00 6C 00 64 00 21 00 00 00 00 00 00 00	r.l.d.!.
004092C0	48 00 00 00 00 00 00 00 00 00 00 00 00 00	H.
004092D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00
004092E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00
004092F0	00 00 00 00 00 00 00 00 00 00 00 00 04 A0 40 00



Hello World

문자열 패치



The screenshot shows a debugger window titled "CPU - main thread, module HelloWor". The assembly list on the left shows instructions from address 00401000 to 00401051. The instruction at 00401000 is highlighted in red. The instruction at 00401007 is "PUSH 0". The instruction at 0040100C is "PUSH 0". The instruction at 0040100E is "CALL DWORD PTR DS:[<&USER32.MessageBoxW". The instruction at 00401014 is "XOR EAX,EAX". The instruction at 00401016 is "RETN". The instruction at 00401017 is "CMP ECX,DWORD PTR DS:[40A004]". The instruction at 0040101D is "JNZ SHORT HelloWor.00401021". The instruction at 0040101F is "PREFIX REP:". The instruction at 00401020 is "RETN". The instruction at 00401021 is "JMP HelloWor.004011AA". The instruction at 00401026 is "MOV EDI,EDI". The instruction at 00401028 is "PUSH EBP". The instruction at 00401029 is "MOV EBP,ESP". The instruction at 0040102B is "CMP DWORD PTR DS:[40AC48],2". The instruction at 00401032 is "JE SHORT HelloWor.00401039". The instruction at 00401034 is "CALL HelloWor.004017E3". The instruction at 00401039 is "PUSH DWORD PTR SS:[EBP+8]". The instruction at 0040103C is "CALL HelloWor.00401638". The instruction at 00401041 is "PUSH 0FF". The instruction at 00401046 is "CALL HelloWor.00401384". The instruction at 00401048 is "POP ECX". The instruction at 0040104C is "POP ECX". The instruction at 0040104D is "POP EBP". The instruction at 0040104E is "RETN". The instruction at 0040104F is "PUSH 14". The instruction at 00401051 is "PUSH HelloWor.004093D0".

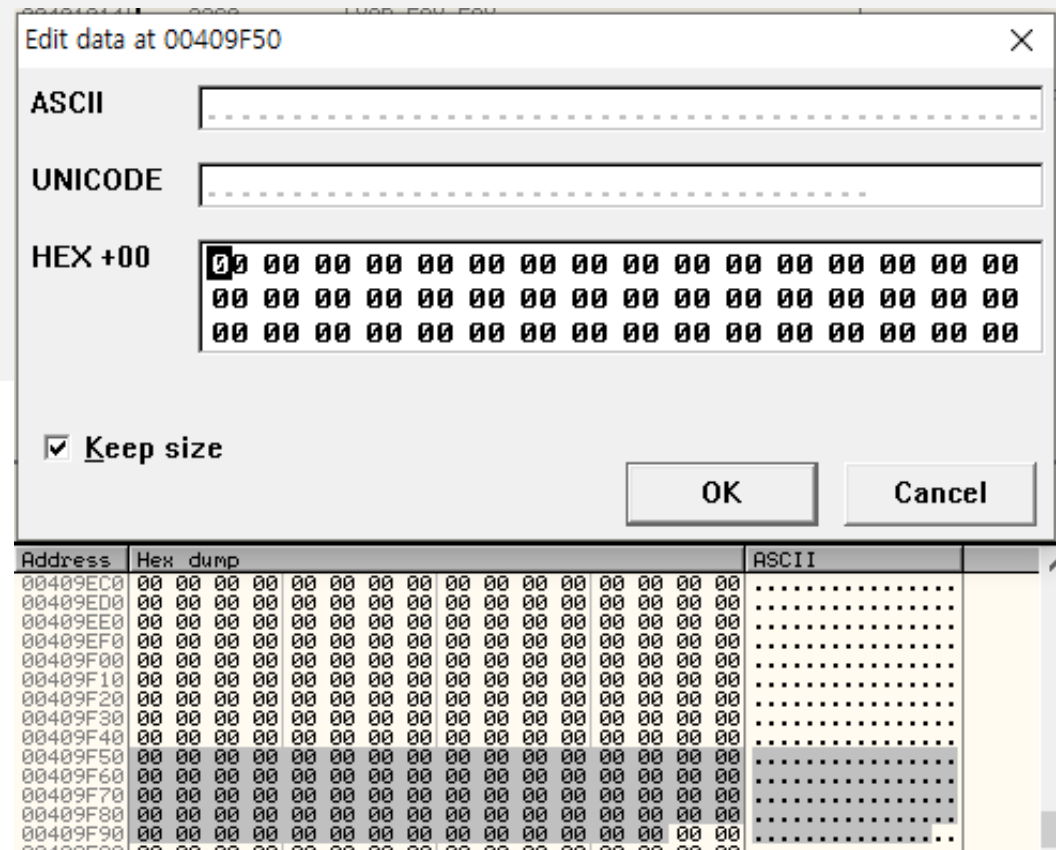
On the right, a "MessageBoxW" dialog box is shown. The title bar is "www.reversecore.com". The text is "I Love SCP!". The owner is NULL. The style is MB_OK|MB_APPLMODAL. The button is "확인".



Hello World

문자열 변경 (2)

원본보다 더 긴 문자열로 패치하기 위해선
다른 메모리에 패치하고자 하는 문자열을 적고
MessageBoxW() 함수에게 해당 주소를 넘겨줘야한다.



Hello World

문자열 변경 (2)

Edit data at 00409F50

ASCII: 2.0.2.1. 0.7.2.7. S.C.P. Presentation

UNICODE: 2021 0727 SCP Presentation.....

HEX +2F: 32 00 30 00 32 00 31 00 20 00 30 00 37 00 32 00
37 00 20 00 53 00 43 00 50 00 20 00 50 00 72 00
65 00 73 00 65 00 6E 00 74 00 61 00 74 00 69 00

☒ Keep size

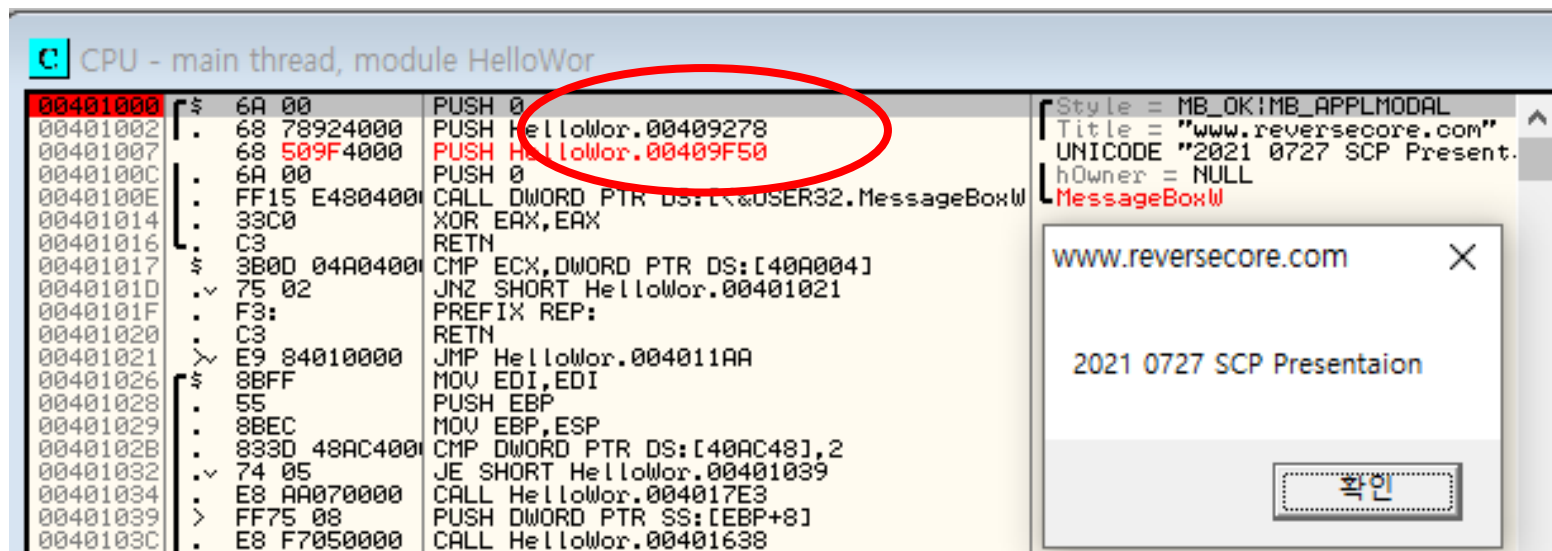
OK Cancel

Address	Hex dump	ASCII
00409EC0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409ED0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409EE0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409EF0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F30	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F70	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409F90	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00409FA0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00



Hello World

문자열 패치 (2)



The screenshot shows a debugger window titled "CPU - main thread, module HelloWor". The assembly code is displayed in a list with addresses, hex values, and instructions. A red circle highlights the instruction at address 00401007: `PUSH HelloWor.00409278`. The instruction at address 00401008 is `PUSH HelloWor.00409F50`. The instruction at address 0040100E is `CALL DWORD PTR DS:[40A00000]`. The instruction at address 00401017 is `CMP ECX, DWORD PTR DS:[40A0041]`. The instruction at address 0040101D is `JNZ SHORT HelloWor.00401021`. The instruction at address 00401021 is `JMP HelloWor.004011AA`. The instruction at address 00401026 is `MOV EDI, EDI`. The instruction at address 00401028 is `PUSH EBP`. The instruction at address 00401029 is `MOV EBP, ESP`. The instruction at address 0040102B is `CMP DWORD PTR DS:[40AC48], 2`. The instruction at address 00401032 is `JE SHORT HelloWor.00401039`. The instruction at address 00401034 is `CALL HelloWor.004017E3`. The instruction at address 00401039 is `PUSH DWORD PTR SS:[EBP+8]`. The instruction at address 0040103C is `CALL HelloWor.00401638`. On the right side of the debugger window, a MessageBox dialog is open with the title "www.reversecore.com" and the text "2021 0727 SCP Presentaion". The dialog has a "확인" (OK) button.

Address	Hex	Instruction
00401000	6A 00	PUSH 0
00401002	68 78924000	PUSH HelloWor.00409278
00401007	68 509F4000	PUSH HelloWor.00409F50
0040100C	6A 00	PUSH 0
0040100E	FF15 E4804000	CALL DWORD PTR DS:[40A00000]
00401014	33C0	XOR EAX, EAX
00401016	C3	RETN
00401017	3B0D 04A04000	CMP ECX, DWORD PTR DS:[40A0041]
0040101D	75 02	JNZ SHORT HelloWor.00401021
0040101F	F3:	PREFIX REP:
00401020	C3	RETN
00401021	E9 84010000	JMP HelloWor.004011AA
00401026	8BFF	MOV EDI, EDI
00401028	55	PUSH EBP
00401029	8BEC	MOV EBP, ESP
0040102B	833D 48AC4000	CMP DWORD PTR DS:[40AC48], 2
00401032	74 05	JE SHORT HelloWor.00401039
00401034	E8 AA070000	CALL HelloWor.004017E3
00401039	FF75 08	PUSH DWORD PTR SS:[EBP+8]
0040103C	E8 F7050000	CALL HelloWor.00401638

Style = MB_OK!MB_APPLMODAL
Title = "www.reversecore.com"
UNICODE "2021 0727 SCP Presentaion"
hOwner = NULL
MessageBoxW

www.reversecore.com X

2021 0727 SCP Presentaion

확인





Q & A