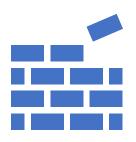


리버싱 이론 : 스택 프레임

이지훈

목차



Stack Frame

Last Week

ESP & EBP

Stack Frame



Stack Frame.exe

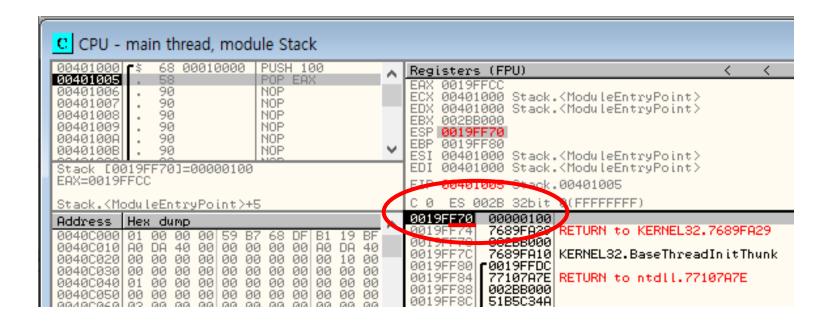
Stack Frame.cpp

main()

aqq()

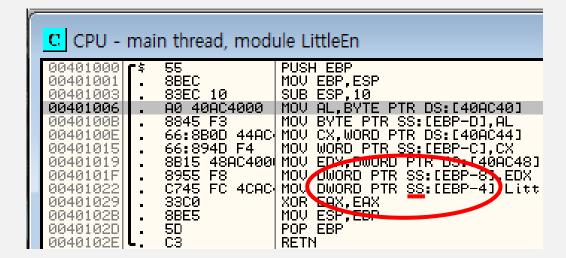


Last Week





ESP & EBP



ESP : Extended Stack Pointer, 스택의 최상단 주소값을 가리키는 레지스터

EBP: Extended Base Pointer, 스택 프레임의 베이스 주소를 가리키는 레지스터

SS: Stack Segment, 해당 메모리가 어떤 세그먼트에 소속되어 있는지 표기

(DS: Data Segment, ES: Extra data Segment)



Stack Frame

Stack Frame 기법 : ESP가 아닌 EBP 레지스터를 사용하여 스택 내의 로컬 변수, 파라미터, 복귀 주소에 접근하는 기법.

ESP의 경우 프로그램 안에서 수시로 변경되기에 CPU가 정확한 위치를 참고하기 어려움 -> 기준점, 즉 EBP가 필요하다!



Stack Frame.cpp

```
#include "stdio.h"
 4 long add(long a, long b)
      long x = a, y = b;
    return (x + y);
 8
    int main(int argc, char* argv[])
11
12
     long a = 1, b = 2;
13
14
       printf("%d\n", add(a, b));
15
16
      return 0;
17 }
```



Stack Frame, exe

```
PUSH EBP
MOV EBP,ESP
SUB ESP,8
                  00401000 r$
                                   8BEC
                  00401001
                                   83EC 08
8B45 08
                  00401003
                                                    MOV EAX, DWORD PTR SS: [EBP+8]
MOV DWORD PTR SS: [EBP-8], EAX
MOV ECX, DWORD PTR SS: [EBP+C]
MOV DWORD PTR SS: [EBP-4], ECX
MOV EAX, DWORD PTR SS: [EBP-8]
ADD EAX, DWORD PTR SS: [EBP-4]
                  00401006
                                   8945 F8
                 00401009
  add()
                 0040100C
                                   8B4D 0C
                                   894D FC
8B45 F8
                  0040100F
                  00401012
                 00401015 .
                                   0345 FC
                                                    MOV ESP,EBP
POP EBP
                                   8BE5
                 0040101A
0040101B
                                   5D
C3
CC
                                                     RETN
                  0040101C
                                                     INTS
                                   CC
CC
CC
55
                  0040101D
                  0040101E
                  0040101F
                                                     INTS
                                                    PÜSH EBP
MOV EBP,ESP
                  00401020 5
                  00401021
                                   8BEC
                                  00401023
                  00401026
                  0040102DI
                  00401034
                 00401037
                                                    PUSH EAX
                                   50
                 00401038
                                                    MOV ECX, DWORD PTR SS: [EBP-4]
                                   8B4D FC
                  0040103B
                                                    PUSH ECX
main()
                                   E8 BFFFFFF
                  0040103C
                                                     CALL StackFra.00401000
                 00401041
                                   83C4 08
                                                    ADD ESP,8
                                                    PUSH EAX
                 00401044
                                   50
                                                    PUSH StackFra.0040B384
CALL StackFra.00401067
                  00401045
                                   68 84B34000
                                   E8 18000000
                  0040104A
                                                    ADD ESP,8
                                   83C4 Ø8
                  0040104F
                                                    XOR EAX, EAX
                  00401052
                                   3300
                                                    MOV ESP, EBP
                  00401054
                                   8BE5
                  00401056
                                                     POP EBP
                                                    RETN
```



main()

Registers (FPU)

EAX 00831CD0 ECX 00000001 EDX 00000001 EBX 00357000 ESP <mark>0019FF28</mark> EBP 0019FF70

401020 : EBP가 본래 가지고 있는 값을 스택에 넣는다.

-> 스택에 본래 EBP 값을 백업하는 것 (19FF70)

401021 : ESP의 값을 EBP로 옮긴다.

-> EBP의 값은 고정된다. (19FF28)

```
8BEC
                                 MOV EBP,ESP
00401021
                                 SUB ESP,8
                83EC 08
00401023
                C745 FC 0100 MOV DWORD PTR SS:[EBP-4],1
C745 F8 0200 MOV DWORD PTR SS:[EBP-8],2
8B45 F8 MOV EAX,DWORD PTR SS:[EBP-8]
00401026
0040102D
00401034
                                 PUSH EAX
00401037
                                 MOV ECX, DWORD PTR SS: [EBP-4]
00401038
                8B4D FC
0040103B
                51
                                 PUSH ECX
                É8 BFFFFFF
                                CALL StackFra.00401000
                83C4 08
50
                                 ADD ESP.8
00401041
00401044
                                 PUSH EAX
                                 PUSH StackFra.0040B384
00401045
                68 84B34000
                E8 18000000
                                CALL StackFra.00401067
ADD ESP,8
0040104A
0040104F
                83C4 08
                                XOR EAX, EAX
00401052
                33C0
                                 MOV ESP,EBP
POP EBP
00401054
00401054
00401056
                8BE5
00401057 4.
                                RETN
```

• • • • • • • •

main()

Registers (FPU)

EAX 00831CD0 ECX 00000001 EDX 00000001 EBX 00357000 ESP 0019FF28 EBP 0019FF28

401020 : EBP가 본래 가지고 있는 값을 스택에 넣는다.

-> 스택에 본래 EBP 값을 백업하는 것 (19FF70)

401021 : ESP의 값을 EBP로 옮긴다.

-> EBP의 값은 고정된다. (19FF28)

```
PUSH EBP
                8BEC
                                MOU EBP.ESP
00401021
                83EC 08
00401023
                                SUB ESP,8
               C745 FC 0100 MOV DWORD PTR SS:[EBP-4],1
C745 F8 0200 MOV DWORD PTR SS:[EBP-8],2
8B45 F8 MOV EAX,DWORD PTR SS:[EBP-8]
00401026
0040102D
00401034
                                PUSH EAX
00401037
                                MOV ECX, DWORD PTR SS: [EBP-4]
00401038
                8B4D FC
0040103B
                51
                                PUSH ECX
                É8 BFFFFFF
                                CALL StackFra.00401000
                83C4 08
50
                                ADD ESP.8
00401041
00401044
                                PUSH EAX
                                PUSH StackFra.0040B384
00401045
                68 84B34000
                E8 18000000
0040104A
                                CALL StackFra.00401067
                                ADD ESP.8
0040104F
                83C4 08
                                XOR EAX, EAX
00401052
                33C0
                                MOV ESP,EBP
POP EBP
00401054
00401054
00401056
                8BE5
00401057 4.
                                RETN
```



main()

EBP ==> 0019FF28 r0019FF70 EBP+4 0019FF2C 00401250 EBP+8 0019FF30 00000001 EBP+C 0019FF34 00831C80 00831CD0 0019FF38 EBP+14 0019FF3C BFC13A62 004012A7 EBP+18 0019FF40 004012A7 EBP+1C 0019FF44 EBP+20 0019FF48 00357000

401020 : EBP가 본래 가지고 있는 값을 스택에 넣는다.

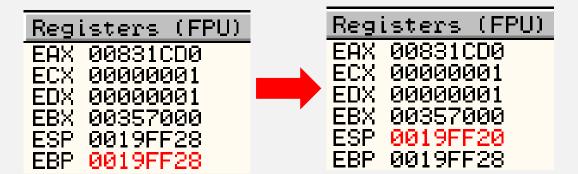
-> 스택에 본래 EBP 값을 백업하는 것 (19FF70)

401021 : ESP의 값을 EBP로 옮긴다.

-> EBP의 값은 고정된다. (19FF28)

00401020 5	55	PUSH EBP	
00401021	8BEC	MOU EBP.ESP	
00401023	83EC 08	SUB ESP,8	
00401026	C745 FC 0100	MOV DWORD PTR SS:[EBP-4],1	
0040102D .	C745 F8 0200	MOV DWORD PTR SS:[EBP-8].2	
00401034	8B45 F8	MOV EAX,DWORD PTR SS:[EBP-8]	
00401037	50	PUSH EAX	
00401038	8B4D FC	MOV ECX.DWORD PTR SS:[EBP-4]	
0040103B .	51	PUSH ECX	
0040103C .	É8 BFFFFFF	CALL StackFra.00401000	
00401041	83C4 Ø8	ADD ESP.8	
00401044	50	PUSH EAX	
00401045	68 84B34000	PUSH StackFra.0040B384	
0040104A	E8 18000000	CALL StackFra.00401067	
0040104F	83C4 08	ADD ESP,8	
00401052	3300	XOR EAX, EAX	
00401054	8BE5	MOV ESP,EBP	
00401056	5D	POP EBP	
00401057	C3	RETN	
00101001	00	INE I	

long a = 1, b = 2;



401023 : ESP 값에서 8을 뺀다.

- -> main() 함수의 로컬 변수를 스택에 저장하기 위함
- -> 4바이트 크기 의 long 타입 변수 2개 이므로 총 8바이트의 공간이 필요하기 때문

401026 ~ 40102D : 각각의 주소에 1 과 2를 넣는다.

```
PUSH EBP
                                 MOV EBP, ESP
                                 SUB ESP.8
                 83EC
00401023
                C745 FC 0100 MOV DWORD PTR SS:[EBP-4],1
C745 F8 0200 MOV DWORD PTR SS:[EBP-8],2
8B45 F8 MOV EAX,DWORD PTR SS:[EBP-8]
00401034
                                 PUSH EAX
                 8B4D FC
                                 MOV ECX, DWORD PTR SS: [EBP-4]
00401038
0040103B
                 51
                                 PUSH ECX
                É8 BFFFFFF
                                 CALL StackFra.00401000
                83C4 08
50
                                  ADD ESP.8
00401041
00401044
                                 PUSH EAX
00401045
                68 84B34000
                                 PUSH StackFra.0040B384
                E8 18000000
                                 CALL StackFra.00401067
ADD ESP,8
0040104A
0040104F
                83C4 08
                                 XOR EAX, EAX
00401052
                 33C0
                                 MOV ESP,EBP
POP EBP
00401054
                8BE5
00401056 .
                                 RETN
```

long a = 1, b = 2;

EBP-8	0019FF20	000000002
EBP-4	0019FF24	000000001
EBP ==	> 0019FF28	┏ 0019FF70
EBP+4	0019FF2C	00401250
EBP+8	0019FF30	000000001
EBP+C	0019FF34	 00831C80

401023 : ESP 값에서 8을 뺀다.

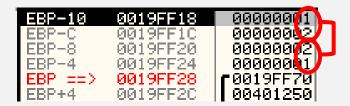
- -> main() 함수의 로컬 변수를 스택에 저장하기 위함
- -> 4바이트 크기 의 long 타입 변수 2개 이므로 총 8바이트의 공간이 필요하기 때문

401026 ~ 40102D : 각각의 주소에 1 과 2를 넣는다.

00401020 rs	55	PUSH EBP
00401021	8BEC	MOV EBP,ESP
00401023 .	83EC 08	SUB FOR,8
00401026 .	C745 FC 0100	
0040102D .		MOO DWORD PTR SS:[EBP-8],2
00401034	8B45 F8	MOV EAK DWORD PTR SS: [FPP 8]
00401037	50	PUSH_EAX
00401038 .	8B4D FC	MOV ECX, DWORD PTR SS: [EBP-4]
0040103B .	51	PUSH ECX
0040103C .	E8 BFFFFFFF	CALL_StackFra.00401000
00401041	83C4 08	ADD ESP,8
00401044	50	PUSH EAX
00401045	68 84B34000	PUSH StackFra.0040B384
0040104A .	E8 18000000 83C4 08	CALL StackFra.00401067
0040104F . 00401052 .	3300	ADD ESP.8 XOR EAX.EAX
00401054	8BE5	MOV ESP,EBP
00401056	5D	POP EBP
00401057	C3	RETN
00701031	00	NETTI

• • • • • • • • •

... add(a,b));



401034~ 40103B : 각각의 변수를 스택에 집어넣는다. -> 함수 파라미터의 역순 저장

40103C : add() 함수 호출

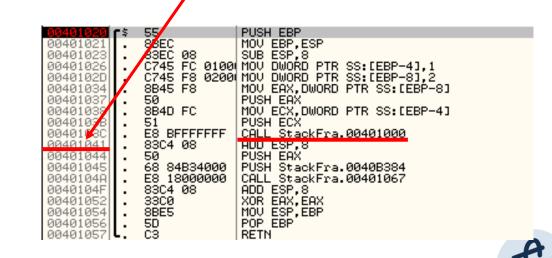
```
PUSH EBP
MOV EBP,ESP
SUB ESP,8
                 83EC
                 C745 FC 0100 MOV DWORD PTR SS:[EBP-4],1
C745 F8 0200 MOV DWORD PTR SS:[EBP-8],2
8B45 F8 HOV EAX,DWORD PTR SS:[EBP-8]
0040102D
                                    HOO EAX,DWORD PTR SS:[EBP-8:
00401034
                                   PUSH EAX
                  8B4D FC
                                   MOV ECX, DWORD PTR SS: [EBP-4]
00401038
0040103B
                  51
                                   PUSH ECX
                 E8 BFFFFFF
83C4 08
50
                                   CALL See
                                   ADD ESP,8
00401041
                                   PUSH EAX
00401044
                 68
                                   PUSH StackFra.0040B384
00401045
                     84B34000
                 E8 18000000
                                   CALL StackFra.00401067
ADD ESP,8
0040104A
                 83C4 08
0040104F
                                   XOR EAX, EAX
00401052
                  33C0
                 8BE5
5D
                                   MOV ESP,EBP
POP EBP
00401054
00401054 .
00401056 .
00401057 .
                                   RETN
```

• • • • • • • • •

... add(a,b));

401034~ 40103B : 각각의 변수를 스택에 집어넣는다. -> 함수 파라미터의 역순 저장

40103C : add() 함수 호출



long add(long a, long b)

EBP-8	0019FF08	000000001
EBP-4	0019FF0C	000000002
EBP ==>	0019FF10	<u> 60019FF28</u>
EBP+4	0019FF14	00401041
EBP+8	0019FF18	000000001
EBP+C	0019FF1C	000000002

main() 함수 때와 방식은 동일하지만, 스택의 위치는 다르다.

80401000 rs 00401001 00401003 00401006 00401009 0040100C	55 8BEC 83EC Ø8 8B45 Ø8 8945 F8 8B4D ØC	PUSH EBP MOV EBP,ESP SUB ESP,8 MOV EAX,DWORD PTR SS:[EBP+8] MOV DWORD PTR SS:[EBP-8],EAX MOV ECX,DWORD PTR SS:[EBP+C]
0040100F 00401012 00401015 00401018 0040101A 0040101B	894D FC 8845 F8 0345 FC 88E5 5D C3	MOU DWORD PTR SS:[EBP-4],ECX MOU EAX,DWORD PTR SS:[EBP-8] ADD EAX,DWORD PTR SS:[EBP-4] MOU ESP,EBP POP EBP RETN



return (x + y);

Registers (FPU) EAX 00000001 ECX 00000002 EDX 00000001 EBX 00357000 ESP 0019FF08

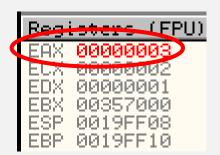
401012: EAX에 변수 x의 값 ([EBP-8] = 1)을 넣는다.

401015 : EAX에 변수 y의 값 ([EBP-4] = 2) 를 더한다.

99491099	55 8BEC 83EC 08 8B45 08 8945 F8 8B4D 0C 894D FC 8B45 F8 0345 FC 8BE5	PUSH EBP MOV EBP,ESP SUB ESP,8 MOV EAX,DWORD PTR SS:[EBP+8] MOV DWORD PTR SS:[EBP-8],EAX MOV ECX,DWORD PTR SS:[EBP+C] MOV DWORD PTR SS:[EBP-4],ECX MOV EAX,DWORD PTR SS:[EBP-8] ADD EAX,DWORD PTR SS:[EBP-4] MOV ESP,EBP
00401018	8BE5	MOV ESP,EBP
0040101A	5D	POP EBP
0040101B	C3	RETN



return (x + y);



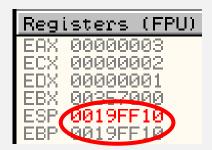
401012: EAX에 변수 x의 값 ([EBP-8] = 1)을 넣는다.

401015 : EAX에 변수 y의 값 ([EBP-4] = 2) 를 더한다.

00401000 r \$	55	PUSH EBP
00401001	8BEC	MOV EBP,ESP
00401003	83EC 08	SUB ESP,8
00401006	8B45 Ø8	MOV EAX, DWORD PTR SS: [EBP+8]
00401009	8945 F8	MOV DWORD PTR SS:[EBP-8],EAX
0040100C .	8B4D ØC	MOV ECX, DWORD PTR SS: [EBP+C]
0040100F .	894D FC	MOV DWORD PTR SS:[EBP-4],ECX
00401012	8B45 F8	MOV EAX, DWORD PTR SS: [EBP-8]
00401015	0345 FC	ADD EAX.DWORD PTR SS:[EBP-4]
00401018	8BE5	MOV ESP,EBP
0040101A .	5D	POP EBP
0040101B L.	Č3	RETN



return (x + y);



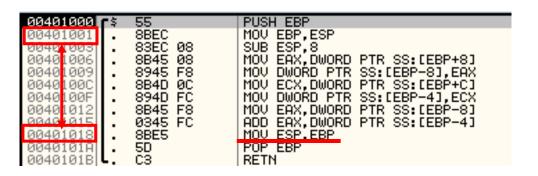
401018 : EBP 값을 ESP에 대입한다.

-> 401001 주소와 대응하는 코드

40101A : 스택에 백업한 EBP 값을 복원한다.

-> 401000 주소와 대응하는 코드

40101B : 스택에 저장된 복귀 주소로 리턴





• • • • • • • • •

return (x + y);

Registers (FPU) EAX 00000003 ECX 00000002 EDX 00000001 EBX 00357000

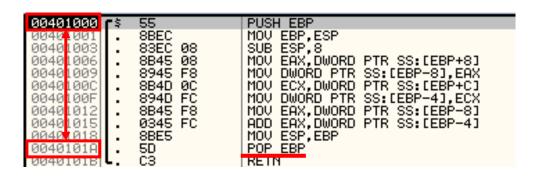
401018 : EBP 값을 ESP에 대입한다.

-> 401001 주소와 대응하는 코드

40101A: 스택에 백업한 EBP 값을 복원한다.

-> 401000 주소와 대응하는 코드

40101B: 스택에 저장된 복귀 주소로 리턴





• • • • • • • • •

return (x + y);

EBP-10	0019FF18	000000001
EBP-C	0019FF1C	000000002
EBP-8	0019FF20	000000002
EBP-4	0019FF24	000000001
EBP ==>	0019FF28	┌ 0019FF70
EBP+4	0019FF2C	00401250

401018 : EBP 값을 ESP에 대입한다.

-> 401001 주소와 대응하는 코드

40101A : 스택에 백업한 EBP 값을 복원한다.

-> 401000 주소와 대응하는 코드

40101B: 스택에 저장된 복귀 주소로 리턴

00401000	C\$	55	PUSH EBP
00401001	:	8BEC	MOV EBP,ESP
00401003		83EC 08	SUB ESP,8
00401006	:	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]
00401009		8945 F8	MOV DWORD PTR SS:[EBP-8],EAX
0040100C		8B4D 0C	MOV ECX,DWORD PTR SS:[EBP+C]
0040100F	:	894D FC	MOV DWORD PTR SS:[EBP-4],ECX
00401012		8B45 F8	MOV EAX,DWORD PTR SS:[EBP-8]
00401015	:	0345 FC	ADD EAX,DWORD PTR SS:[EBP-4]
00401018		8BE5	MOV ESP,EBP
0040101A 0040101B	l:	5D C3	POP EBP



main()

EBP-8	0019FF20	00000002
EBP-4	0019FF24	000000001
EBP ==>	0019FF28	┌ 0019FF70
EBP+4	0019FF2C	00401250
EBP+8	0019FF30	000000001
EBP+C	0019FF34	 00831080

401041 : ESP에 8을 더한다.

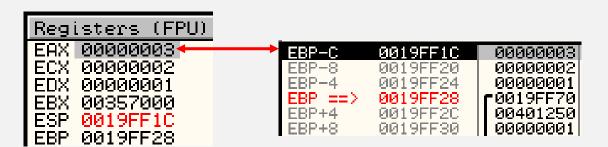
-> add() 함수에게 넘겨준 파라미터를 정리하는 것

401044 : EAX 값에는 add()함수에서 저장된 리턴 값이 있음

401045 ~ 40104F : printf() 함수 내용. (C 표준 라이브러리)

	\$ 55	PUSH EBP
00401021 00401023 00401026 00401034 00401037 00401038 00401038 00401038 00401041 00401044 00401044 00401045 00401047	. 8BEC . 83EC 08 . C745 FC 0100 . C745 F8 0200 . 8B45 F8 . 50 . 8B4D FC . 51 . E8 BFFFFFFF . 83C4 08 . 50 . 68 84B34000 . E8 18000000 . 83C4 08 . 33C0 . 8BE5 . 5D	MOV EBP,ESP SUB ESP,8 MOV DWORD PTR SS:[EBP-4],1 MOV DWORD PTR SS:[EBP-8],2 MOV EAX,DWORD PTR SS:[EBP-8] PUSH EAX MOV ECX,DWORD PTR SS:[EBP-4] PUSH ECX CALL StackFra.00401000 ADD ESP.8 PUSH EHX PUSH EAX PUSH EAX CALL StackFra.00401067 ADD ESP,8 XOR EAX,EAX MOV ESP,EBP POP EBP RETN



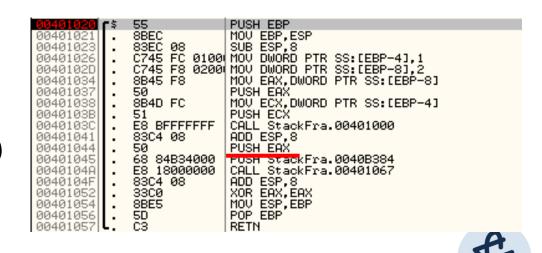


401041 : ESP에 8을 더한다.

-> add() 함수에게 넘겨준 파라미터를 정리하는 것

401044 : EAX 값에는 add()함수에서 저장된 리턴 값이 있음

401045 ~ 40104F : printf() 함수 내용. (C 표준 라이브러리)



return 0;

Registers (FPU) EAX 00000000 ECX 00401102 Stac EDX 0019E324 EBX 003F7000 ESP 0019FF2C EBP 0019FF70

401052 : main()함수의 리턴 값 세팅 (0)

-> XOR 명령어의 경우 실행 속도가 빨라 레지스터를 초기화할 때 자주 사용된다.

401054 ~ 401057 : add() 함수 때와 방식은 동일하다.

```
PUSH EBP
MOV EBP,ESP
SUB ESP,8
                 8BEC
                83EC 08
                C745 FC 0100 MOV DWORD PTR SS:[EBP-4],1
C745 F8 0200 MOV DWORD PTR SS:[EBP-8],2
8B45 F8 MOV EAX,DWORD PTR SS:[EBP-8]
0040102D
00401034
00401037
                                  PUSH EAX
00401038
                                  MOV ECX, DWORD PTR SS: [EBP-4]
                 8B4D FC
0040103B
                 51
                                  PUSH ECX
                É8 BFFFFFF
                                  CALL StackFra.00401000
                                  ADD ESP,8
00401041
                 83C4 Ø8
00401044
                                  PUSH EAX
00401045
                68
                    84B34000
                                  PUSH StackFra.0040B384
                E8 18000000
                                  CALL StackFra.00401067
ADD ESP,8
0040104A
0040104F
                 83C4 08
00401052
                                  XOR EAX.EAX
                 33C0
00401054
                                 MOV ESP,EBP
POP EBP
00401056
00401057 4.
                                  RETN
```



Q&A

