Stack Demo

A walkthrough of how the stack works

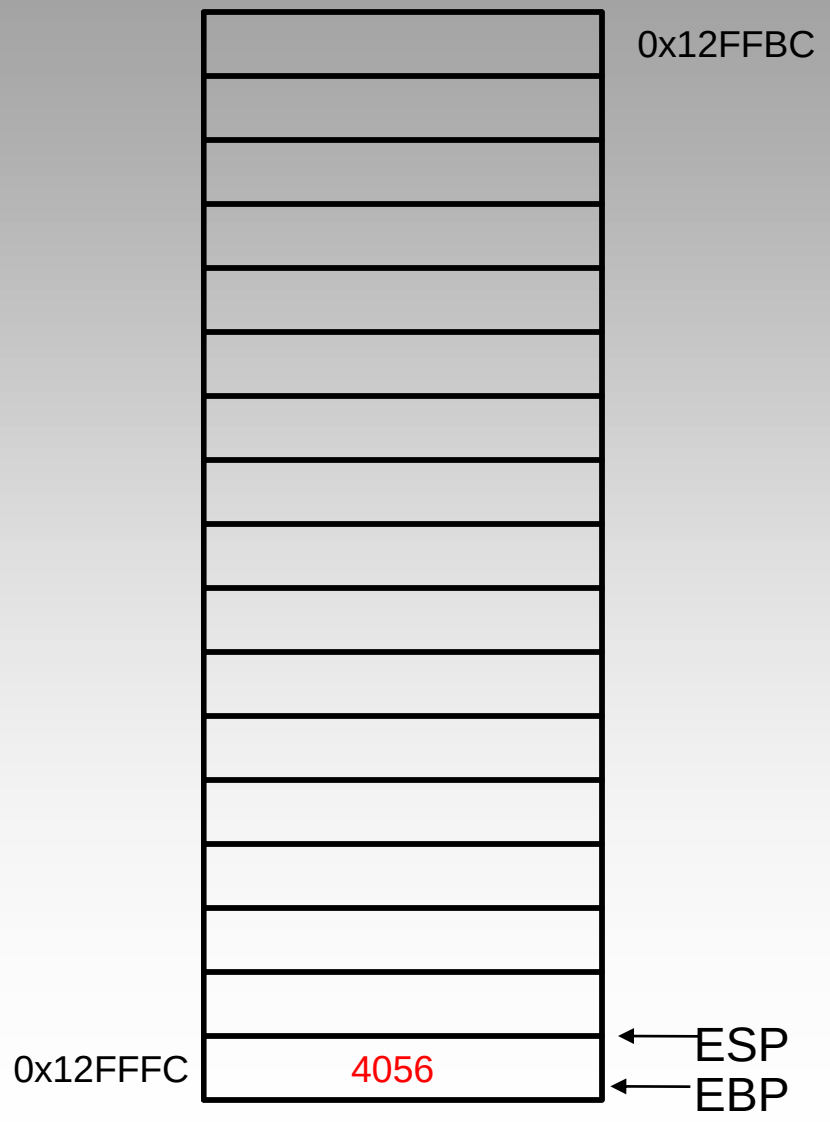
Red text indicates something that changed since the previous slide

Stack Demo

- Push: place value on the stack and subtract four bytes from ESP

eax = 0x1234
ebx = 0x0

push 0x4056
push eax
push ebx

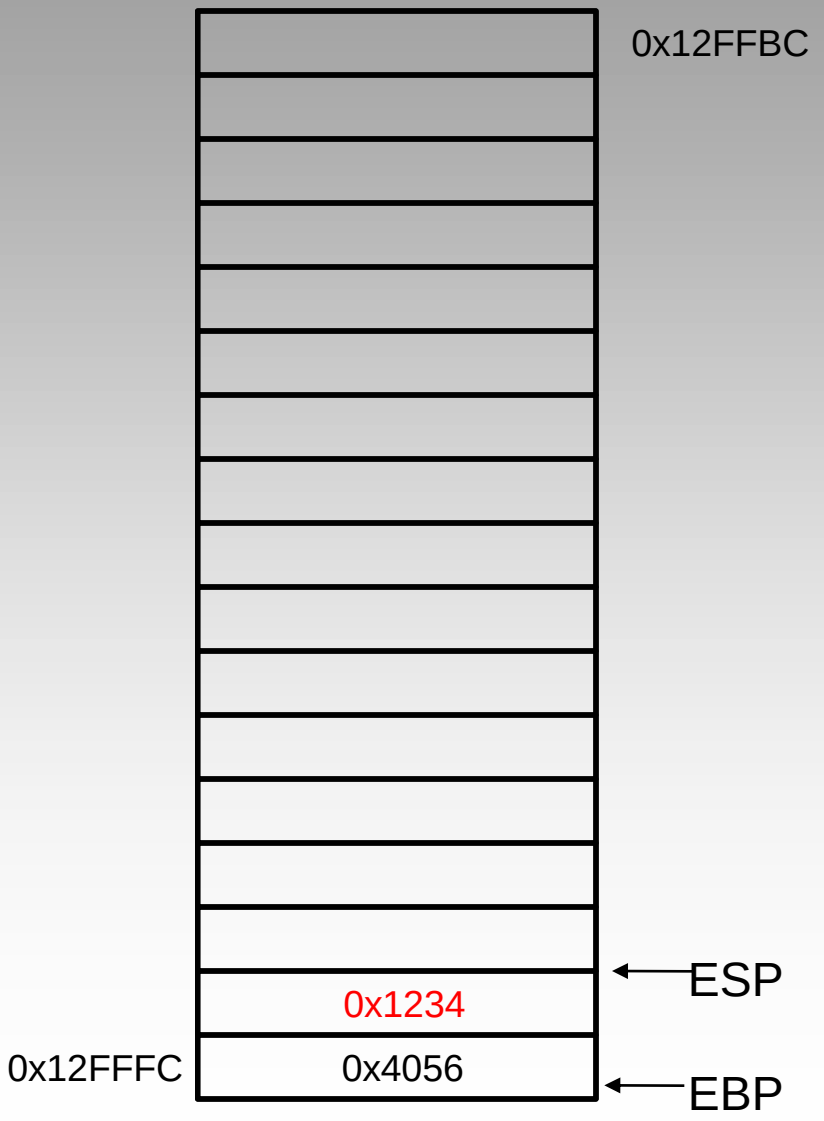


Stack Demo

- Push: place value on the stack and subtract four bytes from ESP

eax = 0x1234
ebx = 0x0

push 0x4056
push eax
push ebx



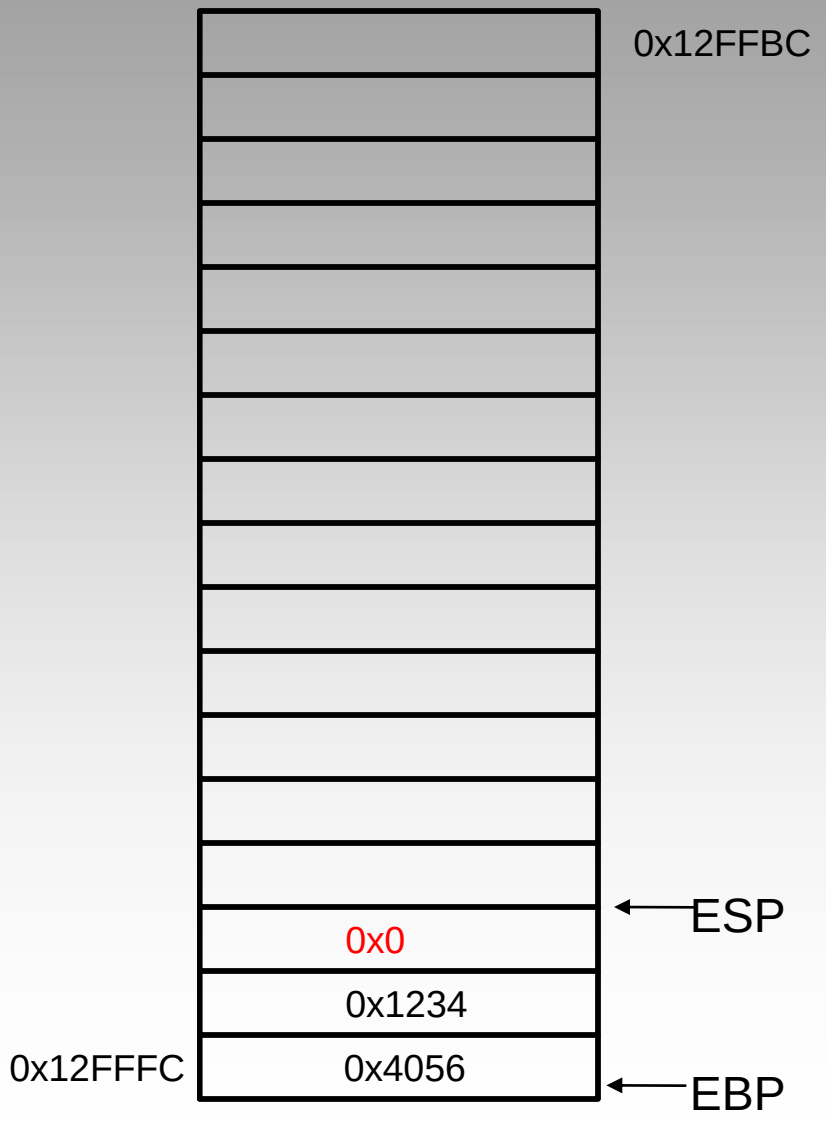
CSRgroup
computer security consultants

Stack Demo

- Push: place value on the stack and subtract four bytes from ESP

eax = 0x1234
ebx = 0x0

push 0x4056
push eax
push ebx

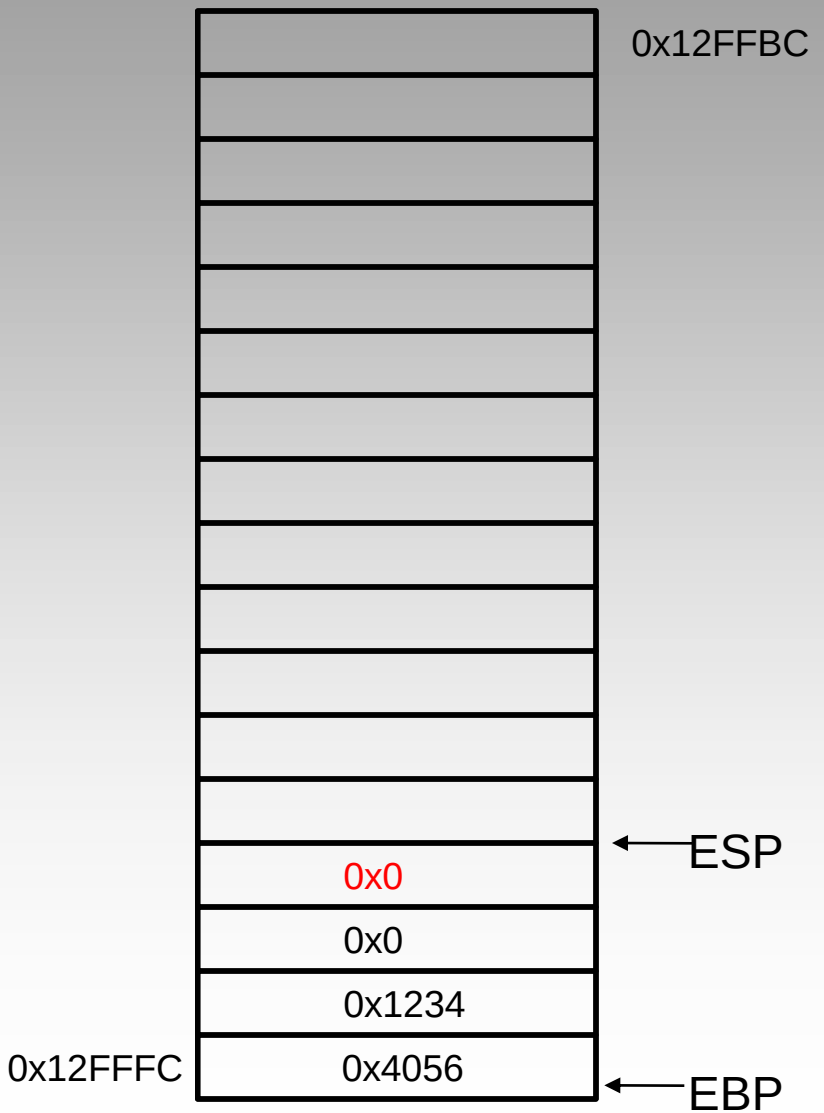


Stack Demo

- Pop: take value off the stack and add four bytes to ESP

eax = 0x1234
ebx = 0x0
ecx = 0x15

push ebx
push ecx
pop ebx
pop ecx

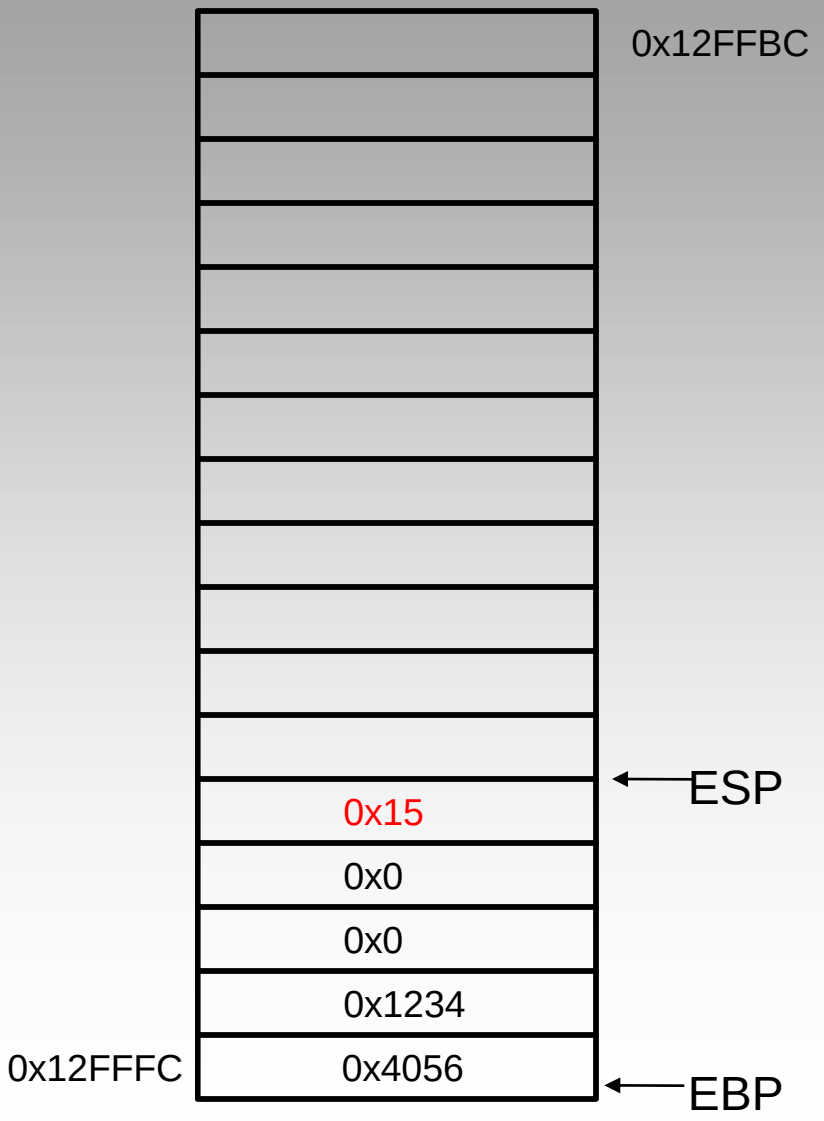


Stack Demo

- Pop: take value off the stack and add four bytes to ESP

eax = 0x1234
ebx = 0x0
ecx = 0x15

push ebx
push ecx
pop ebx
pop ecx

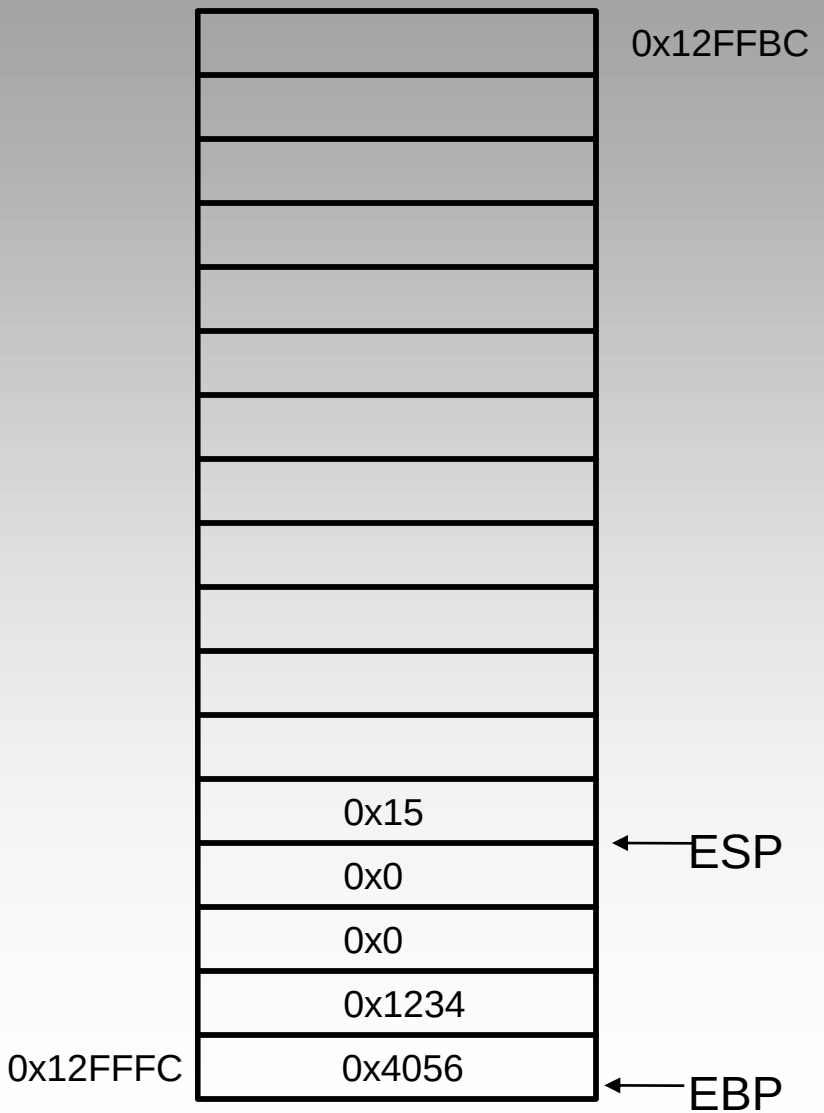


Stack Demo

- Pop: take value off the stack and add four bytes to ESP

eax = 0x1234
ebx = 0x15
ecx = 0x15

push ebx
push ecx
pop ebx
pop ecx



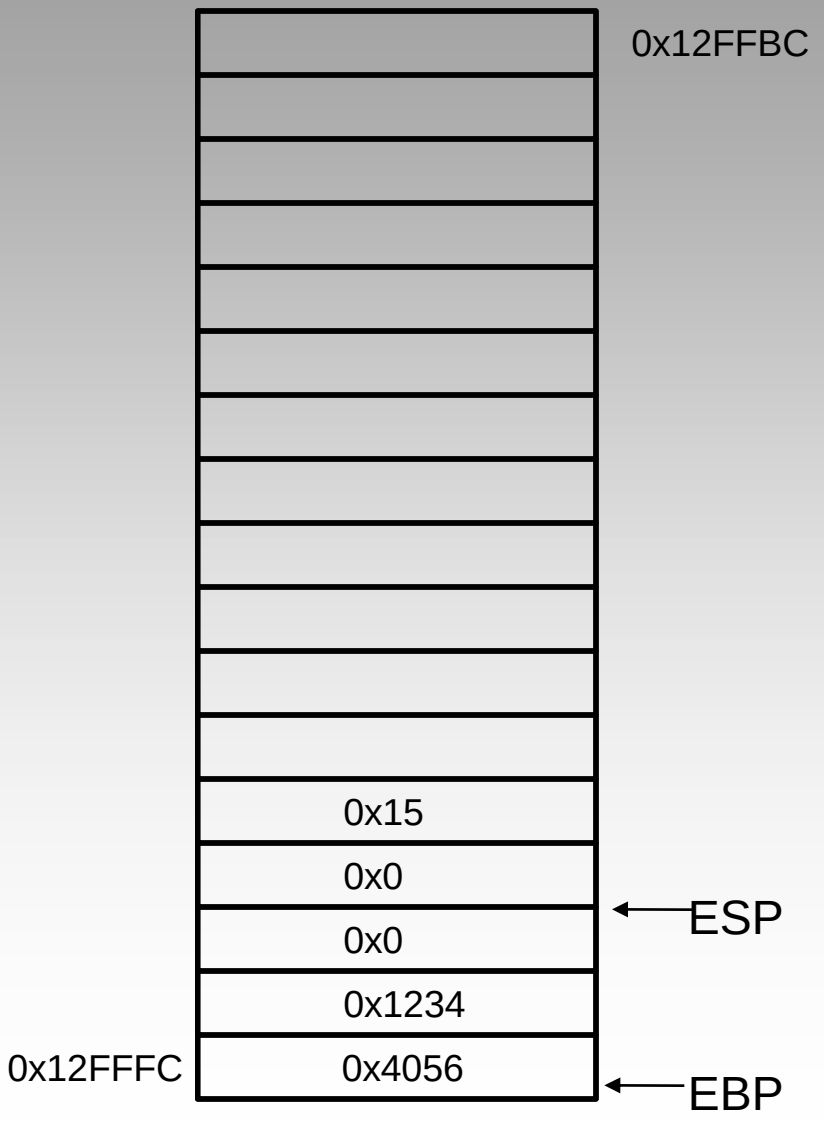
Stack Demo

- Pop: take value off the stack and add four bytes to ESP

eax = 0x1234
ebx = 0x15
ecx = 0x0

push ebx
push ecx
pop ebx
pop ecx

** Key point **
Just because ESP moves, data doesn't automatically disappear from the stack



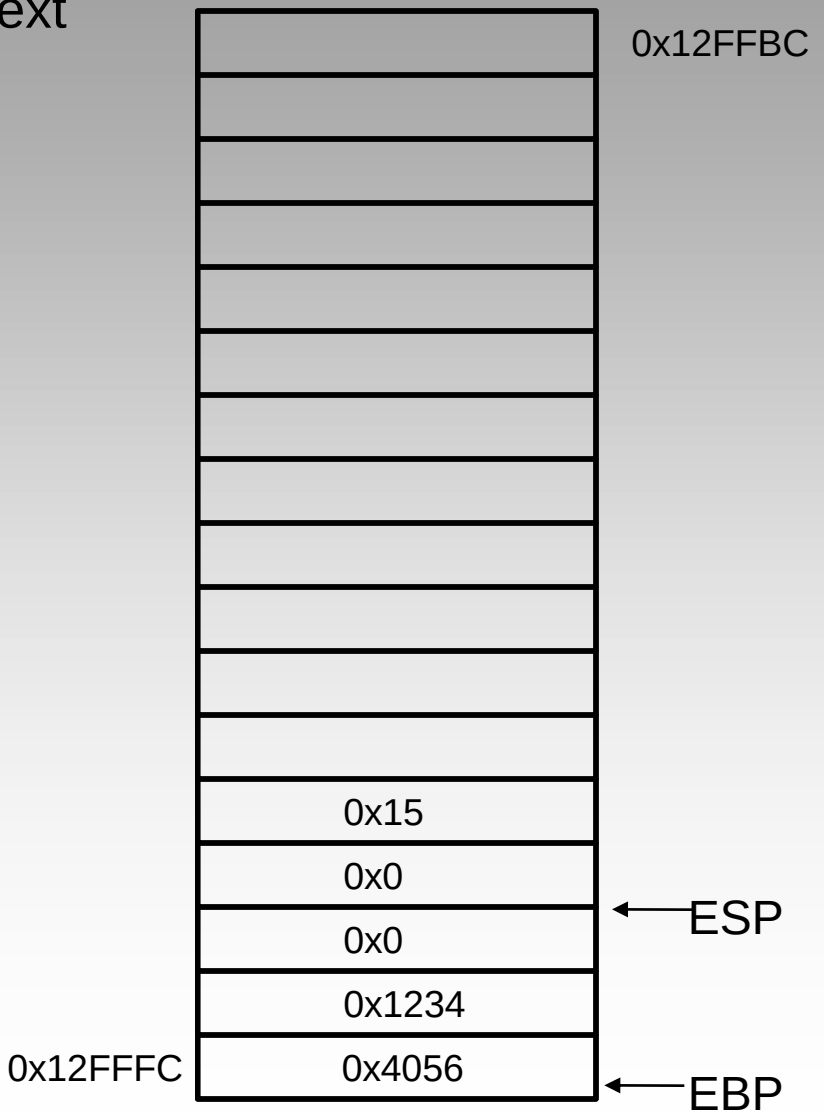
Stack Demo

- Call: push the location of the next instruction onto the stack and transfer control to call address

sub_403804(0x402B0C, 0x15)

eip = 401986
ebx = 0x15

401981 mov ebx, 0x15
401986 push ebx
401988 push 0x402B0C
40198D call 0x403804
401993 cmp eax, eax



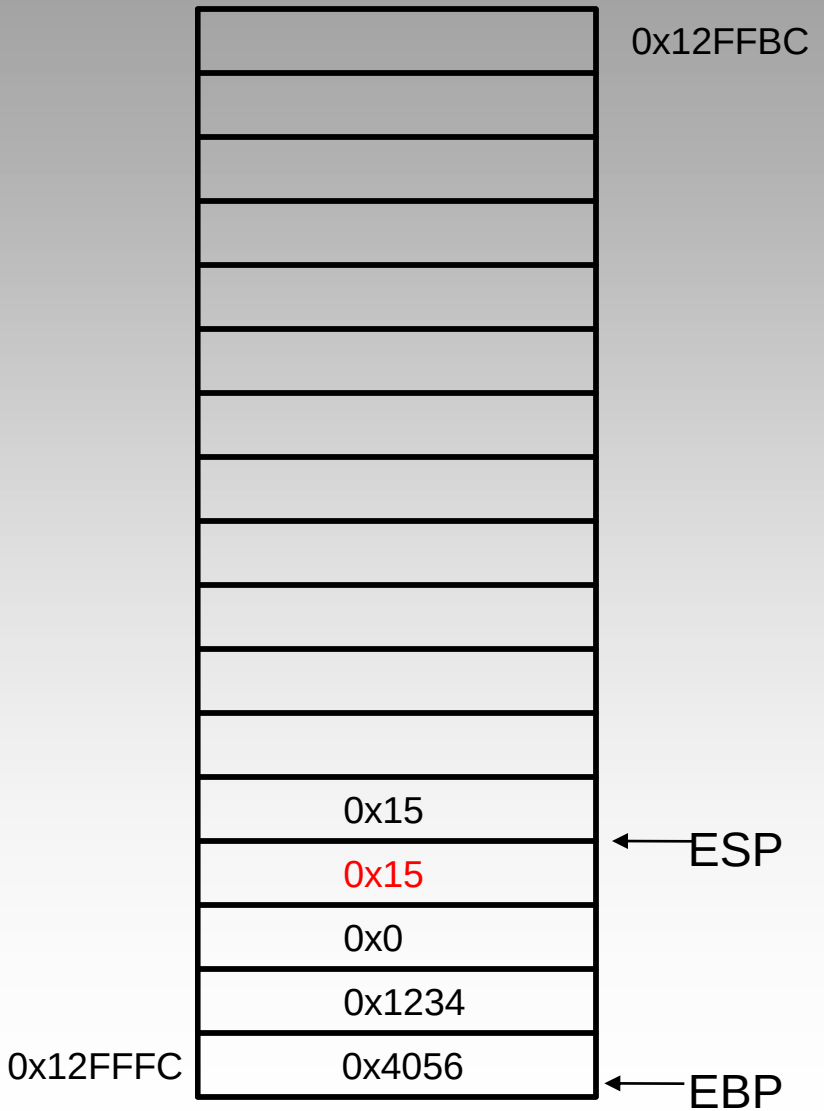
Stack Demo

- Call example

sub_403804(0x402B0C, 0x15)

eip = 401988
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
403804    push ebp
403805    mov ebp, esp
403807    sub esp, 0x10
```



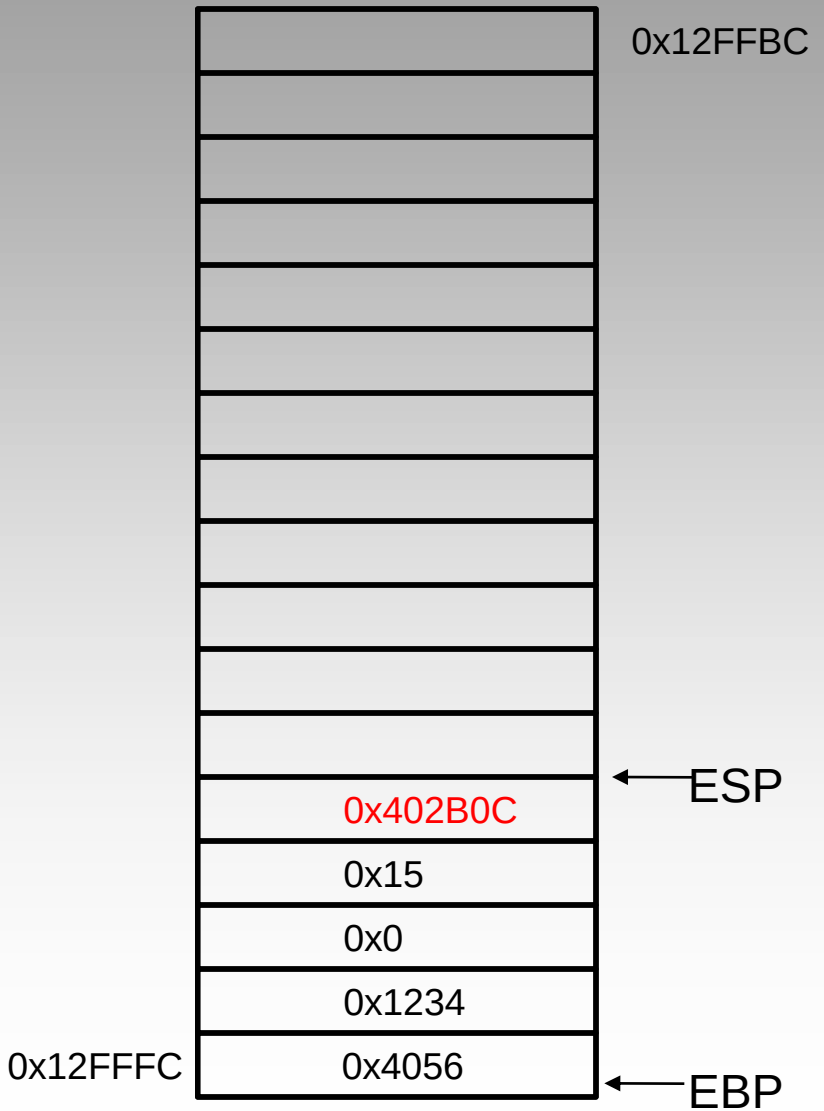
Stack Demo

- Call example

sub_403804(0x402B0C, 0x15)

eip = 40198D
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
403804    push ebp
403805    mov ebp, esp
403807    sub esp, 0x10
```



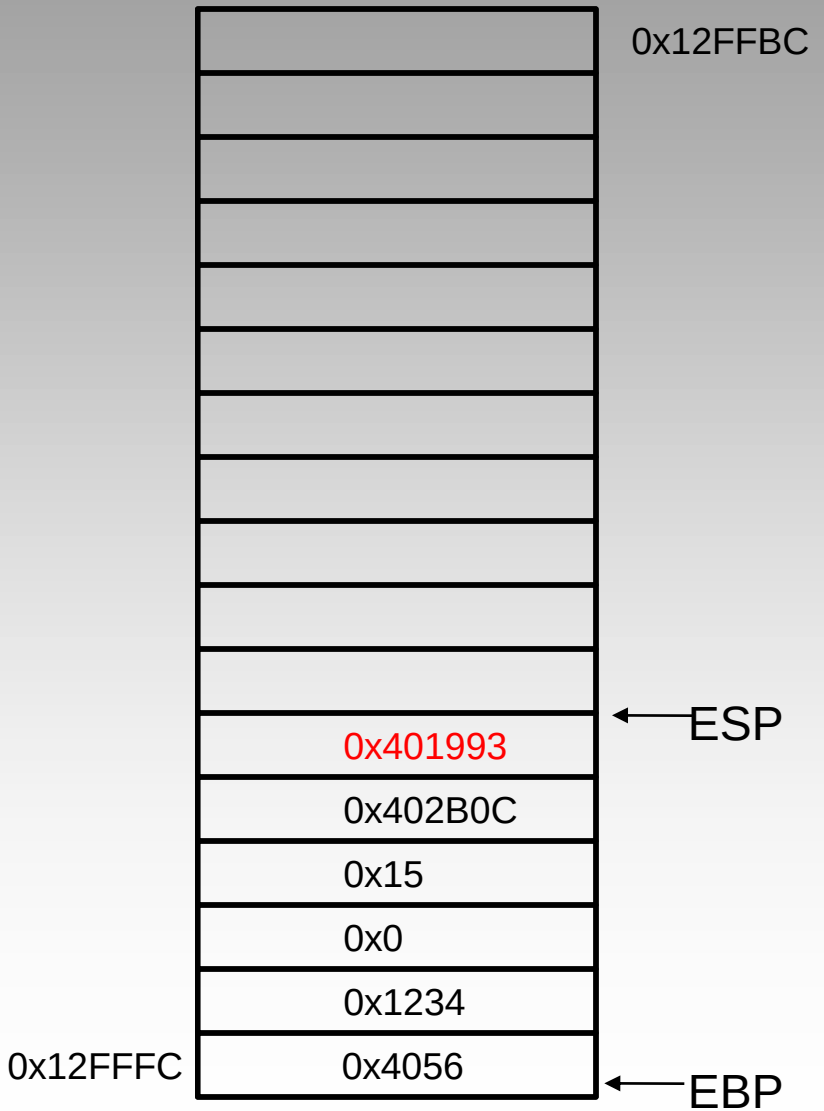
Stack Demo

- Call example

sub_403804(0x402B0C, 0x15)

eip = 403804
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
403804    push ebp
403805    mov ebp, esp
403807    sub esp, 0x10
```



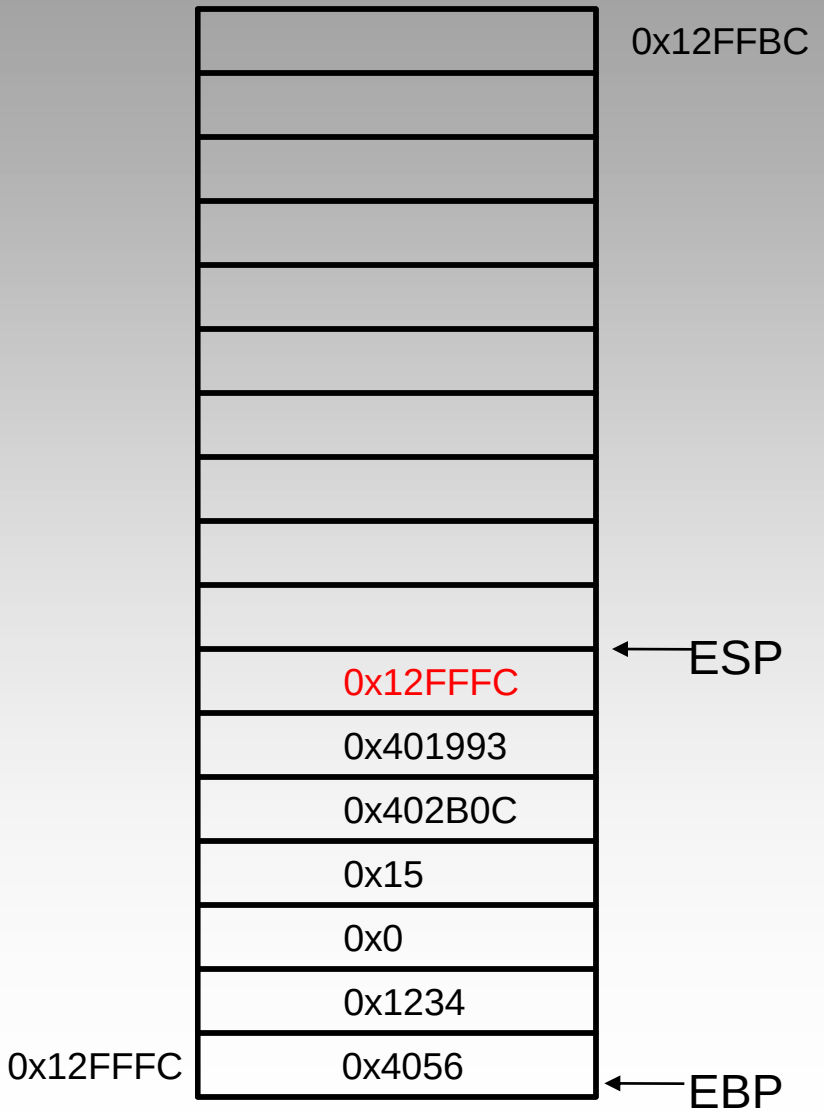
Stack Demo

- Call example

sub_403804(0x402B0C, 0x15)

eip = 403805
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
403804    push ebp
403805    mov ebp, esp
403807    sub esp, 0x10
```



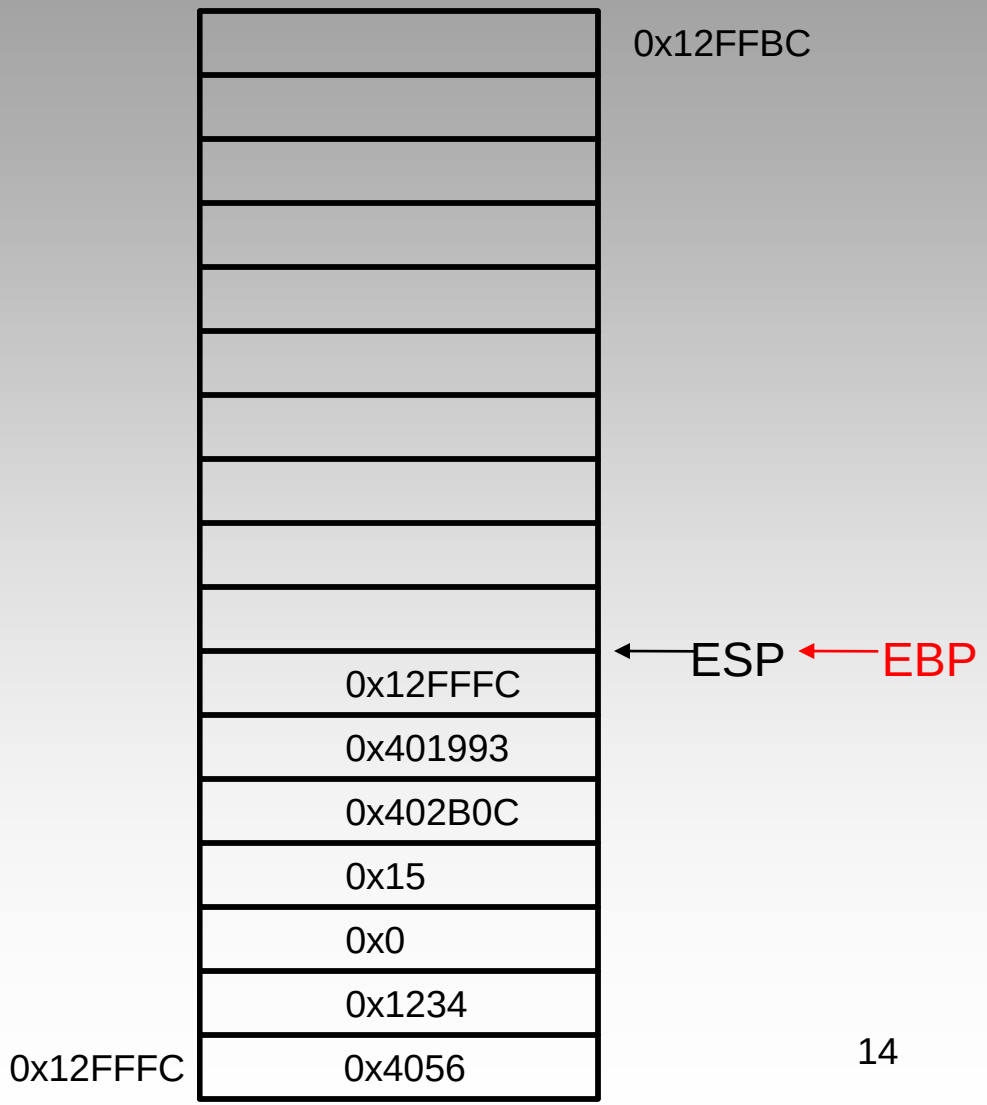
Stack Demo

- Call example

sub_403804(0x402B0C, 0x15)

eip = 403807
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
403804    push ebp
403805    mov ebp, esp
403807    sub esp, 0x10
```



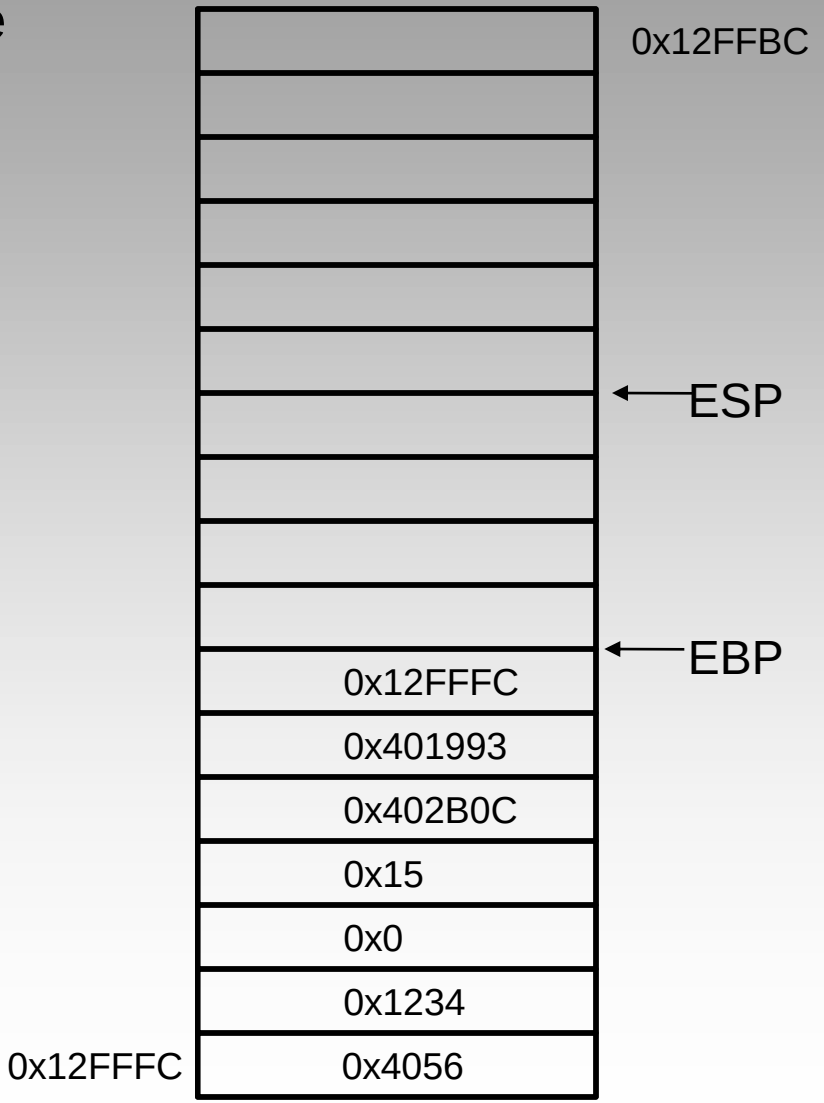
Stack Demo

Subtract 10 bytes from esp (make space on the stack)

sub_403804(0x402B0C, 0x15)

eip = 40380A
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
403804    push ebp
403805    mov ebp, esp
403807    sub esp, 0x10
```



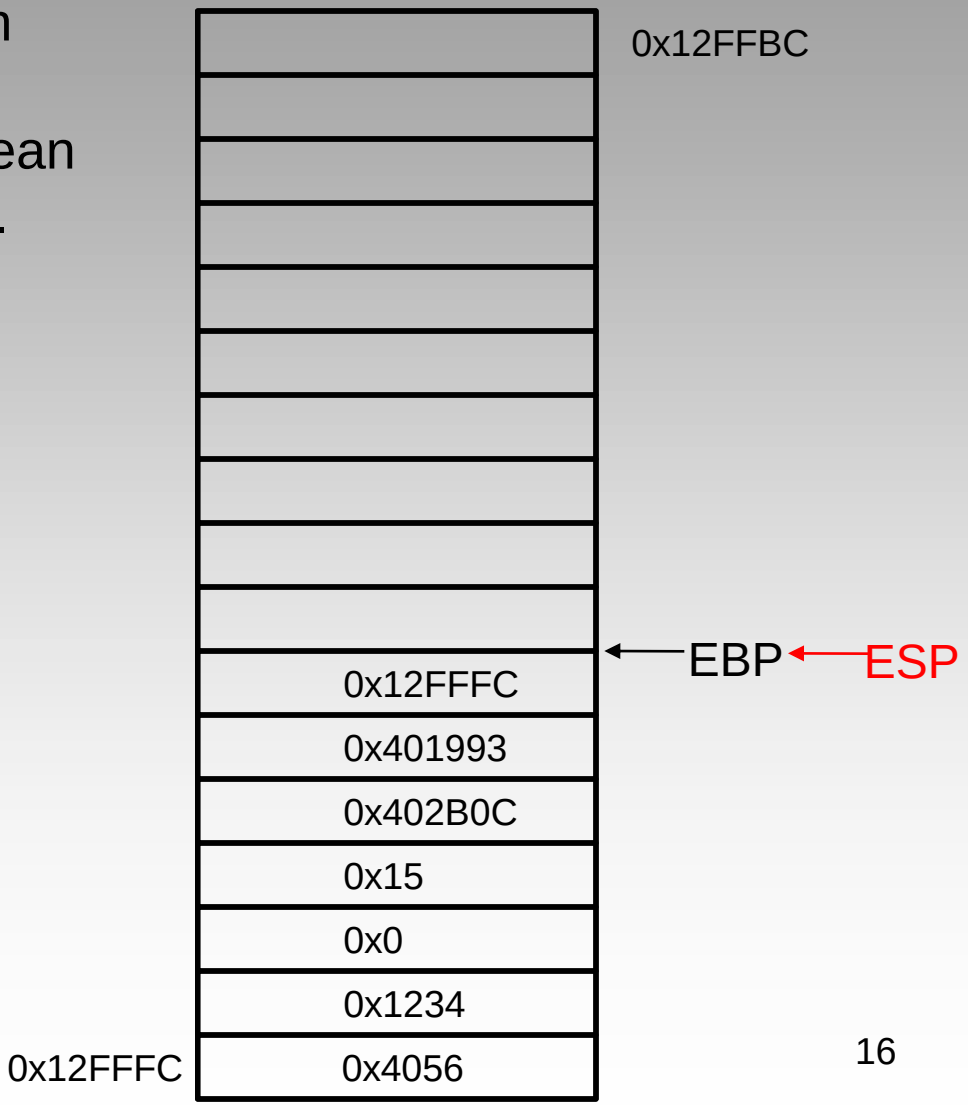
Stack Demo

After some instructions have been executed in the body of the function, add 0x10 to esp to clean up local variables on the stack.

sub_403804(0x402B0C, 0x15)

eip = 40391D
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
40391A    add esp, 0x10
40391D    pop ebp
40391F    ret
```



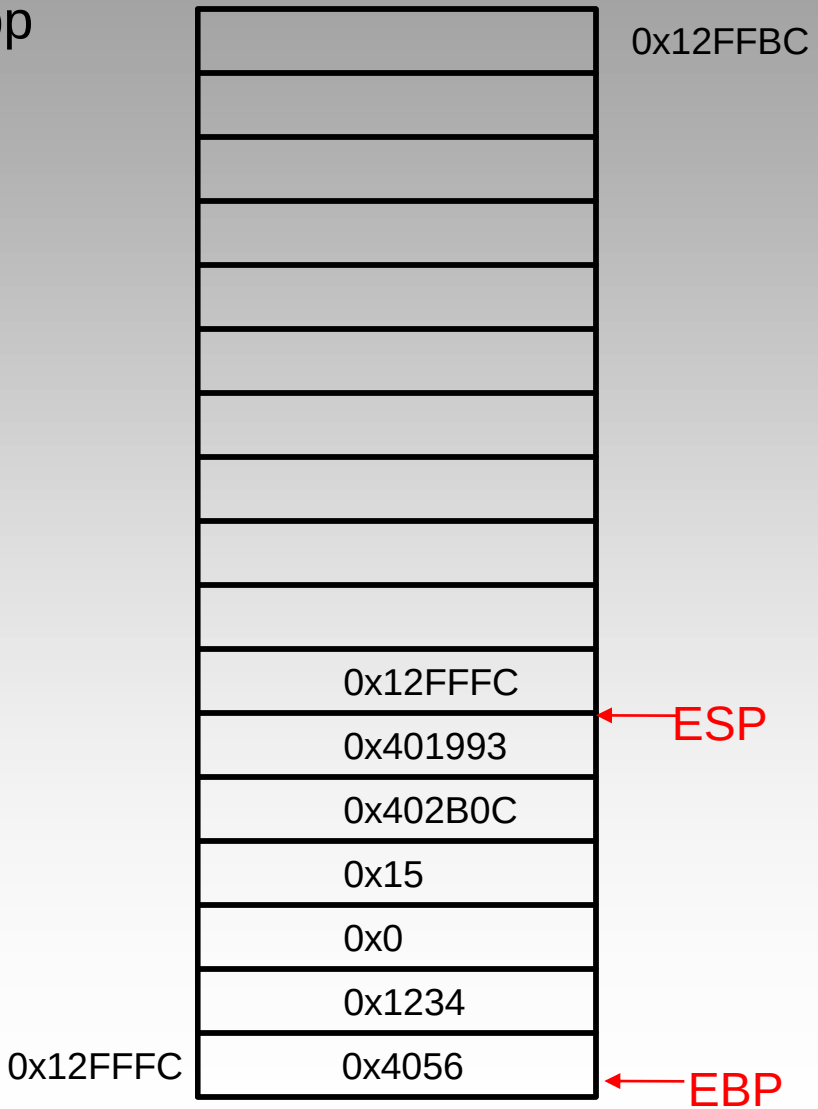
Stack Demo

Pop: retrieve the old version of ebp from the stack

sub_403804(0x402B0C, 0x15)

eip = 40391F
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
40391A    add esp, 0x10
40391D    pop ebp
40391F    ret
```



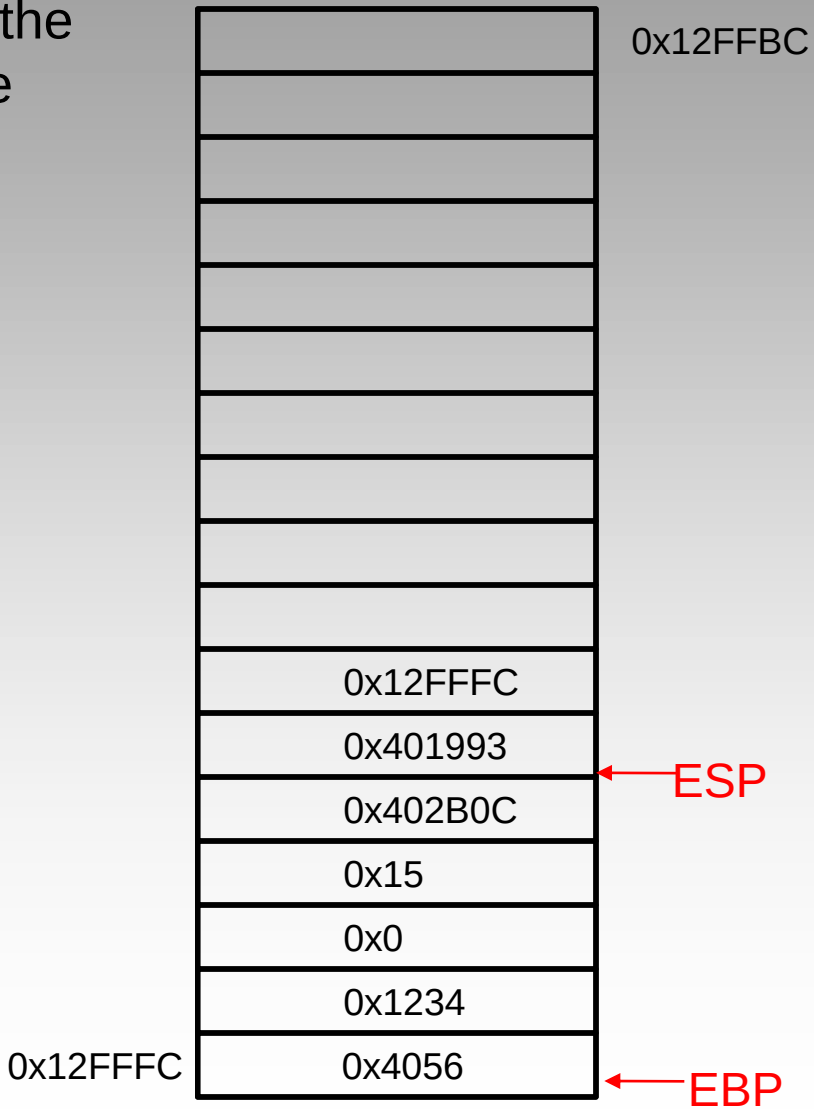
Stack Demo

ret – pop the return address from the stack and transfer control there (note the change in EIP)

sub_403804(0x402B0C, 0x15)

eip = 401993
ebx = 0x15

```
401981    mov ebx, 0x15
401986    push ebx
401988    push 0x402B0C
40198D    call 0x403804
401993    cmp eax, eax
.....
40391A    add esp, 0x10
40391D    pop ebp
40391F    ret
```



Questions/need help reverse engineering malware?

Contact Jake Williams

jwilliams@csr-group.com