Buflab

Introduction

- Individual project, help you develop a detailed understanding of IA-32 calling convention and stack organization.
- We generated the lab using gcc's -m32 flag your machine should have the 32bit library to work on it therefore use the new VM image.
- All code follows the IA-32 rule

Hand Out Instructions

- You can get your buffer lab from:

  http://sysprog.csap.snu.ac.kr:64321/

  - bufbomb: The buffer bomb program you will attack.
  - makecookie: Generates a "cookie" based on your student number.
  - hex2raw: A utility to help convert between string formats.

Hand in Instructions

- First hand in - exploit strings for the different levels that are directly sent to the Buffer Lab's server.
  - The server will automatically validate your submission and update a score table where you can check your current score.

    http://sysprog.csap.snu.ac.kr:64321/scoreboard
- Second handin - a report in PDF format.
  - describe for each of the solved (or attempted) levels how you composed your exploit string

Userids and Cookies

- The correct solution is based on your student number.
  - A cookie is a string of eight hexadecimal digits that is unique to your student number. You can generate your cookie with the makecookie program giving your student number as the argument. For example:

```
→  buflab-handout ./makecookie 2018-111111
0x214fe797
```

  - In four of your five buffer attacks, your objective is to make your cookie show up in places where it ordinarily would not.

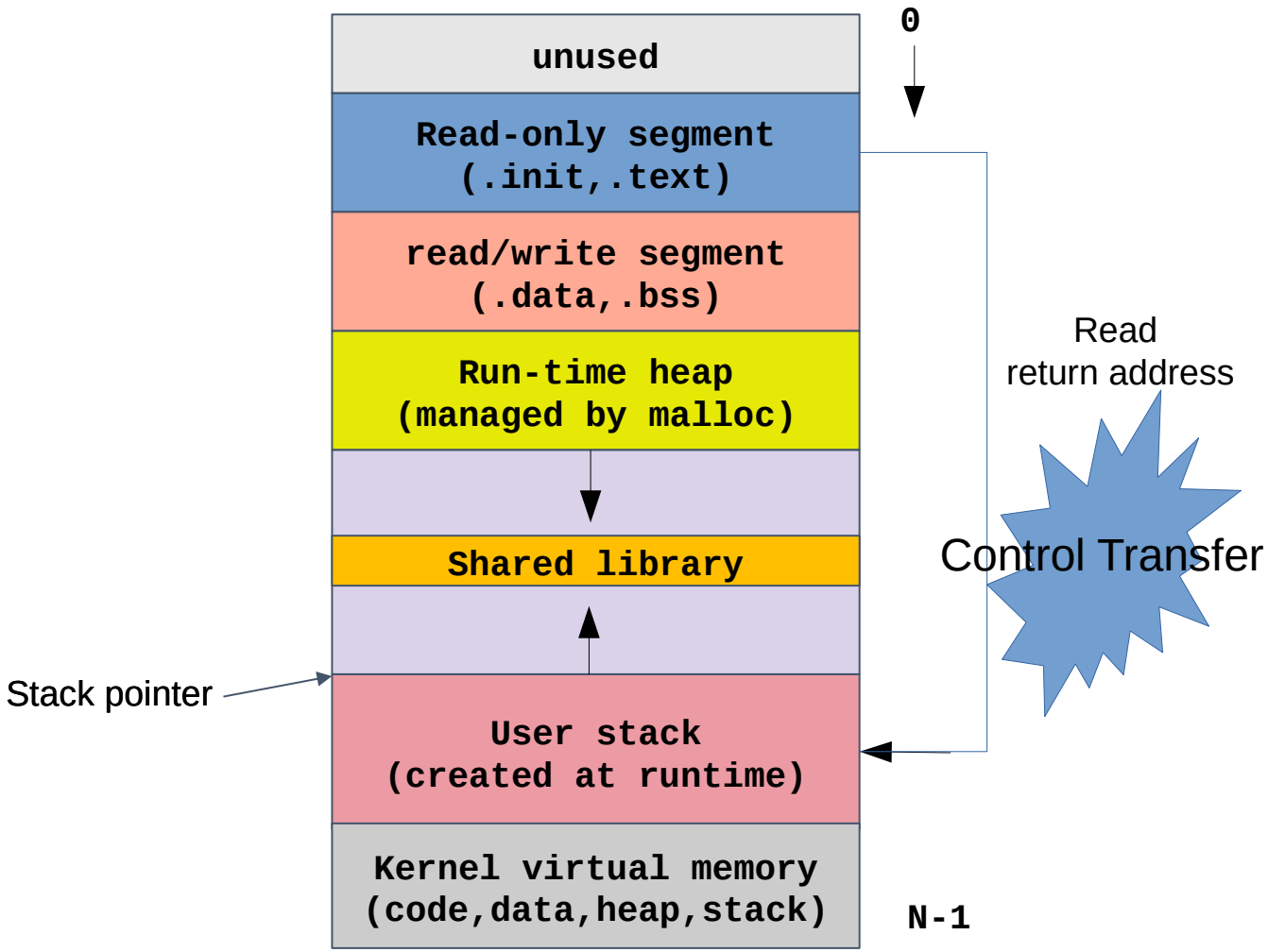
Buflab Tutorial

Buffer Overflow Attack

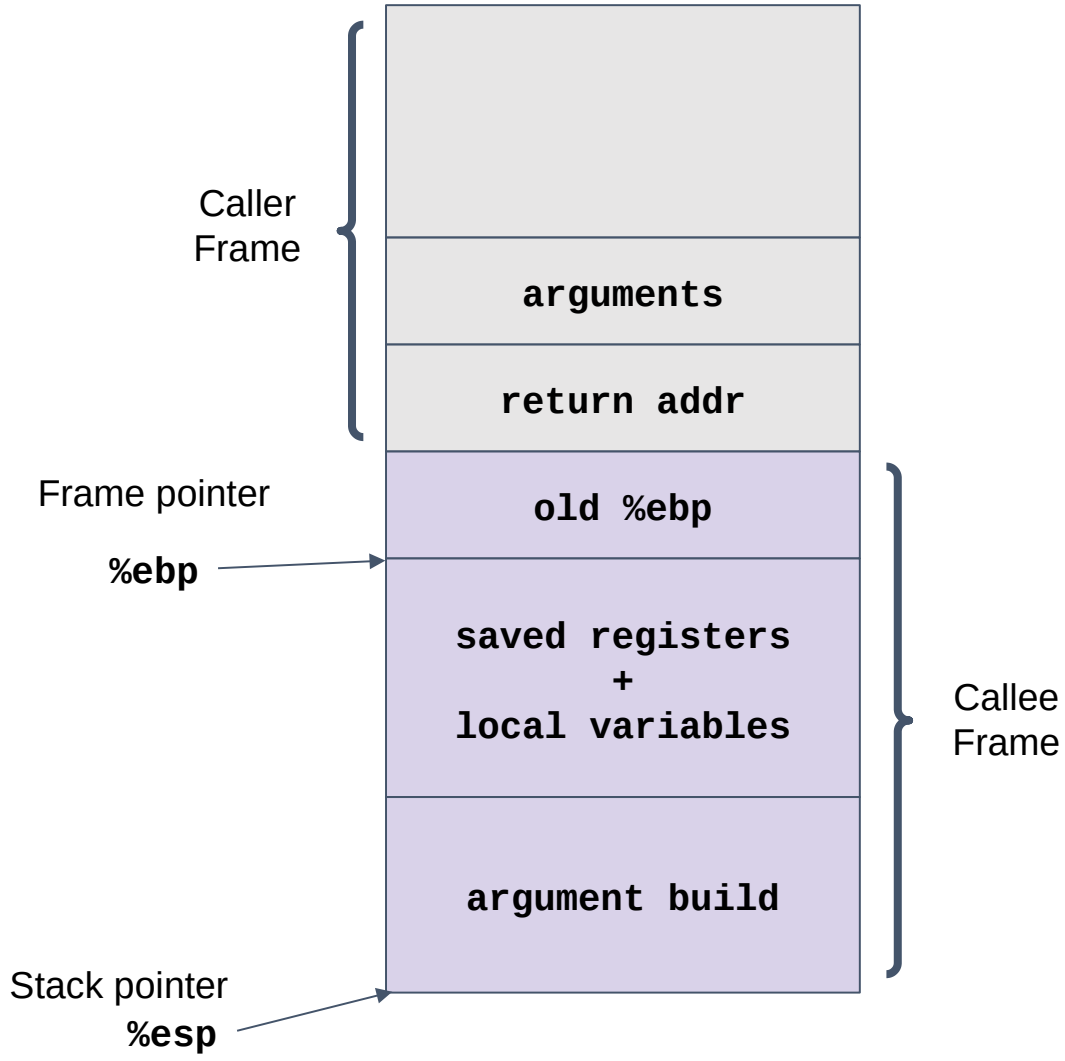
- What is buffer overflow?
  - while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations. (wikipedia)

- How to exploit buffer overflow vulnerability for attack?
  - change the control of a program by overwriting the return address
  - write exploit code on the buffer and make the function return to our code

Linux Virtual Address Space



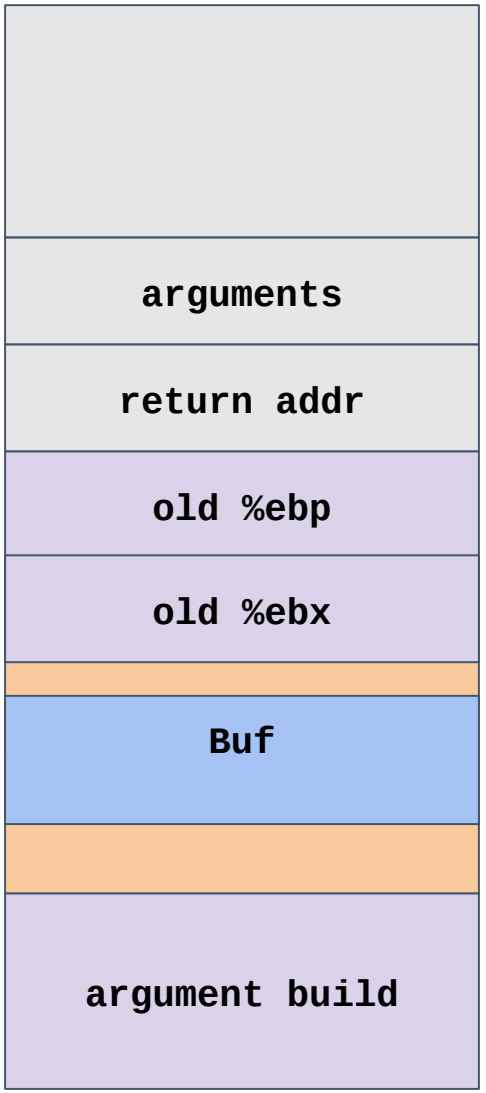
IA32/Linux Stack Frame



Buffer Overflow

```
int getbuf()
{
    char buf[SIZE];
    Gets(buf);
    return 1;
}
```

write data to buf



Buffer Overflow (cont'd)

```
int getbuf()
{
    char buf[SIZE];
    Gets(buf);
    return 1;
}
```

write data to buf

Buffer Overflow (cont'd)

```
int getbuf()
{
    char buf[SIZE];
    Gets(buf);
    return 1;
}
```

write data to buf

Buffer Overflow (cont'd)

```
int getbuf()
{
    char buf[SIZE];
    Gets(buf);
    return 1;
}
```

Level 0: Candle, Let's Make an Exploit String

Your job is call `smoke` by exploiting the buffer overflow attack!

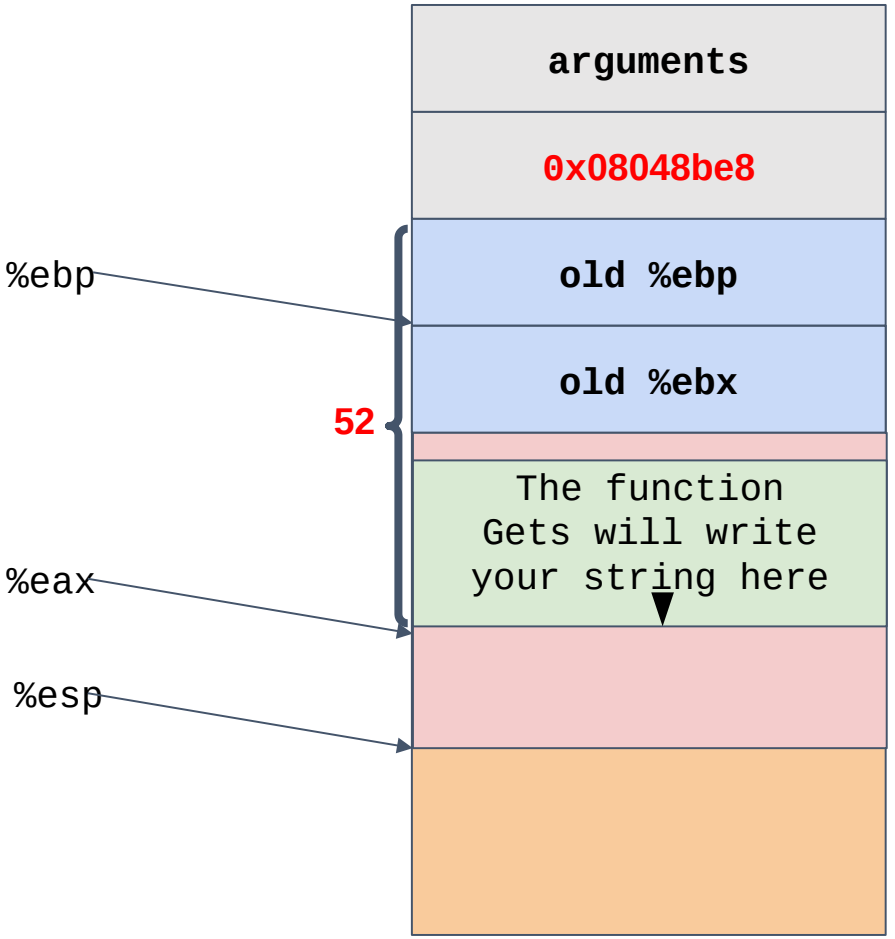
```
08048be8 <smoke>:
 55                      push   %ebp
 89 e5                   mov    %esp,%ebp
 83 ec 18                sub    $0x18,%esp
 c7 04 24 ff a2 04 08    movl   $0x804a2ff,(%esp)
 e8 76 fc ff ff          call   8048870 <puts@plt>
 c7 04 24 00 00 00 00    movl   $0x0,(%esp)
 e8 7e 06 00 00          call   8049284 <validate>
 c7 04 24 00 00 00 00    movl   $0x0,(%esp)
 e8 8e fc ff ff          call   80488a0 <exit@plt>
```

Level 0: Candle, Let's Make an Exploit String

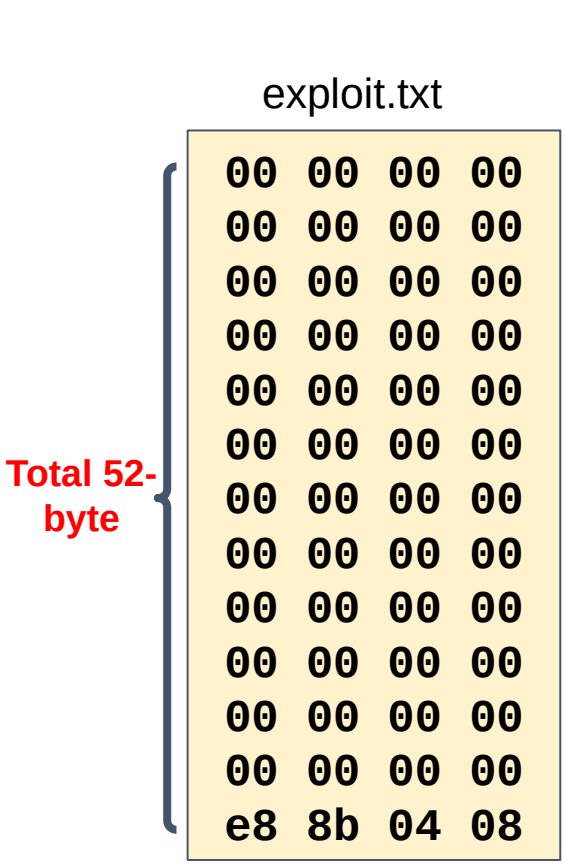
Your job is call `smoke` by exploiting the buffer overflow attack!

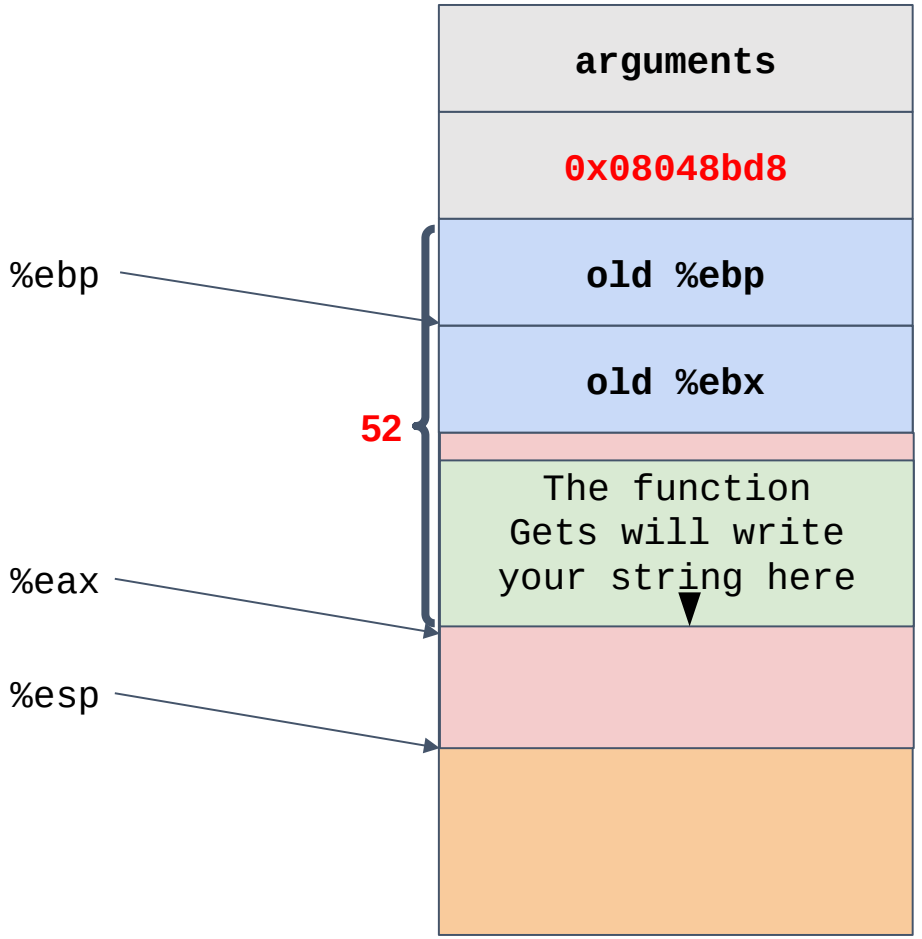
08048be8 <smoke>:

55	push	%ebp
89 e5	mov	%esp,%ebp
83 ec 18	sub	$0x18,%esp
c7 04 24 ff a2 04 08	movl	$0x804a2ff,(%esp)
e8 76 fc ff ff	call	8048870 <puts@plt>
c7 04 24 00 00 00 00	movl	$0x0,(%esp)
e8 7e 06 00 00	call	8049284 <validate>
c7 04 24 00 00 00 00	movl	$0x0,(%esp)
e8 8e fc ff ff	call	80488a0 <exit@plt>



Level 0: Candle, The Exploit String



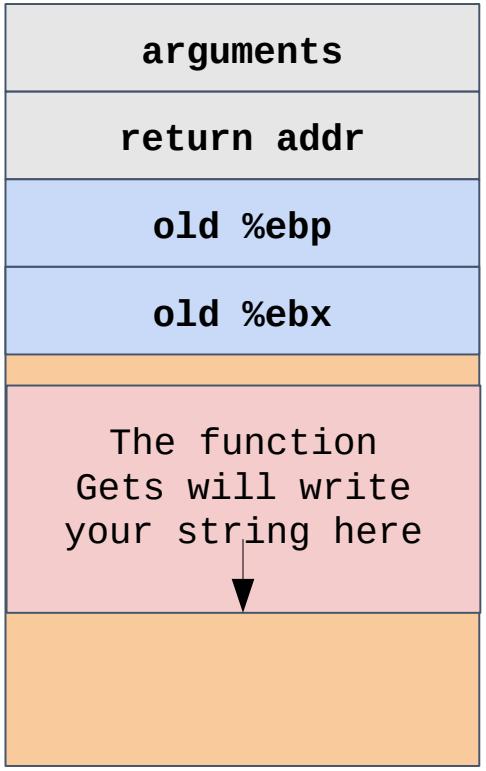


Level 0: Candle, Do the Attack

```
➜ buflab-handout cat explot.txt | ./hex2raw | ./bufbomb -u 2017-111111
Userid: 2017-111111
Cookie: 0x23975c80
Type string:Smoke!: You called smoke()
VALID
NICE JOB!
```

Level 1: Sparkling

- You need to pass the arguments with proper values to the function **`fizz`**
- Do it by yourself! Good Luck!



Don’t forget

- Start early
- Study and follow both the `Git` and `Bufferlab` slides.

- If you have questions or problems please contact the TAs (**sysprog@csap.snu.ac.kr**) - answer in one working day (no holidays or weekends)

- **You should use the VM provided to do this and the other labs. Otherwise, strange errors can happen.**
  - Submitting to the server with -s argument
  - Write a report using the report template and add it in your git repository under the right section.

Buflab Tutorial

`getbuf` details

Caller Function

```
void test()
{
    int val;
    /* Put canary on stack to detect possible corruption */
    volatile int local = uniqueval();
    getbuf(&val);
    /* check for corrupted stack */
    if (local != uniqueval()) {
        printf("Sabotaged!: the stack has been corrupted\n");
    else if (val == cookie) {
        printf("Boom!: getbuf returned 0x%x\n", val);
        validate(3);
    }
    else {
        printf("Dud: getbuf returned 0x%x\n", val);
    }
}
```

getbuf Function

```
int getbuf(int *val)
{
    char buf[40];
    Gets(buf);
    if (val != NULL)
        *val = 1;
}
```

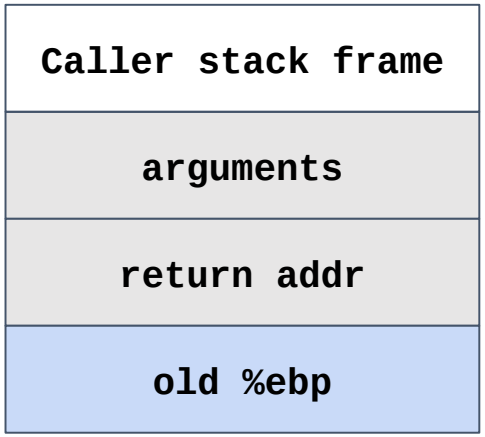
```
08048d7c <getbuf>:
55                      push   %ebp
89 e5                   mov    %esp,%ebp
53                      push   %ebx
83 ec 44                sub    $0x44,%esp
8b 5d 08                mov    0x8(%ebp),%ebx
8d 45 d0                lea    -0x30(%ebp),%eax
89 04 24                mov    %eax,(%esp)
e8 55 ff ff ff          call   8048ce6 <Gets>
85 db                   test   %ebx,%ebx
74 06                   je     8048d9b <getbuf+0x1f>
c7 03 01 00 00 00       movl   $0x1,(%ebx)
83 c4 44                add    $0x44,%esp
5b                      pop    %ebx
5d                      pop    %ebp
c3                      ret
```

Execute the following command in your terminal to get the disassembled code

```
$ objdump -d bufbomb > bufbomb.disas
```

Stack Structure of `getbuf`

```
08048d7c <getbuf>:
 55                   push %ebp
 89 e5                mov  %esp,%ebp
 53                   push %ebx
 83 ec 44             sub  $0x44,%esp
 8b 5d 08             mov  0x8(%ebp),%ebx
 8d 45 d0             lea  -0x30(%ebp),%eax
 89 04 24             mov  %eax,(%esp)
 e8 55 ff ff ff       call 8048ce6 <Gets>
 85 db                test %ebx,%ebx
 74 06                je   8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl $0x1,(%ebx)
 83 c4 44             add  $0x44,%esp
 5b                   pop  %ebx
 5d                   pop  %ebp
 c3                   ret
```

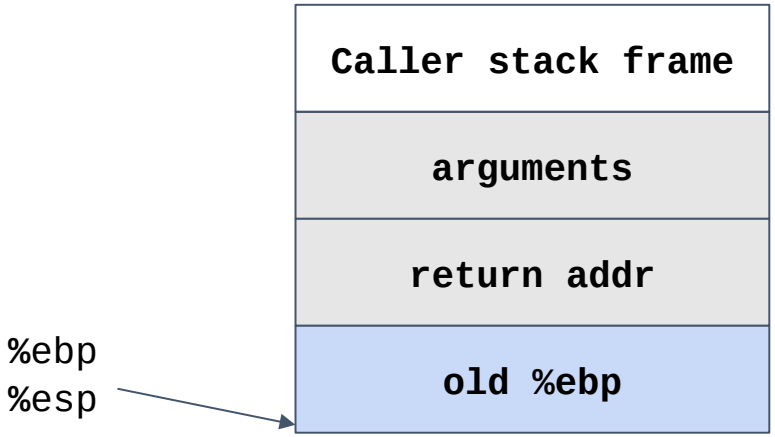



```
void getbuf(int *val)
{
    char buf[NORMAL_BUFFER_SIZE];
    Gets(buf);
    if (val != NULL)
        *val = 1;
}
```

Hex	→	Dec
0x44	→	68
0x30	→	48

Stack Structure of `getbuf`

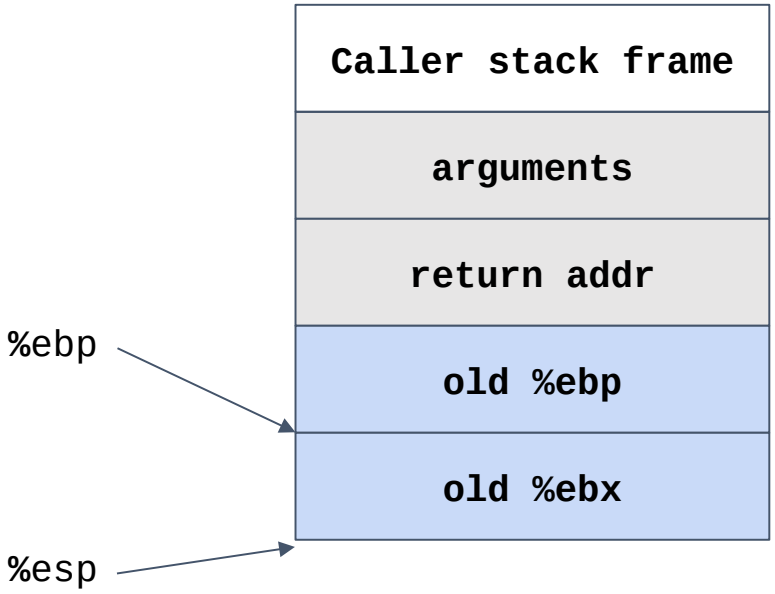
```
08048d7c <getbuf>:
 55                   push  %ebp
 89 e5                mov   %esp,%ebp
 53                   push  %ebx
 83 ec 44             sub   $0x44,%esp
 8b 5d 08             mov   0x8(%ebp),%ebx
 8d 45 d0             lea   -0x30(%ebp),%eax
 89 04 24             mov   %eax,(%esp)
 e8 55 ff ff ff       call  8048ce6 <Gets>
 85 db                test  %ebx,%ebx
 74 06                je    8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl  $0x1,(%ebx)
 83 c4 44             add   $0x44,%esp
 5b                   pop   %ebx
 5d                   pop   %ebp
 c3                   ret
```



Hex	→	Dec
0x44	→	68
0x30	→	48

Stack Structure of `getbuf`

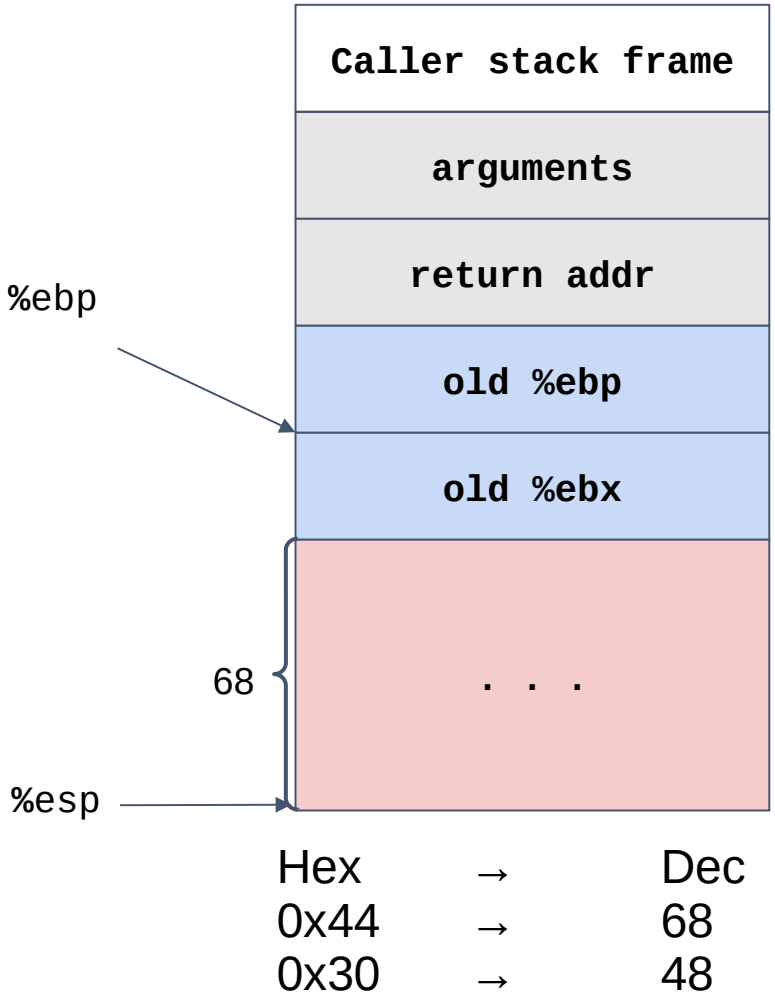
```
08048d7c <getbuf>:
 55                   push   %ebp
 89 e5                mov    %esp,%ebp
 53                   push   %ebx
 83 ec 44             sub    $0x44,%esp
 8b 5d 08             mov    0x8(%ebp),%ebx
 8d 45 d0             lea    -0x30(%ebp),%eax
 89 04 24             mov    %eax,(%esp)
 e8 55 ff ff ff       call   8048ce6 <Gets>
 85 db                test   %ebx,%ebx
 74 06                je     8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl   $0x1,(%ebx)
 83 c4 44             add    $0x44,%esp
 5b                   pop    %ebx
 5d                   pop    %ebp
 c3                   ret
```



Hex	→	Dec
0x44	→	68
0x30	→	48

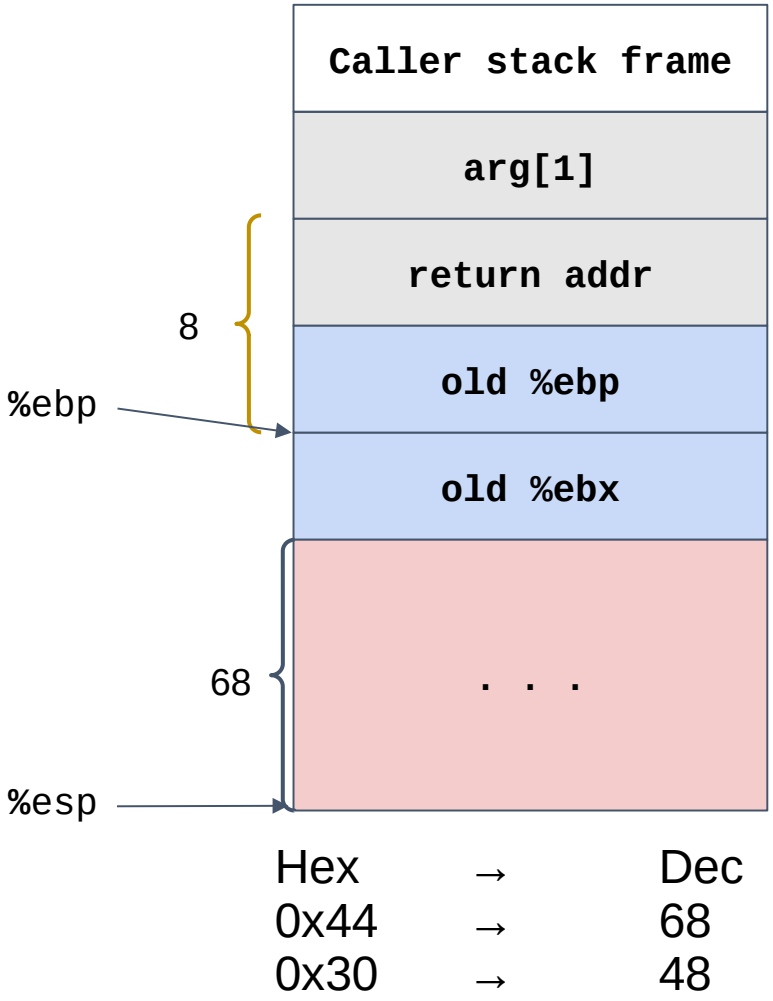
Stack Structure of `getbuf`

```
08048d7c <getbuf>:
 55                   push   %ebp
 89 e5                mov    %esp,%ebp
 53                   push   %ebx
 83 ec 44             sub    $0x44,%esp
 8b 5d 08             mov    0x8(%ebp),%ebx
 8d 45 d0             lea    -0x30(%ebp),%eax
 89 04 24             mov    %eax,(%esp)
 e8 55 ff ff ff       call   8048ce6 <Gets>
 85 db                test   %ebx,%ebx
 74 06                je     8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl   $0x1,(%ebx)
 83 c4 44             add    $0x44,%esp
 5b                   pop    %ebx
 5d                   pop    %ebp
 c3                   ret
```



Stack Structure of `getbuf`

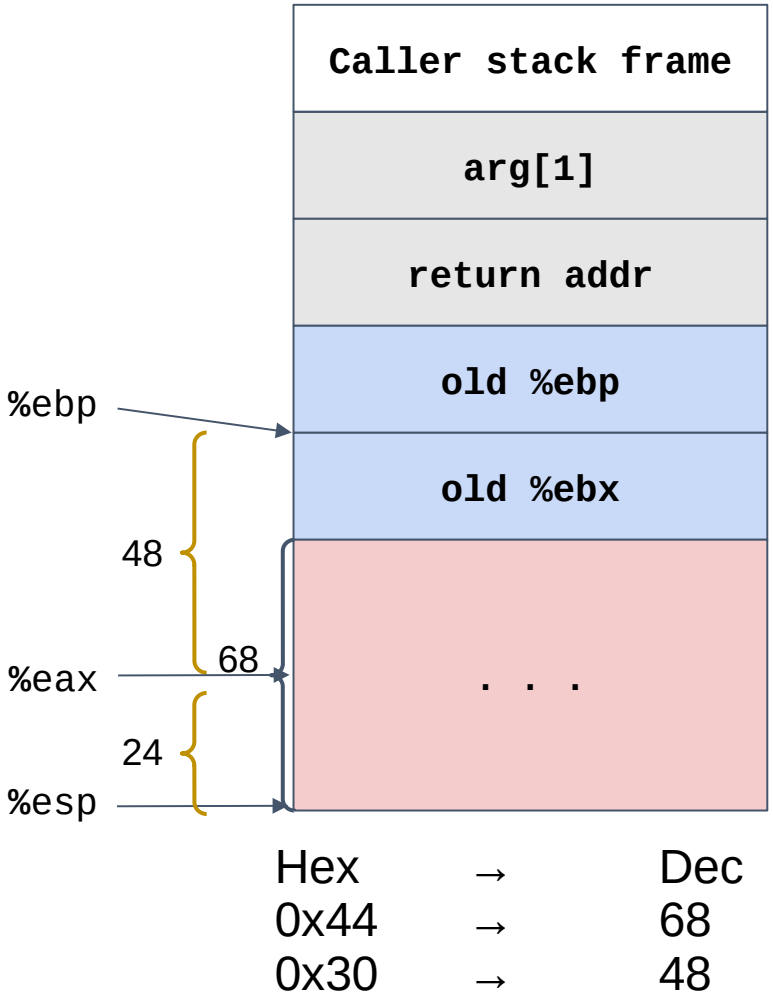
```
08048d7c <getbuf>:
 55                   push   %ebp
 89 e5                mov    %esp,%ebp
 53                   push   %ebx
 83 ec 44             sub    $0x44,%esp
 8b 5d 08             mov    0x8(%ebp),%ebx
 8d 45 d0             lea    -0x30(%ebp),%eax
 89 04 24             mov    %eax,(%esp)
 e8 55 ff ff ff       call   8048ce6 <Gets>
 85 db                test   %ebx,%ebx
 74 06                je     8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl   $0x1,(%ebx)
 83 c4 44             add    $0x44,%esp
 5b                   pop    %ebx
 5d                   pop    %ebp
 c3                   ret
```



%ebx	arg[1]

Stack Structure of `getbuf`

```
08048d7c <getbuf>:
 55                   push   %ebp
 89 e5                mov    %esp,%ebp
 53                   push   %ebx
 83 ec 44             sub    $0x44,%esp
 8b 5d 08             mov    0x8(%ebp),%ebx
 8d 45 d0             lea    -0x30(%ebp),%eax
 89 04 24             mov    %eax,(%esp)
 e8 55 ff ff ff       call   8048ce6 <Gets>
 85 db                test   %ebx,%ebx
 74 06                je     8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl   $0x1,(%ebx)
 83 c4 44             add    $0x44,%esp
 5b                   pop    %ebx
 5d                   pop    %ebp
 c3                   ret
```

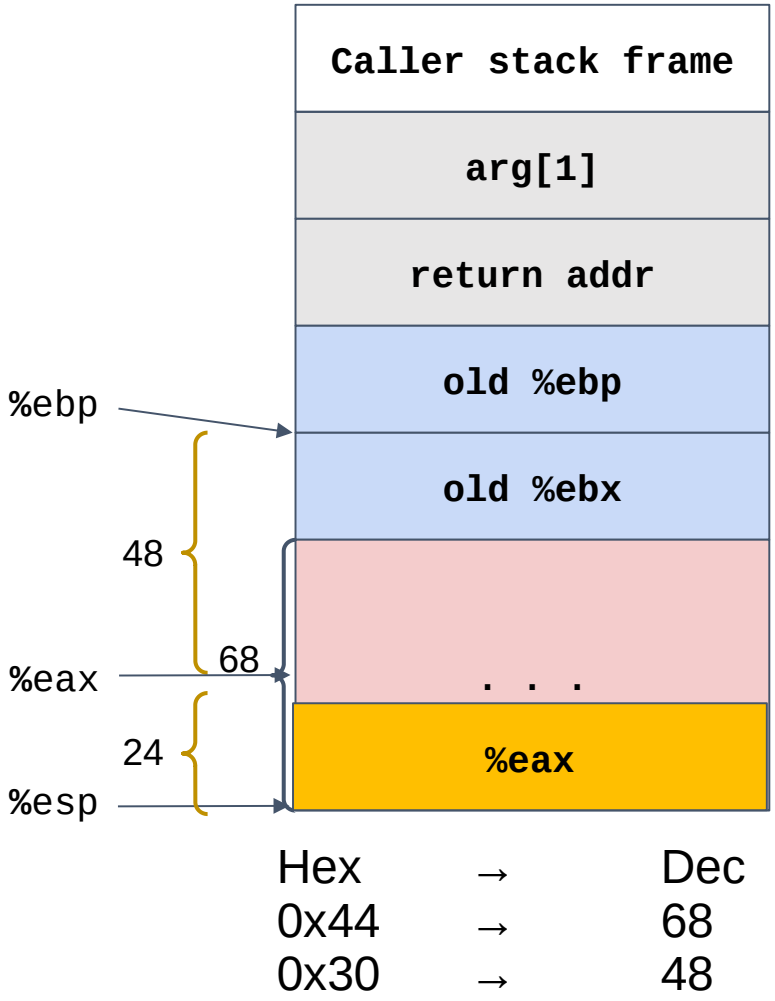


%ebx	arg[1]

%eax	ebp+48

Stack Structure of `getbuf`

```
08048d7c <getbuf>:
 55                   push  %ebp
 89 e5                mov   %esp,%ebp
 53                   push  %ebx
 83 ec 44             sub   $0x44,%esp
 8b 5d 08             mov   0x8(%ebp),%ebx
 8d 45 d0             lea   -0x30(%ebp),%eax
 89 04 24             mov   %eax,(%esp)
 e8 55 ff ff ff       call  8048ce6 <Gets>
 85 db                test  %ebx,%ebx
 74 06                je    8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl  $0x1,(%ebx)
 83 c4 44             add   $0x44,%esp
 5b                   pop   %ebx
 5d                   pop   %ebp
 c3                   ret
```

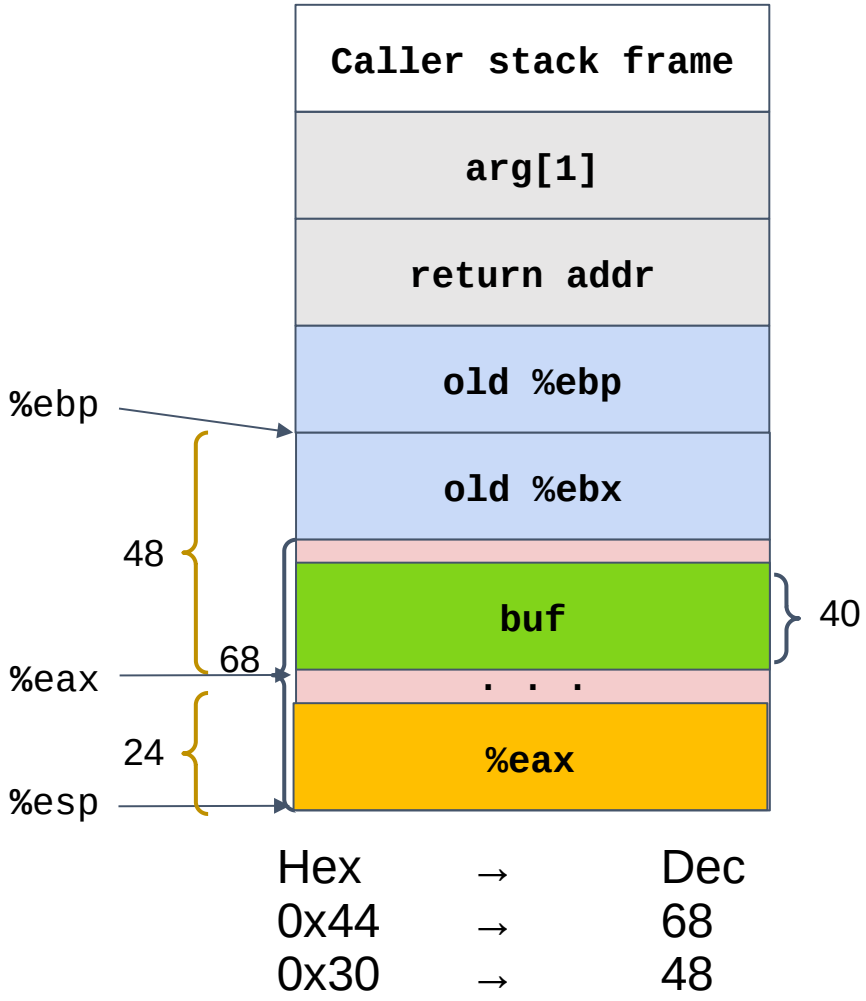


%ebx	arg[1]

%eax	ebp+48

Stack Structure of `getbuf`

```
08048d7c <getbuf>:
 55                   push   %ebp
 89 e5                mov    %esp,%ebp
 53                   push   %ebx
 83 ec 44             sub    $0x44,%esp
 8b 5d 08             mov    0x8(%ebp),%ebx
 8d 45 d0             lea    -0x30(%ebp),%eax
 89 04 24             mov    %eax,(%esp)
 e8 55 ff ff ff       call   8048ce6 <Gets>
 85 db                test   %ebx,%ebx
 74 06                je     8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl   $0x1,(%ebx)
 83 c4 44             add    $0x44,%esp
 5b                   pop    %ebx
 5d                   pop    %ebp
 c3                   ret
```

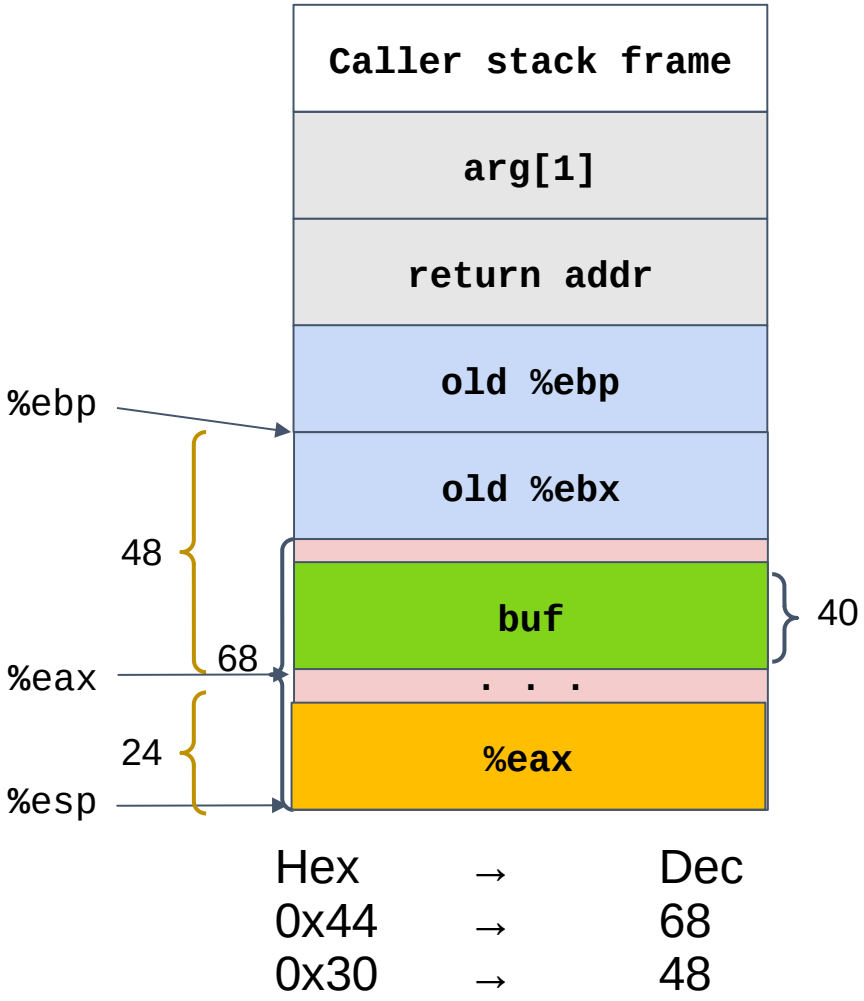


%ebx	arg[1]

%eax	ebp+48

Stack Structure of `getbuf`

```
08048d7c <getbuf>:
 55                  push   %ebp
 89 e5               mov    %esp,%ebp
 53                  push   %ebx
 83 ec 44            sub    $0x44,%esp
 8b 5d 08            mov    0x8(%ebp),%ebx
 8d 45 d0            lea    -0x30(%ebp),%eax
 89 04 24            mov    %eax,(%esp)
 e8 55 ff ff ff      call   8048ce6 <Gets>
 85 db               test   %ebx,%ebx
 74 06               je     8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00   movl   $0x1,(%ebx)
 83 c4 44            add    $0x44,%esp
 5b                  pop    %ebx
 5d                  pop    %ebp
 c3                  ret
```



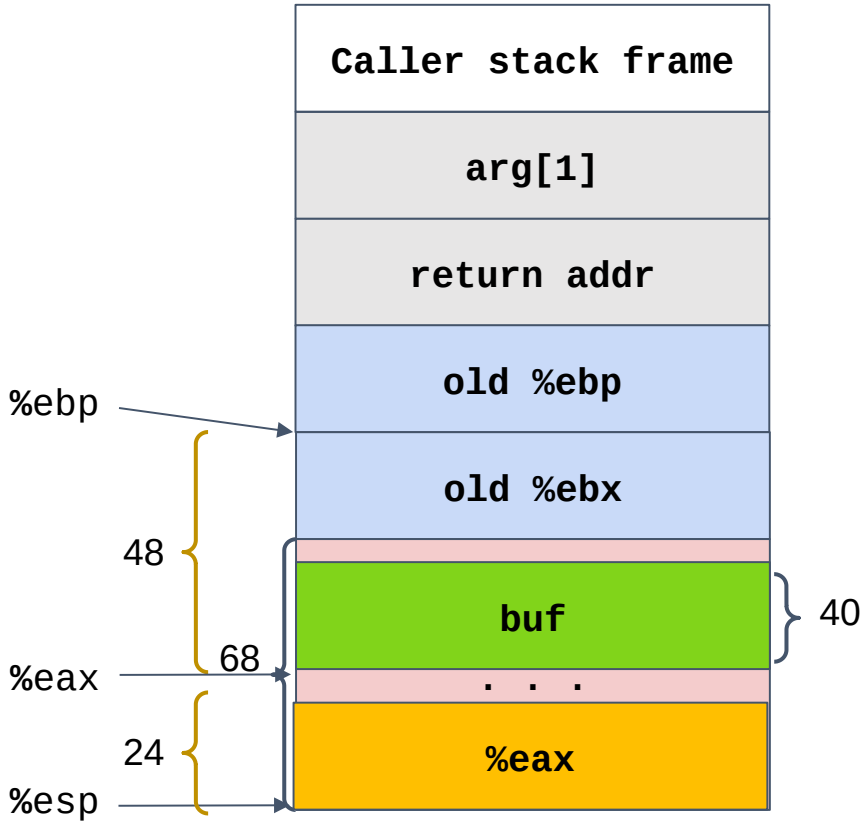
%ebx	arg[1]

%eax	ebp+48

Stack Structure of getbuf

If ebx != NULL

```
08048d7c <getbuf>:
 55                 push   %ebp
 89 e5              mov    %esp,%ebp
 53                 push   %ebx
 83 ec 44           sub    $0x44,%esp
 8b 5d 08           mov    0x8(%ebp),%ebx
 8d 45 d0           lea    -0x30(%ebp),%eax
 89 04 24           mov    %eax,(%esp)
 e8 55 ff ff ff     call   8048ce6 <Gets>
 85 db              test   %ebx,%ebx
 74 06              je     8048d9b <getbuf+0x1f>
 c7 03 01 00        movl   $0x1,(%ebx)
 83 c4 44           add    $0x44,%esp
 5b                 pop    %ebx
 5d                 pop    %ebp
 c3                 ret
```



Hex	→	Dec
0x44	→	68
0x30	→	48

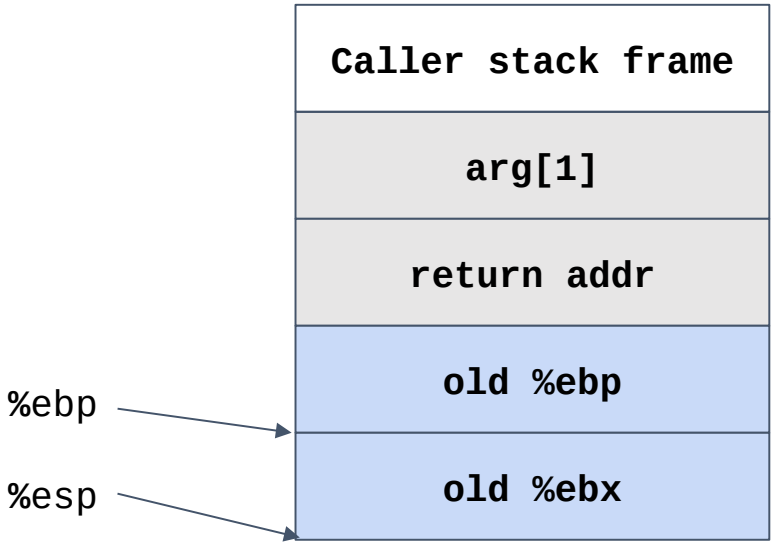
%ebx	arg[1]

%eax	ebp+48

arg[1]	1

Stack Structure of `getbuf`

```
08048d7c <getbuf>:
 55                  push  %ebp
 89 e5               mov   %esp,%ebp
 53                  push  %ebx
 83 ec 44            sub   $0x44,%esp
 8b 5d 08            mov   0x8(%ebp),%ebx
 8d 45 d0            lea   -0x30(%ebp),%eax
 89 04 24            mov   %eax,(%esp)
 e8 55 ff ff ff      call  8048ce6 <Gets>
 85 db               test  %ebx,%ebx
 74 06               je    8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00   movl  $0x1,(%ebx)
 83 c4 44            add   $0x44,%esp
 5b                  pop   %ebx
 5d                  pop   %ebp
 c3                  ret
```



Hex	→	Dec
0x44	→	68
0x30	→	48

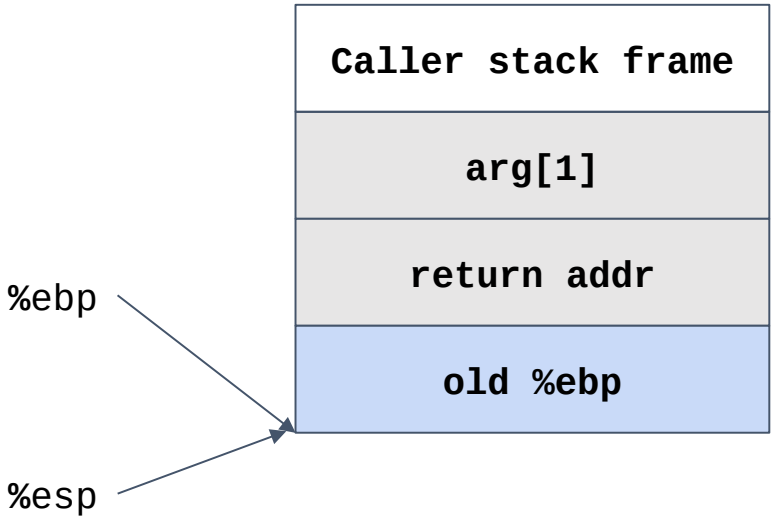
%ebx	arg[1]

%eax	ebp+48

arg[1]	1

Stack Structure of `getbuf`

```
08048d7c <getbuf>:
 55                   push   %ebp
 89 e5                mov    %esp,%ebp
 53                   push   %ebx
 83 ec 44             sub    $0x44,%esp
 8b 5d 08             mov    0x8(%ebp),%ebx
 8d 45 d0             lea    -0x30(%ebp),%eax
 89 04 24             mov    %eax,(%esp)
 e8 55 ff ff ff       call   8048ce6 <Gets>
 85 db                test   %ebx,%ebx
 74 06                je     8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl   $0x1,(%ebx)
 83 c4 44             add    $0x44,%esp
 5b                   pop    %ebx
 5d                   pop    %ebp
 c3                   ret
```



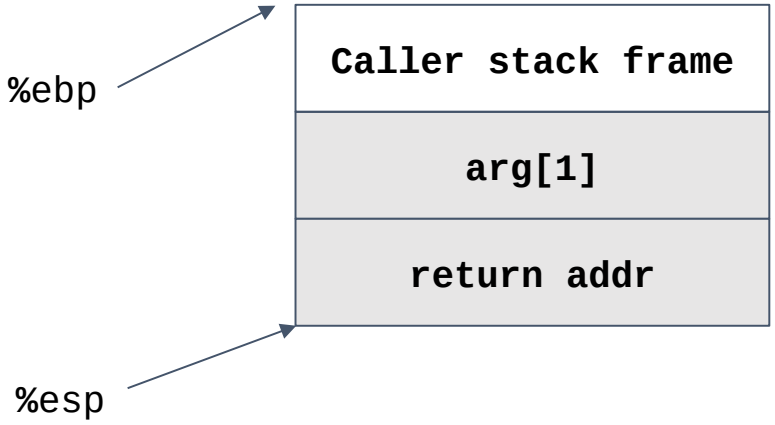
Hex	→	Dec
0x44	→	68
0x30	→	48

%ebx	Old ebx

%eax	ebp+48

arg[1]	1

Stack Structure of `getbuf`




```
08048d7c <getbuf>:
 55                   push   %ebp
 89 e5                mov    %esp,%ebp
 53                   push   %ebx
 83 ec 44             sub    $0x44,%esp
 8b 5d 08             mov    0x8(%ebp),%ebx
 8d 45 d0             lea    -0x30(%ebp),%eax
 89 04 24             mov    %eax,(%esp)
 e8 55 ff ff ff       call   8048ce6 <Gets>
 85 db                test   %ebx,%ebx
 74 06                je     8048d9b <getbuf+0x1f>
 c7 03 01 00 00 00    movl   $0x1,(%ebx)
 83 c4 44             add    $0x44,%esp
 5b                   pop    %ebx
 5d                   pop    %ebp
 c3                   ret
```

Hex	→	Dec
0x44	→	68
0x30	→	48

%ebx	Old ebx

%eax	ebp+48

arg[1]	1