ELEC 377
Operating Systems

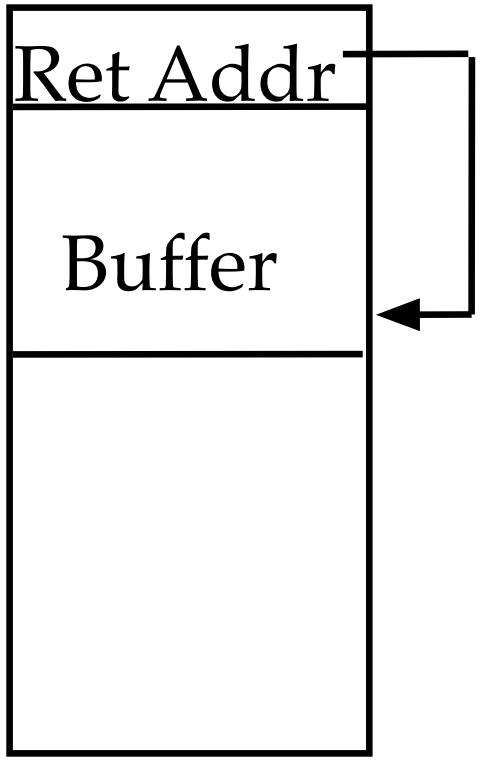
Week 11
Lab Tutorial

Program Threats

- Buffer Overflow
  - ◊ Most common attack
  - ◊ exploit bug as security hole

1. Write binary code into buffer, ending with a value that overwrites the return address and points into the buffer
2. Subroutine returns into the stack instead of to calling program

protection: don't allow stack space to be executable!! don't put buffers on the stack!!

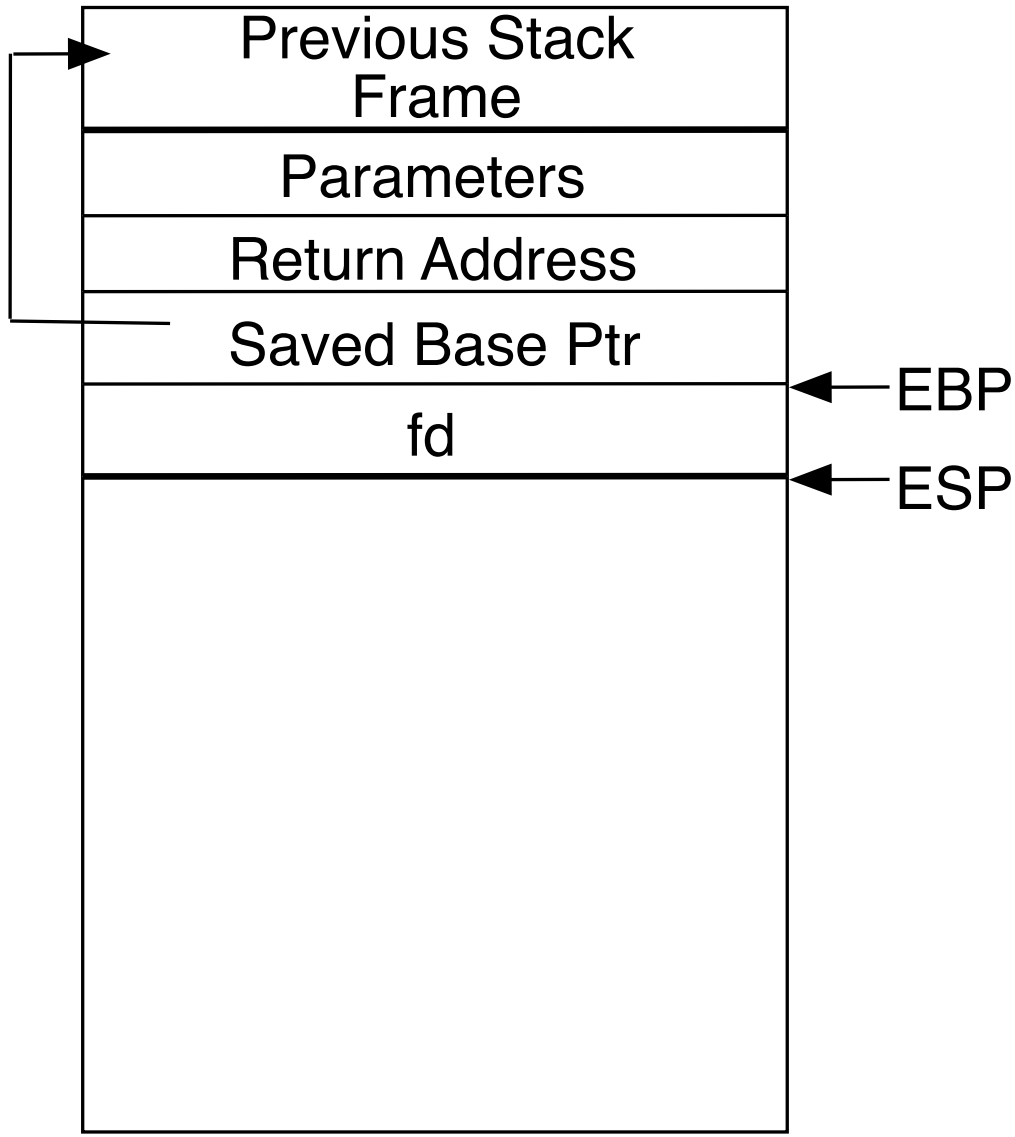


Pentium Stack Layout

```
fd = open(“theFile”, O_RDONLY, 0744);

push 0744
push O_RDONLY
pushd PtrToString

call open

mov [ebp-fd],eax
add esp,12
```

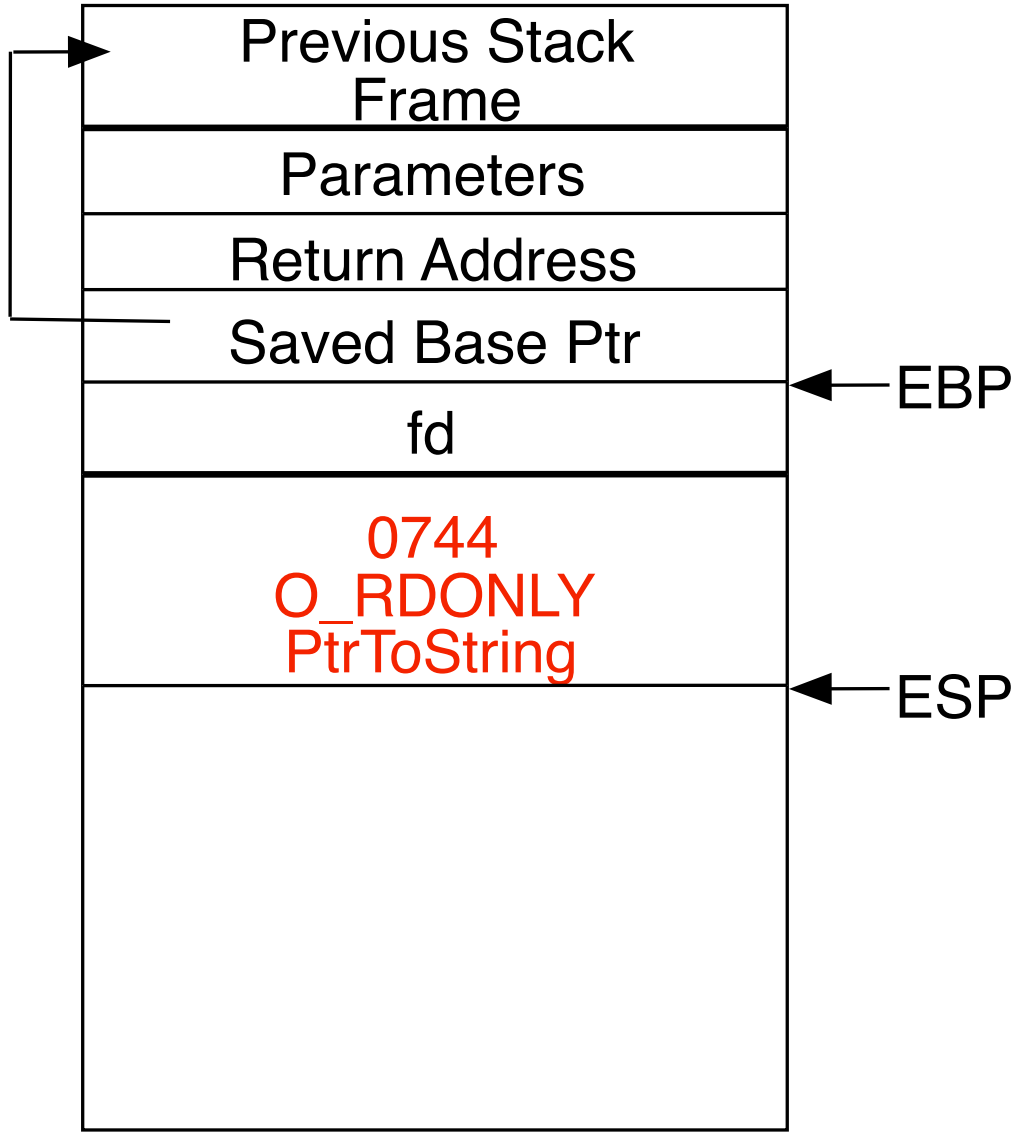
Pentium Stack Layout

```
push 0744
push O_RDONLY
pushd PtrToString

call open

mov [ebp-fd],eax
add esp,12
```



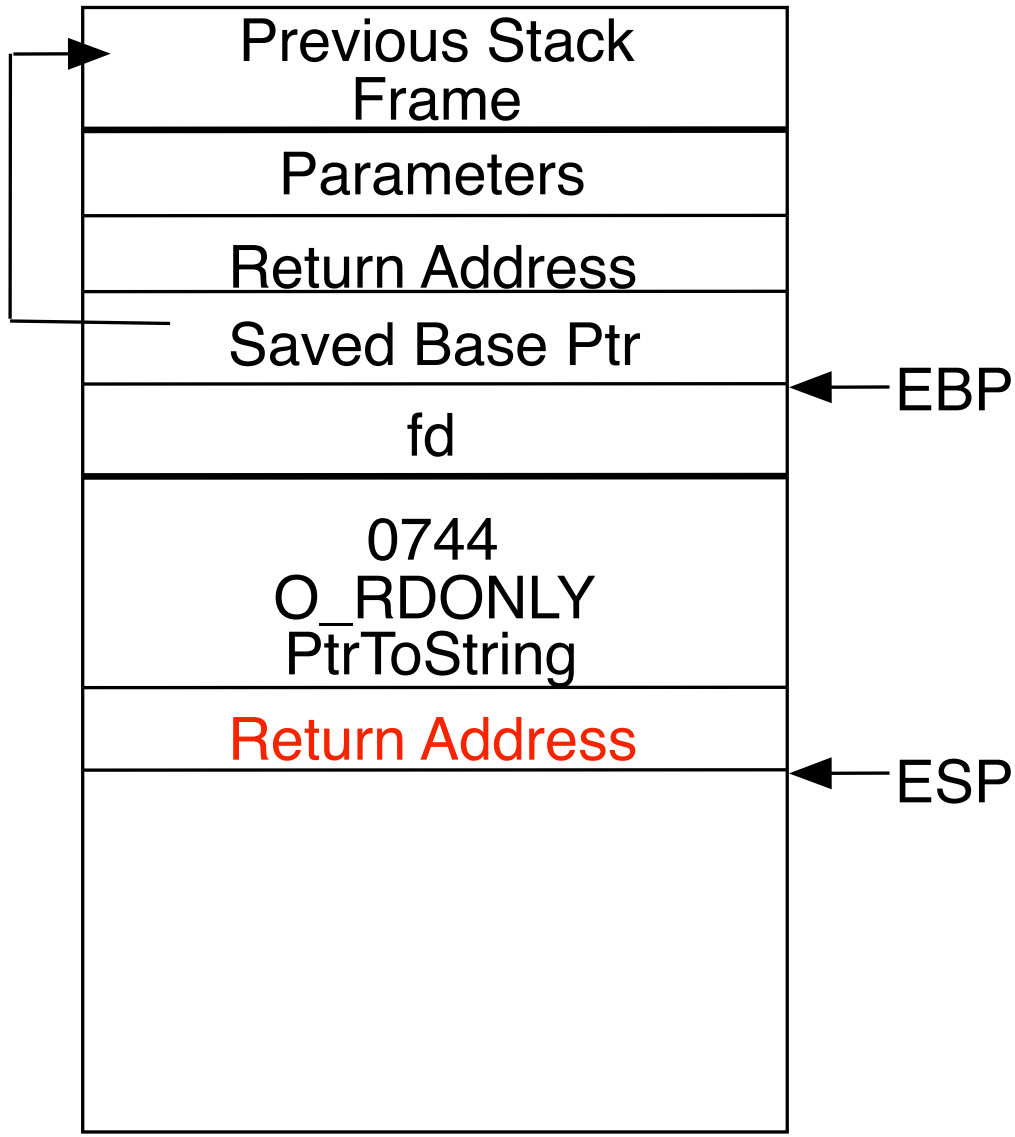
Pentium Stack Layout

```
push 0744
push O_RDONLY
pushd PtrToString

call open

mov [ebp-fd],eax
add esp,12
```



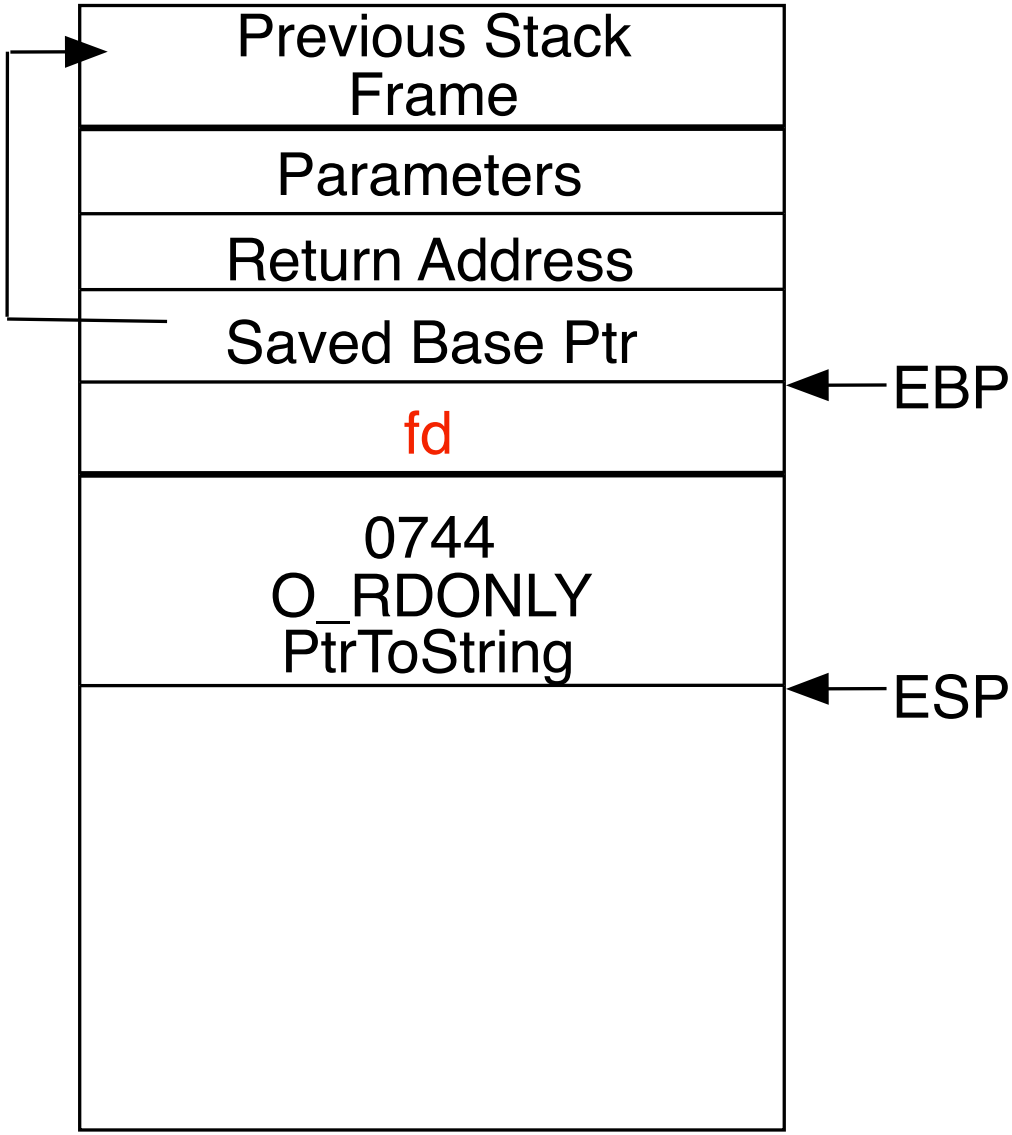
Pentium Stack Layout

```
push 0744
push O_RDONLY
pushd PtrToString

call open

mov [ebp-fd],eax
add esp,12
```



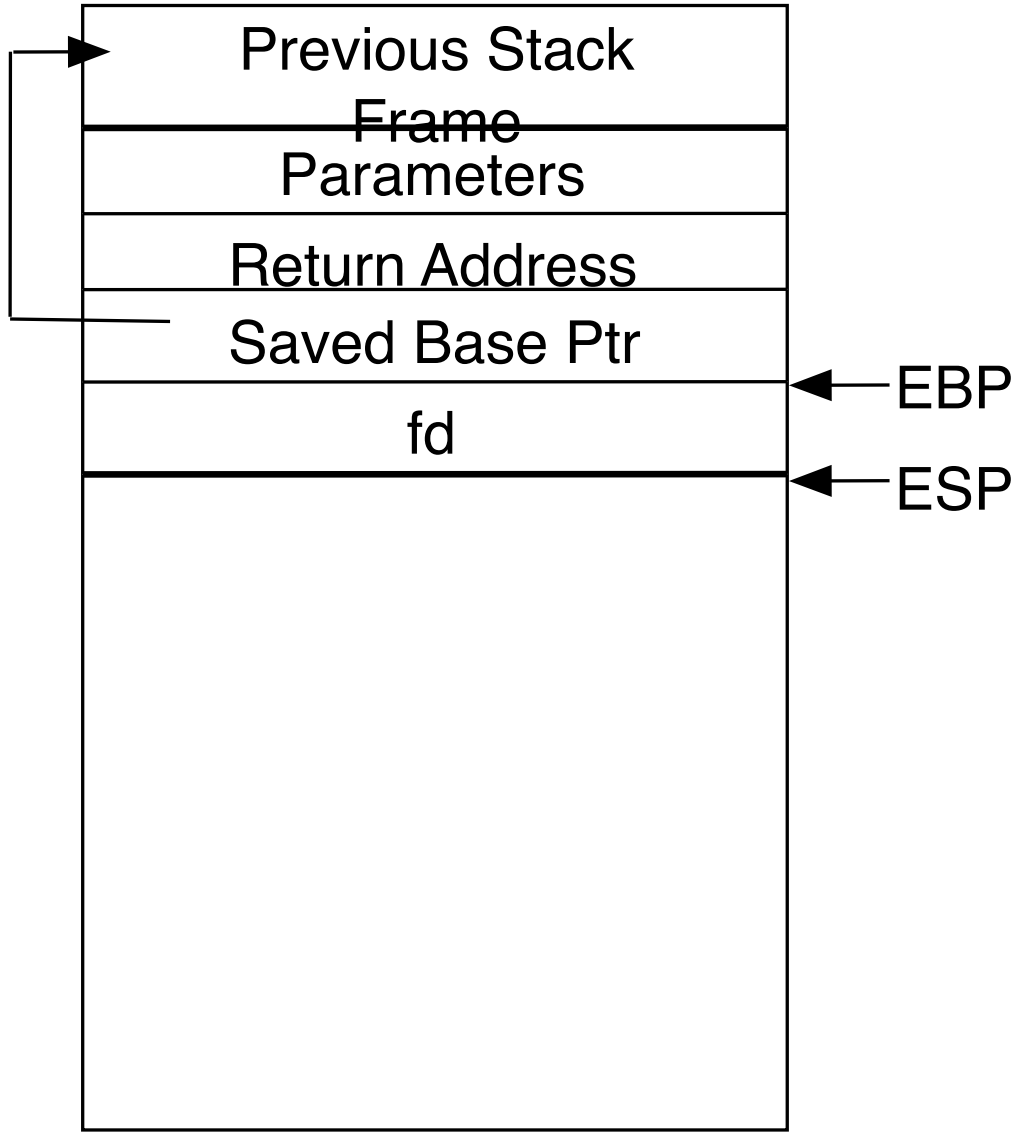
Pentium Stack Layout

```
push 0744
push O_RDONLY
pushd PtrToString

call open

mov [ebp-fd],eax
add esp,12
```



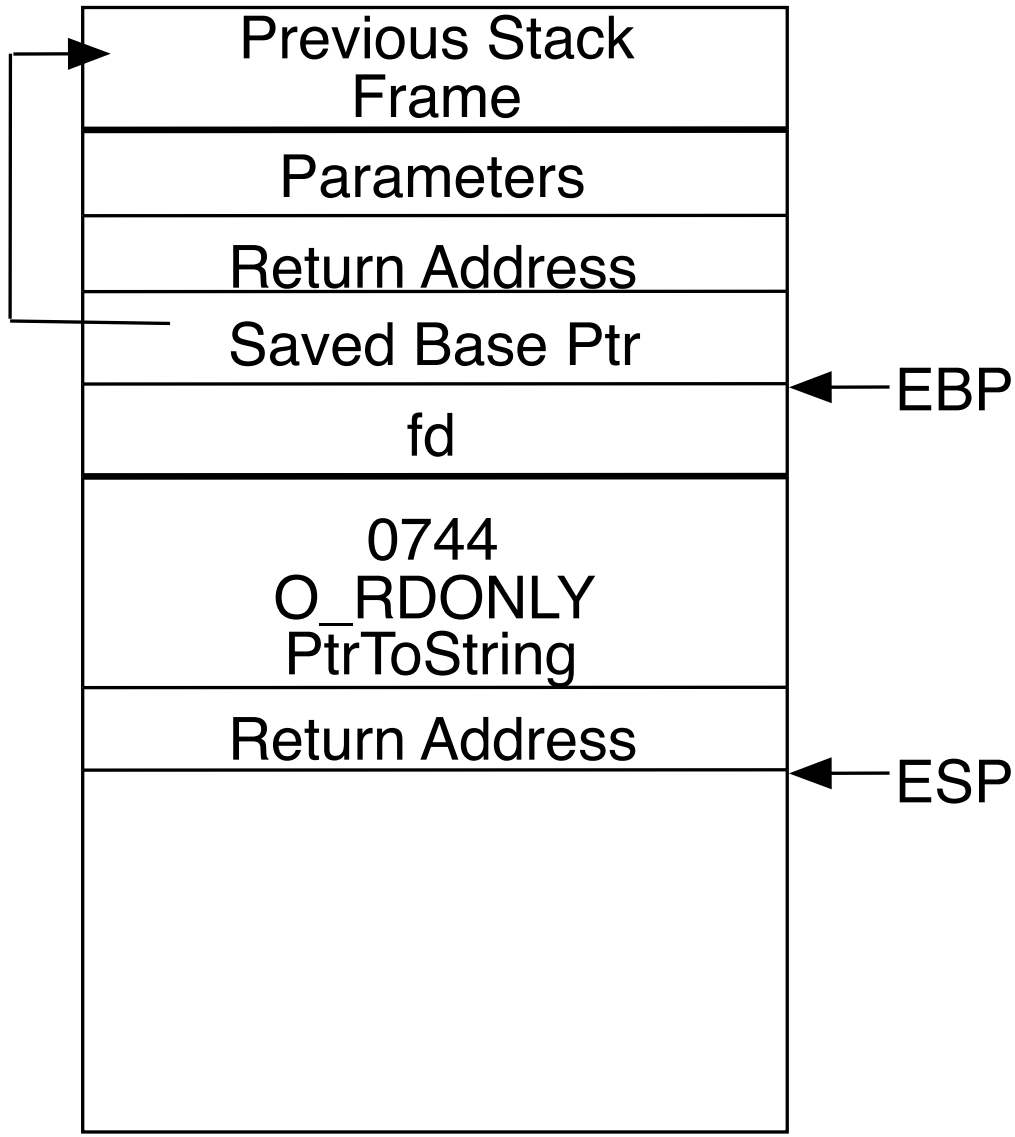
Pentium Stack Layout

```
push 0744
push O_RDONLY
pushd PtrToString

call open

mov [ebp-fd],eax
add esp,12
```



Pentium Stack Layout

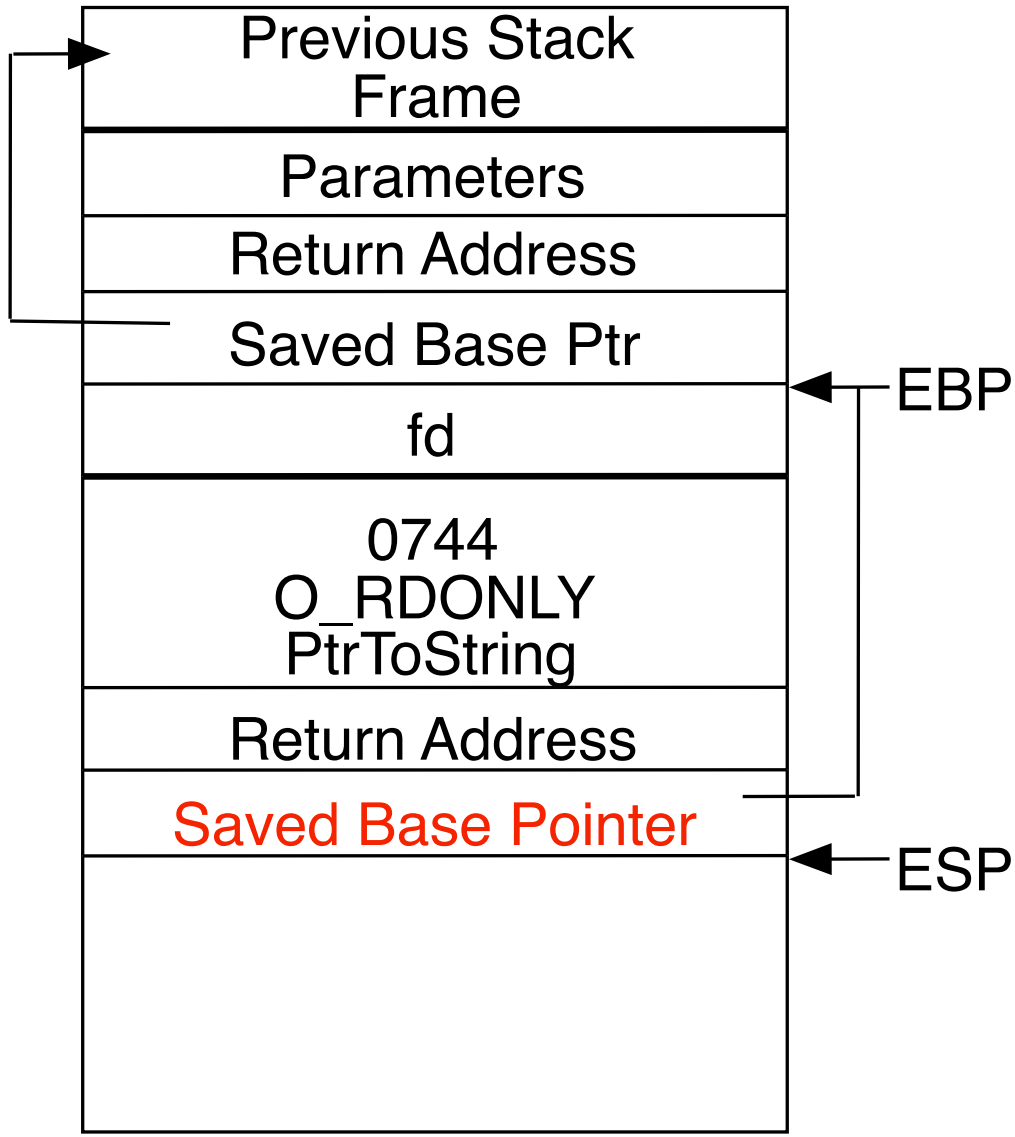
```
push ebp
mov ebp,esp
add esp,NumLocals
```




```
leave
ret
```

Pentium Stack Layout

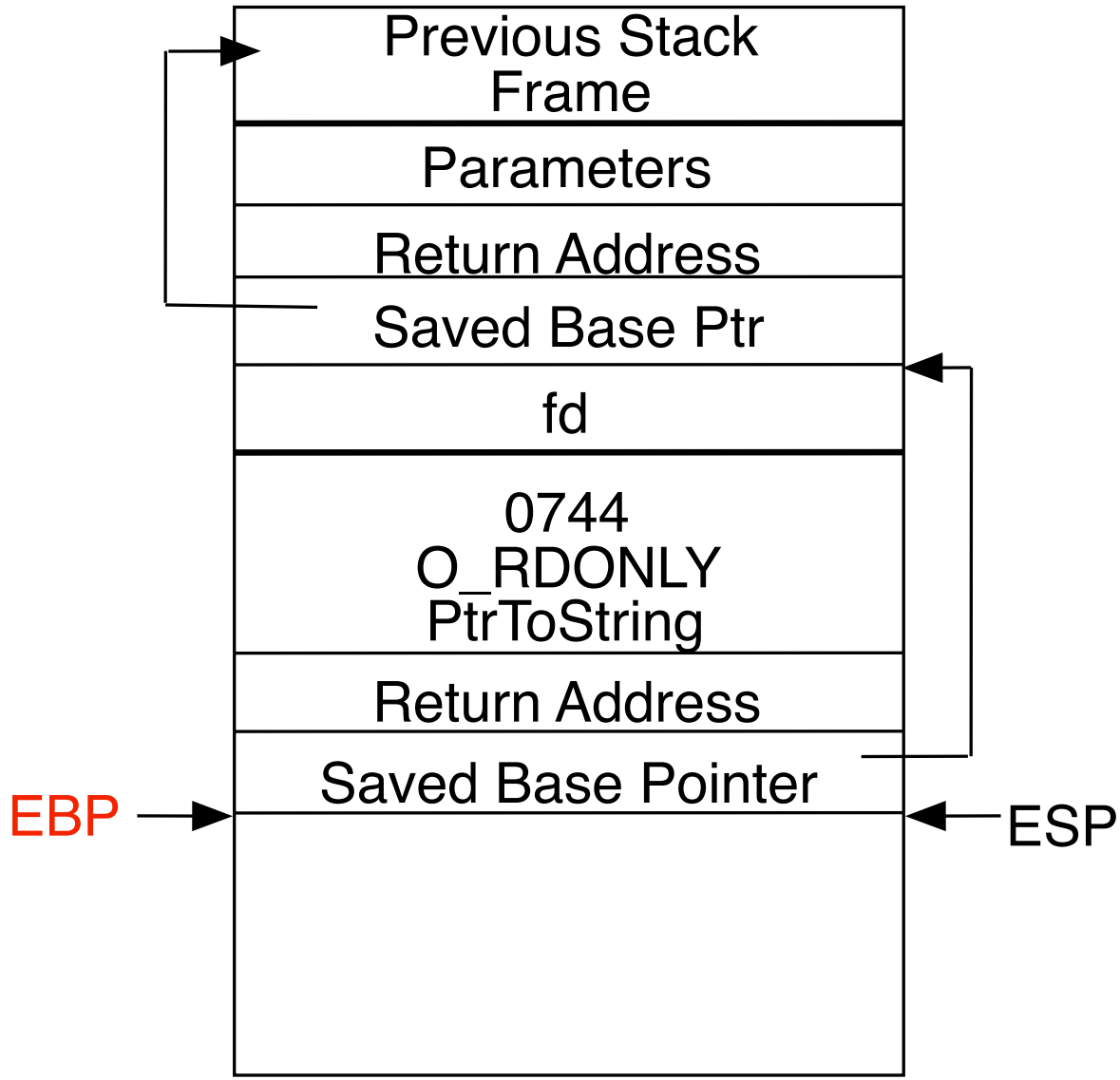
```
push ebp
mov ebp,esp
add esp,NumLocals
```




```
leave
ret
```

Pentium Stack Layout

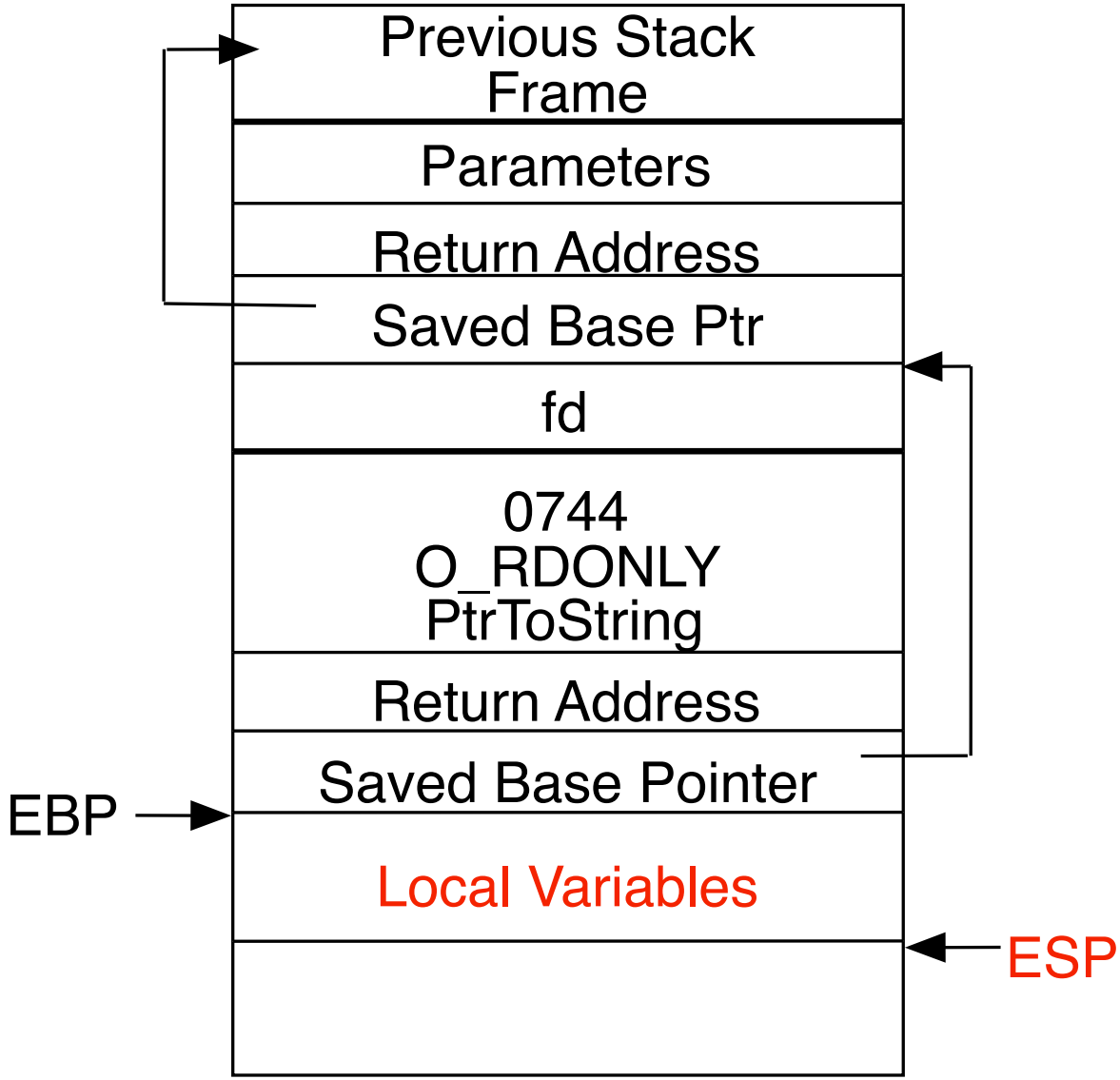
```
push ebp
mov ebp,esp
add esp,NumLocals
```




```
leave
ret
```

Pentium Stack Layout

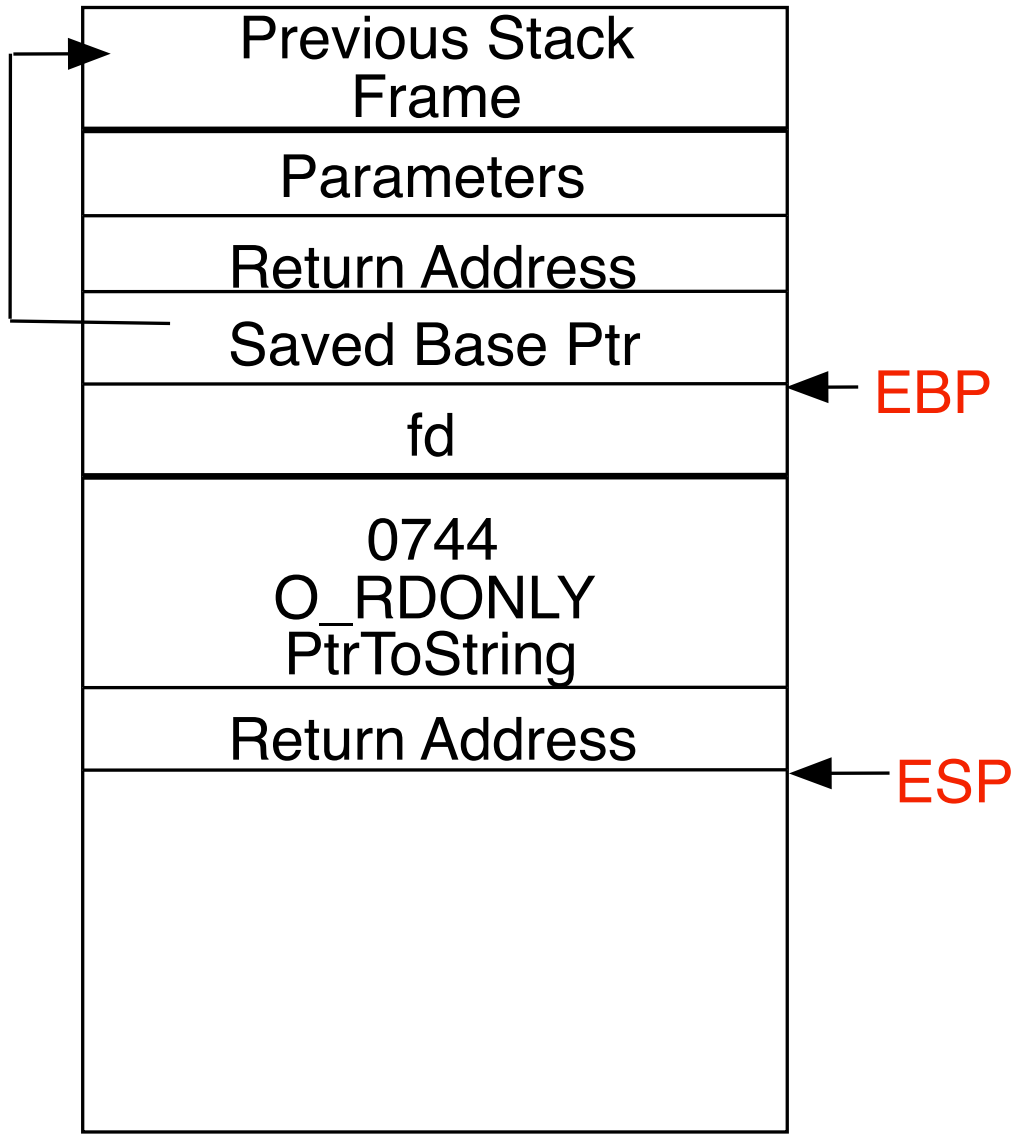
```
push ebp
mov ebp,esp
add esp,NumLocals
```




```
leave
ret
```

Pentium Stack Layout

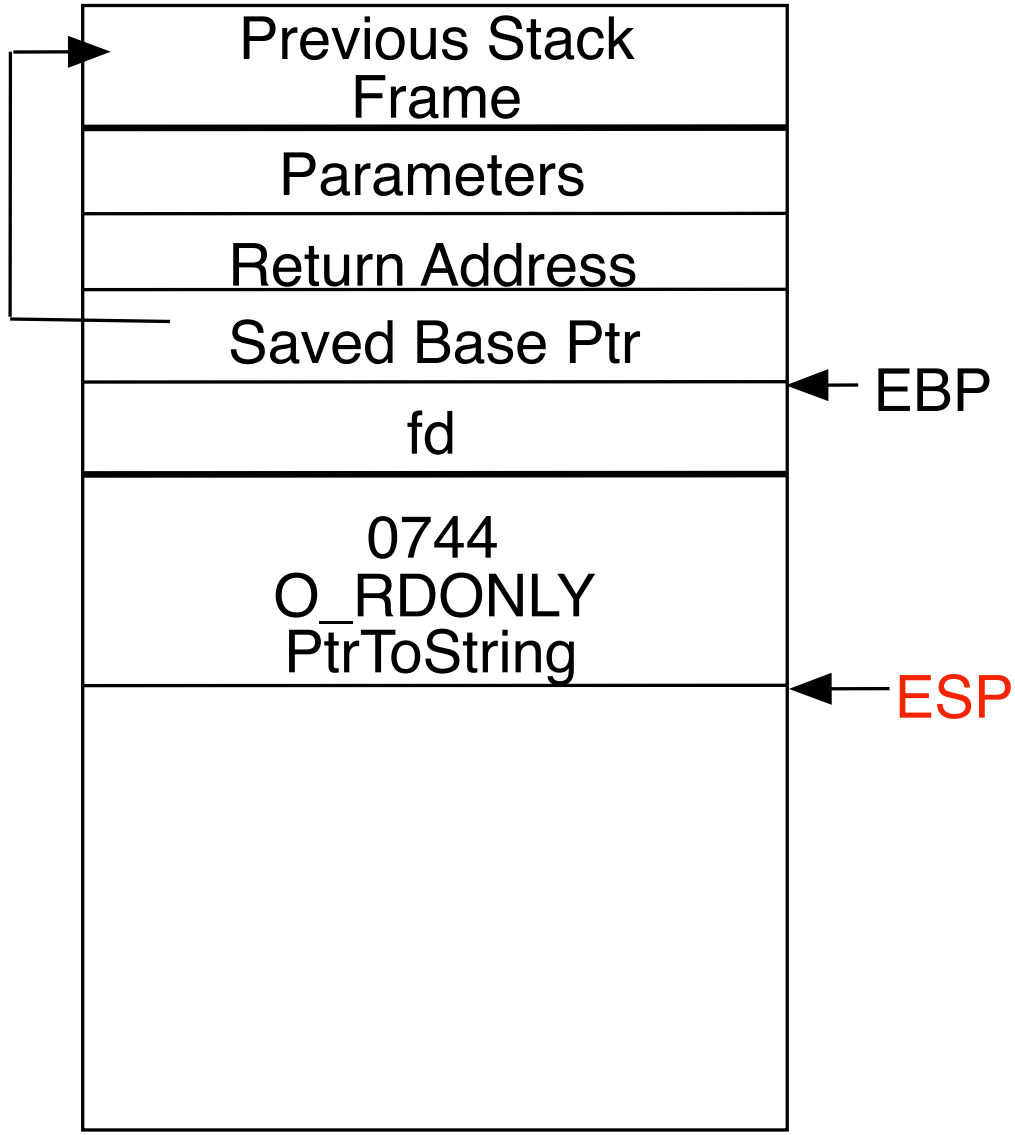
```
push ebp
mov ebp,esp
add esp,NumLocals
```




```
leave
ret
```

Pentium Stack Layout

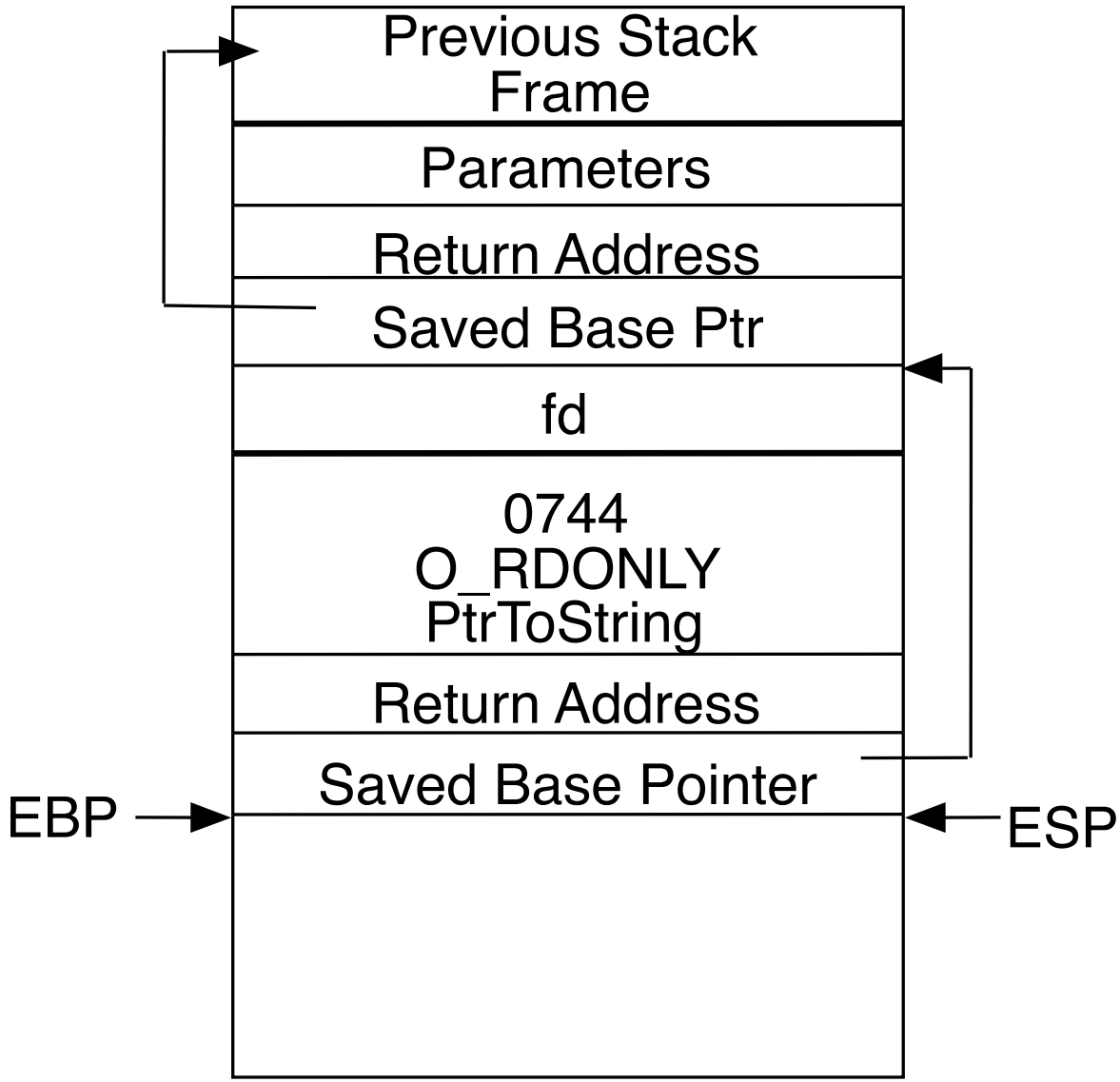
```
push ebp
mov ebp,esp
add esp,NumLocals
```




```
leave
ret
```

Open Function

```
push ebp
mov ebp,esp

mov eax,5
mov ebx,ebp+16
mov ecx,ebp+20
mov edx,ebp+24
int 0x80

leave
ret
```



Stack Overflow Attack

```
char * GetLine(){
    char buffer[130];
    gets(buffer);
}
```

```
getLine:
    push  ebp
    mov   ebp,esp
    sub   esp,152
    lea   eax,-152(ebp)
    pushl eax
    call  gets
    add   esp,4
    leave
    ret
```

Stack Overflow Attack

```
char * GetLine(){
    char buffer[130];
    gets(buffer);
}
```

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```

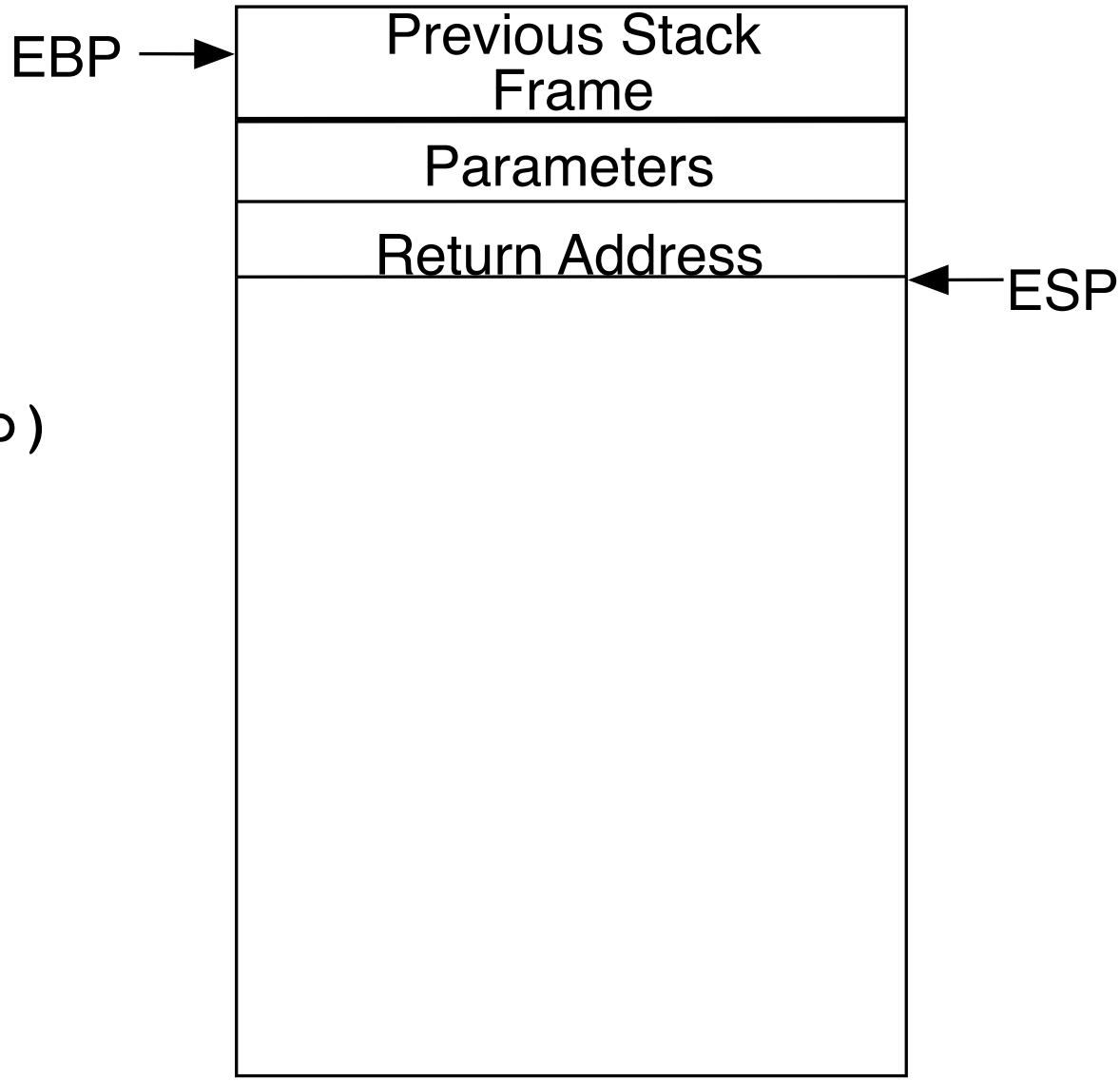
Stack Overflow Attack

```
char * GetLine(){
    char buffer[130];
    gets(buffer);
}
```

```
getLine:
    push  ebp
    mov   ebp,esp
    sub   esp,152
    lea   eax,-152(ebp)
    pushl eax
    call  gets
    add   esp,4
    leave
    ret
```

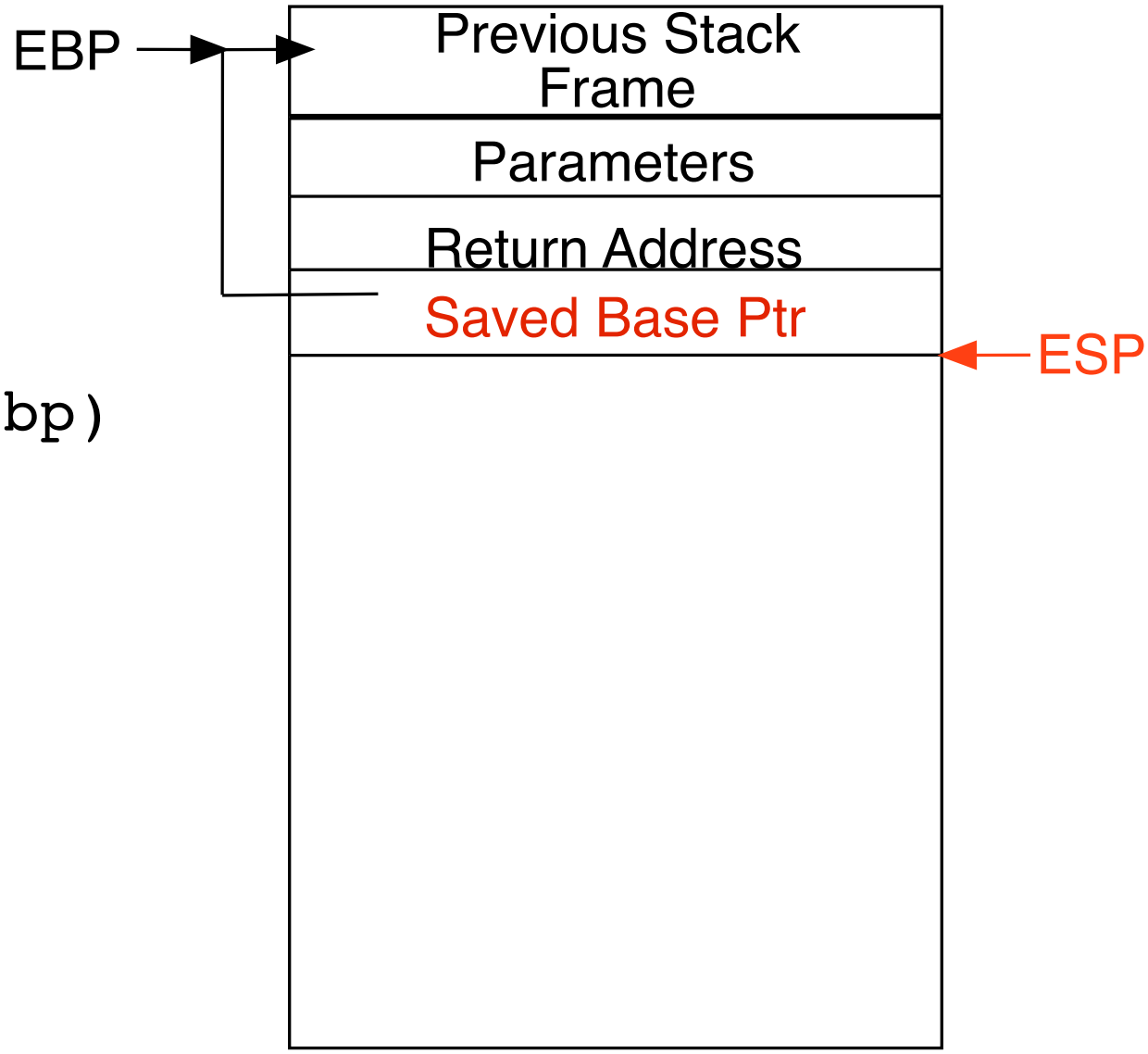
Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



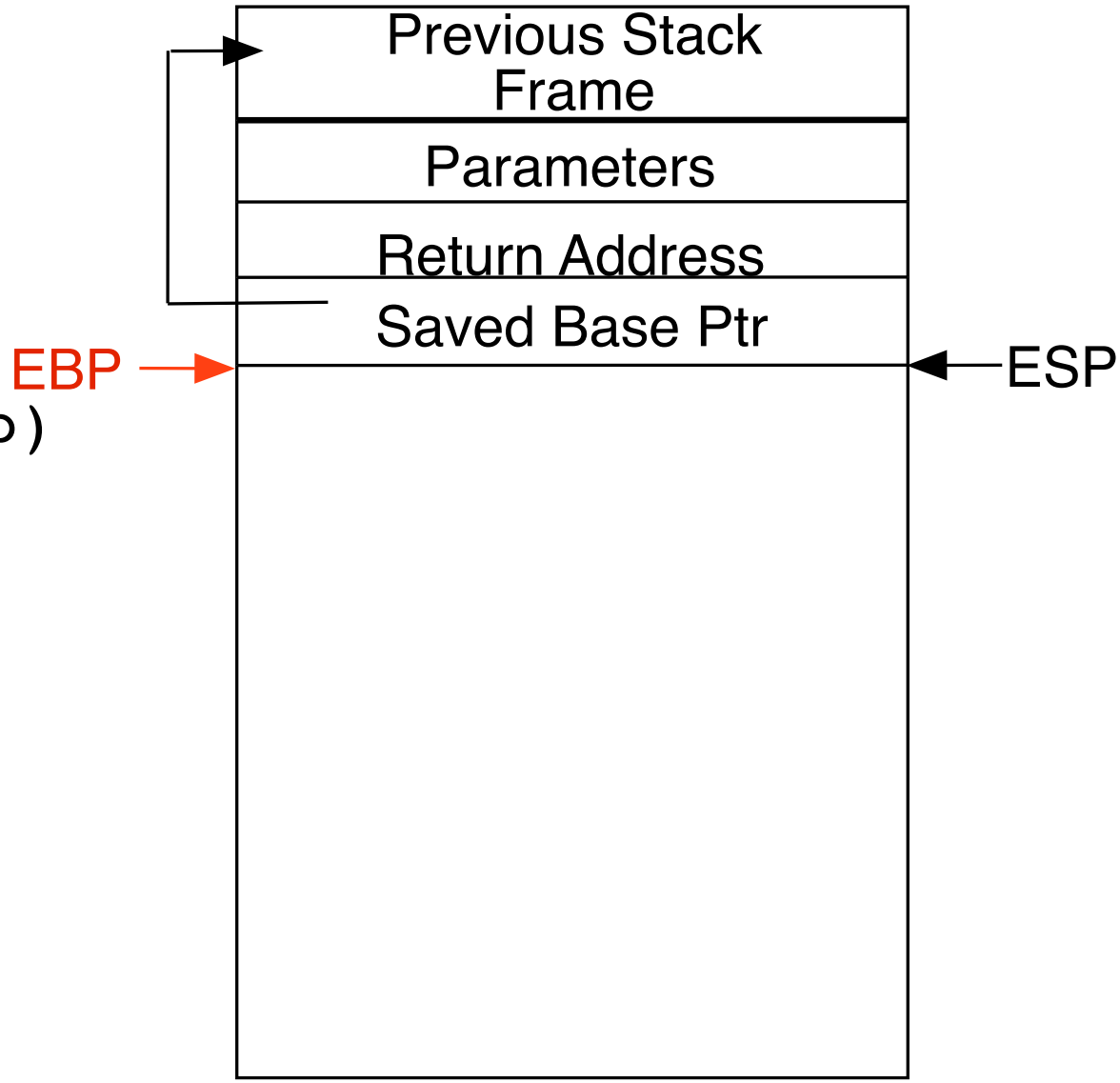
Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



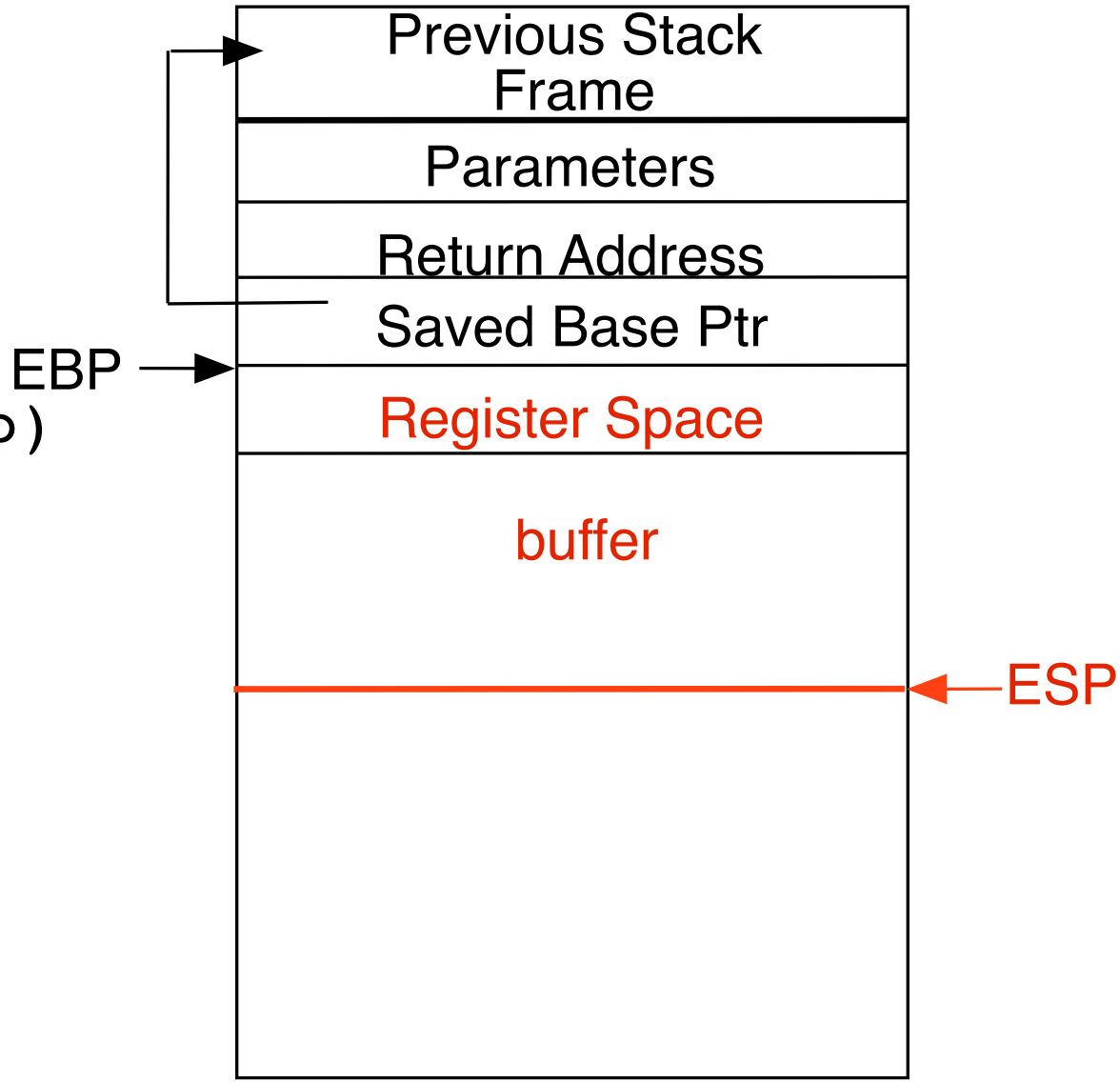
Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



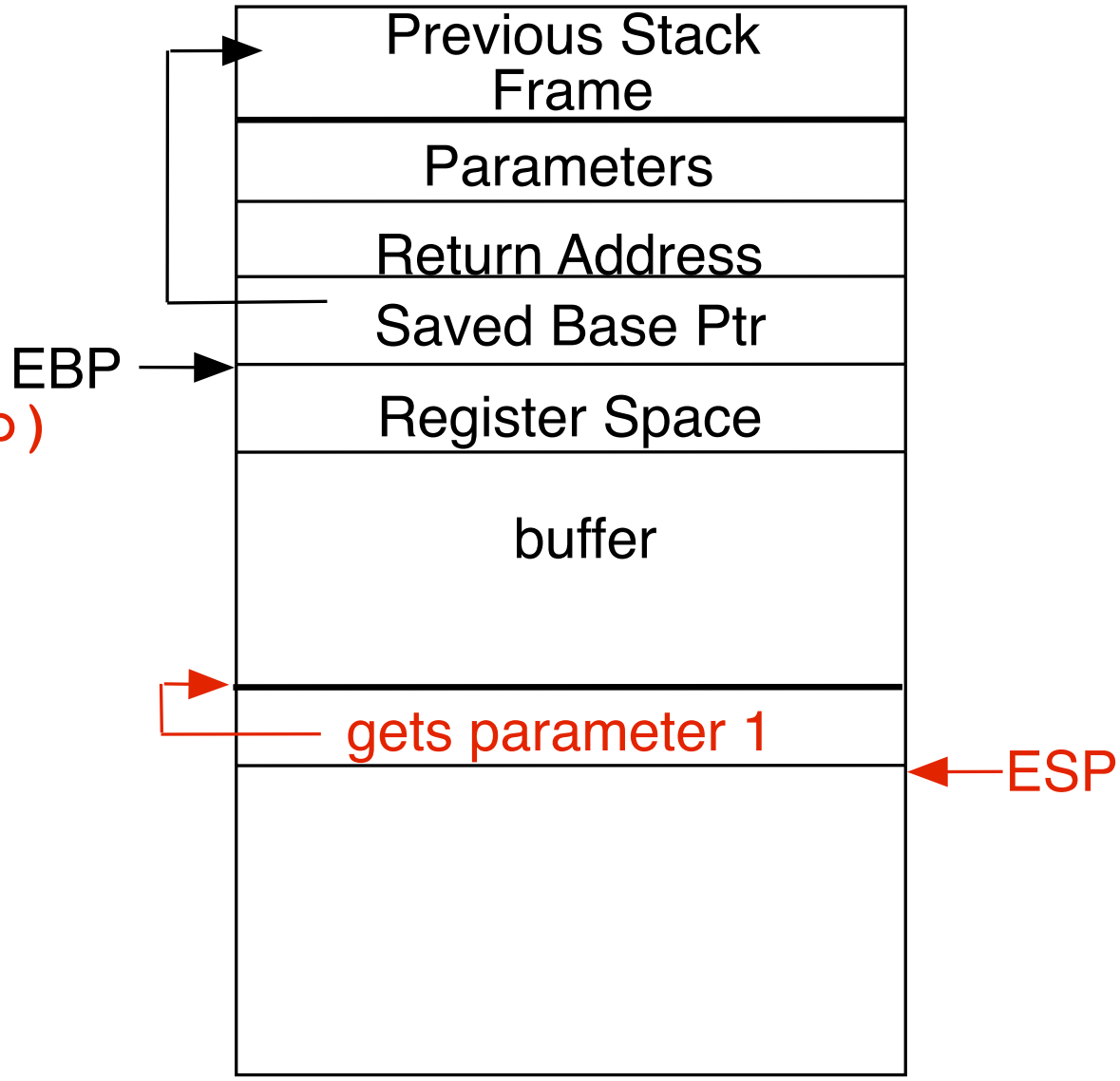
Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



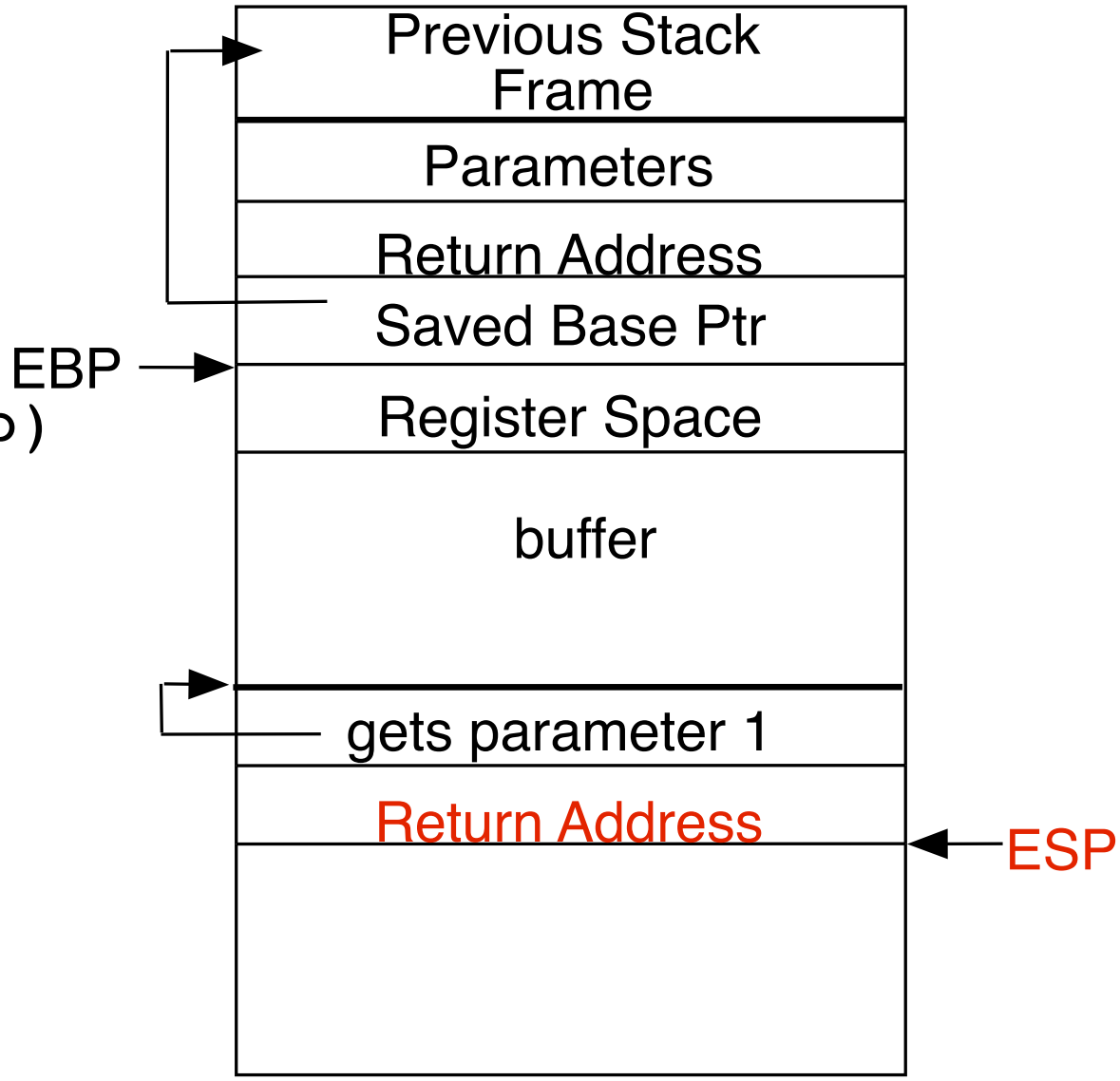
Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



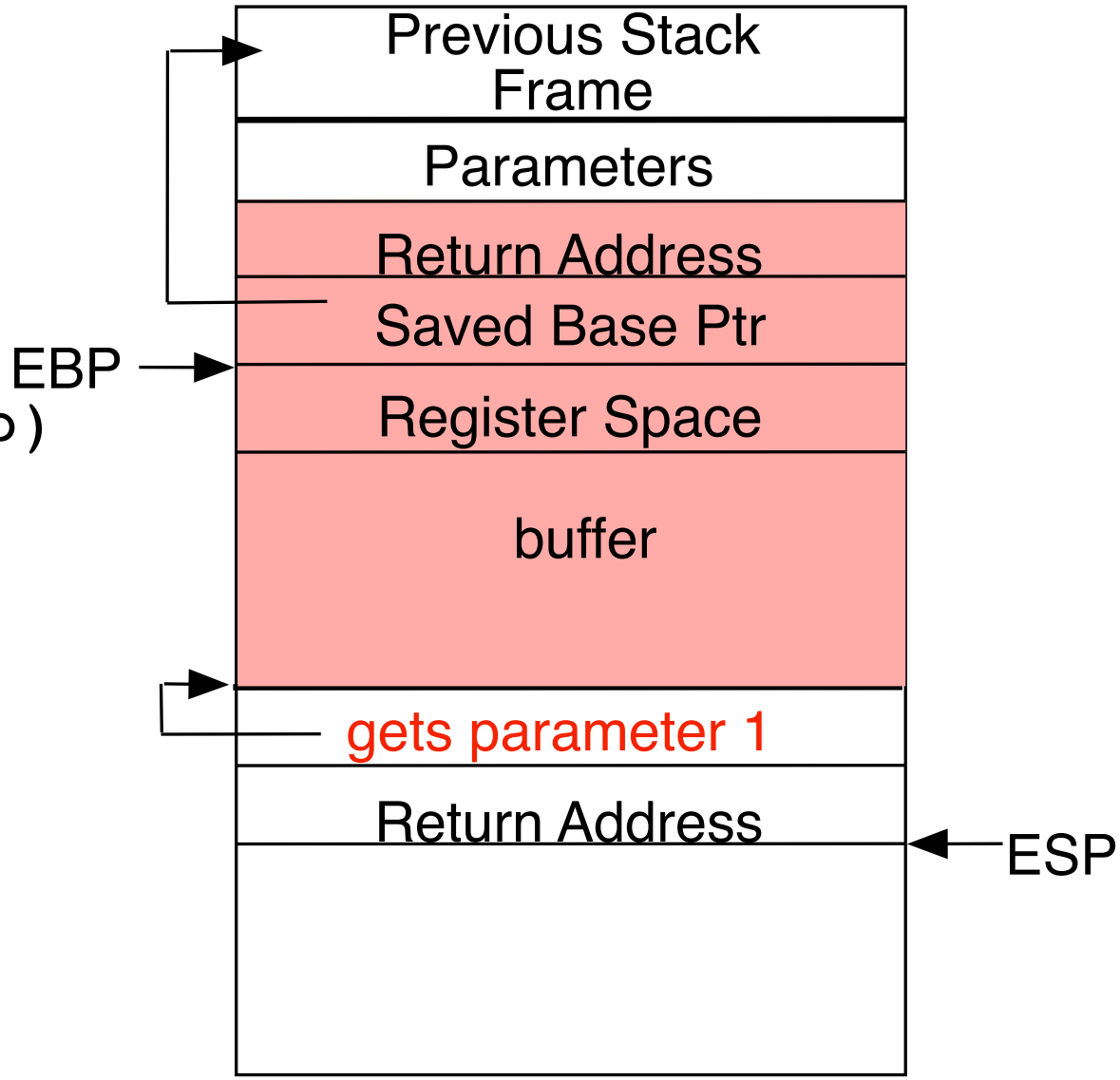
Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



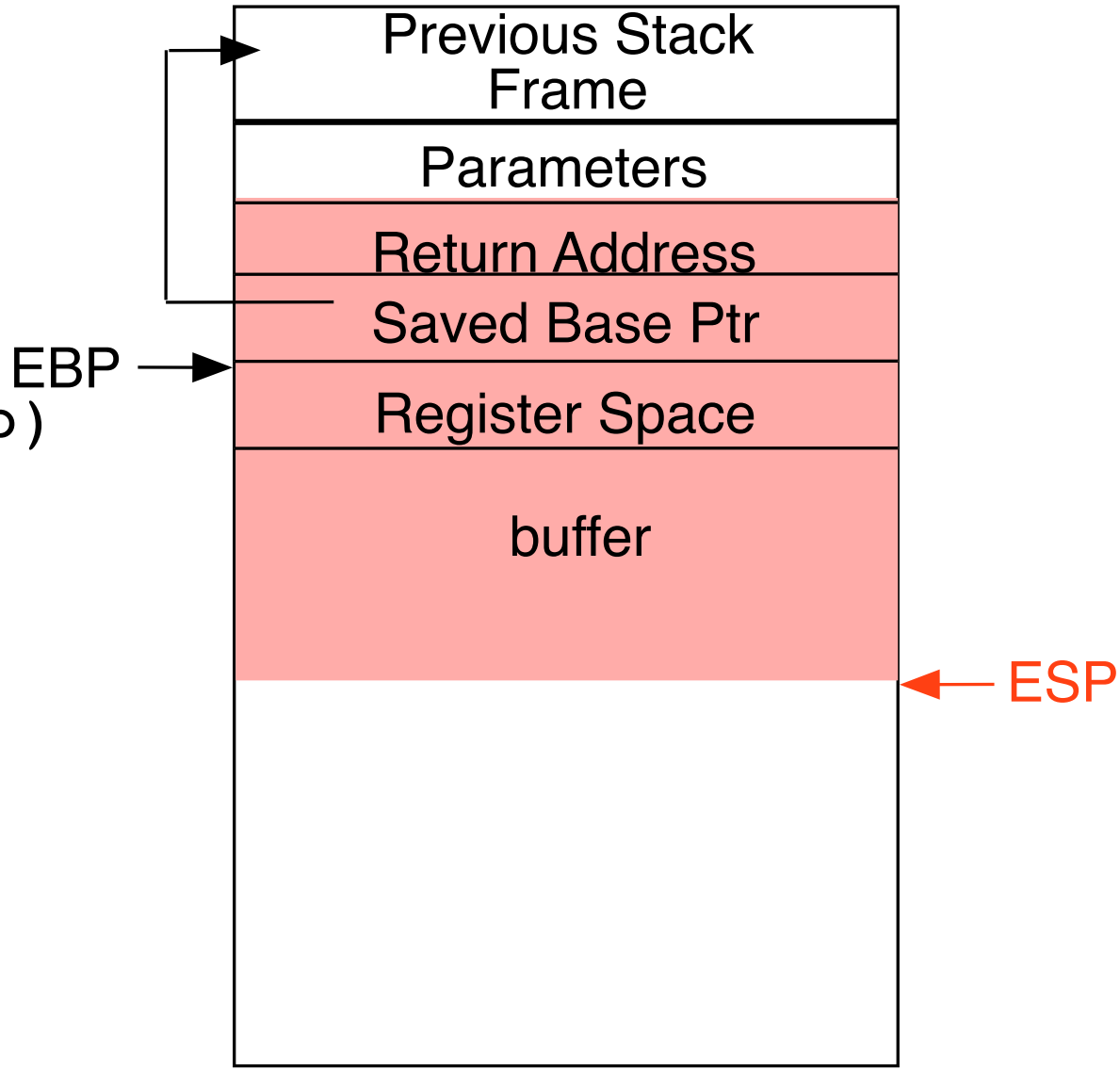
Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



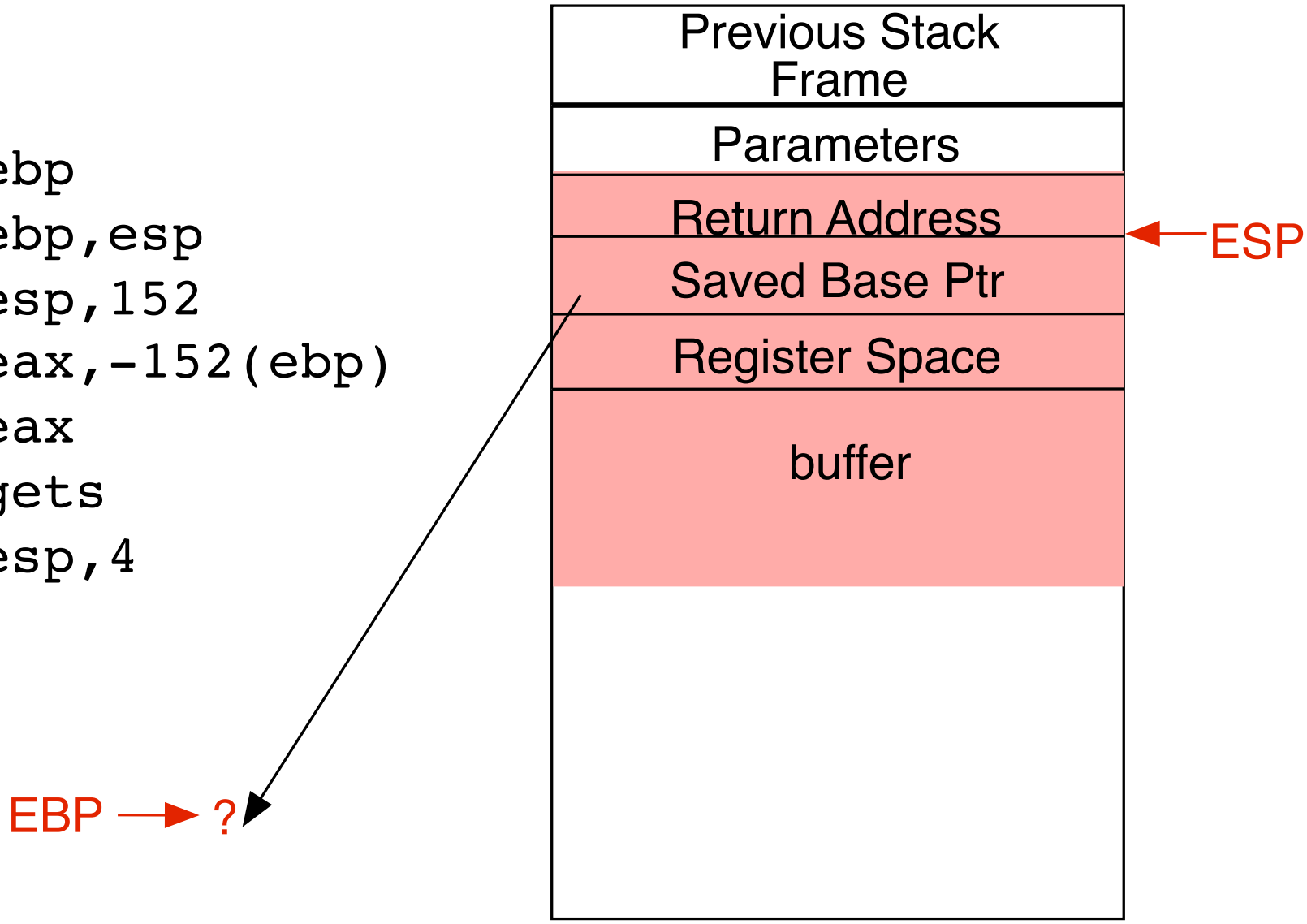
Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



Stack Overflow Attack

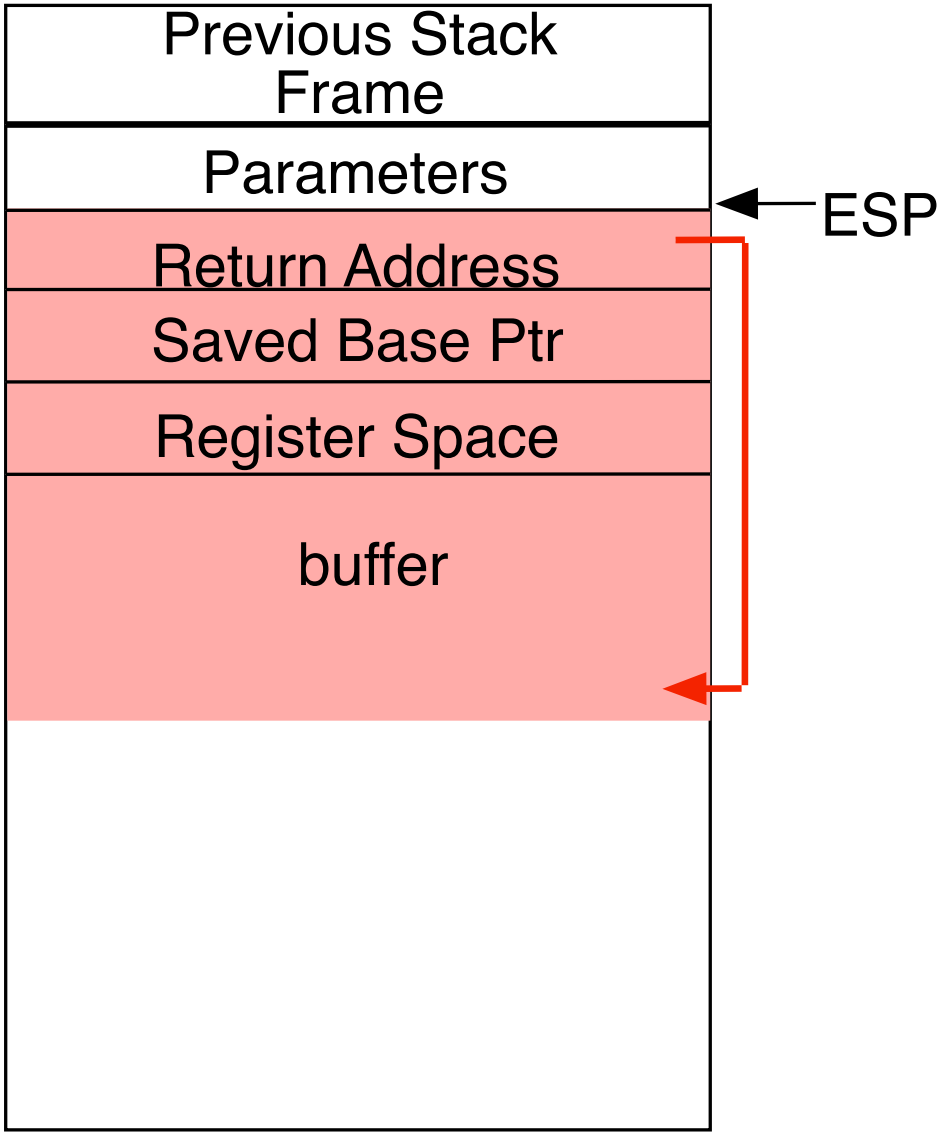
```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```



Stack Overflow Attack

```
getLine:
    push   ebp
    mov    ebp,esp
    sub    esp,152
    lea    eax,-152(ebp)
    pushl  eax
    call   gets
    add    esp,4
    leave
    ret
```

EBP → ?



Shell Code

- Properties
  - ◊ will be be read by a gets or fgets
  - ◊ may not contain null characters
  - ◊ may not contain newline or carriage return chars
- Small simple code
  - ◊ limited to the size of the buffer
- Download password file:
  - ◊ open file (check for error)
  - ◊ loop until end of file
  - ◊ read some chars
  - ◊ write some chars
  - ◊ exit

Shell Code

- Small simple code
  - ◊ limited to the size of the buffer
- Download password file:
  - ◊ sh -c "cat /etc/passwd; exit"

```
result = execve(const char * path, char *const argv[], char *const envp[]);
```

- Position independent
  - ◊ relative addressing
  - ◊ can be adapted to other attacks by only changing the address written to the return addresss.