Return Oriented Programming

pwning time

Buffer Overflows

What is wrong with this program?

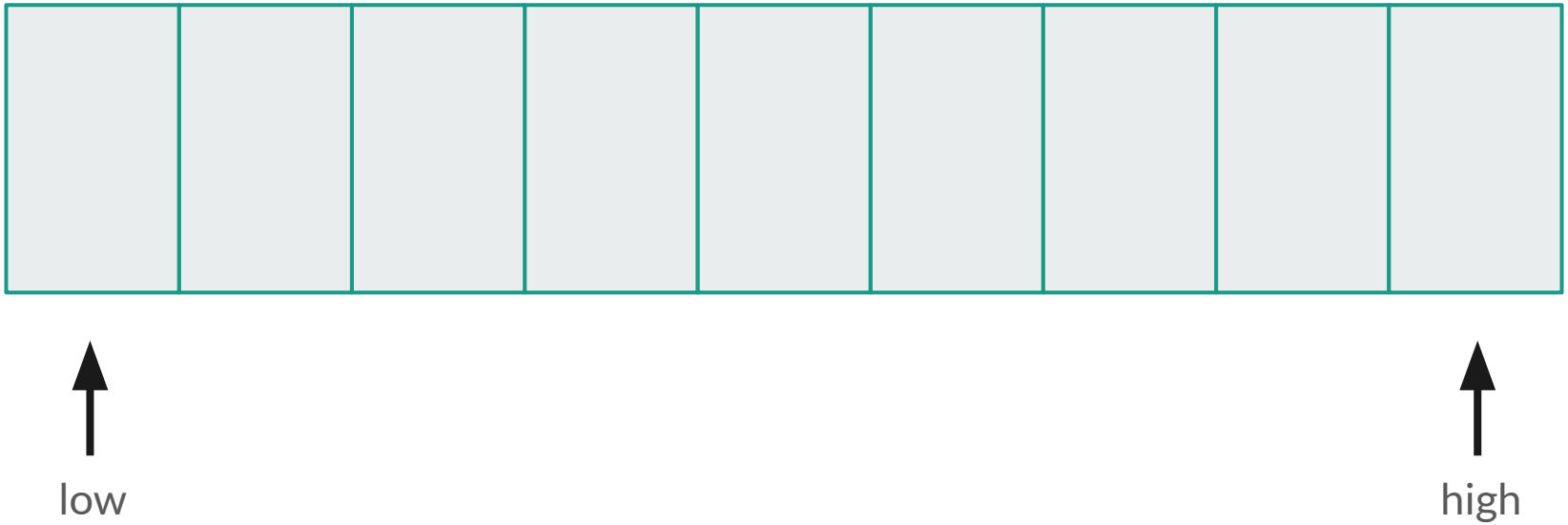
```c
#include <stdio.h>
#include <stdlib.h>

void admin() { system("/bin/sh"); }

void main() {
  int check = 0x11223344;
  char buf[4];

  setbuf(stdout, NULL);

  printf("Input: ");

  fgets(buf, 20, stdin);

  if (check == 0xdeadbeef) {
    puts("Correct!");
    admin();
  } else {
    puts("Nah.");
  }
}
```

What is wrong with this program?

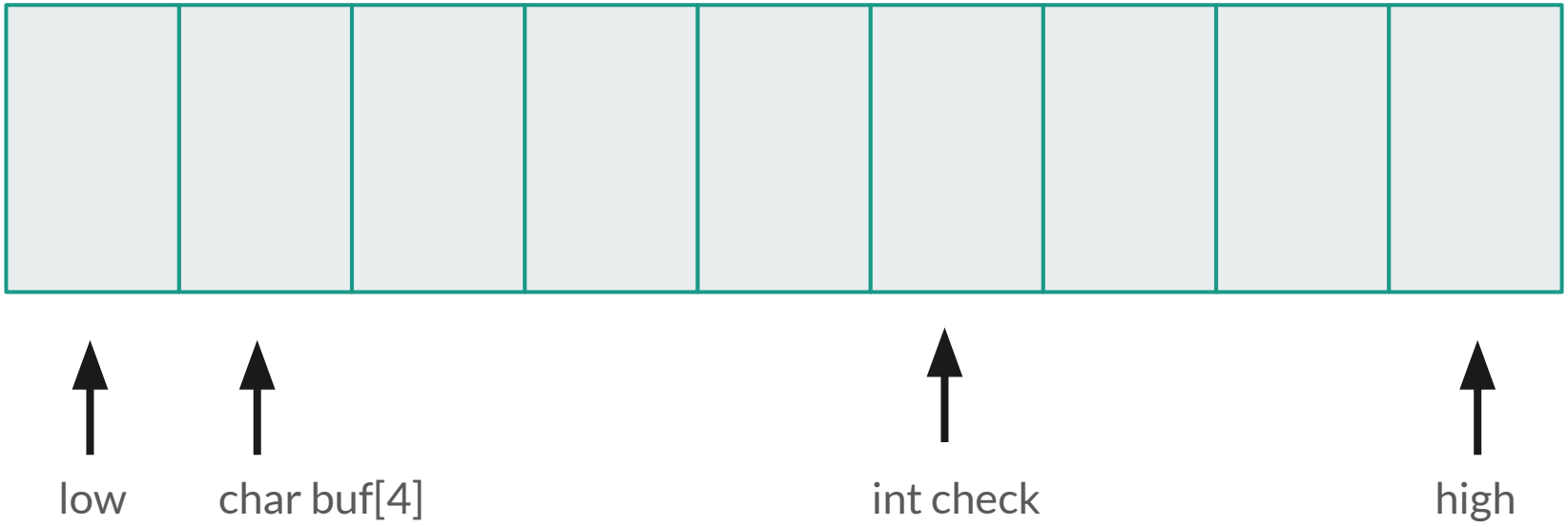
```c
#include <stdio.h>
#include <stdlib.h>

void admin() { system("/bin/sh"); }

void main() {
  int check = 0x11[illegible]3544;
  char buf[4];

  setbuf(stdout, NULL);

  printf("Input: ");

  fge[illegible](buf, 20, stdin);

  if (check == 0xdeadbeef) {
    puts("Correct!");
    admin();
  } else {
    puts("Nah.");
  }
}
```

its pwning time

Stack Visualization

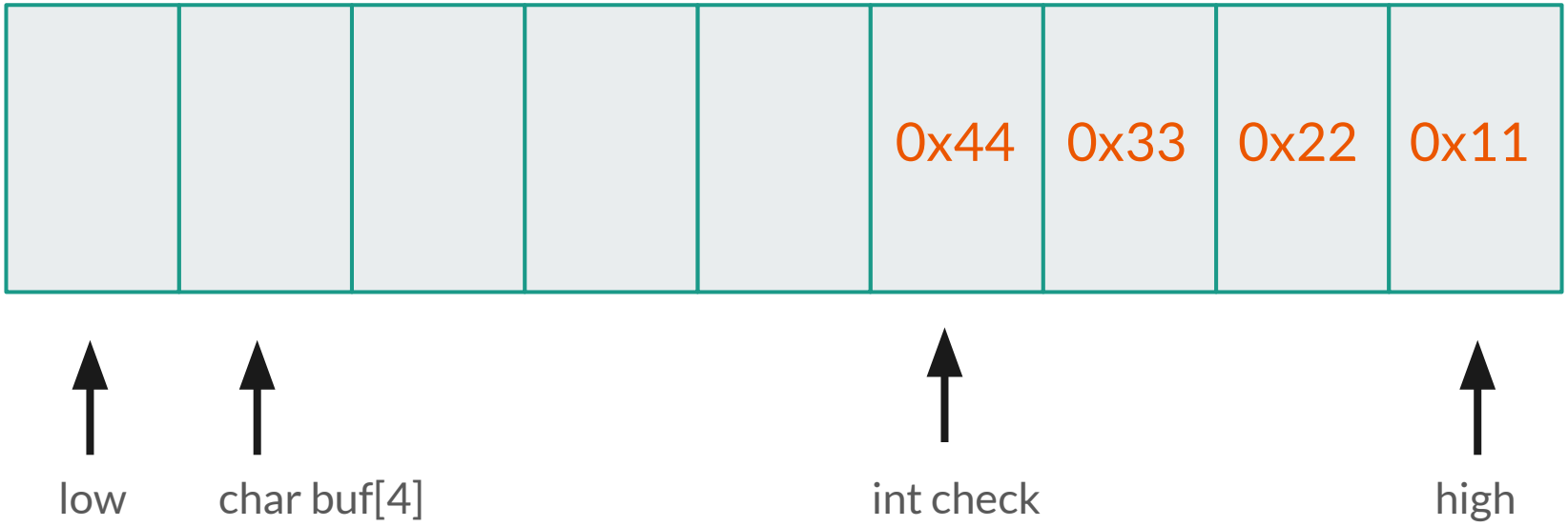


Stack Visualization



Stack Visualization

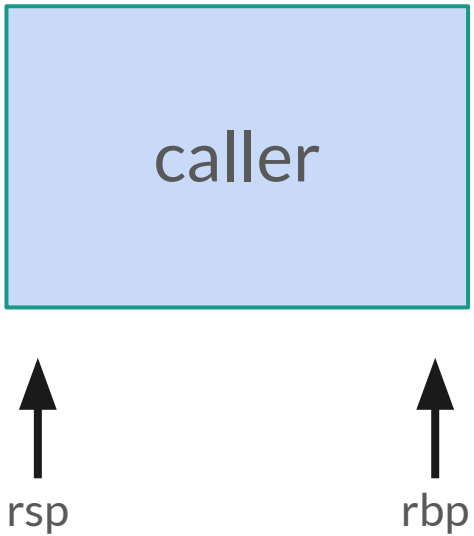
Little endian!



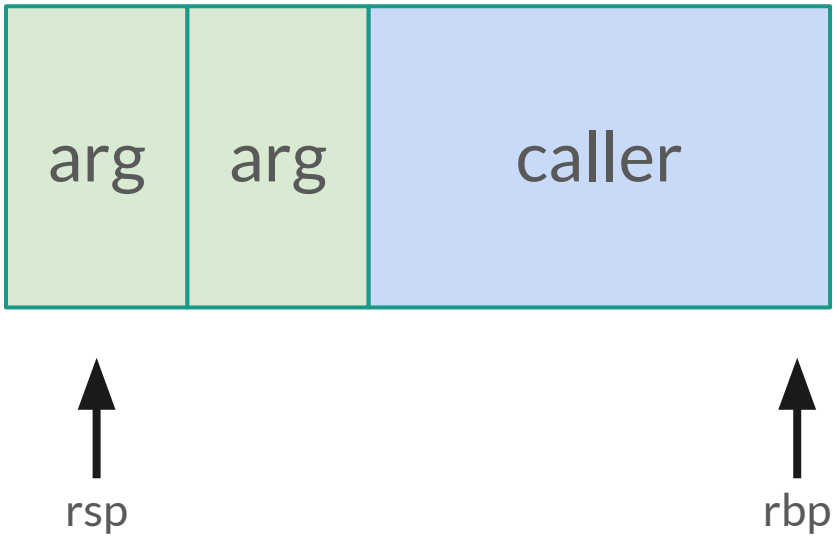
demo 1!

Code redirection

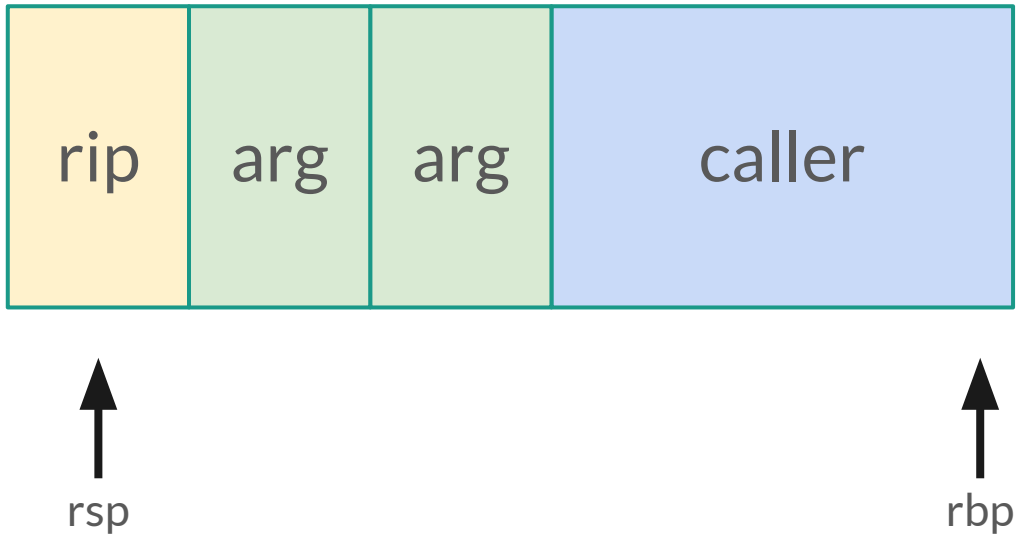
Stack Layout - Caller's Frame



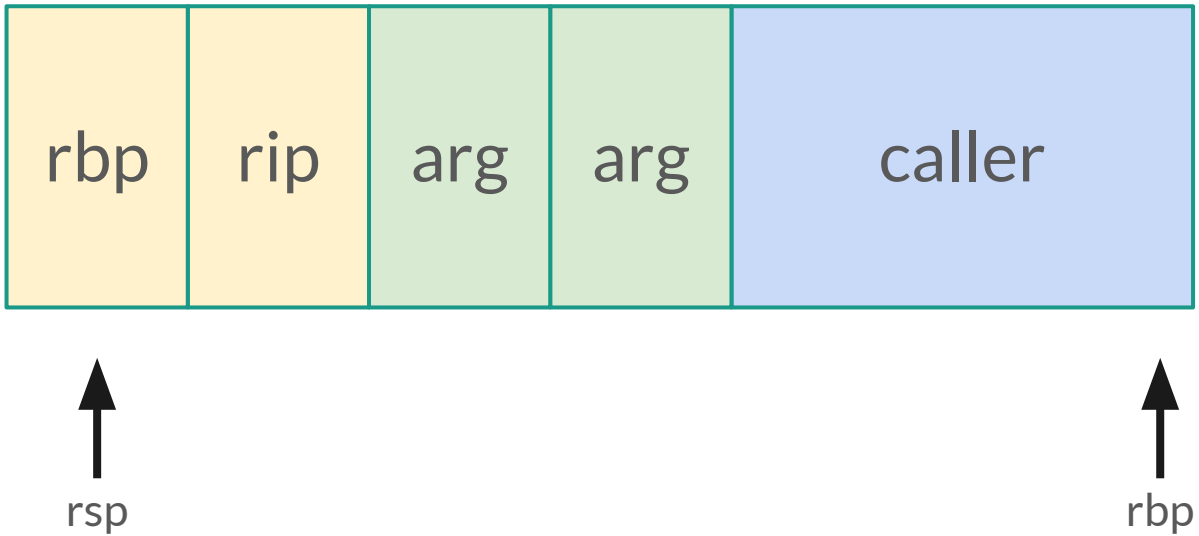
Stack Layout - Arguments



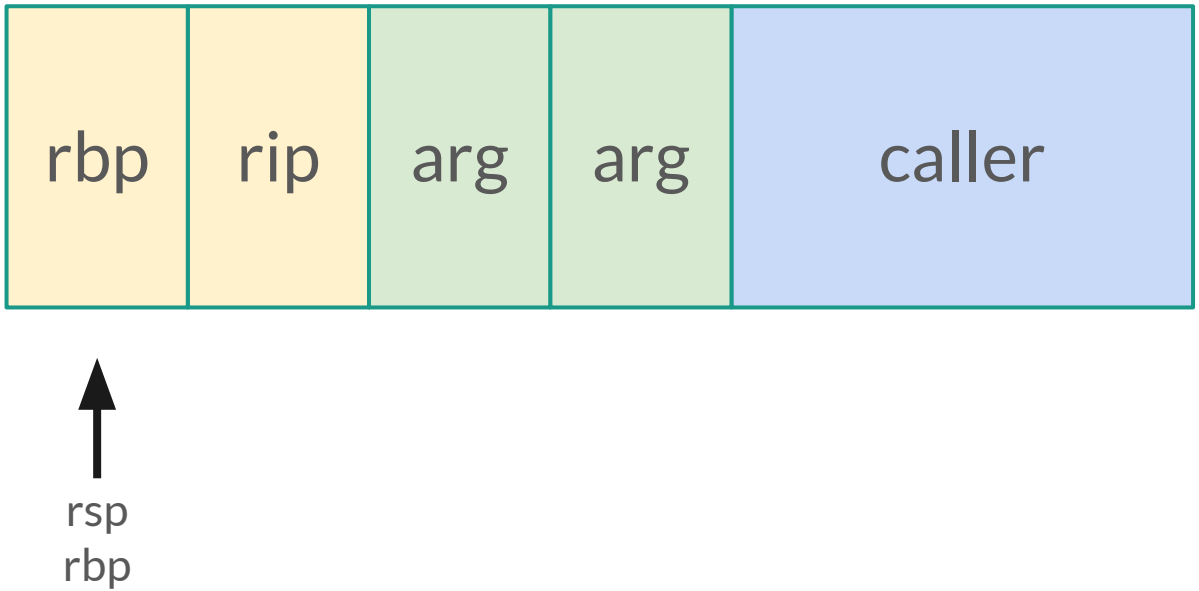
Stack Layout - Return Pointer



Stack Layout - Callee saves `rbp`



Stack Layout - Callee prologue

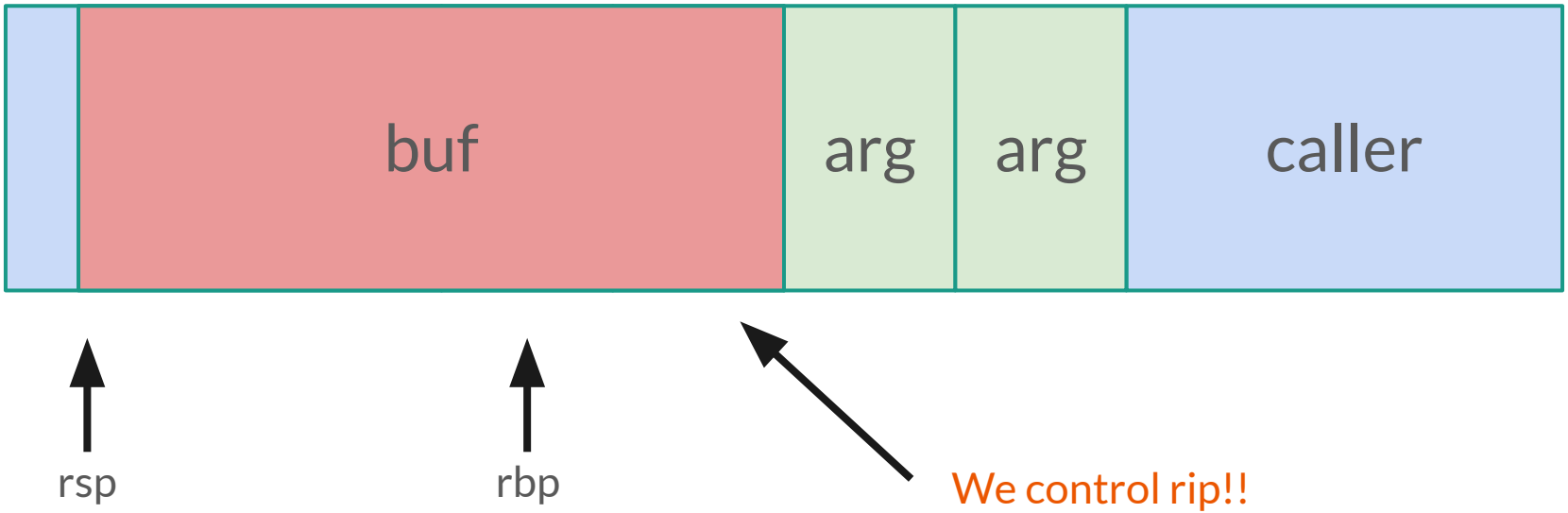


Stack Layout - Callee's frame

Stack Layout - Buffer overflow

Stack Layout - Overwrite variables

Stack Layout - !!!



demo pt2

we hack some more things

ROP

ok but we still need a function to jump to

```c
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void admin() { system("/bin/sh"); }

int main(void) {
  char buf[15];

  setbuf(stdin, NULL);
  setbuf(stdout, NULL);

  printf("Enter your name: ");
  fgets(buf, 100, stdin);

  return 0;
}
```

ok but we still need a function to jump to

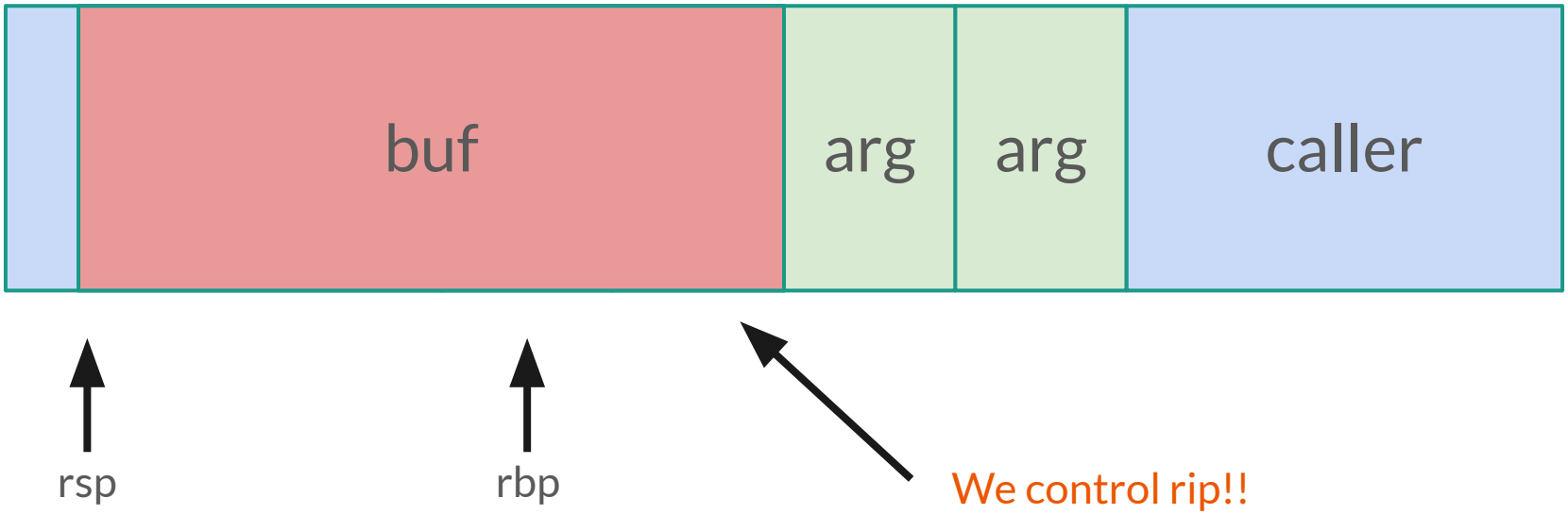
```c
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void admin() { system("/bin/sh"); }

int main(void) {
  char buf[15];

  setbuf(stdin, NULL);
  setbuf(stdout, NULL);

  printf("Enter your name: ");
  fgets(buf, 100, stdin);

  return 0;
}
```

ok but we still need a function to jump to

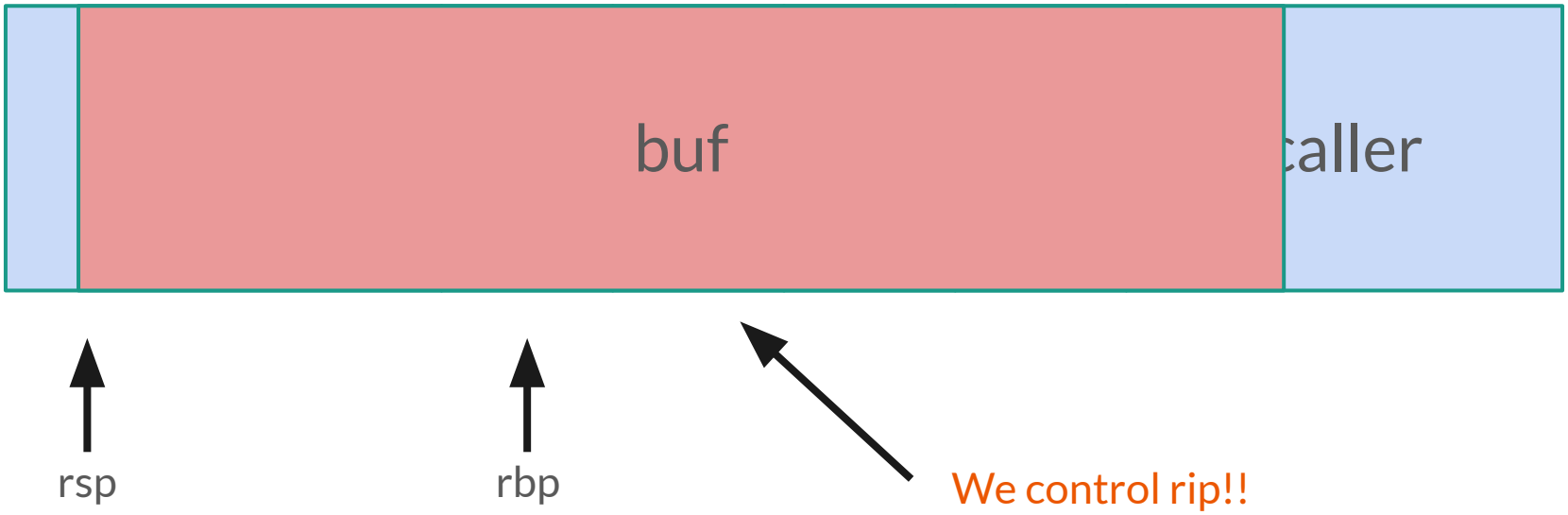
```c
#include <stdio.h>
#include <stdlib.h>

int main() {
  char name[100];

  setbuf(stdout, NULL);

  printf("Enter your name: ");
  fgets(name, 0x100, stdin);
}
```

Totally safe… right?!

Buffer Overflow Layout



But we control even more...



Gadgets

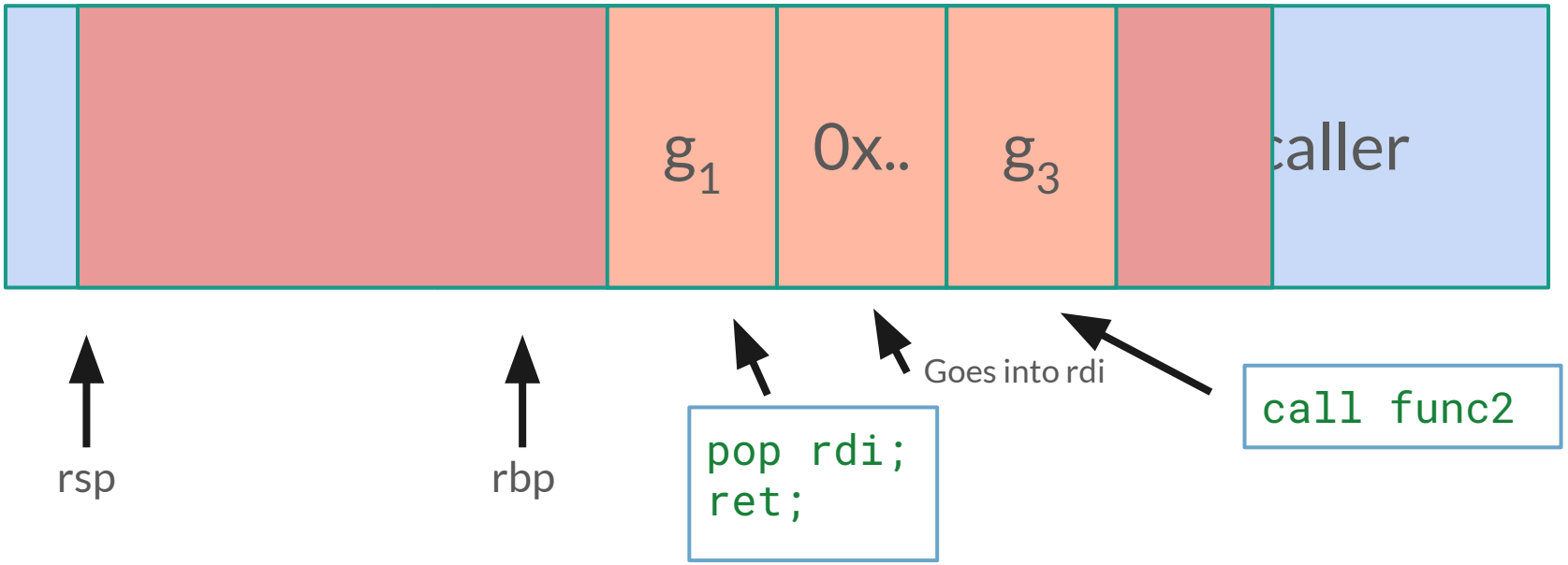
ROP Gadgets: small snippets of code which (typically) end in a `ret`

```
pop rdi;
ret;
```

```
mov [rbx], rax;
ret;
```

```
mov rsi, r13;
call [r15 + rbx*8];
```

Chaining Gadgets



demo pt3