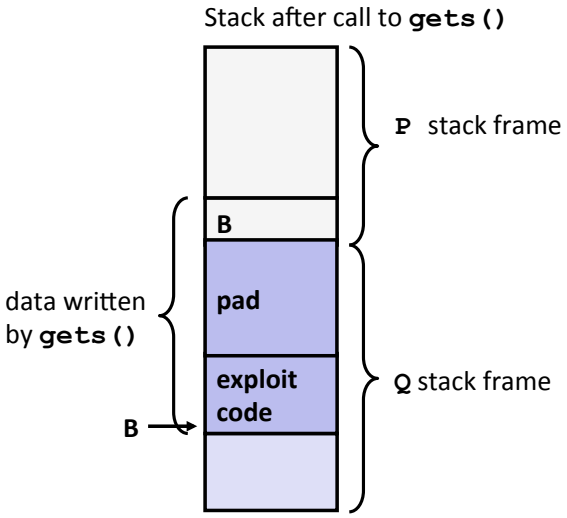
Code Injection Attacks

```
void P(){
  Q();
  ...
}
```

return address A

```
int Q() {
  char buf[64];
  gets(buf);
  ...
  return ...;
}
```

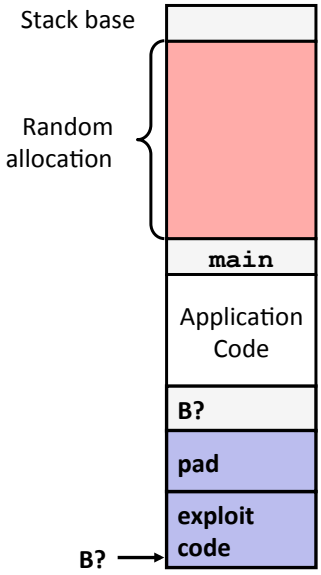


Write Secure Code (!)

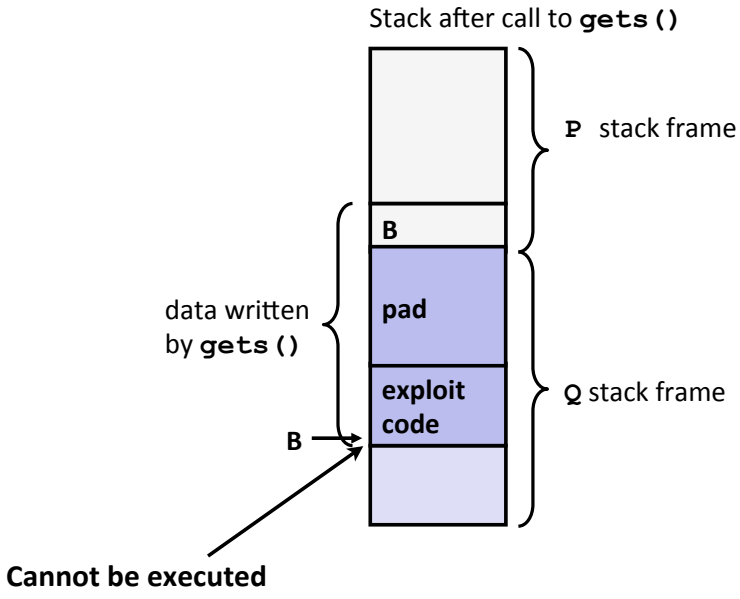
- Example: length-limited string routines
  - fgets, strncpy, ...
- No %s in scanf

```
/* Echo Line */
void echo() {
    char buf[4];  /* Way too small! */
    fgets(buf, 4, stdin);
    puts(buf);
}
```

Stack Randomization (ASLR)



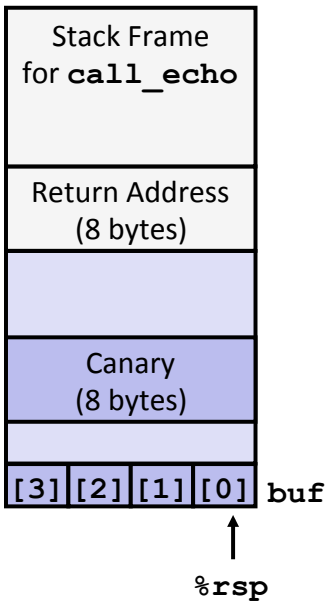
Non-executable Memory Segments



Stack Canaries

Stack Canary Example

Before call to gets

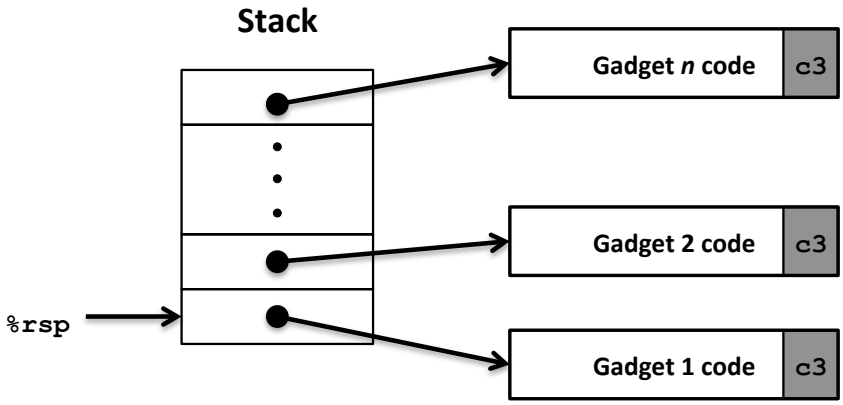



```
/* Echo Line */
void echo() {
    char buf[4];  /* Way too small! */
    gets(buf);
    puts(buf);
}
```

echo:

```
40072f:  sub    $0x18,%rsp
400733:  mov    %fs:0x28,%rax   # Get canary
40073c:  mov    %rax,0x8(%rsp)  # Put on stack
400741:  xor    %eax,%eax
400743:  mov    %rsp,%rdi
400746:  callq  4006e0 <gets>
40074b:  mov    %rsp,%rdi
40074e:  callq  400570 <puts@plt>
400753:  mov    0x8(%rsp),%rax  # Retrieve canary
400758:  xor    %fs:0x28,%rax   # Check canary
400761:  je     400768 <echo+0x39>
400763:  callq  400580 <__stack_chk_fail@plt>
400768:  add    $0x18,%rsp
40076c:  retq
```

Return-Oriented Programming



Gadget Example: Function Ends

```
long ab_plus_c(long a, long b, long c) {
   return a * b + c;
}
```

```
00000000004004d0 <ab_plus_c>:
  4004d0:  48 0f af fe  imul %rsi,%rdi
  4004d4:  48 8d 04 17  lea (%rdi,%rdx,1),%rax
  4004d8:  c3           retq
```

rax ← rdi + rdx

Gadget address = 0x4004d4

Gadget Example 2: Repurposing

```
void setval(unsigned *p) {
    *p = 3347663060u;
}
```