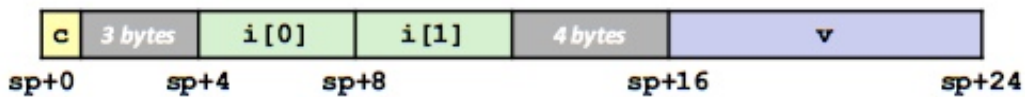
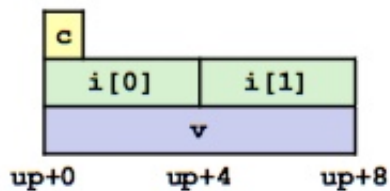


Union Allocation

- allocate according to largest element
- can only use on field at a time

```
1 union U1 {  
2     char c;  
3     int i[2];  
4     double v;  
5 } *up;
```

```
1 struct S1 {  
2     char c;  
3     int i[2];  
4     double v;  
5 } *sp
```



Using Union to Access Bit Patterns



```
1 typedef union {  
2     float f;  
3     unsigned u;  
4 } bit_float_t;
```

```
1 float bit2float(unsigned u) {  
2     bit_float_t arg;  
3     arg.u = u;  
4     return arg.f;  
5 }
```

```

1 unsigned float2bit(float f) {
2     bit_float_t arg;
3     arg.f = f;
4     return arg.u;
5 }

```

Byte Ordering Example

```

1 union {
2     unsigned char c[8];
3     unsigned short s[4];
4     unsigned int i[2];
5     unsigned long l[1];
6 } dw;

```

32-bit

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]
s[0]		s[1]		s[2]		s[3]	
i[0]				i[1]			
l[0]							

64-bit

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]
s[0]		s[1]		s[2]		s[3]	
i[0]				i[1]			
l[0]							

Buffer Overflow

- string library code for functions like `gets`, `scanf`, etc. would not check for buffer overflow
- hackers take advantage of this to precisely change the return address in the stack
 - changes program flow to run what the hacker wants

Before call to gets

Stack Frame for <code>call_echo</code>			
00	00	00	00
00	40	06	f6
20 bytes unused			
[3]	[2]	[1]	[0]

buf ← %rsp

```
void echo()
{
    char buf[4];
    gets(buf);
    . . .
}
```

```
echo:
    subq $24, %rsp
    movq %rsp, %rdi
    call gets
    . . .
```

call_echo:

```
. . .
4006f1: callq 4006cf <echo>
4006f6: add $0x8,%rsp
. . .
```

Doesn't cause a problem

After call to gets

Stack Frame for <code>call_echo</code>			
00	00	00	00
00	40	06	f6
00	32	31	30
39	38	37	36
35	34	33	32
31	30	39	38
37	36	35	34
33	32	31	30

buf ← %rsp

```
void echo()
{
    char buf[4];
    gets(buf);
    . . .
}
```

```
echo:
    subq $24, %rsp
    movq %rsp, %rdi
    call gets
    . . .
```

call_echo:

```
. . .
4006f1: callq 4006cf <echo>
4006f6: add $0x8,%rsp
. . .
```

```
unix> ./bufdemo-nsp
Type a string: 01234567890123456789012
01234567890123456789012
```

Changes return address

After call to gets

Stack Frame for <code>call_echo</code>			
00	00	00	00
00	40	00	34
33	32	31	30
39	38	37	36
35	34	33	32
31	30	39	38
37	36	35	34
33	32	31	30

`buf` ← `%rsp`

```
void echo()  
{  
    char buf[4];  
    gets(buf);  
    . . .  
}
```

```
echo:  
    subq    $24, %rsp  
    movq    %rsp, %rdi  
    call    gets  
    . . .
```

`call_echo:`

```
. . .  
4006f1:    callq   4006cf <echo>  
4006f6:    add     $0x8, %rsp  
. . .
```

```
unix> ./bufdemo-nsp  
Type a string: 0123456789012345678901234  
Segmentation Fault
```

Avoiding Overflow Vulnerabilities

- use `fgets` instead of `gets`
- `strncpy` instead of `strcpy`
- don't use `scanf` with `%s` conversion specification
 - use `fgets` to read the string
 - or use `%ns` where `n` is a suitable integer