Recitation2

Outline

Datalab

Bomblab

Assembly

gdb

Example

QOTW

“Commenting your code is like cleaning your bathroom — you never want to do it, but it really does create a more pleasant experience for you and your guests.”

– Ryan Campbell

Datalab

Results

182 got 61/61

50 were in the 50's

42 had below a 50

Nice result!

But remember to start early on labs!

Grading

Graded by next Monday

Scores will show up on Autolab

A random TA will grade your lab

Bomblab

“Hacking the binary...”
“I'm in!”

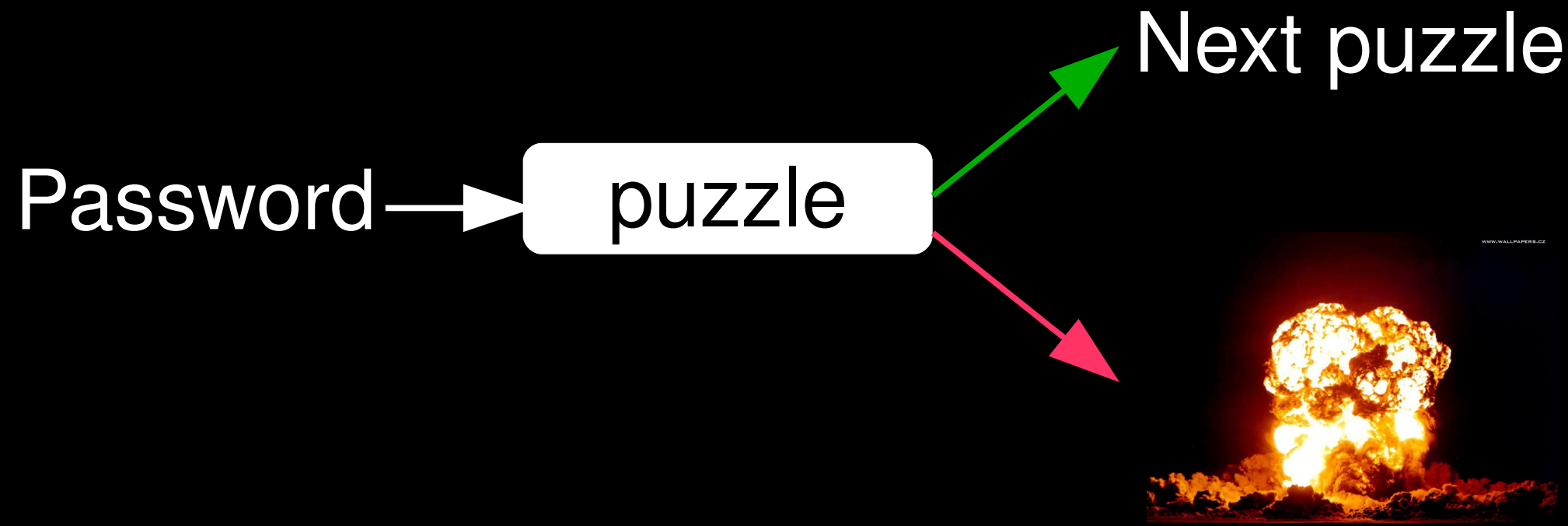
Bomblab

Set of puzzles where you have to enter the correct password

A minor inconvenience

Wrong password blows up bomb

Lowers your grade by half a point



So you don't have to this is what it looks like:

```
[jprimero@flounder bomb174]$ ./bomb
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
blow up bomb

BOOM!!!
The bomb has blown up.
Your instructor has been notified.
[jprimero@flounder bomb174]$
```

How to solve

We give you a compiled binary

You read the assembly code to figure out the passwords

But how do we stop a bomb from exploding?

Assembly

x86 Assembly

```
0000000000400448 <fac>:
  400448:       55                      push   %rbp
  400449:       48 89 e5                mov    %rsp,%rbp
  40044c:       48 83 ec 08             sub    $0x8,%rsp
  400450:       89 7d fc                mov    %edi,0xfffffffffffffffc(%rbp)
  400453:       83 7d fc 01             cmpl   $0x1,0xfffffffffffffffc(%rbp)
  400457:       75 09                   jne    400462 <fac+0x1a>
  400459:       c7 45 f8 01 00 00 00    movl   $0x1,0xfffffffffffffff8(%rbp)
  400460:       eb 14                   jmp    400476 <fac+0x2e>
  400462:       8b 7d fc                mov    0xfffffffffffffffc(%rbp),%edi
  400465:       83 ef 01                sub    $0x1,%edi
  400468:       e8 db ff ff ff          callq  400448 <fac>
  40046d:       89 c2                   mov    %eax,%edx
  40046f:       0f af 55 fc             imul   0xfffffffffffffffc(%rbp),%edx
  400473:       89 55 f8                mov    %edx,0xfffffffffffffff8(%rbp)
  400476:       8b 45 f8                mov    0xfffffffffffffff8(%rbp),%eax
  400479:       c9                      leaveq
  40047a:       c3                      retq

000000000040047b <main>:
  40047b:       55                      push   %rbp
  40047c:       48 89 e5                mov    %rsp,%rbp
  40047f:       bf 03 00 00 00          mov    $0x3,%edi
  400484:       e8 bf ff ff ff          callq  400448 <fac>
  400489:       c9                      leaveq
  40048a:       c3                      retq
  40048b:       90                      nop
  40048c:       90                      nop
  40048d:       90                      nop
  40048e:       90                      nop
  40048f:       90                      nop
```

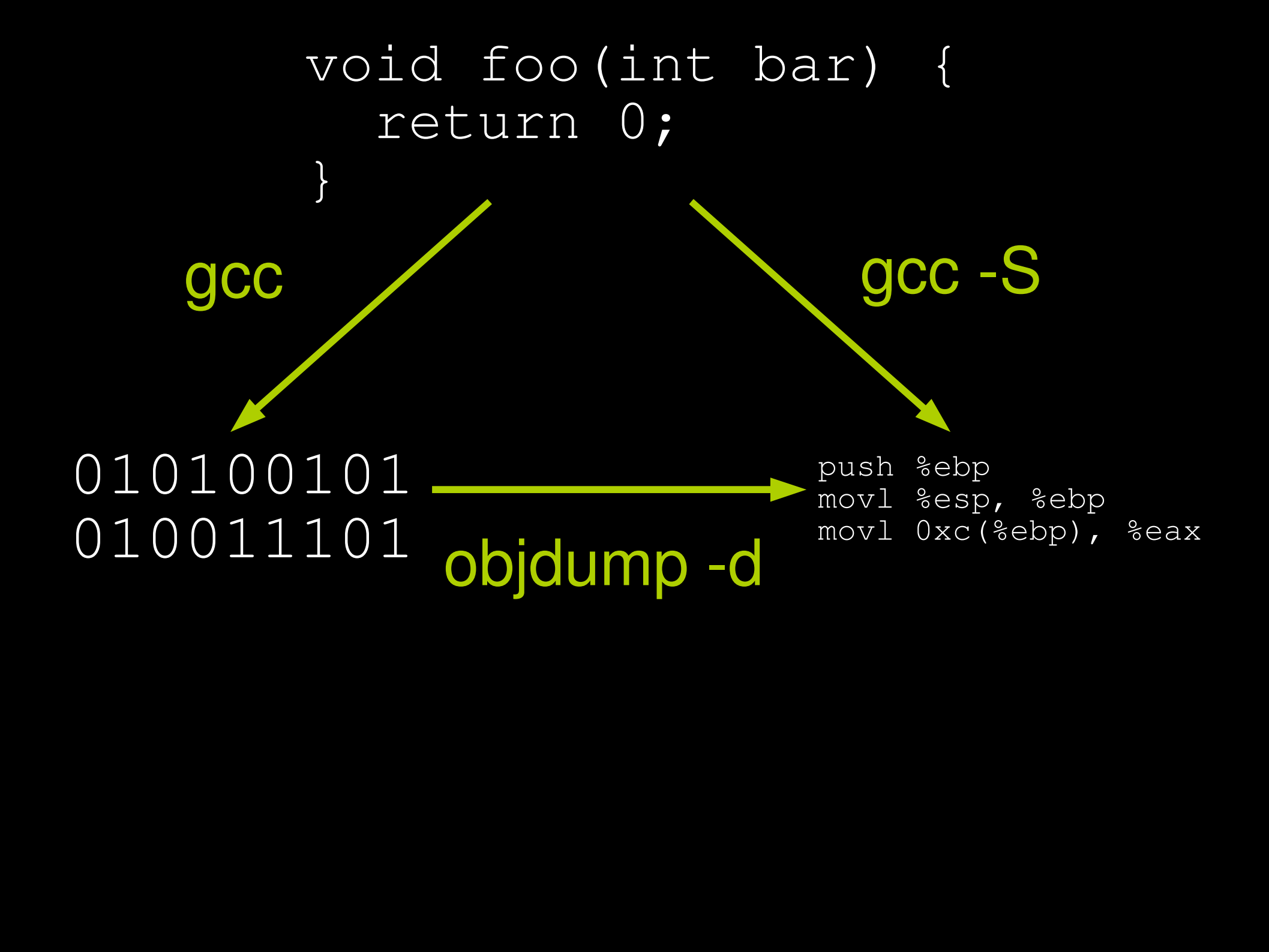
```
void foo(int bar) {
   return 0;
}
```

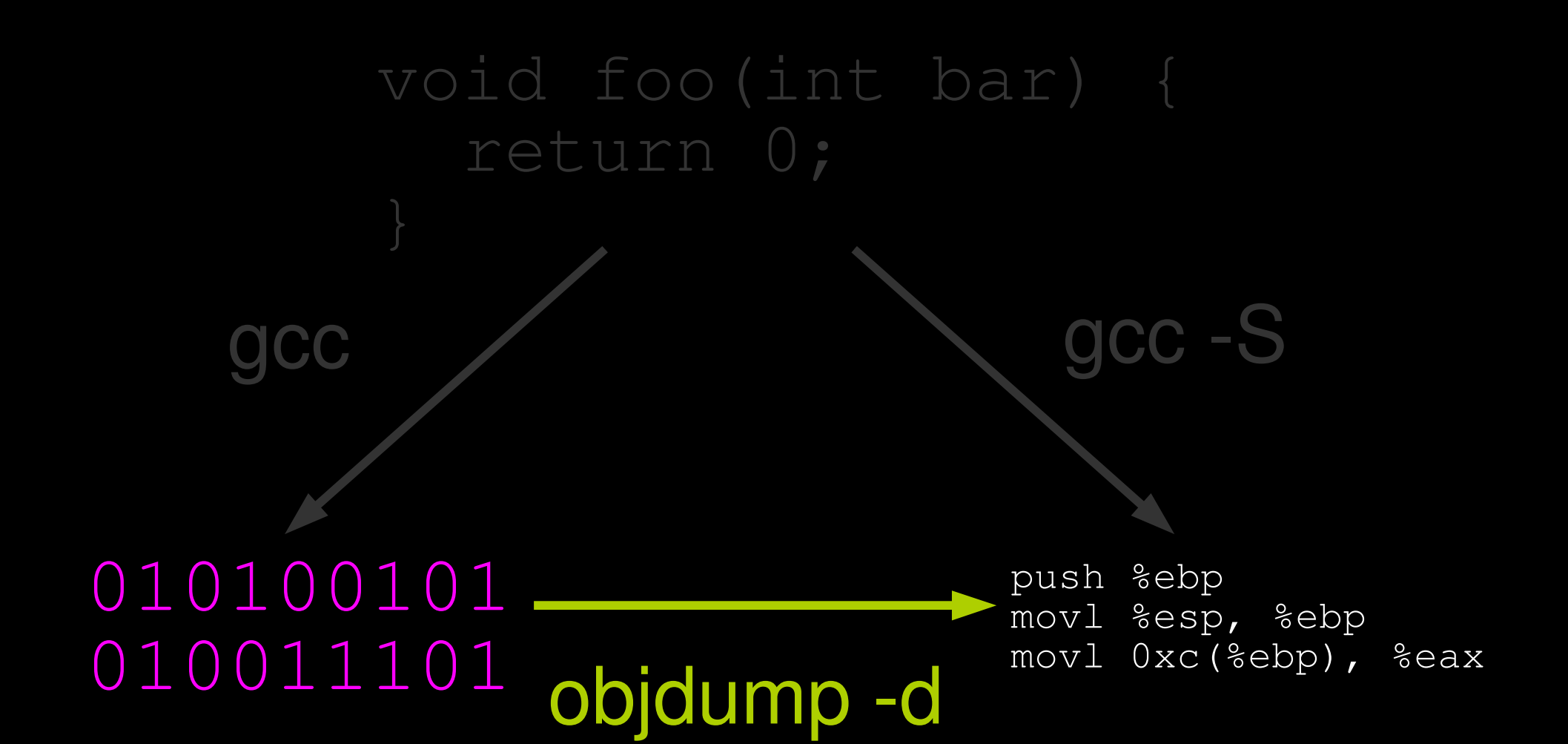
compile

```
010100101
010011101
```

One to One

```
push %ebp
movl %esp, %ebp
movl 0xc(%ebp), %eax
```


void foo(int bar) {
return 0;
}
gcc
gcc -S
010100101
010011101
objdump -d
push %ebp
movl %esp, %ebp
movl 0xc(%ebp), %eax



For bomblab we only give the binary

Bomblab Hints

`objdump -d bomb` will print out the assembly code of bomb

`objdump -d bomb > bomb.s` will place the bomb assembly code into the file bomb.s

Assembly Code Example

x86 assembly flash demo

http://www.ece.cmu.edu/~jprimero/Impress.swf

gdb

Gnu DeBugger

Step through program execution

Examine program execution

Set breakpoints to halt execution at any point (hrmmmm...)

Gnu DeBugger

Very important tool for debugging so learn it!

Reference at:
http://www.cs.cmu.edu/~213/resourses.html

Last Thoughts

Start Early!