Buffer Overflows
Classic Security Vulnerability

• See “Smashing The Stack For Fun And Profit”
void func(char *user_input) {
    char name[100];
    ...
}

- Callee’s local variables.
- Caller’s Frame Pointer
- Return Address
- First Argument to Callee
- ...
- Final Argument to Callee
- Caller’s local variables.
- ...
- Older stack frames.
- ...
Classic Security Vulnerability

```c
void func(char *user_input) {
    char name[100];
    ...
}
```

![Diagram of stack frames and local variables]

- func’s local variables.
- Caller’s Frame Pointer
- Return Address
- First Argument to Callee
  - ...
- Final Argument to Callee
- Caller’s local variables.
- ...
- Older stack frames.
  - ...
void func(char *user_input) {
    char name[100];
    ...
}

Suppose we asked a user to input their name. Is it safe to copy that into our “name” char array?

Why or why not?

A. Safe
B. Not safe
Is it safe? It depends...

• What function are we using to do the copy?
  – strcpy? When does it stop copying?

• What happens if we copy too much?
  – Does C ensure that we don’t go beyond the buffer?
  – Does strcpy?
  – What will we overwrite?

• Can we take advantage of that behavior?
A well intentioned program...

char name[100]
A well intentioned program...

If used properly, with a reasonable name, no problem here.
A well intentioned program...

What if my cat steps on the keyboard and types in a name of:

```
asdfweffewawerrr3f322frtfgfdfgrgrdgrgdgvcdllliuiyytylj;jouiyiytuytrf
bbncbvcxvcxzvn,mn.,n.,mloiyuytytjgkghgfdgtreyteretdgfhjdfsd
```
A well intentioned program...

What if my cat steps on the keyboard and types in a name of:

```
asdfweffewerrrr3f322frtfgfgdfgfrgrdrgdvgcdllliuiyytj;jouiyiyuytrf
bbncbvxvrvxznv,mn.,n..,mloijuytytjgjkgfghdftreytieredffhdjfdvsf
```

Set PC to this value on return!

What’ll happen?
Cat, performing the classic “Denial of Pizza” attack.

• Is crashing the program the worst we can do?
A well intentioned program...

Suppose I want to change the return address to do Evil™

Fake_name_that’s_really_long_to_fill_100_characters

Set PC to this value on return!

What’ll happen?

Does this help me be evil?

char name[100]

Stack
Memory

name[0]  

...  

name[99]  

FE  32  78  12
A well intentioned program...

If I can set the return address to be an arbitrary pointer, I can control what gets executed next!

If only I could add my own instructions in memory somewhere...

cchar name[100]
A well intentioned program...

Suppose I want to change the return address to do Evil™

[Do nothing (NOP)]...[Do nothing (NOP)]
[Evil™ Code that sends all your secrets to me]0xFE327812

Suppose I want to change the return address to do Evil™

Stack Memory

cchar name[100]  

Stack Memory

char name[100]  

Aim for RA to point within this region.
One careless street... Yours?

Remember—Only you can prevent buffer overflows!