Buffer Overflows
Classic Security Vulnerability

• “Smashing The Stack For Fun And Profit”

Callee’s frame.

Callee’s local variables.

Caller’s Frame Pointer

Return Address

First Argument to Callee

... 

Final Argument to Callee

Caller’s local variables.

Caller’s frame.

Shared by caller and callee.

Older stack frames.
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.

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

- 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]

Stack
Memory

name[0]
...
name[99]

Ret  urn  add  ress
```
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:

```plaintext
asdfweffewerrr3f322ftrfgfgfgfgrgrdgrgdgvcdllliuiyytylj;jouiyiuytvtrf
bbncbvcxvcxzvn, mn., n., mloijuymtytytgjkhghgfldtdretyeteretdgfdjdjsd
```

```plaintext
<table>
<thead>
<tr>
<th>char name[100]</th>
</tr>
</thead>
<tbody>
<tr>
<td>name[0]</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>name[99]</td>
</tr>
</tbody>
</table>

Stack Memory

Ret urn add ress
A well intentioned program...

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

```
asdfweeffewerrr3f322frtfgfdgdfgrgdgrgdgvcdbliliuyytyl;jouiyiuytuytrf
bbncbvcxvcxzv,nn,.,n,,mloijuytytjgkhgfdgtreyteretdghfhdjfsdfsds
```

char name[100]

Stack
Memory

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_________
__________________________________________0xFE327812

char name[100]

Stack
Memory

Set PC to this value on return! What’ll happen?

Does this help me be evil?
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...

char name[100] →

Stack
Memory

[Diagram showing memory layout with pointers and addresses]
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

char name[100] →

Stack Memory

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

Remember—Only you can
PREVENT BUFFER OVERFLOWS!