

Hi there

(I'm Angela)

# Don't worry, you got this

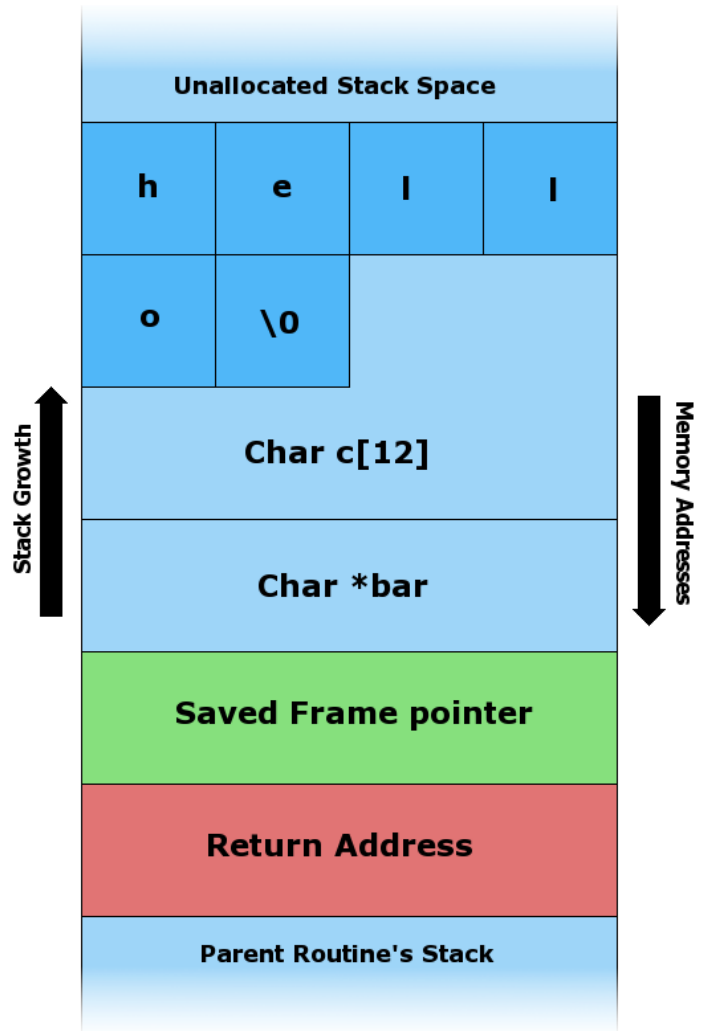
- Buffer overflow attacks
- CS50 library
- HTTP
- HTML
- CSS

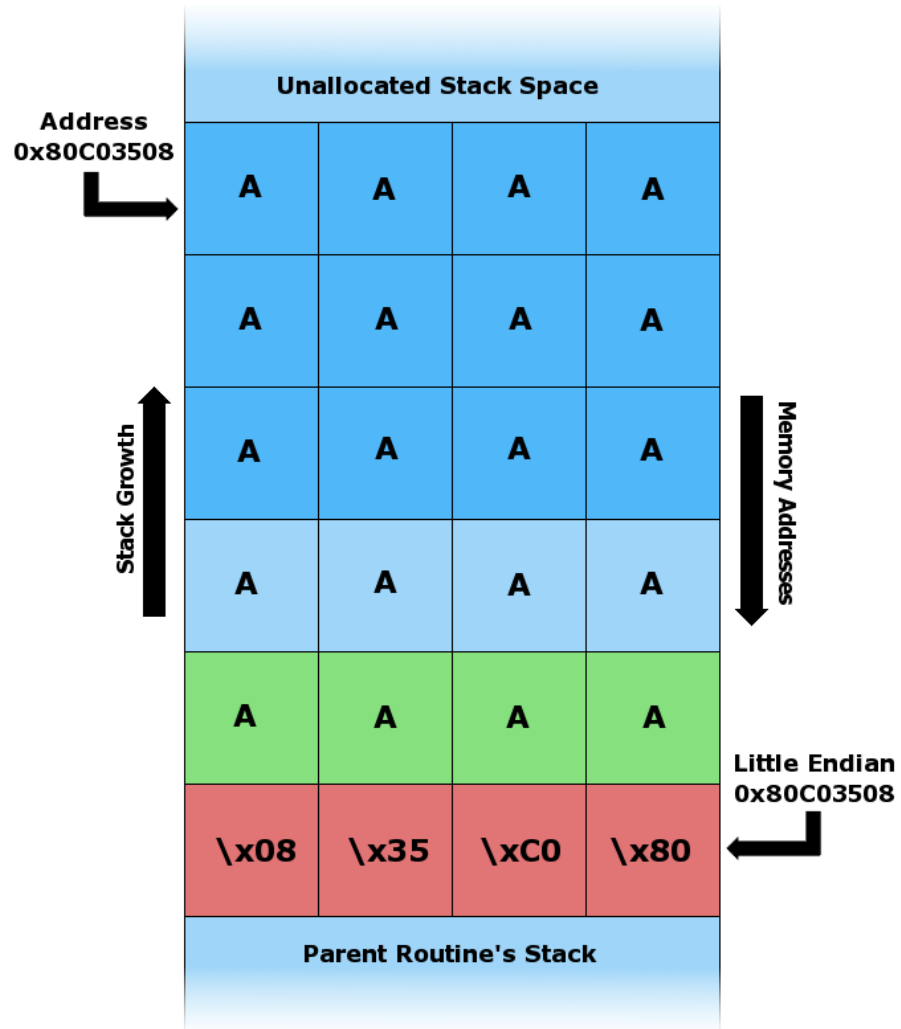
Buffer overflow

# Buffer overflow

- Asking for trouble:

```
void foo(char* bar)
{
    char c[12];
    memcpy(c, bar, strlen(bar));
}
```





CS50 library

# CS50 library

- `#include <cs50.h>`
- `GetString()`, `GetInt()`, `GetFloat()`, ...
- `typedef char* string;`



# CS50 library

```
string GetString(void)
{
    // ...
    while ((c = fgetc(stdin)) != '\n' && c != EOF)
    {
        if (n + 1 > capacity)
            buffer = realloc(buffer, capacity * 2);
        buffer[n++] = c;
    }
    // ...
}
```

# CS50 library

```
int GetInt(void)
{
    while (true)
    {
        string line = GetString();
        if (line == NULL)
            return INT_MAX;

        int n; char c;
        if (sscanf(line, " %i %c", &n, &c) == 1)
        {
            free(line);
            return n;
        }
        else
        {
            free(line);
            printf("Retry: ");
        }
    }
}
```

HTTP

# HTTP

- **H**yper**T**ext **T**ransfer **P**rotocol
- Heart and soul of the world wide web
- **Hypertext**: HTML!
- **Transfer**: requests (from clients) and responses (from servers)
- **Protocol**: set of standards or conventions

# HTTP requests

- GET /me HTTP/1.1

Host: www.facebook.com

...

# HTTP responses

- HTTP/1.1 200 OK

Content-Type: text/html; charset=UTF-8

Server: Apache

...

# HTTP status codes

- 200 OK
- 403 Forbidden
- 404 Not Found
- 500 Internal Server Error
- *Angela's favorite: 418 I'm a teapot*

HTML



# HTML

- **H**yper**T**ext **M**arkup **L**anguage
- Defines **structure** and **semantics** of webpages
  - (Not style—that's CSS!)
- Series of nested tags, or elements

# HTML example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Hello, world!</title>
```

```
    <link href="style.css" rel="stylesheet" />
```

```
    <script src="script.js"></script>
```

```
  </head>
```

```
  <body>
```

```
    <p></p>
```

```
  </body>
```

```
</html>
```

CSS

# CSS

- **Cascading Style Sheets**
- Used to **style** webpages
  - (Webpage **structure** first defined with HTML)

# Including CSS

- Acceptable method: between `<style>` tags

```
<html>
  <head>
    <style>
      /* CSS here */
    </style>
  </head>
  <body></body>
</html>
```

# Including CSS

- Better method: linked external stylesheet

```
<link href="style.css" rel="stylesheet" />
```

- In **style.css**:

```
body
{
    background-color: #000000;
    /* ... */
}
```

# Including CSS

- Bad method: inline styles

```
<div style="background: #000000; color: #FFFFFF"></div>
```

- Avoid inline styling!
  - Why?
  - Muddles **structure** (HTML) and **style** (CSS)

# CSS example

```
body
{
    font-family: "Comic Sans";
}
```

```
#main
{
    margin: 20px;
    text-align: center;
}
```

```
.section
{
    background-color: #61D2D6;
}
```