Building a Better Doorbot

Why?

- Wifi can be spotty
- Zulip on mobile doesn’t have a great UI
- If we have a webservice, we can integrate with twilio

What we need

- Check for a password
- Get the Hacker School phone to open the door
- Wrap this in a web service
- Put it online (outside the Hacker School wifi)

C == low level

```c
int main(int argc, char **argv) {
    puts("Hi there!\nWelcome to Hacker School.");
    if (check_password()) {
        open_door();
    } else {
        puts("I can't let you do that Dave");
    }
    return 0;
}
```

```c
char* SECRET_HACKERSCHOOL_PASSWORD = "hunter1";

int check_password() {
    int success = 0;
    char password_guess[32];
    puts("Enter the password to open the door");
    scanf("%[^\n]", password_guess);
    if (strcmp(password_guess,
        SECRET_HACKERSCHOOL_PASSWORD) == 0) {
        success = 1;
    }
    return success;
}
```

```
void open_door() {
    puts("Opening Hacker School Door.");
    //TODO learn how to solder
}
```

Demo Time

Let’s wrap this in a web service

Python/Flask == easy

```
@app.route("/")
def
    return """
<html>
<body>
Hi, enter a password to unlock the door
<form action="/unlock" method="get">
<input type="text" name="password">
<input type="submit">
</form>
</body>
</html>
"""
```

```
@app.route("/unlock")
def unlock():
    password = request.args.get('password', '')
    result = subprocess.check_output(
"echo {0} | ./doorbot; exit 0”.format(password),
                shell=True, stderr=subprocess.STDOUT)
    result = result.replace("\n","<br>")
    return result
```

Demo Time

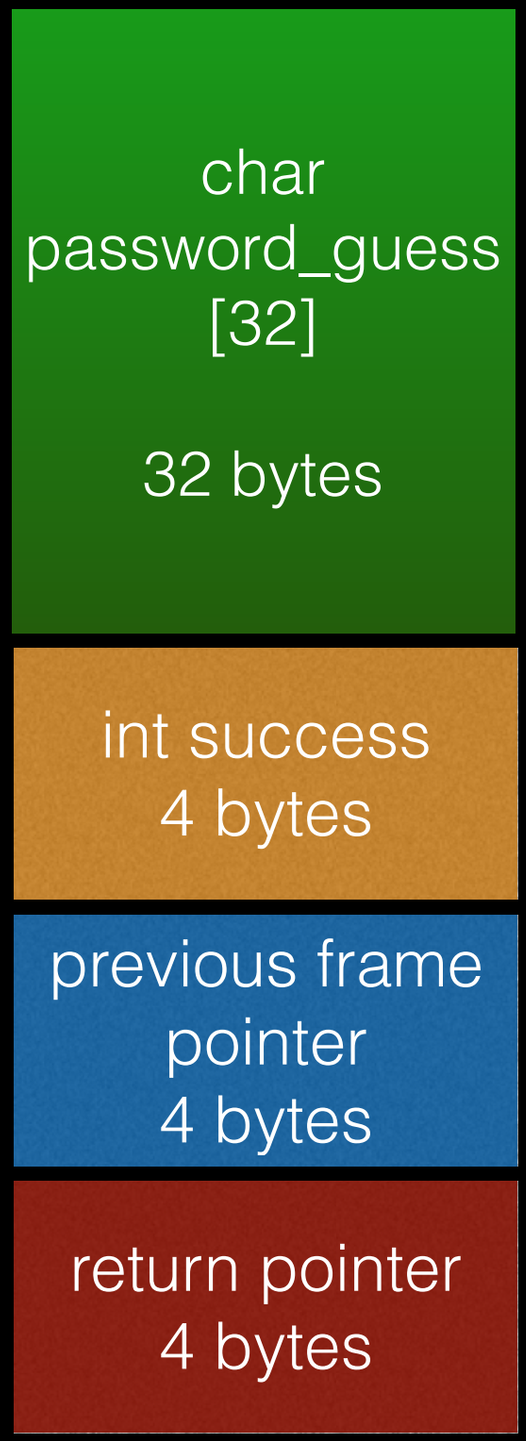
Well that was awkward

This talk is actually about exploiting memory corruption vulnerabilities

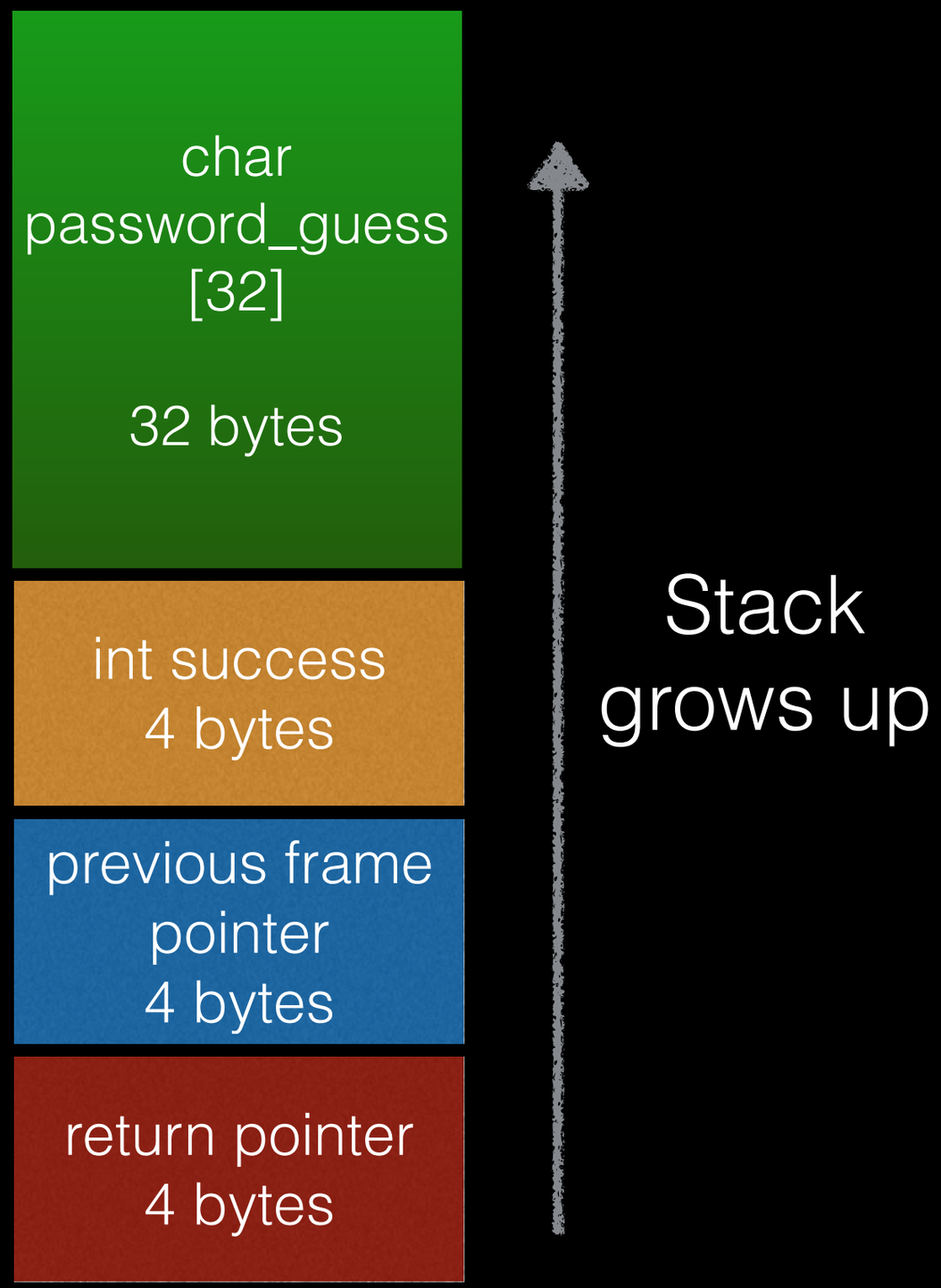
A segmentation fault occurs when a program attempts to access a memory location that it is not allowed to access, or attempts to access a memory location in a way that is not allowed (for example, attempting to write to a read-only location, or to overwrite part of the operating system).

Wikipedia (https://en.wikipedia.org/wiki/Segmentation_fault)

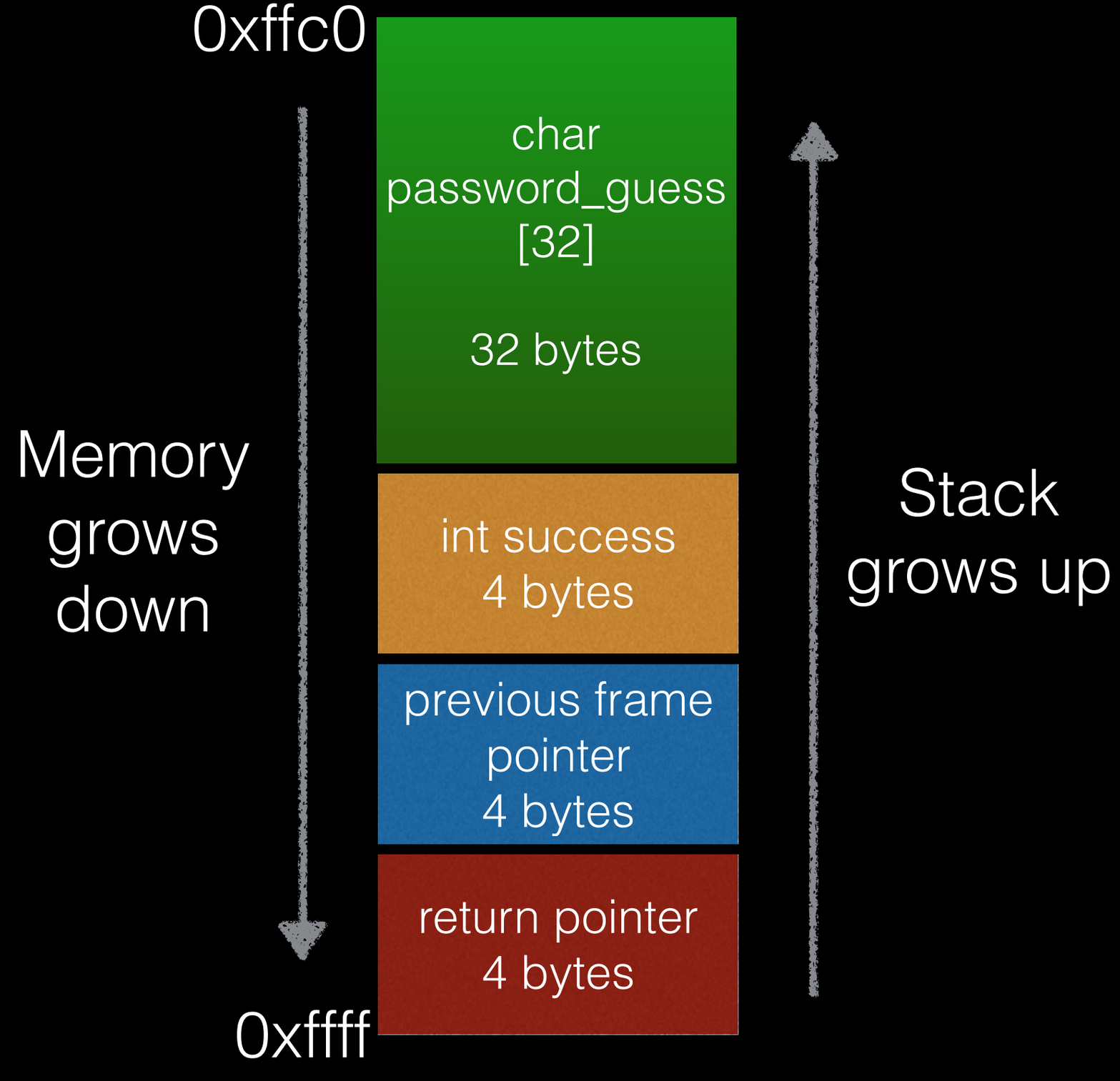
The Stack


char
password_guess
[32]
32 bytes
int success
4 bytes
previous frame
pointer
4 bytes
return pointer
4 bytes

The Stack


char
password_guess
[32]
32 bytes
int success
4 bytes
previous frame
pointer
4 bytes
return pointer
4 bytes
Stack
grows up

The Stack

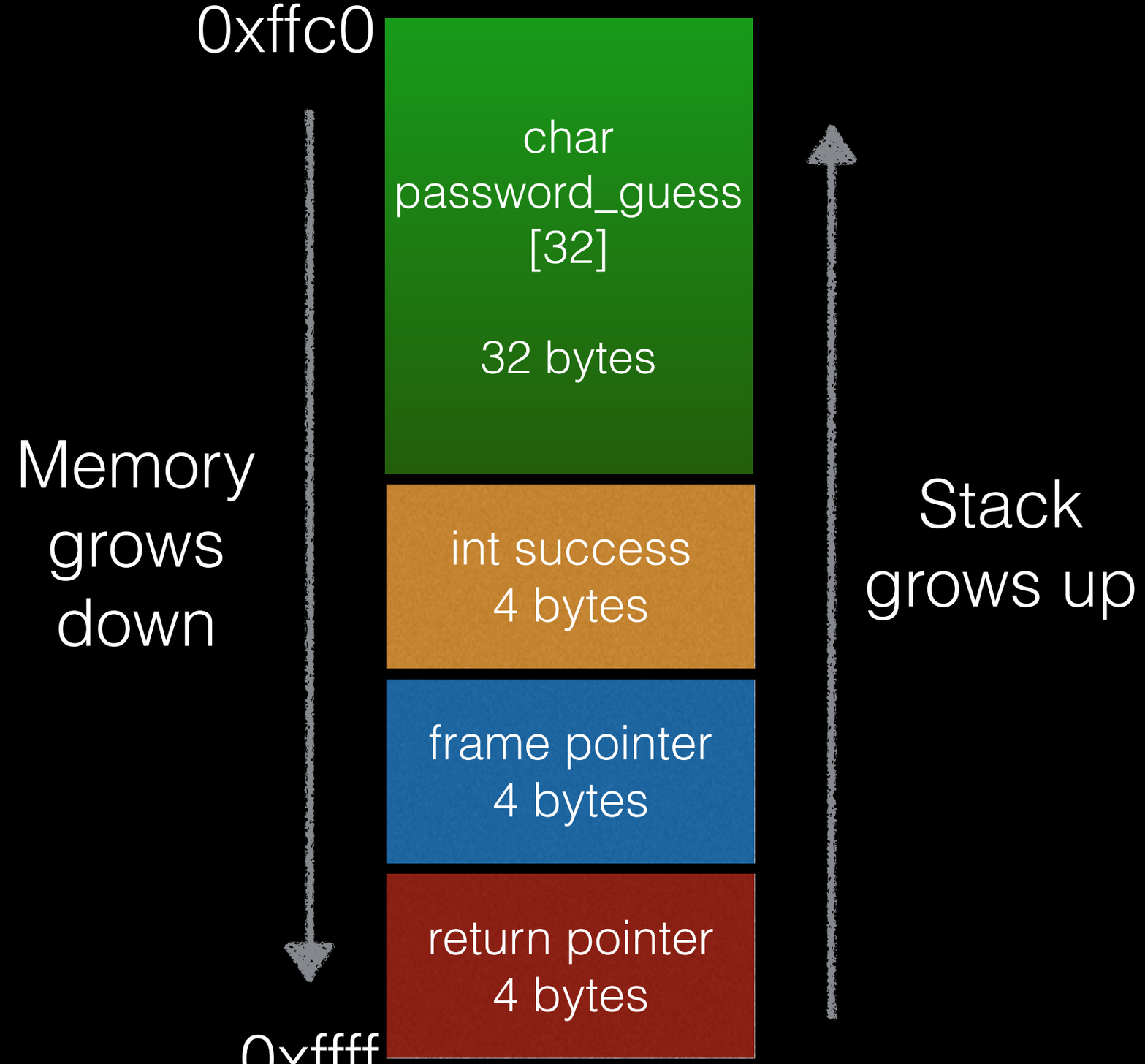

0xffc0
char password_guess [32]
32 bytes
int success
4 bytes
previous frame pointer
4 bytes
return pointer
4 bytes
0xffff
Memory grows down
Stack grows up

```c
char* SECRET_HACKERSCHOOL_PASSWORD = "hunter1";

int check_password() {
    int success = 0;
    char password_guess[32];
    puts("Enter the password to open the door");
    scanf("%[^\n]", password_guess);
    if (strcmp(password_guess,
        SECRET_HACKERSCHOOL_PASSWORD) == 0) {
        success = 1;
    }
    return success;
}
```

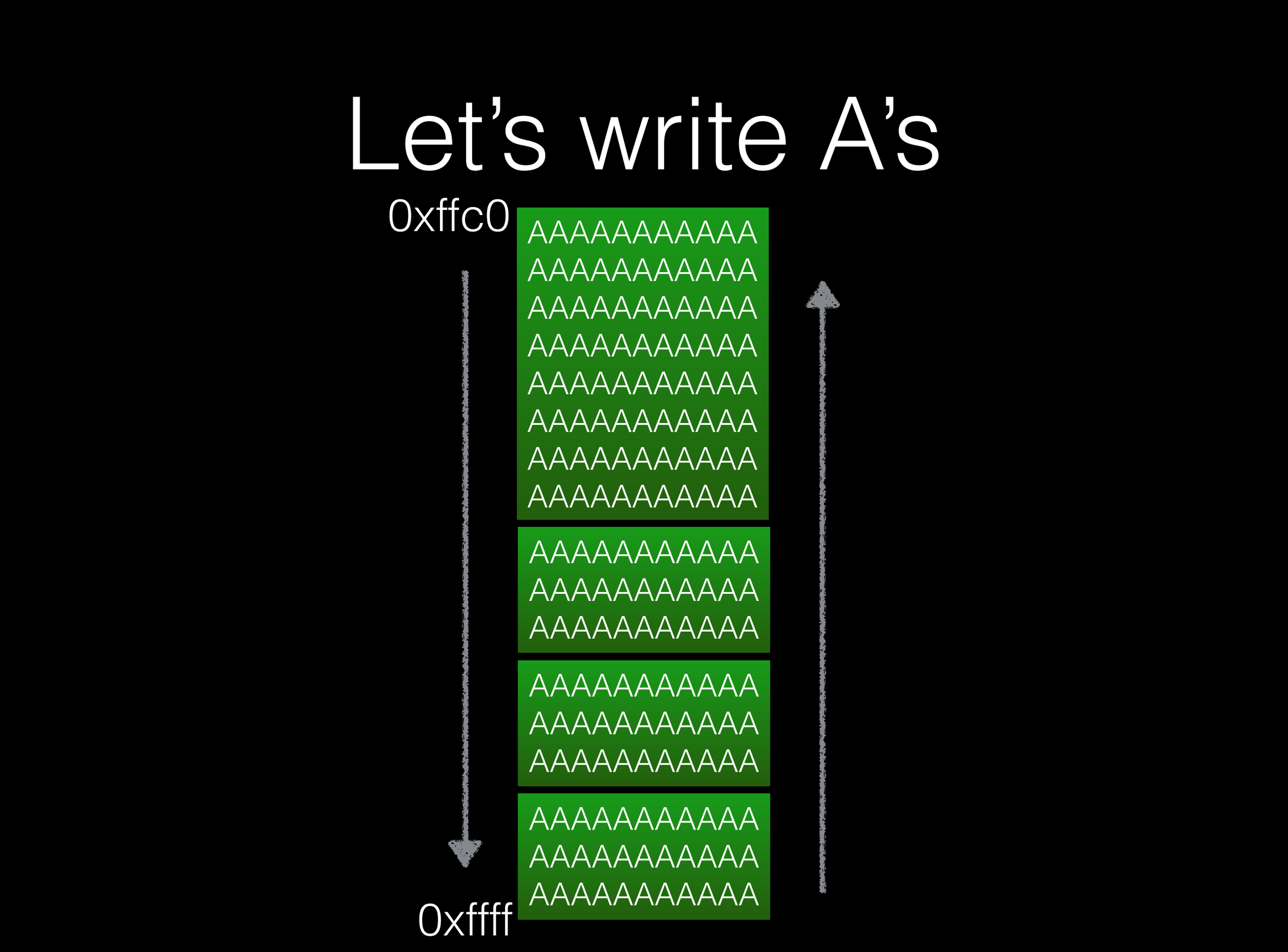
```
char* SECRET_HACKERSCHOOL_PASSWORD = "hunter1";

int check_password() {
    int success = 0;
    char password_guess[32];
    puts("Enter the password to open the door");
    scanf("%[^\n]", password_guess);
    if (strcmp(password_guess,
        SECRET_HACKERSCHOOL_PASSWORD) == 0) {
        success = 1;
    }
    return success;
}
```

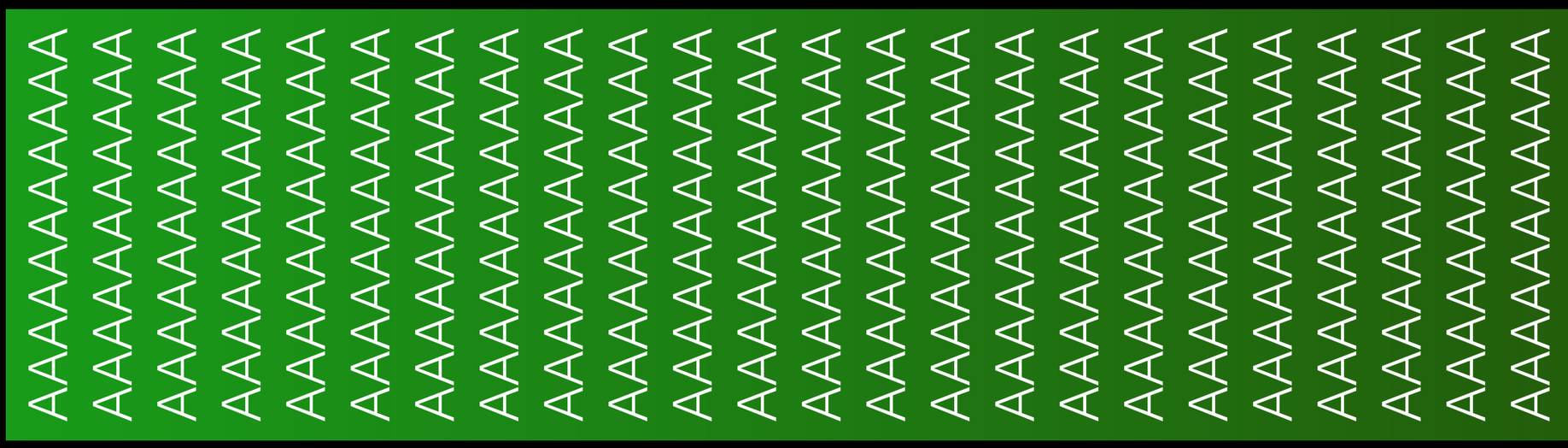
```c
char* SECRET_HACKERSCHOOL_PASSWORD = "hunter1";

int check_password() {
    int success = 0;
    char password_guess[32];
    puts("Enter the password to open the door");
    scanf("%[^\n]", password_guess);
    if (strcmp(password_guess,
        SECRET_HACKERSCHOOL_PASSWORD) == 0) {
        success = 1;
    }
    return success;
}
```

The Stack


0xffc0
char password_guess [32]
32 bytes
int success
4 bytes
frame pointer
4 bytes
return pointer
4 bytes
0xffff
Memory grows down
Stack grows up

Let’s write A’s

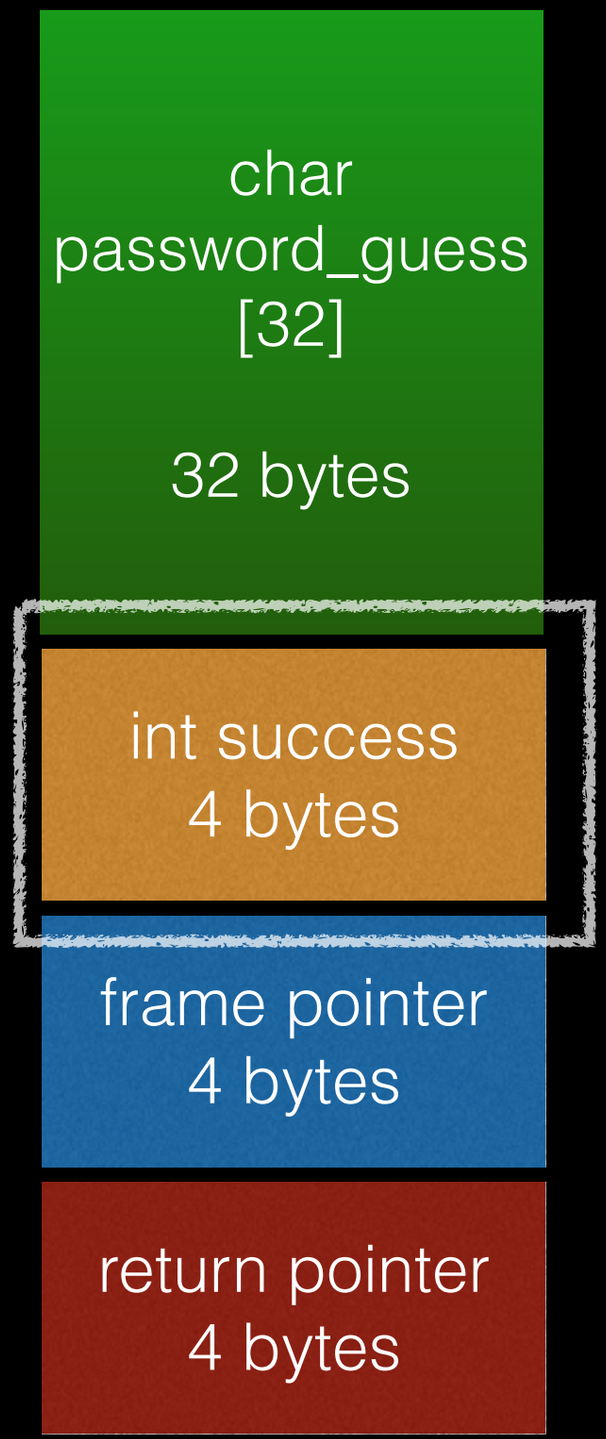




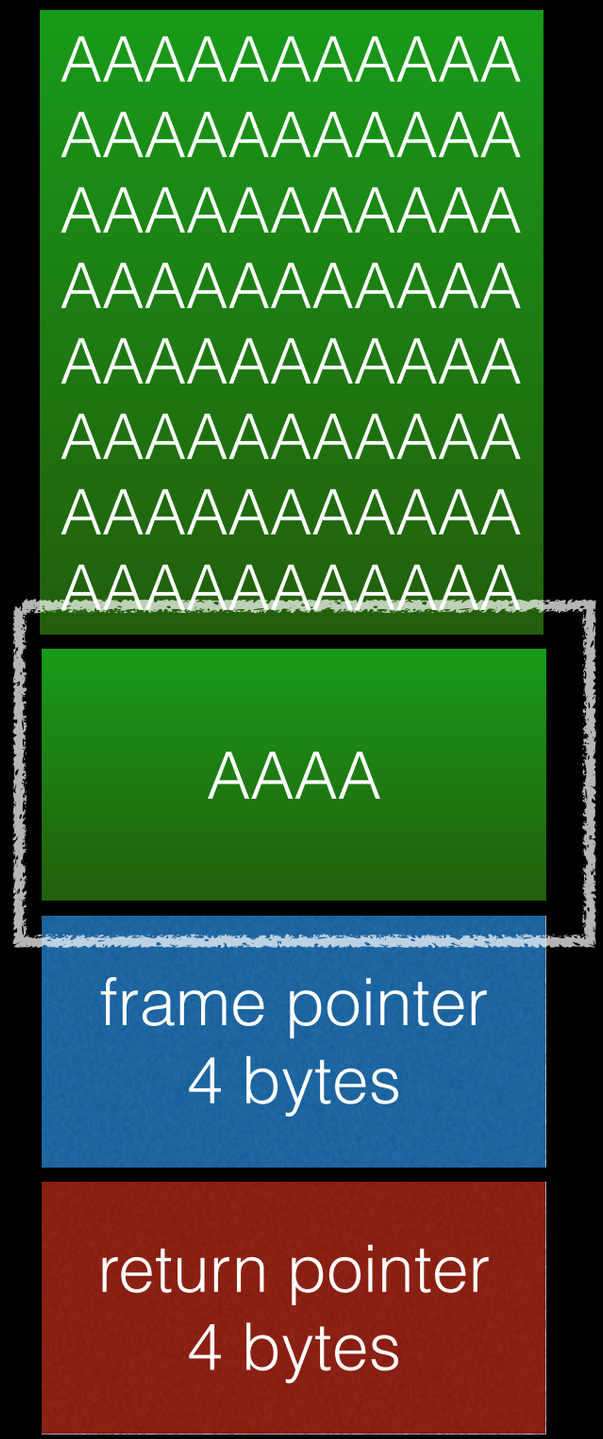

stackoverflow

```c
char* SECRET_HACKERSCHOOL_PASSWORD = "hunter1";

int check_password() {
    int success = 0;
    char password_guess[32];
    puts("Enter the password to open the door");
    scanf("%[^\n]", password_guess);
    if (strcmp(password_guess,
               SECRET_HACKERSCHOOL_PASSWORD) == 0) {
        success = 1;
    }
    return success;
}
```

The Stack


char
password_guess
[32]
32 bytes
int success
4 bytes
frame pointer
4 bytes
return pointer
4 bytes

The Stack


AAAAAAAAAAAA
AAAAAAAAAAAA
AAAAAAAAAAAA
AAAAAAAAAAAA
AAAAAAAAAAAA
AAAAAAAAAAAA
AAAAAAAAAAAA
AAAAAAAAAAAA
AAAA
frame pointer
4 bytes
return pointer
4 bytes

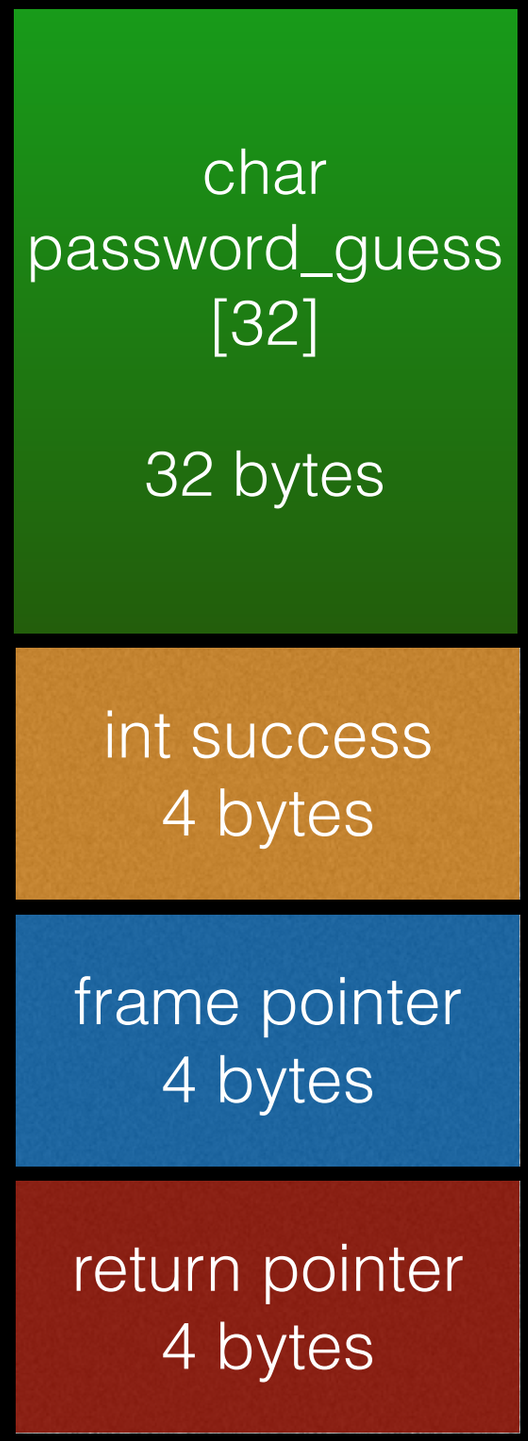
Demo

Fixing it

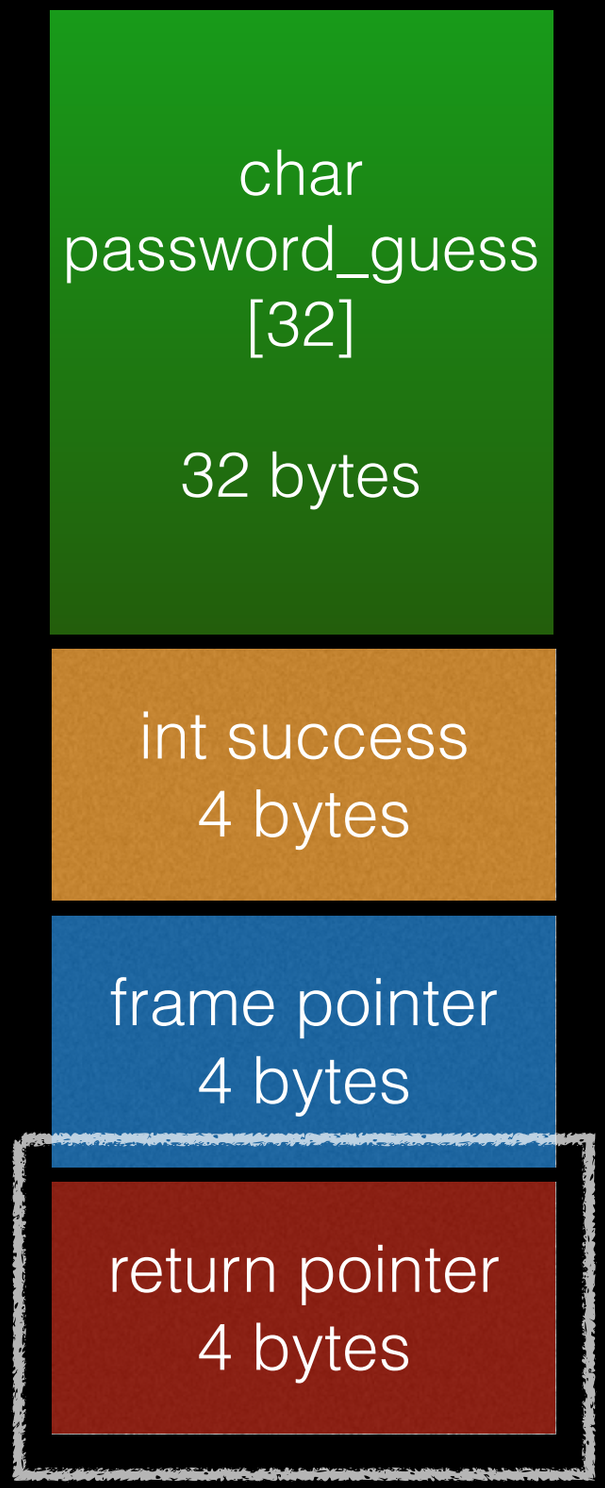
```c
int check_password() {
    char password_guess[32];
    puts("Enter the password to open the door");
    scanf("%[^\n]", password_guess);
    if (strcmp(password_guess,
        SECRET_HACKERSCHOOL_PASSWORD) == 0) {
        return 1;
    } else {
        return 0;
    }
}
```

This is no good

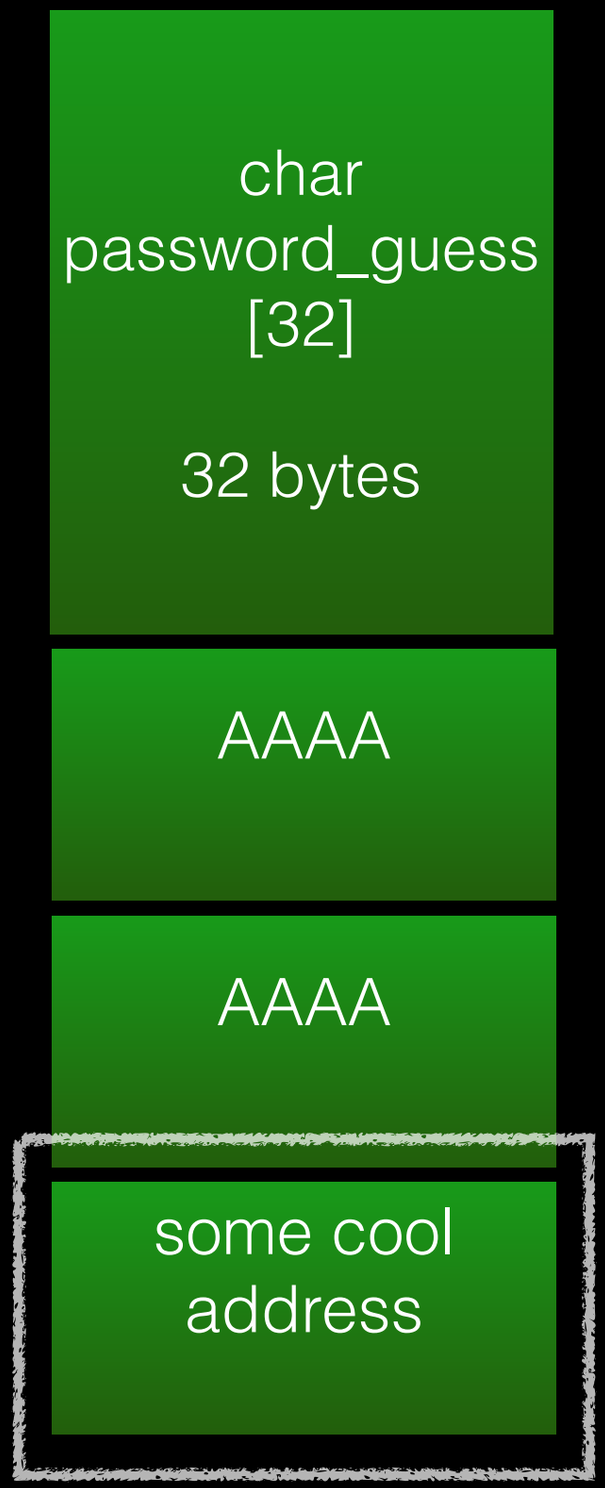
The Stack


char
password_guess
[32]
32 bytes
int success
4 bytes
frame pointer
4 bytes
return pointer
4 bytes

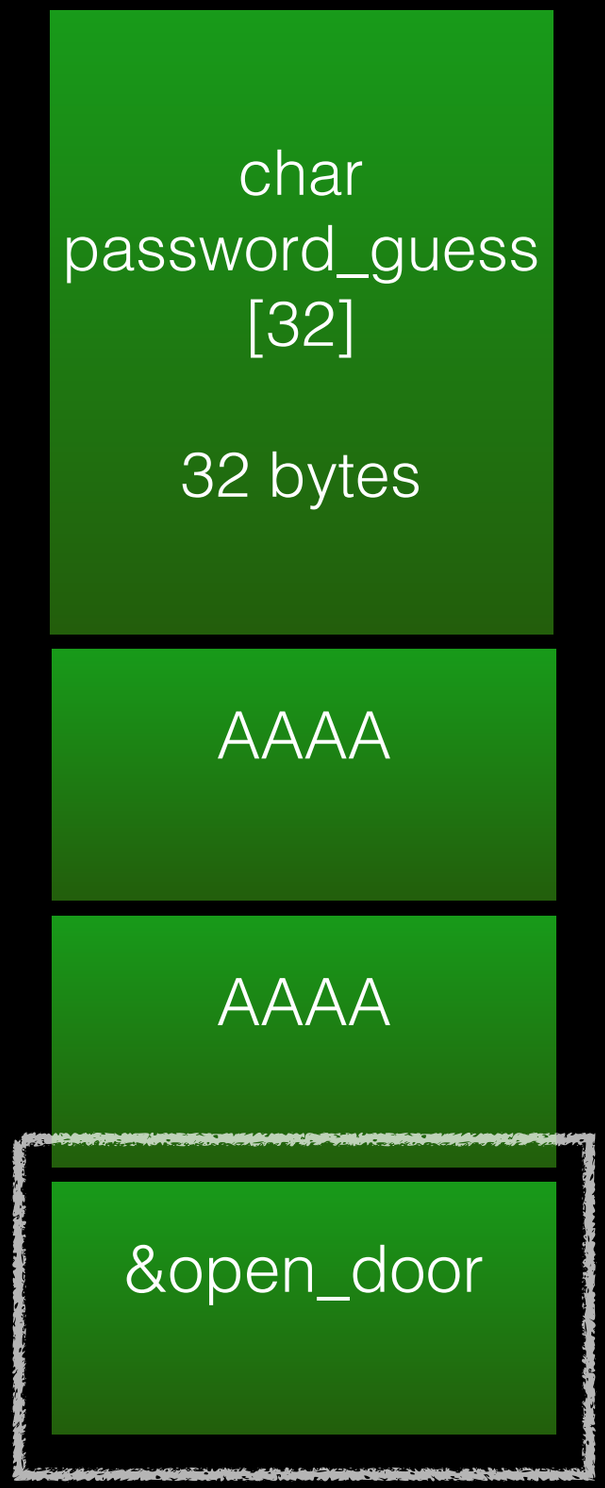
The Stack


char
password_guess
[32]
32 bytes
int success
4 bytes
frame pointer
4 bytes
return pointer
4 bytes

The Stack



The Stack



Demo

Hacking like it's 1997

ASLR

```
echo 0 | sudo tee /proc/sys/kernel/randomize_va_space
```

Stack Canaries

gcc -o doorbot2 -fno-stack-protector -g doorbot2.c

DEP

~~DEP~~

Doesn’t actually affect us

Bonus

```
@app.route("/unlock")
def unlock():
    password = request.args.get('password', '')
    result = subprocess.check_output(
"echo {0} | ./doorbot; exit 0”.format(password),
          shell=True, stderr=subprocess.STDOUT)
    result = result.replace("\n","<br>")
    return result
```