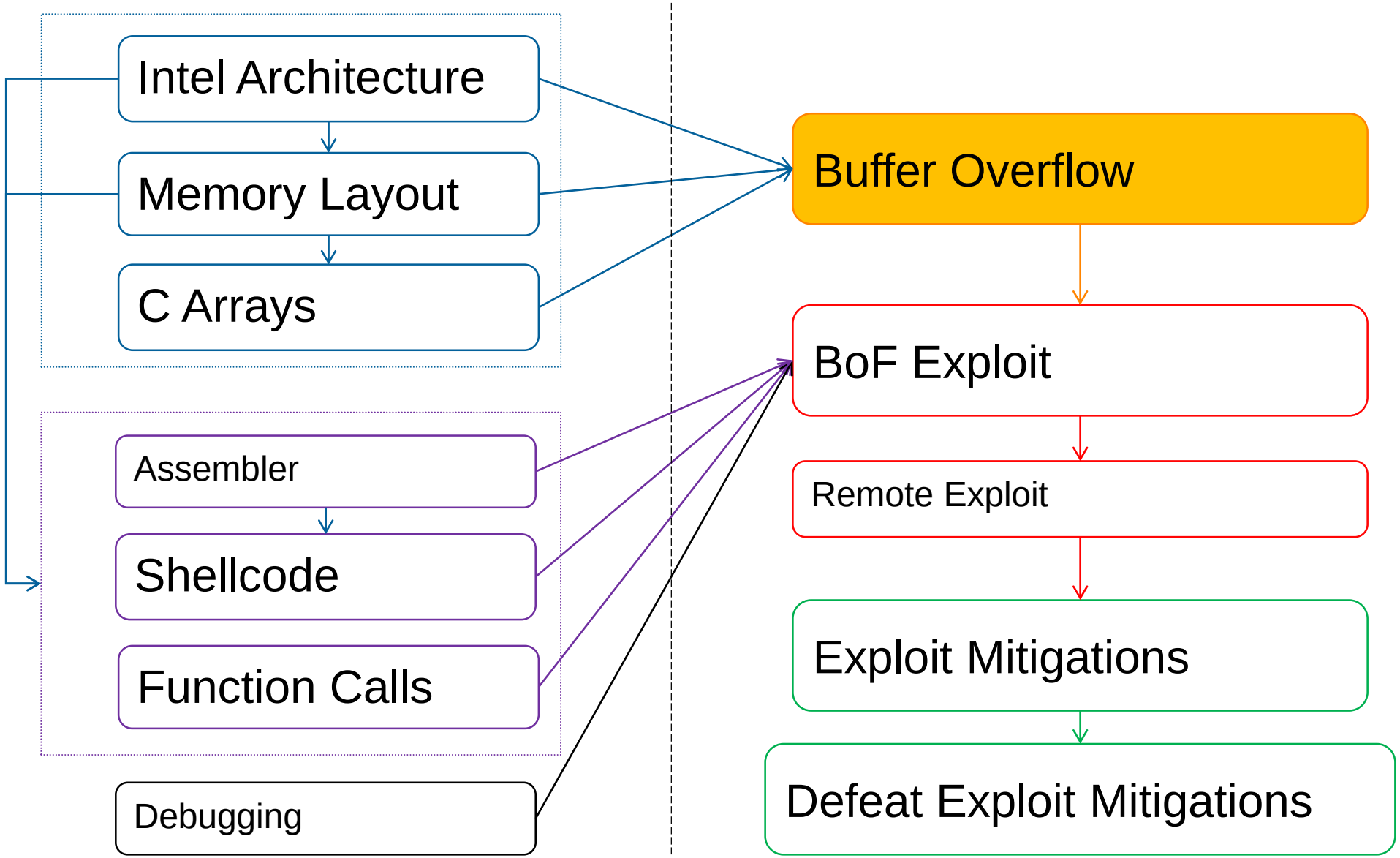
Stack Buffer Overflow

Content


Intel Architecture
Memory Layout
C Arrays
Assembler
Shellcode
Function Calls
Debugging
Buffer Overflow
BoF Exploit
Remote Exploit
Exploit Mitigations
Defeat Exploit Mitigations

Buffer Overflow

Without exploit

Buffalo Overflow

Buffer Overflow

- Challenge10

```
# ./challenge10 <username> <password>

# ./challenge10 someusername somepassword
You are not admin.
Lame.
```

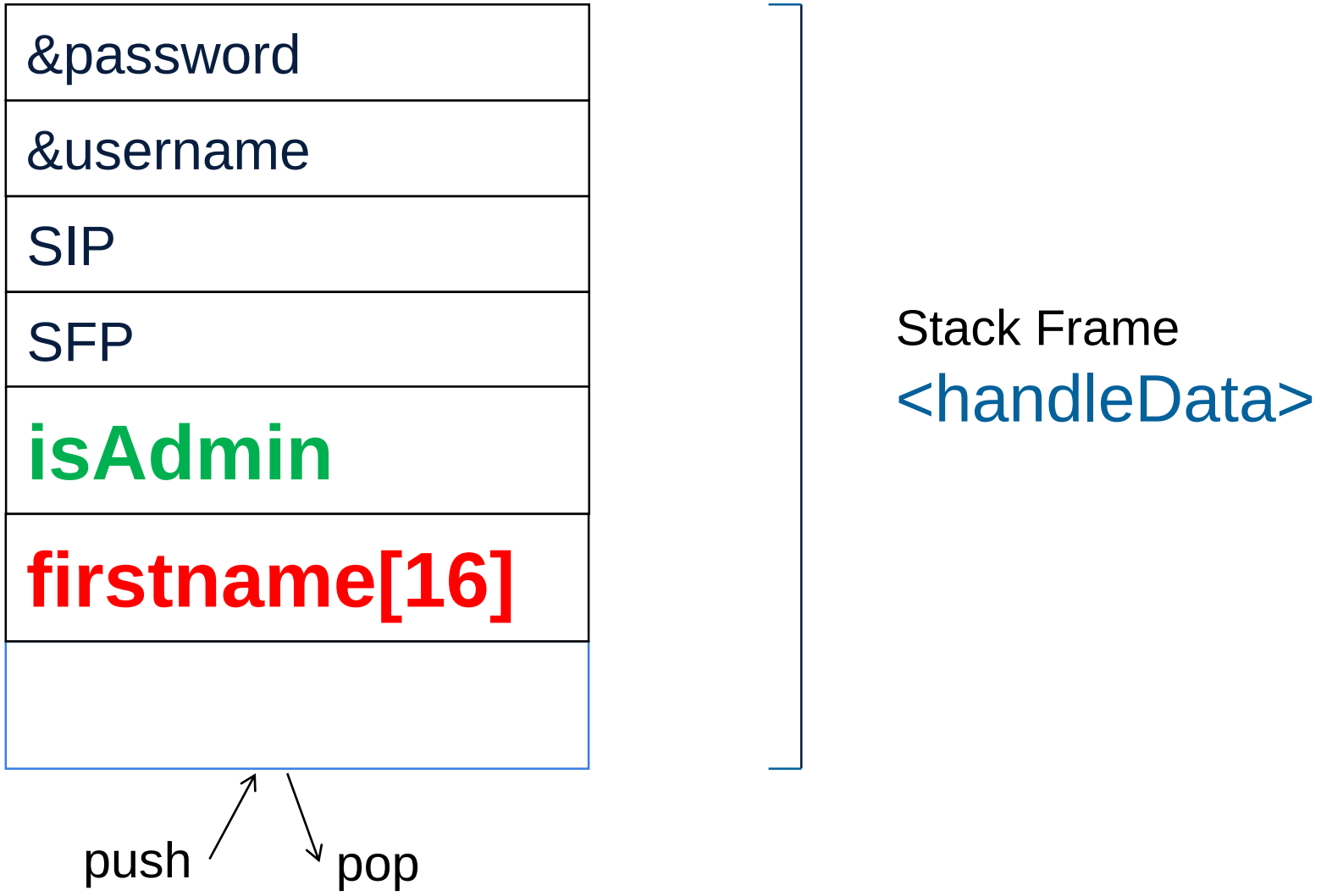
Buffer Overflow

```
void handleData(char *username, char *password) {
	int isAdmin = 0;
	char firstname[16];

	isAdmin = checkPassword(password);
	strcpy(firstname, username);

	if(isAdmin > 0) {
		printf("You ARE admin!");
	} else {
		printf("You are not admin.\nLame.\n");
	}
}
```

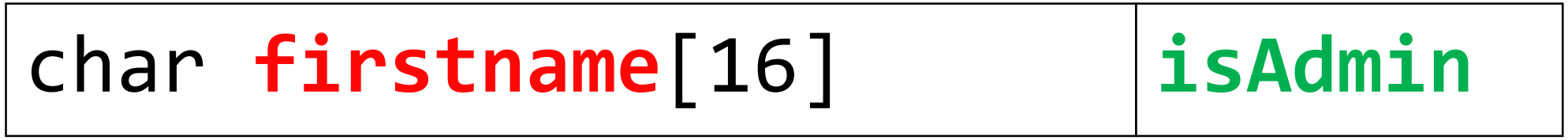
Buffer Overflow

```
const char *adminHash = "$6$saaaaalty$cjw9qyA..";

int checkPassword(char *password) {
      char *hash;

      hash = crypt(password, "$6$saaaaalty");

      if (strcmp(hash, adminHash) == 0) {
            return 1;
      } else {
            return 0;
      }
}
```

Buffer Overflow

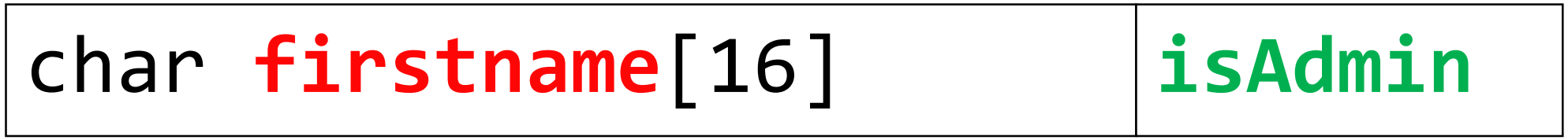


Buffer Overflow - Basic Layout



strcpy(**firstname**, “AAAA AAAA AAAA AAAA”);

Buffer Overflow - Basic Layout



strcpy(**firstname**, “AAAA AAAA AAAA AAAA **B**”);

Buffer Overflow: handleData()

```
void handleData(char *username, char *password) {
      int isAdmin = 0;
      char firstname[16];
(0)
      isAdmin = checkPassword(password);
(1)
      strcpy(firstname, username);
(2)
      if(isAdmin > 0) {
            printf("You ARE admin!");
      } else {
            printf("You are not admin.\nLame.\n");
      }
}
```

Buffer Overflow

char firstname[16]	isAdmin

Buffer Overflow

char firstname[16]	isAdmin
0 <undefined>	<undef>

Buffer Overflow

	char firstname[16]	isAdmin
0	<undefined>	<undef>
1	<undefined>	0x00000000

Buffer Overflow

	char firstname[16]	isAdmin
0	<undefined>	<undef>
1	<undefined>	0x00000000
2	AAAAAAAAAAAAAAAAAAAAAAAA	0x00000000

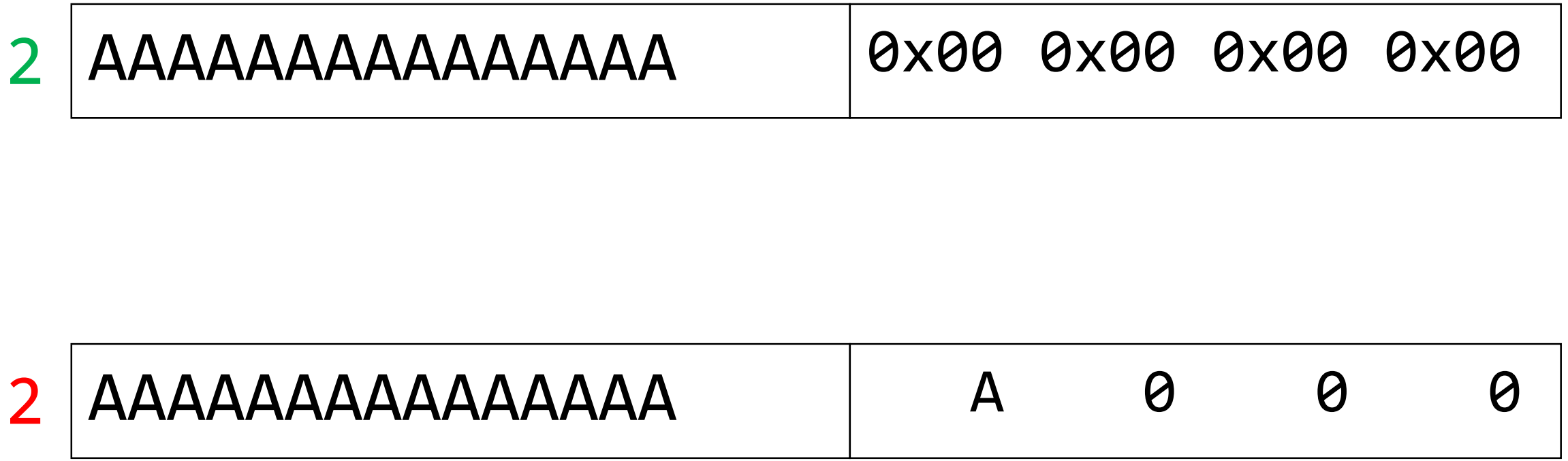
Buffer Overflow

	char firstname[16]	isAdmin
0	<undefined>	<undef>
1	<undefined>	0x00000000
2	AAAAAAAAAAAAAAAAAAAAAAAA	0x00000000
2	AAAAAAAAAAAAAAAAAAAAAAAA	0x00000041

Buffer Overflow

2	AAAAAAAAAAAAAAAA	0x00 0x00 0x00 0x00

Buffer Overflow



Buffer Overflow

2	AAAAAAAAAAAAAAAA	0x00 0x00 0x00 0x00

2	AAAAAAAAAAAAAAAA	A 0 0 0
2	AAAAAAAAAAAAAAAA	0x41 0x00 0x00 0x00

Buffer Overflow

```
./challenge1 compass superpassword
You are not admin.

./challenge1 01234567890123456790123456 78 test
You are not admin.

./challenge1 01234567890123456790123456 78A test
You ARE admin!
isAdmin: 0x41

./challenge1 01234567890123456790123456 78AB test
You ARE admin!
isAdmin: 0x4241
```

Typical bugs

Typical bugs

Solaris Xsun:

```
buf = malloc(1024);
strcpy(buf, user_supplied)
```

Solaris Login:

```
buf = (char **) malloc(BUF_SIZE);
while (user_buf[i] != 0) {
     buf[i] = malloc(strlen(user_buf[i]) + 1);
  i++
}
```

Typical bugs

Samba:

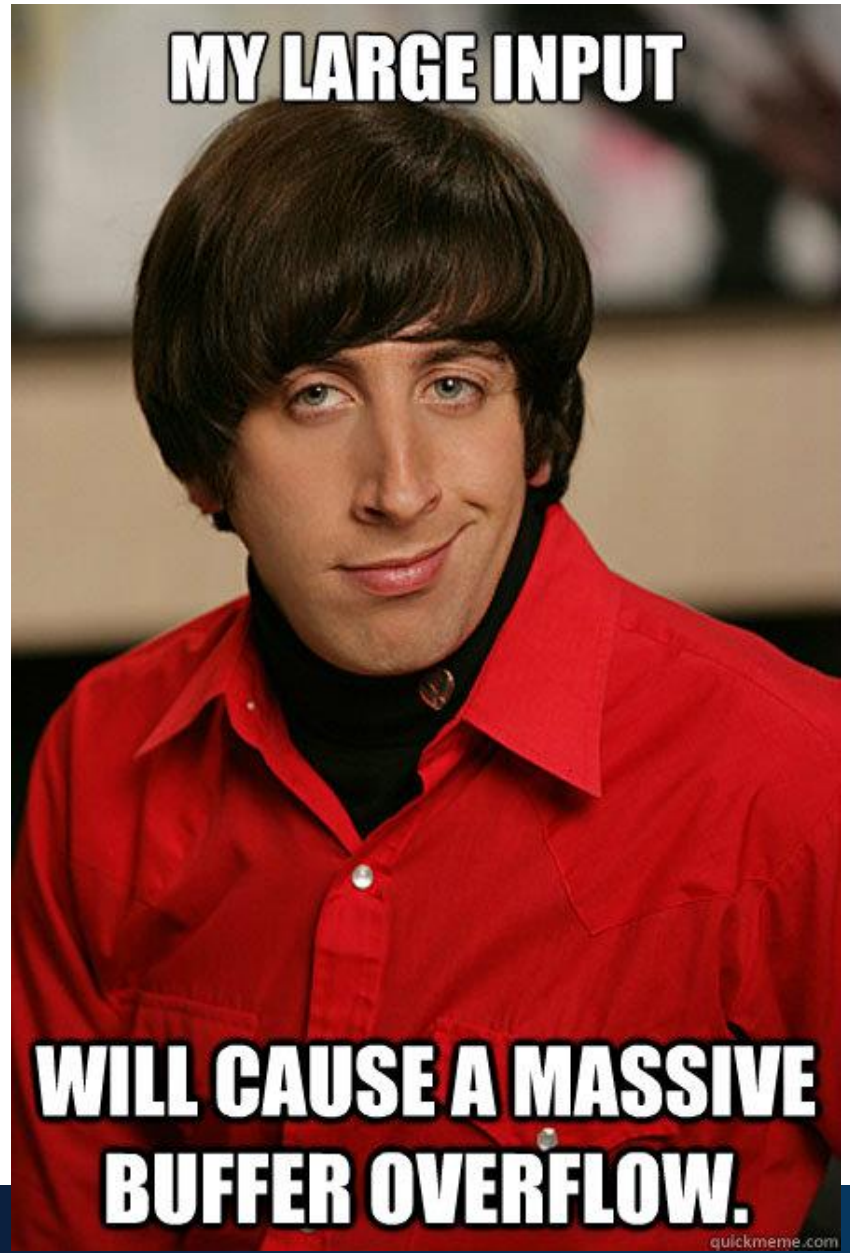
```
memcpy(
   array[user_supplied_int],
      user_supplied_buffer,
      user_supplied_int2)
```

Buffer Overflow

Recap:

- Local variables of a function (buffers) are allocated adjectant to each other
- One after another, as written in the source code (first initialized first allocated)

Buffer Overflow



References

References:

- https://www.uperesia.com/buffer-overflow-explained
- https://www.youtube.com/watch?v=1S0aBV-Waeo

`Buffer Overflow Attack - Computerphile`