

Brent Turner

CSC 495

Lab 1

## Buffer Overflow

TARGET 1: (target1.c found at the bottom)

A buffer Overflow is present when, if the exploit exists, a piece of memory, a buffer, is given more data to which it can actually hold, and then proceeds to fill that data in other places of the buffer, that are most defiantly not intended, to the point where, if the code is present, the hacker can get the program to move to a specific return address in the code.

### 1 & 2

---

**Parameters:** password – (size of 8 \* size of pointer 4) = 32 bytes

**return address:** 0x000005bc (successfulLogin) 4bytes

0x000005e7 (incorrectPasswordError) – 4bytes

**local variable:** usersPassword – (size of 8 \* size of pointer 4) = 32 bytes

### 3

---

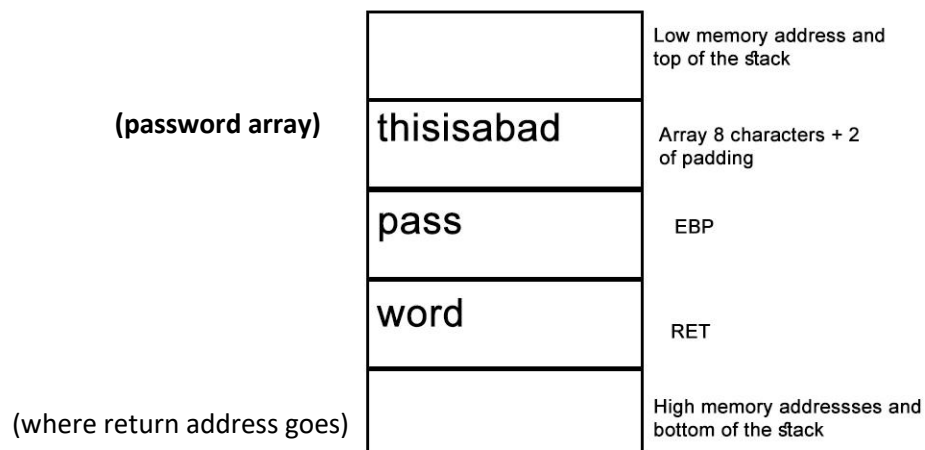
To get the overflow buffer one must type in 18 characters then the shell code

(8 (array size) + 2 (padding) + 4 (size of EBP) + 4 (size of RET)) These are the base 18 characters and then the shell code can proceed

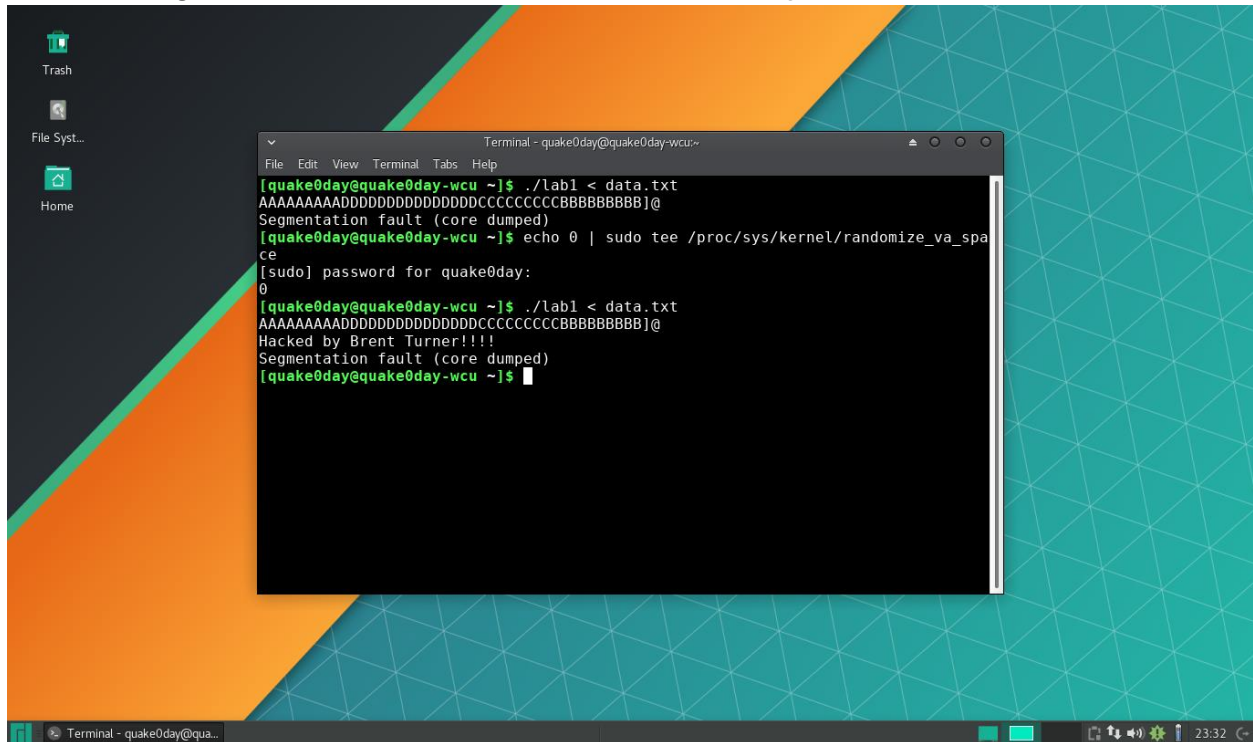
### 4

---

The overflow direction is down words to get access to memory address



DATA.txt String: AAAAAAAAAADDDDDDDDDDDDDDDCCCCCCCCBBBBBBBBBB] @



Target1.c

```
#include<stdio.h>
```

```
#include <string.h>
```

```
void successfulLogin();
```

```
void incorrectPasswordError();
```

```
main()
```

```
{
```

```
    char usersPassword[8] = "secret99";
```

```
    //need to make a buffer/stack in order to overflow
```

```
    char password[8];
```

```
    gets(password);
```

```
    if(password == usersPassword)
    {
        successfulLogin();
    }
    else
    {
        incorrectPasswordError();
    }
}
```

```
void successfulLogin()
{
    printf("You password is correct, You have now logged in");
}
```

```
void incorrectPasswordError()
{
    printf("The entered password was incorrect, please try again");
}
```