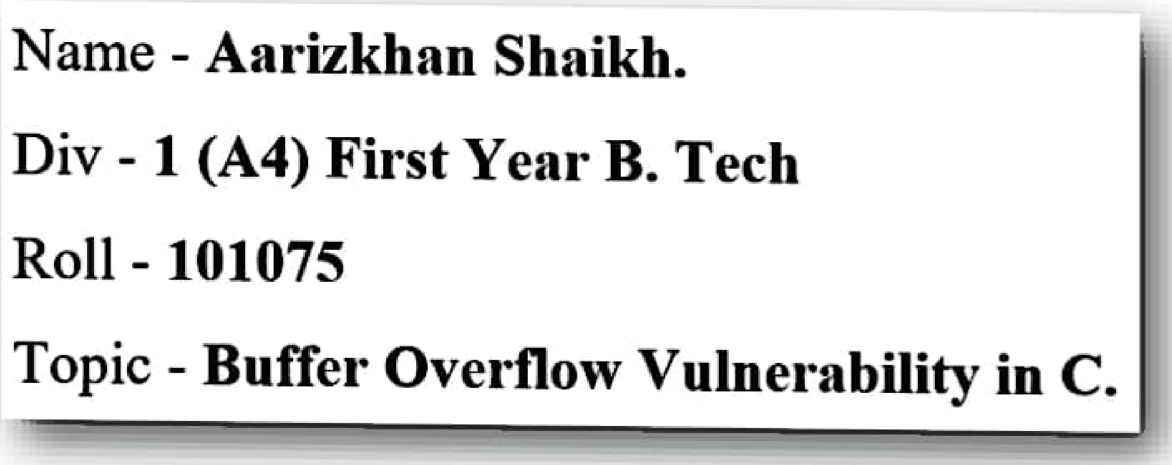


Name - **Aarizkhan Shaikh.**

Div - **1 (A4) First Year B. Tech**

Roll - **101075**

Topic - **Buffer Overflow Vulnerability in C.**

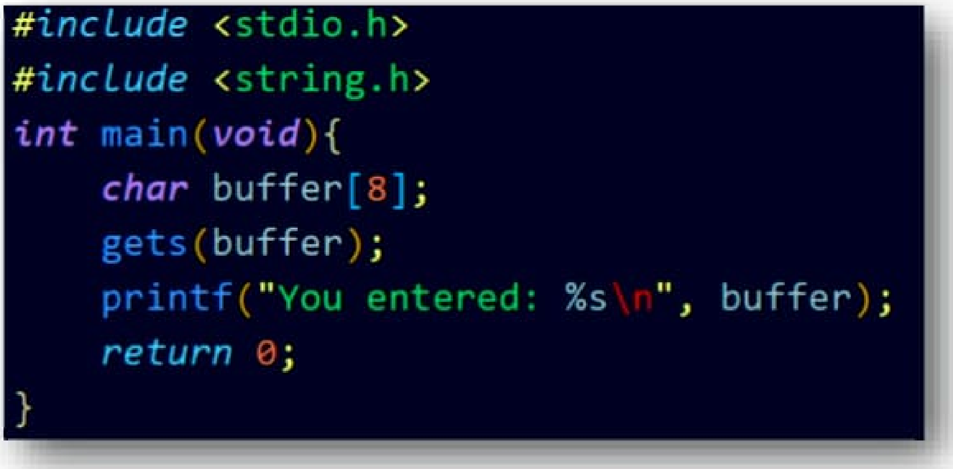
Active Learning Report

Topic: Buffer Overflow Vulnerability in C.

A buffer overflow is a type of vulnerability that can occur in software programs written in the C programming language. This vulnerability occurs when a program attempts to write data to a memory buffer, but the data is larger than the buffer can hold. The result is that the extra data overwrites adjacent memory locations, potentially causing the program to crash or allowing an attacker to execute malicious code.

One common way that buffer overflows can occur is through the use of the gets() function in C. This function is used to read a line of text from standard input (usually the keyboard) and store it in a character array. However, gets() does not check the length of the input, so if the user enters more data than the array can hold, a buffer overflow will occur.

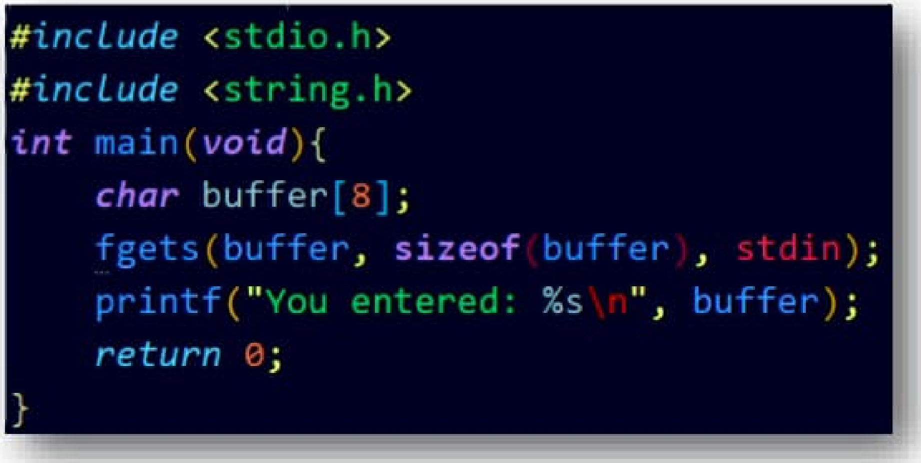
For example, consider the following code:



```c
#include <stdio.h>
#include <string.h>
int main(void){
    char buffer[8];
    gets(buffer);
    printf("You entered: %s\n", buffer);
    return 0;
}
```

In this code, the buffer array is declared with a size of 8 characters. This means that it can hold a string of up to 7 characters, plus a null terminator. However, if the user enters a string longer than 7 characters, the gets() function will write past the end of the buffer array, potentially causing a buffer overflow.

To avoid this vulnerability, it is recommended to use the fgets() function instead of gets(). This function takes an additional argument that specifies the maximum number of characters to read from the input, preventing buffer overflows from occurring. For example, the code above could be rewritten as follows:



```
#include <stdio.h>
#include <string.h>
int main(void){
    char buffer[8];
    fgets(buffer, sizeof(buffer), stdin);
    printf("You entered: %s\n", buffer);
    return 0;
}
```

In this revised code, the fgets() function is used to read a maximum of 8 characters (the size of the buffer array) from standard input. This prevents a buffer overflow from occurring, even if the user enters a string longer than 7 characters.

Takeaways –

- Buffer Overflow in C.
- Secure Coding Practices.
- Cybersecurity Awareness.
- Real Life Applications & Examples.
- Info on some standard C library functions.

Summary:
The use of the gets() function in C can result in buffer overflow vulnerabilities. To avoid these vulnerabilities, it is recommended to use the fgets() function instead. This function prevents buffer overflows by limiting the amount of data that can be read from the input.