Network Attacks
Buffer overflow Attack

- Buffer overflow is common in program when data exceeds the boundary of the buffer.
- Attacker tries to store too much information in an undersized receptacle – most exploits are based on buffer overflow

```c
char *ptr = (char*) malloc(10);
ptr[10] = 'c';
```

Cause crash/seg fault/etc

- Overflow the stack or heap
Buffer Overflow Example

```c
int main(void)
{
    char buff[15];
    int pass = 0;

    printf("\n Enter the password : \n");
    gets(buff);

    if(strcmp(buff, "thegeekstuff"))
    {
        printf("\n Wrong Password \n");
    }
    else
    {
        printf("\n Correct Password \n");
        pass = 1;
    }

    if(pass)
    {
        /* Now Give root or admin rights to user*/
        printf("\n Root privileges given to the user \n");
    }

    return 0;
}
```

Compare password if correct – thegeekstuff – escalate privilege

Gets() function does not check array bounds
Length of string could be greater than the size of buffer

Supply input length > buffer size (20 ‘h’ there);
Overwrites the memory of integer ‘pass’
‘pass’ becomes nonzero, 8 bytes of excessive ‘h’ -> int 8 bytes

Use fgets() instead of gets(), strncmp() instead of strcmp(), strncpy() instead of strcpy()
Stack Overflow

- Smashes the return address of a function to some custom code

```c
#include <stdio.h>

void secretFunction()
{
    printf("Congratulations!\n");
    printf("You have entered in the secret function!\n");
}

void echo()
{
    char buffer[20];

    printf("Enter some text:\n");
    scanf("%s", buffer);
    printf("You entered: %s\n", buffer);
}

int main()
{
    echo();

    return 0;
}
```