

## Aim: Buffer OverFlow Attack vulnerability detection using splint.

### CODE:

```
#include<stdio.h>
int main(){

    // declaring a character of fixed size that can store name of the user
    // This is the array which can be overflowed.
    char name[10];

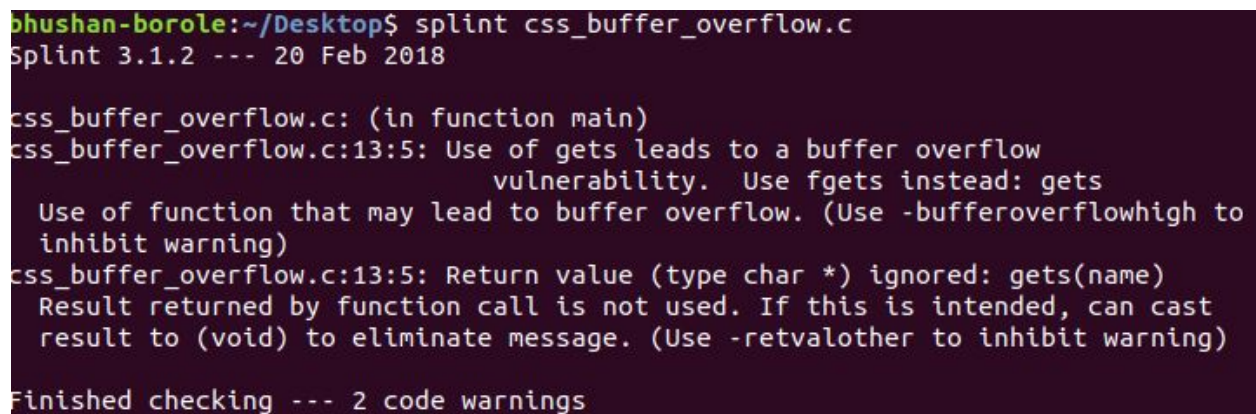
    // prompt user for the name;
    printf("Enter your name: ");

    // using gets to enable user to allow give a string bigger than the
    // buffer size of name array.
    gets(name);

    // try to print the name
    // This will result in a buffer overflow error when user has entered a
    // string of length greater than the specified buffer array.
    //
    // This will result in a segmentation error causing dumped core in
    // case of buffer overflow attack.
    printf("Hello, %s", name);

    // using int main and return 0 because,
    // i like it and also splint told to do so.
    return 0;
}
```

### OUTPUT:



```
bhushan-borole:~/Desktop$ splint css_buffer_overflow.c
Splint 3.1.2 --- 20 Feb 2018

css_buffer_overflow.c: (in function main)
css_buffer_overflow.c:13:5: Use of gets leads to a buffer overflow
                           vulnerability. Use fgets instead: gets
    Use of function that may lead to buffer overflow. (Use -bufferoverflowhigh to
    inhibit warning)
css_buffer_overflow.c:13:5: Return value (type char *) ignored: gets(name)
    Result returned by function call is not used. If this is intended, can cast
    result to (void) to eliminate message. (Use -retvalother to inhibit warning)

Finished checking --- 2 code warnings
```

## 2) Simulating DOS attack using hping3 and monitor using wireshark.

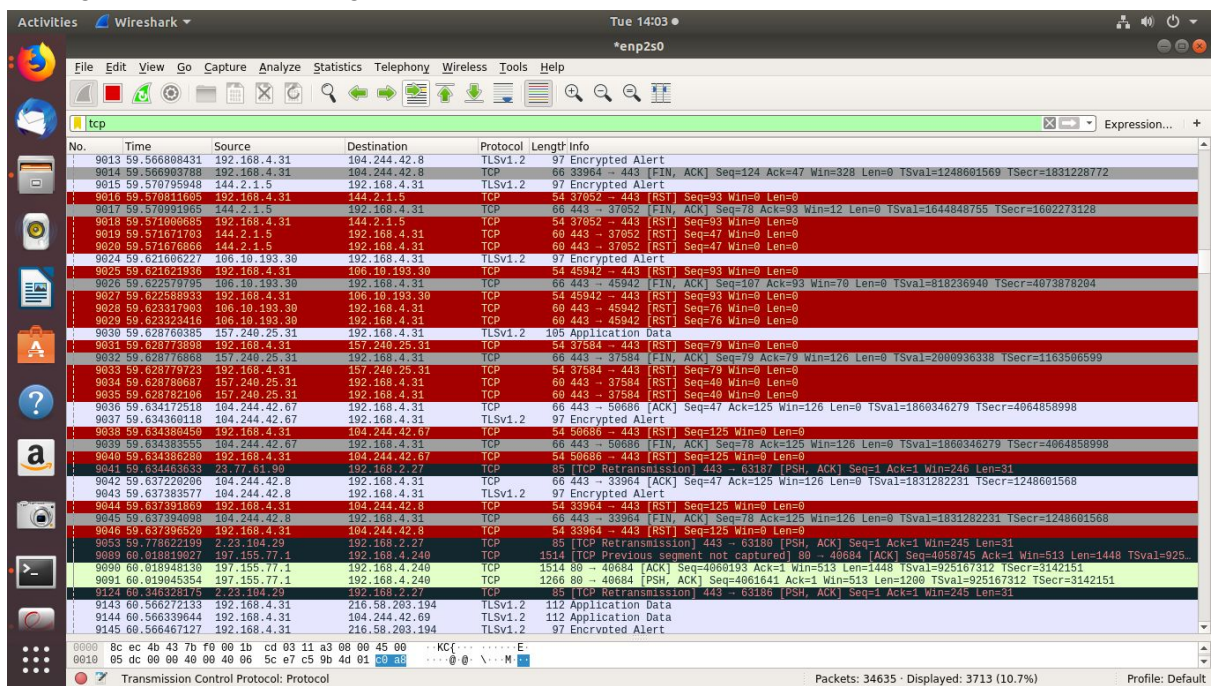
- a) Getting the ip address of current machine:

```
bhushan-borole:~/Desktop$ ifconfig | grep "inet 192"
    inet 192.168.2.102 netmask 255.255.248.0 broadcast 192.168.7.255
```

- b) Using the address of the local machine to perform dos attack using hping3 command with --flood option.

```
bhushan-borole:~/Desktop$ sudo hping3 192.168.4.31 --flood
HPING 192.168.4.31 (enp2s0 192.168.4.31): NO FLAGS are set, 40 headers + 0 data
bytes
hping in flood mode, no replies will be shown
```

- c) Seeing the dos attack using wireshark's interface.



3) Detecting the arp spoofing attack using arpwatrch and nmap.

a) Installing arpwatrch:

```
bhushan-borole:~/Desktop$ sudo apt install arpwatrch
Reading package lists... Done
Building dependency tree
Reading state information... Done
arpwatch is already the newest version (2.1a15-6).
0 upgraded, 0 newly installed, 0 to remove and 363 not upgraded.
```

b) Calling the arpwatrch command for a given eth port

```
bhushan-borole:~/Desktop$ arpwatrch -i eth8
bhushan-borole:~/Desktop$
```

c) Showing there is no messages present in the logs

```
bhushan-borole:~/Desktop$ cat /var/log/messages
cat: /var/log/messages: No such file or directory
```

d) Output in an alternate BSD style output format(with no fixed columns).

```
bhushan-borole:~/Desktop$ arp -a
_gateway (192.168.7.254) at 00:1b:cd:03:11:a3 [ether] on enp2s0
? (192.168.4.31) at d8:cb:8a:b8:83:c6 [ether] on enp2s0
? (192.168.2.32) at d8:cb:8a:b3:64:c3 [ether] on enp2s0
```