Secure Coding Cse 2010
Lab-9

K. Charan teja
18BCN7141

1. Find out the vulnerabilities of the code and fix it.

```
#include <stdio.h>
int main () {
   char user_name[10];
   int allow = 0;
   printf("Enter your username, please: ");
   gets(user_name);
   if (grantAccess(user_name)) {
      allow = 1;
   }
   if (allow != 0) {
               printf("%s",user);
     // privilegedAction();
   }
   return 0;
}

int grantAccess(char *name)
{
char name1[]="vitap";
if(!strcmp(name1,name))
  return 1;
else
   return 0;
 }
```

2. Find out the vulnerabilities of the code and fix it.

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

enum { BUFFER_SIZE = 10 };

int main() {
```

```
    char buffer[BUFFER_SIZE];
    int check = 0;

    sprintf(buffer, "%s", "This string is too long!");

    printf("check: %d", check); /* This will not print 0! */

    return 0;
}
```

3. Find out the vulnerabilities of the code and fix it.

```
#include <stdio.h>
int main () {
    sort int sub_marks[10]; //each subject marks is less than 10
          sort int sum=0;
    int i;
    printf("Enter the 10 subjects marks");
    for(i=0;i<10;i++)
        scanf("%d",&roll_numbers[i]);

    for(i=0;i<10;i++)
                    sum=sub_marks[i]+sum;

          printf("sum of all the subject marks %d",sum);

    return 0;
}
```

4. Find out the vulnerabilities of the code and fix it.

```
#include <stdio.h>
int main (int argc, char *argv[]) {
    char name1[10]; //each subject marks is less than 10
          char name2[10];

          strcpy(name1,argv[1]);
          strcypy(name2,argv[2]);

          puts(name1);
```

```
        printf(name2);

}
```