

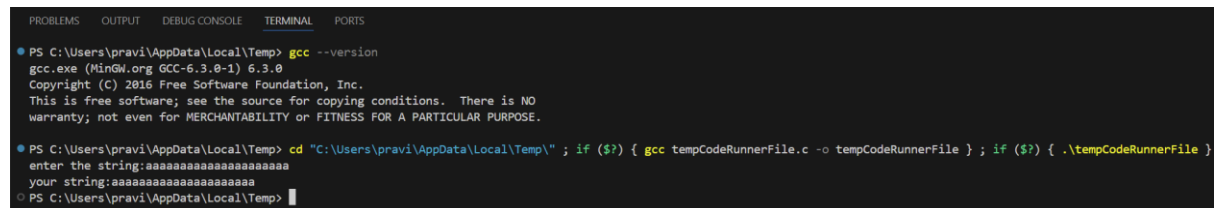
BUFFER OVERFLOW

Declaring a fixed-size character array (buffer) with a size of 10. Then prompt to enter a string and store it in the buffer. Printing the entered string. To demonstrate the buffer overflow vulnerability, include a string input that exceeds the buffer size.

PROGRAM:

```
#include<stdio.h>
int main()
{
    char str[10];
    printf("enter the string:");
    scanf("%s",str);
    printf("your string:%s\n",str);
}
```

OUTPUT:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\pravi\AppData\Local\Temp> gcc --version
gcc.exe (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

PS C:\Users\pravi\AppData\Local\Temp> cd "C:\Users\pravi\AppData\Local\Temp"; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile }; if ($?) { .\tempCodeRunnerFile }
enter the string:aaaaaaaaaaaaaaaaaaaaa
your string:aaaaaaaaaaaaaaaaaaaaa
PS C:\Users\pravi\AppData\Local\Temp>
```

I give the out of the program by printing 20 times of 'a'.then it print the string which the length is greater than size here it undergoes buffer overflow.

EXPLANATION OF BUFFER OVERFLOW

Make some change in program to understand the bufferoverflow clearly.

```
#include<stdio.h>
int main()
{
    char s[]="hello";
    char str[10];
    printf("enter the string:");
    scanf("%s",str);
    printf("your string:%s\n",str);
    for (int i=0;i<20;i++){
        printf("str[%d]=%c  %p\n",i,str[i],&str[i]);
    }
}
```

```

printf("address of str:%p\n",&str);
printf("address of s:%p\n",&s);
printf("s:%s\n",s);
printf("str:%s",str);
}

```

address of s and str(store the input):

address of str:0061FF0C

address of s:0061FF16

here difference is (F16-F0C)=10.

OUTPUT OF THIS PROGRAM:

```

PS C:\Users\pravi\AppData\Local\Temp> cd "C:\Users\pravi\AppData\Local\Temp\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
enter the string:aaaaaaaaaaaaaa
your string:aaaaaaaaaaaaaa
str[0]=a 0061FF0C
str[1]=a 0061FF0D
str[2]=a 0061FF0E
str[3]=a 0061FF0F
str[4]=a 0061FF10
str[5]=a 0061FF11
str[6]=a 0061FF12
str[7]=a 0061FF13
str[8]=a 0061FF14
str[9]=a 0061FF15
str[10]=a 0061FF16
str[11]=a 0061FF17
str[12]=a 0061FF18
str[13]=a 0061FF19
str[14]=a 0061FF1A
str[15]=a 0061FF1B
str[16]=a 0061FF1C
str[17]=a 0061FF1D
str[18]=a 0061FF1E
str[19]=a 0061FF1F
address of str:0061FF0C
address of s:0061FF16
s:aaaaa
str:aaaaaaaaaaaaaa

```

I give the output of aaaaaaaaaaaaaa(15),then I printed given string and also address of each character using for loop to understand.

Giving output more than the size array it also stores the character to next address in sequential order because sizeof char is 1.

Address of s =0062FF16

Address of str[10] == address of s.

In program s[]="hello"

After the execution it changes to s:aaaaa

STRING PRINTING

Printing of string in C is done %s-format specifier

%s-print all the next character until space or null character('\0')

FINAL EXPLANATION

Enter of string in program which is more the number of size of array.it also store the character to next address by sequential order then %s print all the character until it have null character.

ASSIGNMENT DONE BY

PRAVINKUMAR S