

## 什么是格式化字符串？

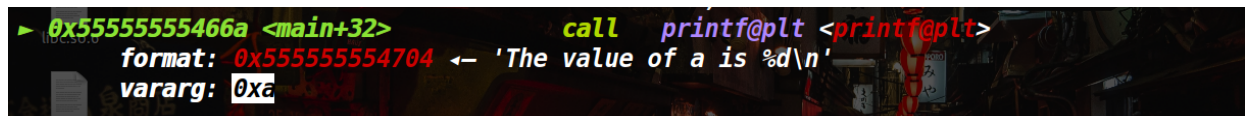
格式化字符串是将现有的字符串按照指定模板嵌入，再生成固定格式的新字符串。看看一个demo

```
int main()
{
    int a = 10;
    printf("The value of a is %d\n", a);
    return 0;
}
```

输出为

```
The value of a is 10
```

调试：



GDB debug output showing the call to printf@plt. The format string is 'The value of a is %d\n' and the vararg is 0xa.

```
> 0x55555555466a <main+32> call printf@plt <printf@plt>
format: 0x555555554704 ← 'The value of a is %d\n'
vararg: 0xa
```

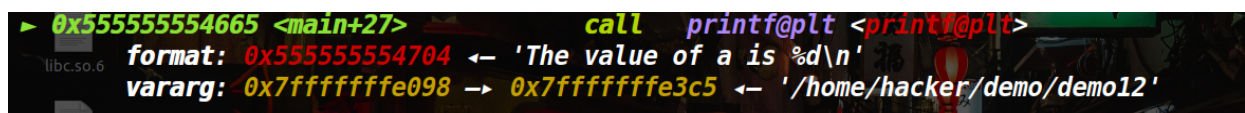
若将其变为

```
int main()
{
    int a = 10;
    printf("The value of a is %d\n");
    return 0;
}
```

输出为

```
The value of a is 1337076520
```

调试



GDB debug output showing the call to printf@plt. The format string is 'The value of a is %d\n' and the vararg is 0x7fffffff098. The output shows the address 0x7fffffff3c5, which is the address of the string '/home/hacker/demo/demo12'.

```
> 0x555555554665 <main+27> call printf@plt <printf@plt>
format: 0x555555554704 ← 'The value of a is %d\n'
vararg: 0x7fffffff098 → 0x7fffffff3c5 ← '/home/hacker/demo/demo12'
```

demo2的源码

### 观察调试：

在执行这个格式化字符串之后，这个地址内的内容会变成什么

即会把%n前面的字符长度输入第二个参数的地址的内容。

- ## 练习：非栈上格式化字符串

该题思路：

1. 绕伪随机数
2. 格式化字符串写入printf的返回地址的下一地址改为ret地址，再将下一地址改为gadgets地址