格式字符串漏洞

S6.1. 编写一个函数，接收数量可变的字符串参数，并输出这些字符串的总长度。

S6.2. 缓冲区溢出和格式字符串漏洞都可能改写保存的返回地址，但二者的写内存方式不同。请说明区别，并判断哪一种攻击受到的限制更少。

S6.3. 能否用 StackGuard 的思路防御格式字符串攻击？

S6.4. 执行 `printf(fmt)` 时，栈从低地址到高地址依次包含下列四字节值，其中第一个值是指向格式字符串的 `fmt`。若攻击者能控制格式字符串，最少使用多少个格式说明符即可确定使程序崩溃？

```
0xAABBCCDD, 0xAABBDDFF, 0x22334455, 0x00000000, 0x99663322
```

S6.5. 服务器程序接收远程用户输入，并把它保存在栈缓冲区中，即图 1 的区域 ❷。缓冲区地址随后赋给局部变量 `fmt`，并执行 `printf(fmt)`。执行时栈布局如图所示。请构造恶意输入，使服务器执行注入代码。无需给出代码的具体字节，用“恶意代码”标出即可，但必须放在正确位置并写出其余输入内容。

S6.6. 问题 S6.5. 的答案会让服务器输出超过十亿个字符，攻击执行很慢。请修改输入，使总输出字符数少于 `80,000`。

S6.7. 本题基于问题 S6.5.。若图 1 中区域 ❷ 与区域 ❸ 相距 26 字节，而不是 28 字节，应如何构造格式字符串？距离不能被 4 整除，但每个 `%x` 都会按四字节参数推进。

S6.8. ★

本题基于问题 S6.5.，但不知道保存的返回地址究竟位于何处，只知道它是以下地址之一：`0xAABBCDA0`、`0xAABBCDA4`、`0xAABBCDA8` 或 `0xAABBCDAC`。

1. 构造一个格式字符串，使真实地址无论是哪一个都能攻击成功，且总输出少于 80,000 个字符。
2. 在同一输出限制下，尽量缩短格式字符串。

S6.9. 编译器若发现 `printf()` 的参数数量与格式说明符数量不一致，可以发出警告。这种防御有什么局限？

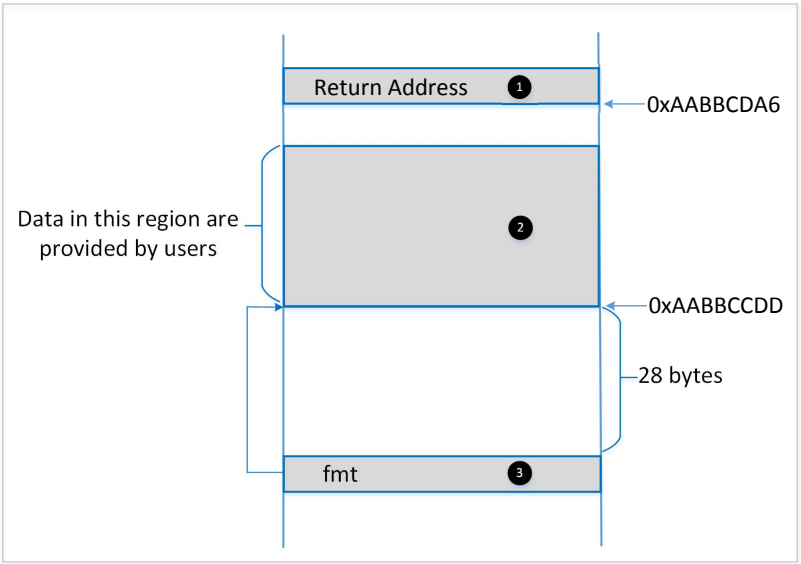

图 1: 问题的堆栈布局 S6.5.

S6.10. 由于 `printf()` 本身不需要特权，可以在执行这条语句前暂时降低程序权限。这样即使格式字符串漏洞被利用，攻击者似乎也无法获得太多权限。请评价这一想法。

S6.11. 若栈被设为不可执行，攻击者还能利用格式字符串漏洞让受害程序启动 shell 吗？

S6.12. 请使用 return-to-libc 技术利用格式字符串漏洞。结合图 1，详细说明需要修改栈上的哪些位置，以及如何用格式字符串完成这些写入。

S6.13. 若要利用格式字符串漏洞向目标地址写入 0，但为推进可变参数位置而使用的 `%x` 已使输出计数大于零，还能写入 0 吗？