D. 当 get_line 返回时，哪个(些)寄存器的值被破坏了？

E. 除了可能会缓冲区溢出以外，get_line 的代码还有哪两个错误？

```
/* This is very low-quality code.
   It is intended to illustrate bad programming practices.
   See Practice Problem 3.46. */
char *get_line()
{
    char buf[4];
    char *result;
    gets(buf);
    result = malloc(strlen(buf));
    strcpy(result, buf);
    return result;
}
```

a) C代码

```
    char *get_line()
1   0000000000400720 <get_line>:
2     400720:  53                    push   %rbx
3     400721:  48 83 ec 10           sub    $0x10,%rsp
    Diagram stack at this point
4     400725:  48 89 e7              mov    %rsp,%rdi
5     400728:  e8 73 ff ff ff        callq  4006a0 <gets>
    Modify diagram to show stack contents at this point
```

b) 对gets调用的反汇编

图 3-41　练习题 3.46 的 C 和反汇编代码

缓冲区溢出的一个更加致命的使用就是让程序执行它本来不愿意执行的函数。这是一种最常见的通过计算机网络攻击系统安全的方法。通常，输入给程序一个字符串，这个字符串包含一些可执行代码的字节编码，称为*攻击代码*(exploit code)，另外，还有一些字节会用一个指向攻击代码的指针覆盖返回地址。那么，执行 ret 指令的效果就是跳转到攻击代码。

在一种攻击形式中，攻击代码会使用系统调用启动一个 shell 程序，给攻击者提供一组操作系统函数。在另一种攻击形式中，攻击代码会执行一些未授权的任务，修复对栈的破坏，然后第二次执行 ret 指令，(表面上)正常返回到调用者。

让我们来看一个例子，在 1988 年 11 月，著名的 Internet 蠕虫病毒通过 Internet 以四种不同的方法获取对许多计算机的访问。一种是对 finger 守护进程 fingerd 的缓冲区溢出攻击，fingerd 服务 FINGER 命令请求。通过以一个适当的字符串调用 FINGER，蠕虫可以使远程的守护进程缓冲区溢出并执行一段代码，让蠕虫访问远程系统。一旦蠕虫获得了对系统的访问，它就能自我复制，几乎完全地消耗掉机器上所有的计算资源。结果，在安全专家制定出如何消除这种蠕虫的方法之前，成百上千的机器实际上都瘫痪了。这种蠕虫的始作俑者最后被抓住并被起诉。时至今日，人们还是不断地发现遭受缓冲区溢出攻击的系统安全漏洞，这更加突显了仔细编写程序的必要性。任何到外部环境的接口都应该是“防弹的”，这样，外部代理的行为才不会导致系统出现错误。