# 分析过程：

x86中参数都是保存在栈上,但在x64中前六个参数依次保存在RDI, RSI, RDX, RCX, R8和 R9寄存器里，如果还有更多的参数的话才会保存在栈上。

利用漏洞程序ggtest将system()在内存中的地址打印出来或者调试获得

（system()在内存中的地址不会被随机化），这样就不需要考虑ASLR的问题了。

利用攻击执行system(“/bin/sh”)，其中system()所需的参数”/bin/sh”保存在rdi寄存器中。

利用ROPGadget在/lib/x86\_64-linux-gnu/libc.so.6中搜素

gadget：pop rdi; ret

命令： ROPgadget --binary /lib/x86\_64-linux-gnu/libc.so.6 --only "pop|ret" | grep rdi

从而获得gadget的地址。

构造payload：首先找到漏洞程序中返回地址rip距离buf的偏移(40)，

然后确定gadget的位置以及参数”/bin/sh”的位置，system()在内存中的地址由漏洞程序发送过来。

payload = "a"\*136 + p64(pop\_ret\_addr) + p64(binsh\_addr) + p64(system\_addr)