

# HCTF writeup week1

## CRYPTO:

### RSA 类:

#### ID18:

$n=p*q$  然后利用解密公式  $m = c^d \bmod n$

不过一开始题目  $d$  和  $e$  搞反了，算出一段 CA 开头的明文，怎么也找不出 flag，浪费了很多时间

#### ID33:

这题有点瞎蒙，利用  $c = m^e \bmod n$

由于  $e=1$ ， $n>m$  时  $m=c$

其次观察  $c$ ，开头为 6867，即 hg 所对应的 ascii 码

便能轻松猜出  $m=c$

### 凯撒密码:

#### ID19:

查看凯撒密码的百科后知道偏移量相同

密文为 flag 格式，很容易猜出 mlfrij 对应 hgame

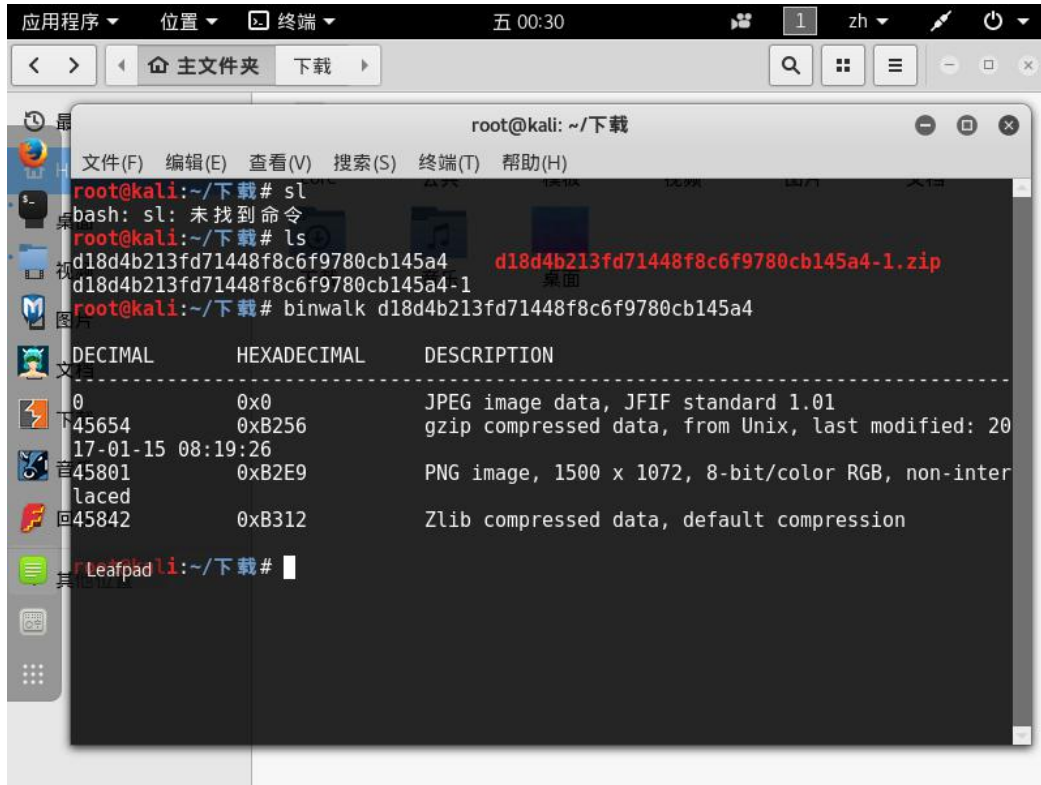
## MISC:

### ID32:

下载文件后改后缀名为.txt 打开，得到 flag

ID31 30:

在 linux 下用 binwalk 打开文件，结果如下图

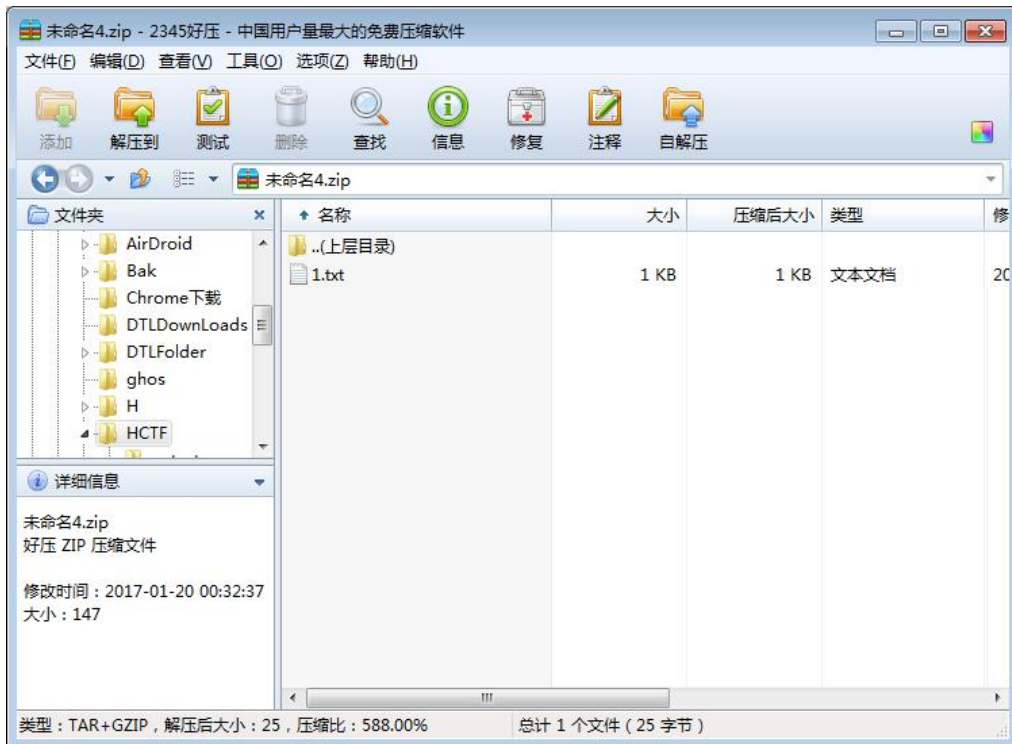


The screenshot shows a terminal window on a Kali Linux desktop. The user is in the root directory of the /Downloads folder. They run the command `binwalk d18d4b213fd71448f8c6f9780cb145a4`. The output shows a table of offsets, hexadecimal values, and descriptions of the file's contents.

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
45654	0xB256	gzip compressed data, from Unix, last modified: 2017-01-15 08:19:26
45801	0xB2E9	PNG image, 1500 x 1072, 8-bit/color RGB, non-interlaced
45842	0xB312	Zlib compressed data, default compression

得知包含 JPEG 图片 PNG 图片 ZIP 压缩文档还有 Zlib 文件

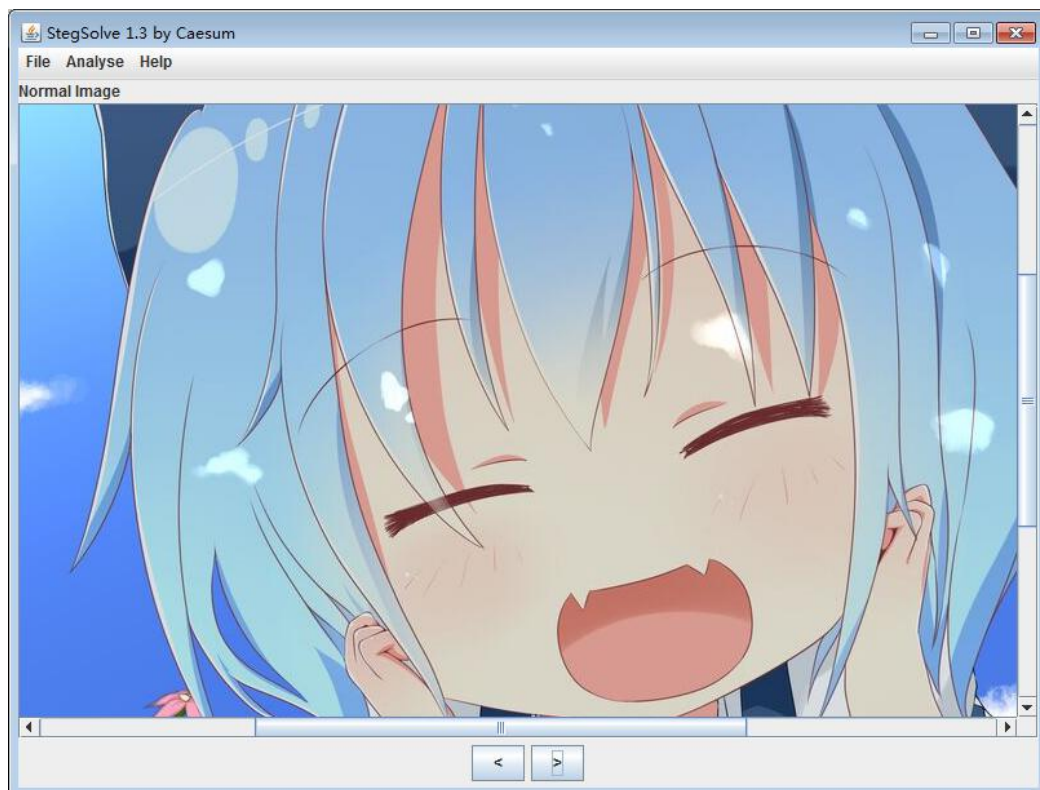
先分离出 ZIP 压缩文档，解压后得到包含 flag 的 txt 文档



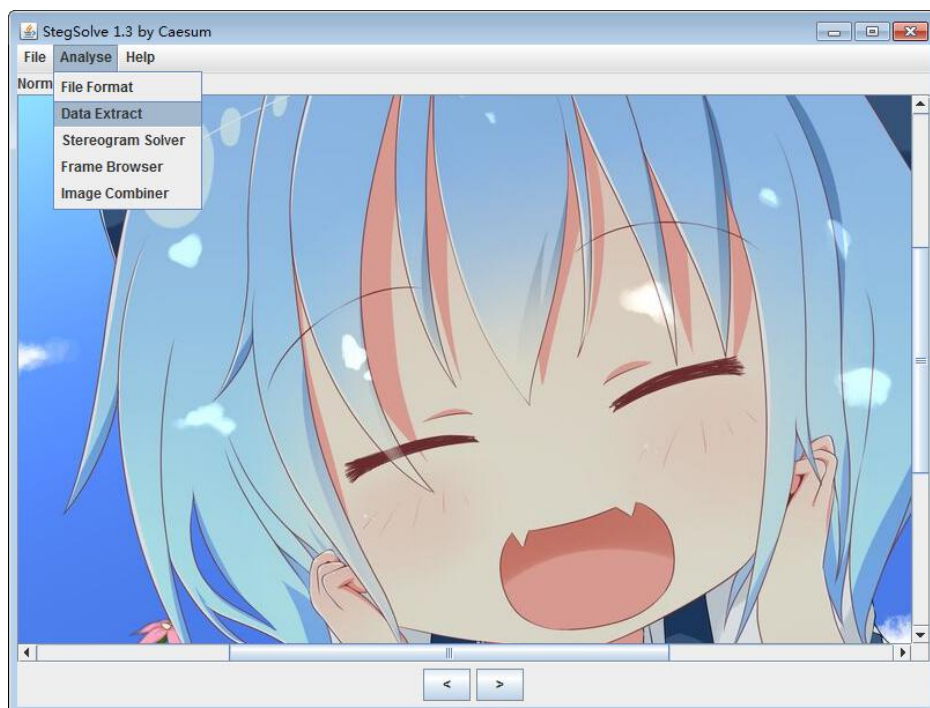
得到第一个 flag

然后分离出剩下所有的文件发现是一张完整的 PNG 图片，

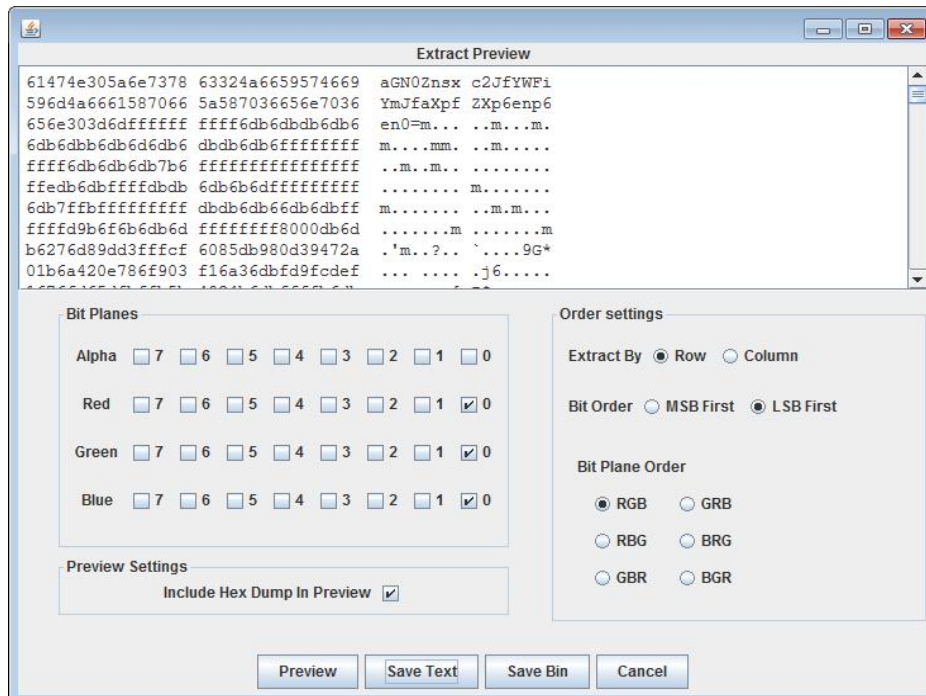
百度得知利用 stegsolve 软件有助于找出 flag



使用 Analyse->Data Extract 功能



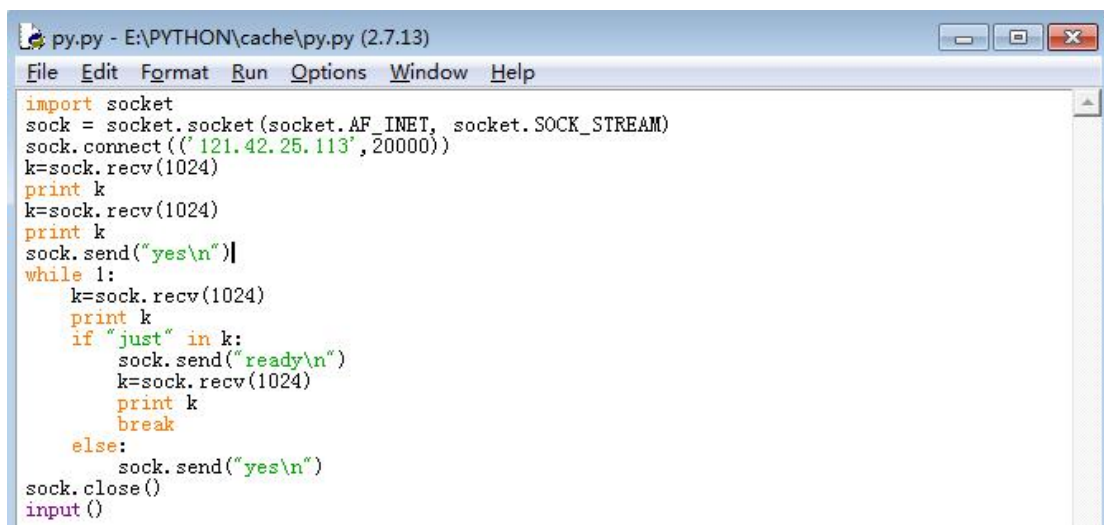
结果如下图



再将开头代码用 base64 解密得到 flag

ID38:

由于有原代码，使用 python 编写 socket 后运行，得到 flag



```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('121.42.25.113', 20000))
k=sock.recv(1024)
print k
k=sock.recv(1024)
print k
sock.send("yes\n")
while 1:
    k=sock.recv(1024)
    print k
    if "just" in k:
        sock.send("ready\n")
        k=sock.recv(1024)
        print k
        break
    else:
        sock.send("yes\n")
sock.close()
input()
```

PENTEST:

ID14:

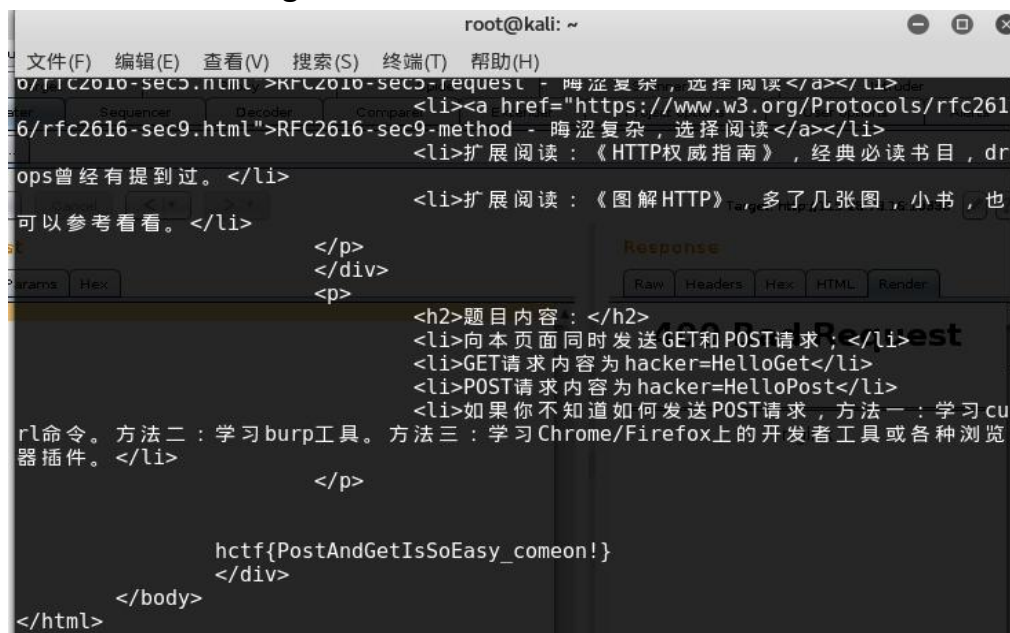
先学习了 curl 命令，无果，后来又尝试使用 burpsuite，发现

很难学会，转回 curl。经高人点拨发现可用

curl -d "hacker=HelloPost" ..../?hacker=HelloGet

同时 Post 与 Get

成功获得 flag



```
root@kali: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
0/rfc2016-sec5.html > RFC2016-sec5-request - 晦涩复杂，选择阅读 </a></li>
6/rfc2616-sec9.html > RFC2616-sec9-method - 晦涩复杂，选择阅读 </a></li>
ops曾经有提到过。 </li>
可以看看。 </li>
</p>
</div>
<p>
</p>
<h2>题目内容：</h2>
<li>向本页面同时发送GET和POST请求：</li>
<li>GET请求内容为hacker=HelloGet</li>
<li>POST请求内容为hacker=HelloPost</li>
<li>如果你不知道如何发送POST请求，方法一：学习cu
rl命令。方法二：学习burp工具。方法三：学习Chrome/Firefox上的开发者工具或各种浏览
器插件。</li>
</p>
hctf{PostAndGetIsSoEasy_comeon!}
</div>
</body>
</html>
```



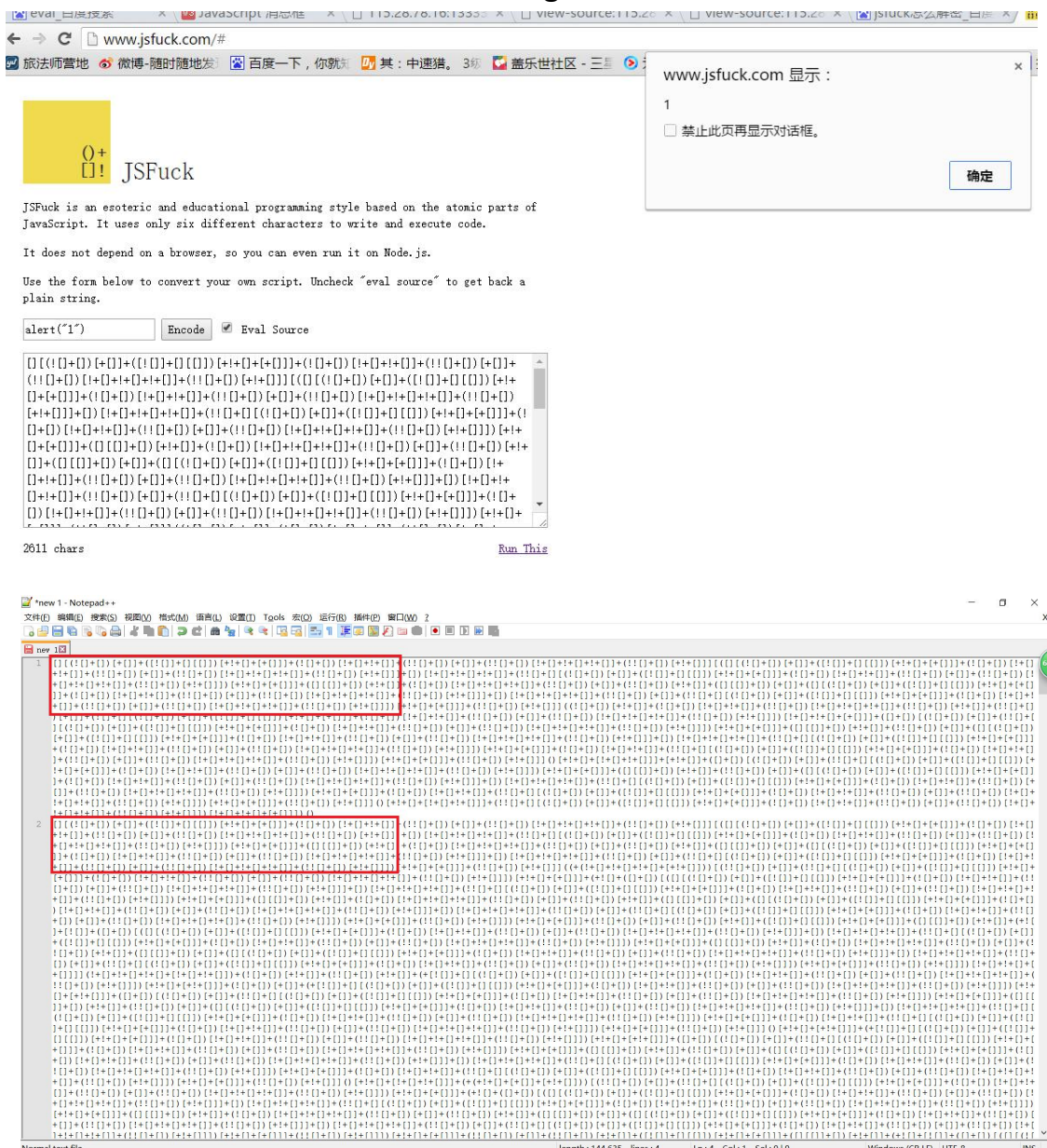
WEB:

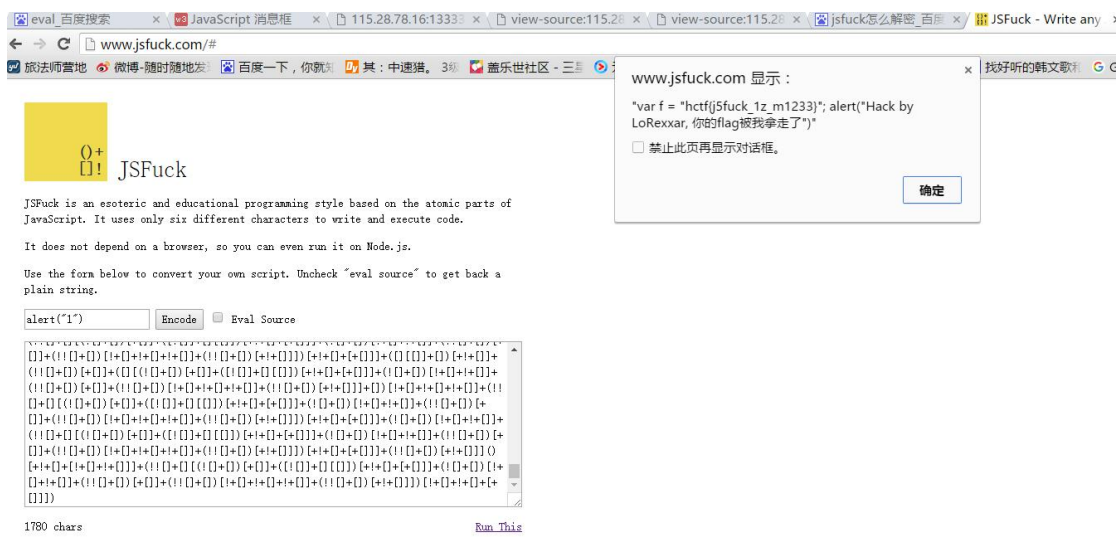
ID21:

标签<script>确定是 js 代码，上网查了之后得知是 jsfuck 加密，进入 jsfuck 官网 [www.jsfuck.com](http://www.jsfuck.com)

一开始有弹窗说明有 alert 函数

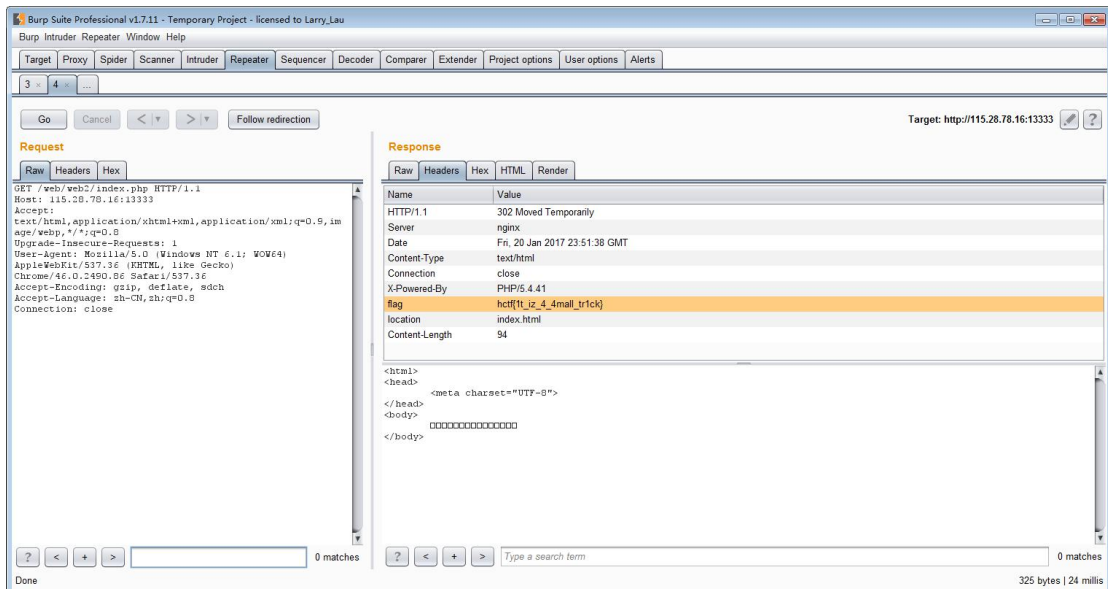
随意尝试一句 alert 函数加密后复制到 notepad 中对比发现前面有一段相同，随后不勾选 eval source 当成字符串跑了一下剩余部分发现后面即原码当成字符串加密的结果，提取题目源码后半段当成字符串跑得到 flag





## ID22:

点击网页后发现自动跳转，使用 burpsuite 打开跳转前网页获得 flag



## ID25:

由原代码可知想拿到 flag 必须满足

```
$req["number"] = intval($req["number"])
```

```
!($req["number"][0] == "+" || $req["number"][0] == "-")
```

```
$n1 = $n2
```

```
!(is_palindrome_number($req["number"]))
```

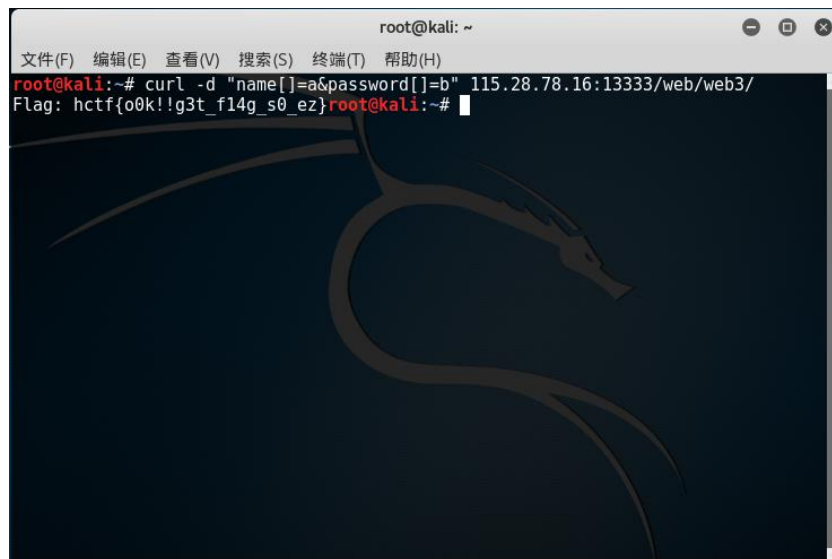
语言中高精度浮点数比较往往会出错

如 C 语言中判断 `1.000000001==1` 为 true

同理尝试 `10000000000.000000000010` 拿到 flag

ID26:

利用 php 中 `sha1()` 函数的漏洞，post “`name[]=a&password[]=b`”  
得到 flag



PENTEST:

ID15:

利用 burpsuite 改变 User-Agent X-Forward-For Referer 内容

```
POST /pentest/02/ HTTP/1.1
Host: 115.28.78.16:13333
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 99_0 like Mac OS X) AppleWebKit/536.26
(KHTML, like Gecko) Version/6.0 Mobile/10A403 Safari/8536.25
Referer: http://google.com
Content-Length: 0
Content-Type: application/x-www-form-urlencoded
Proxy-Connection: Keep-Alive
X-Forwarded-For: 127.0.0.1
```



ID16:

利用 burpsuite 在请求头加上 cookie: admin=1;isLogin=true

得到 flag

```
GET /pentest/03/ HTTP/1.1
Host: 115.28.78.16:13333
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/46.0.2490.86 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Connection: close
Cookie: admin=1;isLogin=true
```

ID37:

根据 hint 字母数字字符处理不同, 先转换字母, 后暴力破解数

字, 字符不变, 得到 flag