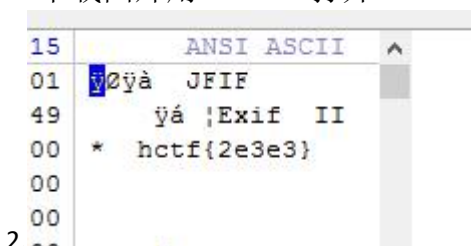


一. Explorer 的图库之一

1. 下载图片用 winhex 打开



拿到 flag

二. 密码学教室入门（二）

发现“{”前有四个字符想到 hctf，推断出偏移量为-5，数字不知道如何处理，但发现 is just for fun 很符合语法规则，试了一下错误，联想 flag 的正确姿势，i 换成 l，回答正确。

三. lightless 的渗透教室入门篇（一）

1. 在 url 后直接加上? hacker=HelloGet 发送 GET 请求

2. 发送 POST 请求

```
<form method="POST">
```

```
  <input type="text" name="hacker">
```

```
  <input type="submit">
```

输入 HelloPost 提交

题目内容：

- 向本页面同时发送GET和POST请求；
 - GET请求内容为hacker=HelloGet
 - POST请求内容为hacker=HelloPost
- url命令。方法二：学习burp工具。方法三：学习C

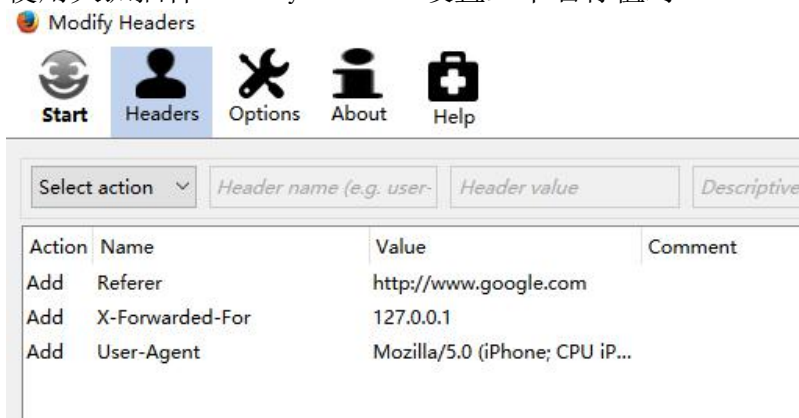
hctf{PostAndGetIsSoEasy_comeon!}



拿到 flag:

四. lightless 的渗透教室入门篇（二）

使用火狐插件 ModifyHeaders 设置三个名称值对



发送后在响应头中找到 flag

▼ 响应头 (0.265 KB)
Connection: "keep-alive"
Content-Encoding: "gzip"
Content-Type: "text/html"
Date: "Thu, 19 Jan 2017 16:49:04 GMT"
Server: "nginx"
Transfer-Encoding: "chunked"
Vary: "Accept-Encoding"
X-Powered-By: "PHP/5.4.41"
flag: "hctf{h77p_He4dEr_50_E4sy_AnD_fUn_ohhouhou}"

五. lightless 的渗透教室入门篇（三）

POST 一个 hint 之后在响应中找到 cookiecontent

▼ 响应头 (0.256 KB)
Connection: "keep-alive"
Content-Encoding: "gzip"
Content-Type: "text/html"
Date: "Thu, 19 Jan 2017 16:51:06 GMT"
Server: "nginx"
Transfer-Encoding: "chunked"
Vary: "Accept-Encoding"
X-Powered-By: "PHP/5.4.41"
cookiecontent: "admin=1 and isLogin=true"

通过 web developer 插件设置两个 cookie
拿到 flag

题目内容：

- 设置cookie，内容自行寻找，都是以前学过的内容。

造cookie，方法一：学习curl命令。方法二：学习burp工具。方法三：学习Chrome/Firefox上的开发者

flag: hctf{hao_hao_kan_zi_liao!!!}, 通过伪造cookie，你可以绕过一些限制，或是伪造身份。

谨记涛涛 dalao 教诲会好好看资料的 TAT

六. 我是谁我在哪？？？

原来的 php 变成了 html

好奇改了一下发现多了一个 get

再去响应里找就捉到了一个野生的 flag

七. 密码学教室番外篇

讲道理各种地方都没有搜到凯撒密码对数字的处理方法，谜。

于是在字母和符号都确定的情况下，数字的各种偏移量都试了一遍。

毕竟怎么说也小于 30？【雾】

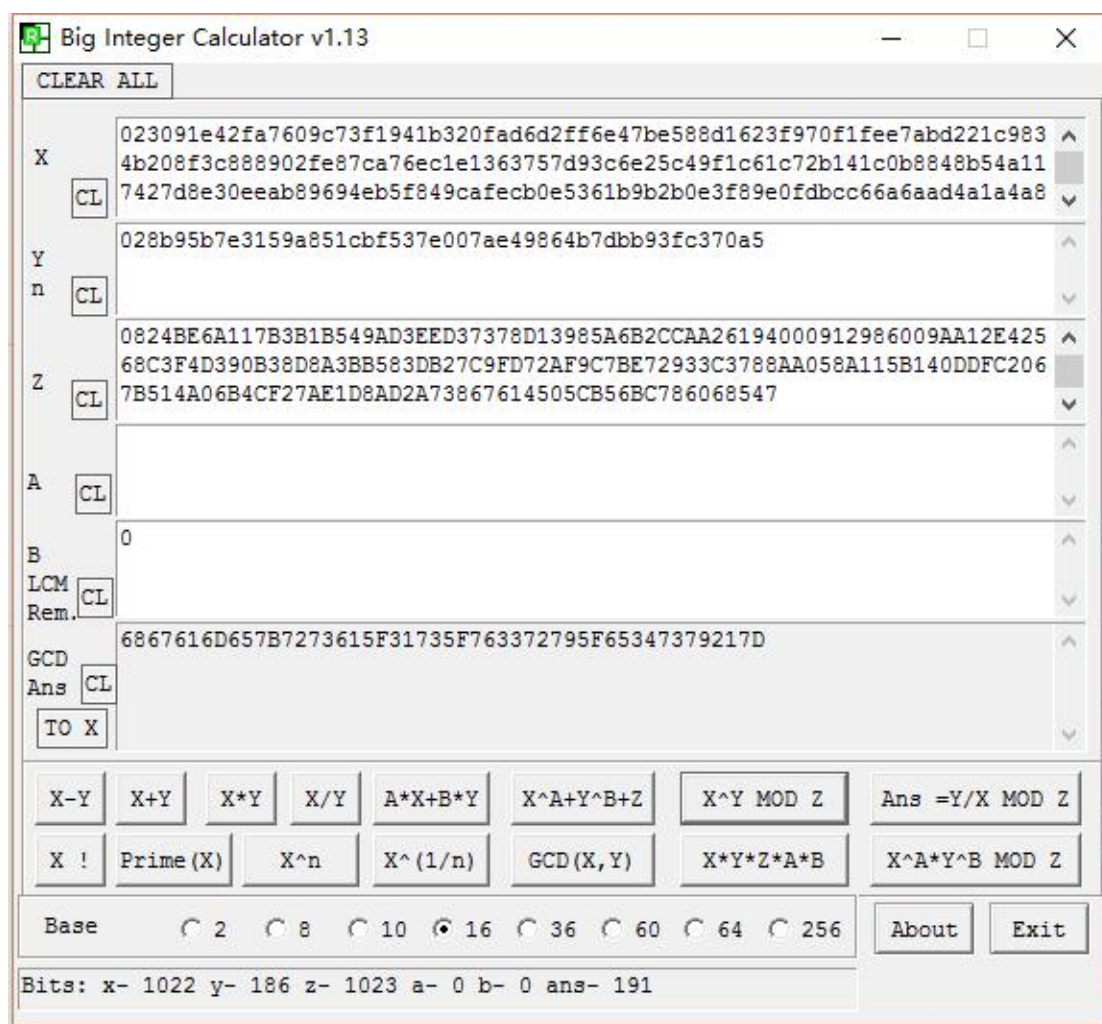
八. 这 TM 是啥

在 Jsfuck 的网站上，跑出了 hctf{ 和 } 的 jsfuck 代码，在源码中查找，这两个中间的代码对应的原文即为 flag 的内容。

九. 密码学教室入门（一）

由 $m = c^d \bmod n$ 以及 $n = p * q$

通过该工具



计算得出 16 进制的结果 转为字符串即为 flag

十. 密码学教室入门（四）

由公式 $m^e = c \bmod n$

以及上题工具解出 16 进制再转为字符串得到 flag

十一. 神奇的数字

`number = intval(number)`

`intval(number) = intval(strrev(number))`

not a palindorme number

以上为 number 需要满足的条件

intval 函数返回值的最大的值取决于操作系统。64 位系统上，最大带符号的 integer 值是 9223372036854775807。然而回文数明显小于 64 位的限制，于是我们在后面加一个 0 然后 POST 就攻略了 orz

十二. 不可能拿到的 flag

`isset($_POST['name']) and isset($_POST['password'])`

`$_POST['name'] != $_POST['password']`

`sha1($_POST['name']) === sha1($_POST['password'])`

Name 和 password 需满足以上条件，由哈希算法无法针对数组使用可发送

POST: name[]=12&password[]=65

即得 flag 一只。