

@Cyril

1-----题目 ID: 21 这 TM 是啥?

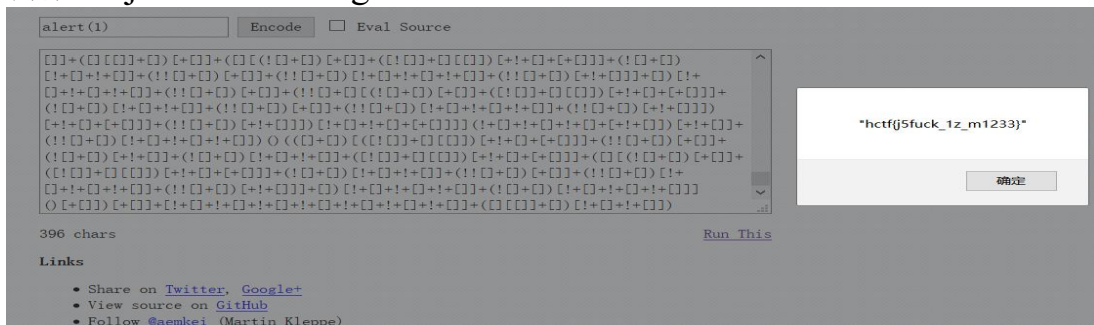
<http://115.28.78.16:13333/web/web1/>进入页面后弹出土土的对话框,F12 发现这段 JS 的代码。



进入 <http://www.jsfuck.com/>，由于 hint 里写到“这段代码是什么”，猜测里面藏了 flag，将这段代码贴到 word，并在 jsfuck 网站中找出“hctf”，“{”，“}”所对应的 js 代码，利用 word 搜索工具找出匹配段落

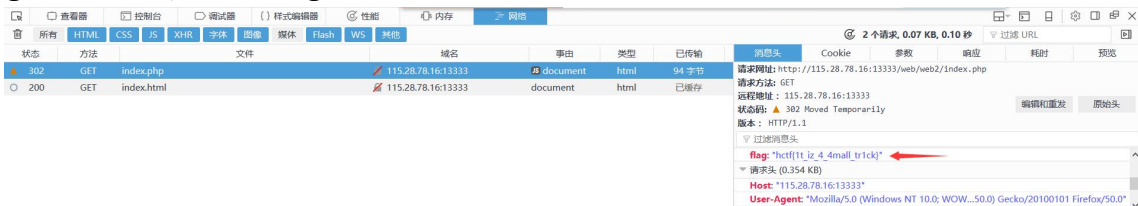
[illegible]

再放入 jsfuck 跑出 flag。

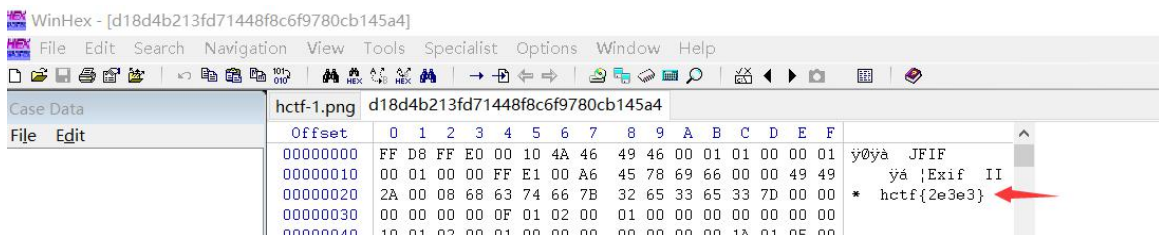


2-----题目 ID: 22 我是谁我在哪???

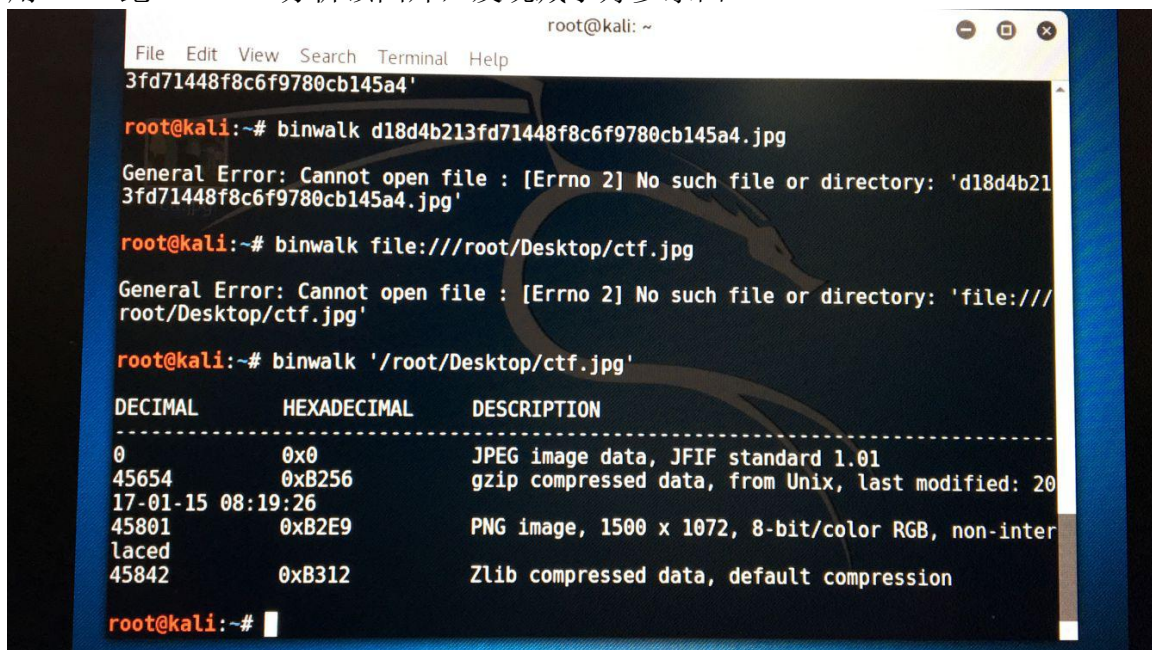
页面进去以后会跳一下后缀，把 html 改回 php，直接开 F12 在可以 get 包里拿到 flag。








3-----题目 ID: 32 Explorer 的图库之一
下载图片，用 winhex 打开直接得到 flag。



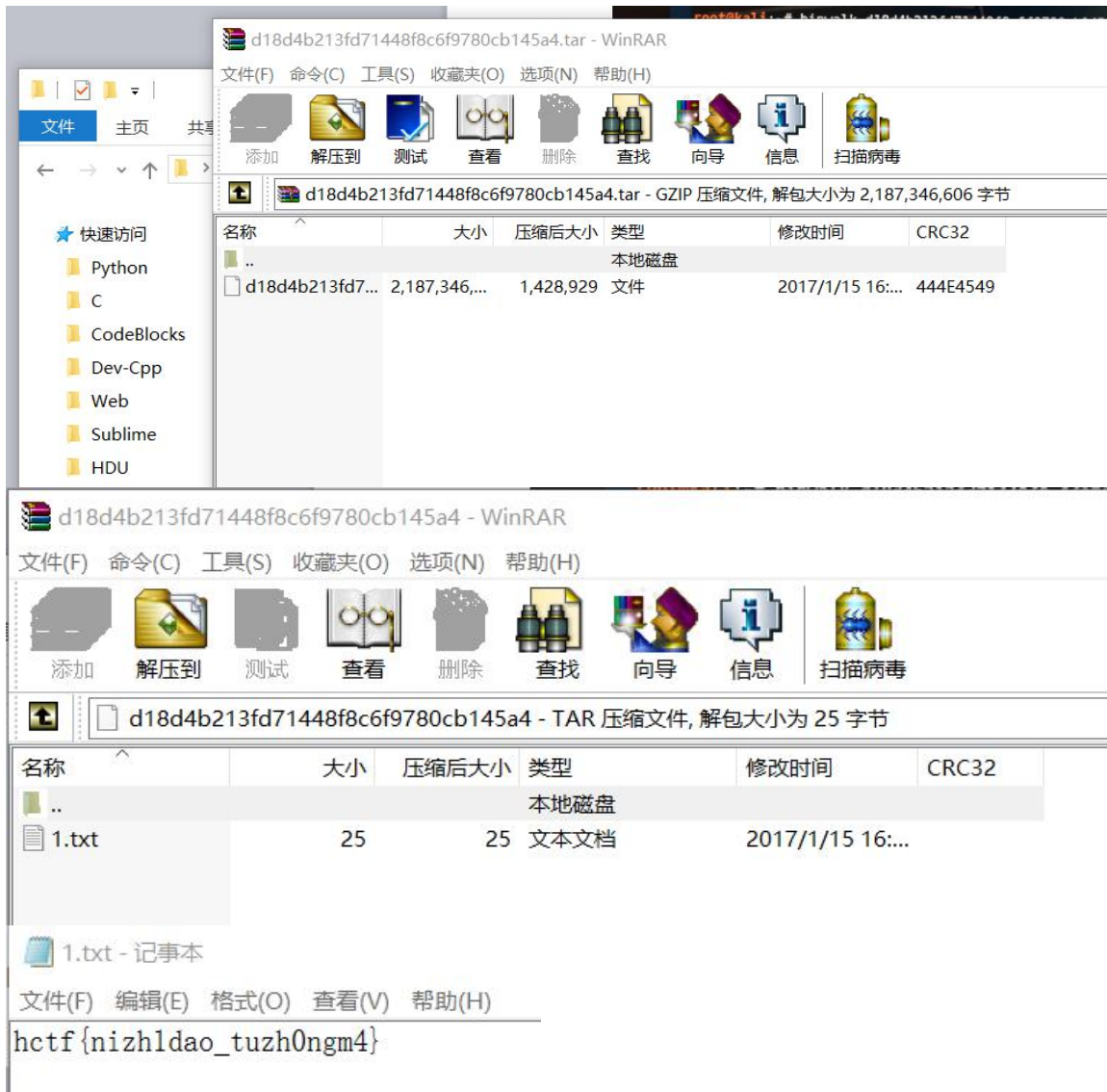
4-----题目 ID: 31 Explorer 的图库之二
用 kali 跑 binwalk 分析该图片，发现藏了好多东西 awww。



用 dd 命令分解这些东西最后得到：

	d18d4b213fd71448f8c6f9780cb145a4	2017/1/18 1:31	文件
	d18d4b213fd71448f8c6f9780cb145a...	2017/1/17 20:13	JPG 文件
	d18d4b213fd71448f8c6f9780cb145a...	2017/1/18 19:20	WinRAR 压缩文件
	hctf-1.png	2017/1/18 19:12	PNG 文件
	hctf-3.Zlib	2017/1/18 19:24	ZLIB 文件

打开 gzip 文件，查看里面藏的 txt 得到 flag。



The screenshot shows a Windows desktop with two windows open. The top window is WinRAR, displaying a tar archive named 'd18d4b213fd71448f8c6f9780cb145a4.tar'. The archive contains a single file named 'd18d4b213fd7...' with a size of 2,187,346 bytes. The bottom window is Notepad, showing the contents of the extracted file '1.txt'. The text in the Notepad window is 'hctf{nizhldao_tuzh0ngm4}'.

WinRAR Window: d18d4b213fd71448f8c6f9780cb145a4 - WinRAR

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加 解压到 测试 查看 删除 查找 向导 信息 扫描病毒

d18d4b213fd71448f8c6f9780cb145a4.tar - GZIP 压缩文件, 解包大小为 2,187,346,606 字节

名称	大小	压缩后大小	类型	修改时间	CRC32
..			本地磁盘		
d18d4b213fd7...	2,187,346,...	1,428,929	文件	2017/1/15 16:...	444E4549

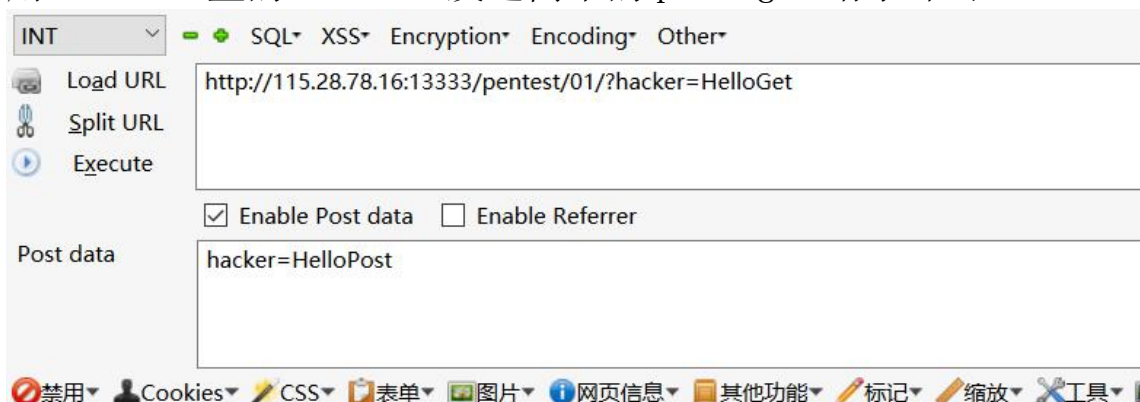
Notepad Window: 1.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

hctf{nizhldao_tuzh0ngm4}

5: 题目 ID: 14 lightless 的渗透教室入门篇 (一)

用 FireFox 里的 hackbar 发送简单的 post get 请求即可。

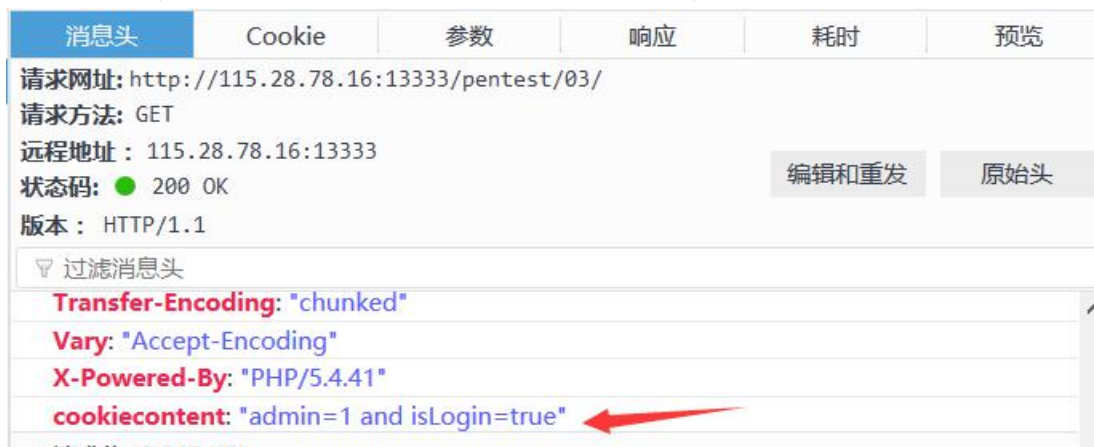


题目内容：

- 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
- 如果你不知道如何发送POST请求，方法一：学习curl命令。方法二：学习burp工具。方法三：学习Chrome/Firefox上的开发者工具或各种浏览器插件。
hctf{PostAndGetIsSoEasy_comeon!}

6-----题目 ID: 16 lightless 的渗透教室入门篇 (三)

F12 查看源码，找出 cookie 所需设置内容



用 Firefox 的 Web Developer 插入两个 cookie。

添加 Cookie

名称(必需): isLogin

值: true

域名/主机: 115.28.78.16

路径: /pentest/03/

有效期: Fri, 20 Jan 2017 05:32:37 GMT

☐ 会话期 Cookie (会话结束立即过期)

☐ 安全的 Cookie (对 Cookie 加密)

确定 取消

添加 Cookie

名称(必需):

admin

值:

1

域名/主机:

115.28.78.16

路径:

/pentest/03/

有效期:

Fri, 20 Jan 2017 05:31:48 GMT

☐ 会话期 Cookie (会话结束立即过期)

☐ 安全的 Cookie (对 Cookie 加密)

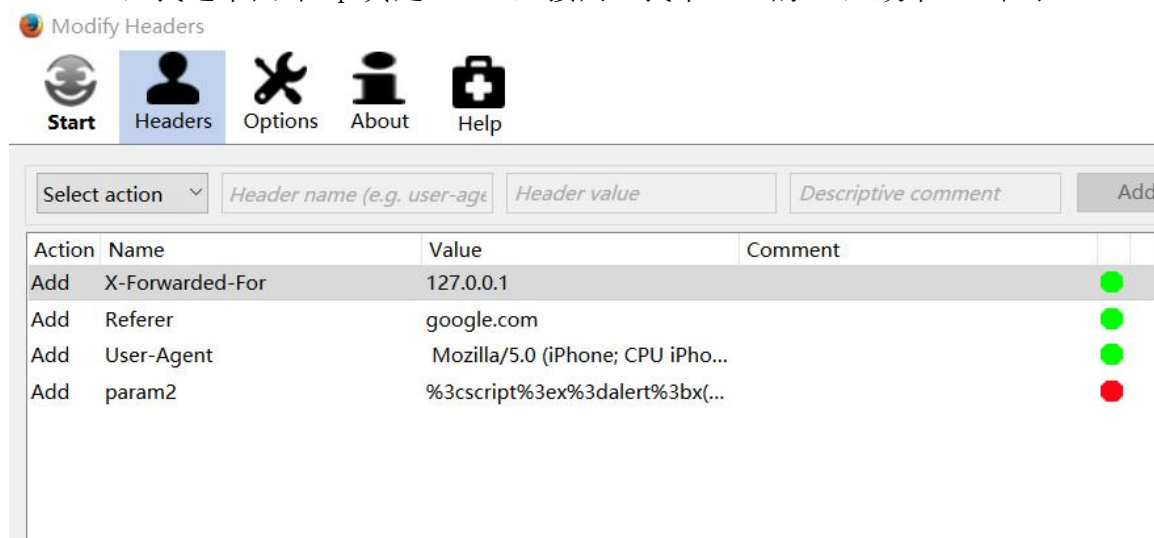
确定

取消

得到 flag。

7-----题目 ID: 15 lightless 的渗透教室
入门篇（二）

在 Modify Headers 中插入 xff (此处的本地 ip 为广义上的本地回环 ip), UA, referer, 找这个回环 ip 真是... UA 直接网上找个 iOS 的 UA, 改个 99 即可。



然后在 html 里发现 hint
跑去 get 里看一眼得到 flag。



8-----题目 ID: 19 密码学教室入门（二）

题目给出 mlfrj{Hfjxfw_hnumjw_8x_ozxy_ktw_kzs}, 按密码这章的 flag 格式解出 hgame{Caesar_cipher_3s_just_for_fun}, 但并不是正确的 flag, 由于数字与字母的处理方式不同, 于是再进行整理得到 hgame{Caesar_cipher_1s_just_for_fun}

9-----题目 ID: 37 密码学教室番外篇

原理同上, 字符不变, 手动解出 hgame{dgfdyhcry42287235413//+/%}。

10-----题目 ID: 25 神奇的数字

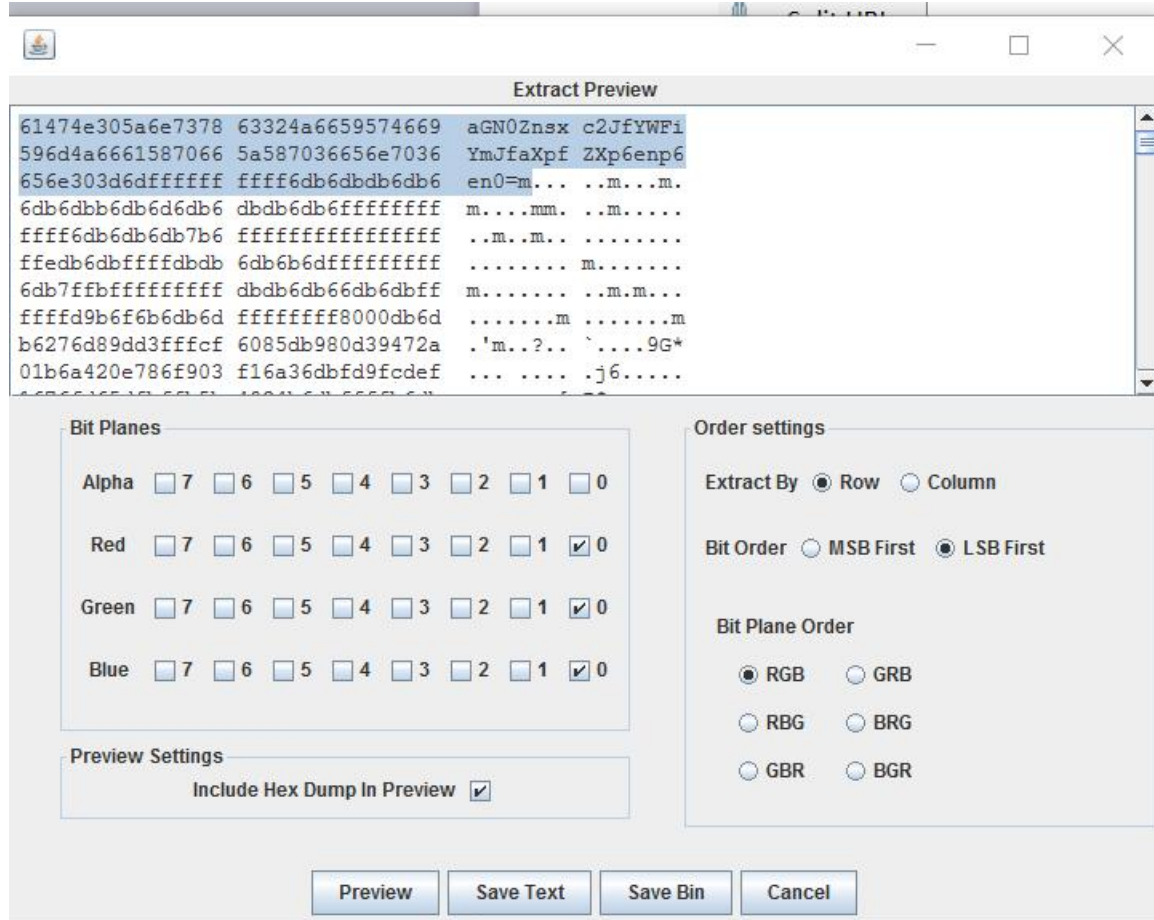
该题借鉴了以前的一些 ctf 题的做法，用了整数溢出的方式绕过。首先查看 php 代码中的所需 post 的 number 所具备的条件：`number = intval(number)`, `intval(number) = intval(strrev(number))`, not a palindorme number，显然这些条件中存在矛盾点，想到函数范围外绕过的方法。在 <http://php.net/manual/zh/function.intval.php> 中查看 intval 函数的限制，发现在 64 位操作系统中最大的 integer 值是 9223372036854775807，但是 post 这个数字发现还是没能拿到 flag，提示“no, this is not a palindrome number!”，因为它的回文数小于 64 位系统的限制，所以我们在这一串数字前补个 0，成功绕过并拿到 flag。



find a strange dongxi: hctf{go0d_job_intv4l_iz_g00d}

11-----题目 ID: 30 Explorer 的图库之三

Binwalk 提取出的文件里还有张 png 没用到，查找基本的一些隐写形式，用 StegSolve 打开，Bit Planes 调至最低位，检测 1sb。



发现 base64 加密语句头，解析一下得到 flag。

请输入要进行编码或解码的字符：

aGN0Znsxc2JfYWFiYmJfaXpfZXp6enp6en0=m

编码

解码

☐ 解码结果以16进制显示

Base64编码或解码结果：

hctf{lsb_aabbb_iz_ezzzzzz}

12-----题目 ID: 18 密码学教室入门(一)

直接根据公式 $m=c^d \bmod n$; $n=p*q$ 解出，再将 16 进制 m 转换成字符串即可。

Big Integer Calculator v1.13

CLEAR ALL

X

023091e42fa7609c73f1941b320fad6d2ff6e47be588d1623f970f1fee7abd221c9834b208f3c888902fe87ca76ec1e1363757d93c6e25c49f1c61c72b141c0b8848b54a117427d8e30eeab89694eb5f849cafeeb0e5361b9b2b0e3f89e0fdbcc66a6aad4a1a4a8

CL

Y

028b95b7e3159a851cbf537e007ae49864b7dbb93fc370a5

CL

Z

0824BE6A117B3B1B549AD3EED37378D13985A6B2CCAA26194000912986009AA12E42568C3F4D390B38D8A3BB583DB27C9FD72AF9C7BE72933C3788AA058A115B140DDFC2067B514A06B4CF27AE1D8AD2A73867614505CB56BC786068547

CL

A

CL

B

0

CL

LCM

CL

Rem.

CL

GCD

6867616D657B7273615F31735F763372795F65347379217D

CL

Ans

CL

TO X

X-Y

X+Y

X*Y

X/Y

A*X+B*Y

X^A+Y^B+Z

X^Y MOD Z

Ans =Y/X MOD Z

X !

Prime (X)

X^n

X^(1/n)

GCD(X,Y)

X*Y*Z*A*B

X^A*Y^B MOD Z

Base

☐ 2
☐ 8
☐ 10
☒ 16
☐ 36
☐ 60
☐ 64
☐ 256

About

Exit

Bits: x- 1022 y- 186 z- 1023 a- 0 b- 0 ans- 191

6867616D657B7273615F31735F763372795F65347379217D

16进制转字符

字符转16进制

清空结果

hgame{rsa_1s_v3ry_e4sy!}

13-----题目 ID: 33 密码学教室入门(四)

根据导出公式 $m^e \equiv c \pmod n$ 计算得到 flag。套路同上题？

Big Integer Calculator v1.13

CLEAR ALL

X	06867616d657b7273615f31735f737469316c5f653473795f6e6f77217d
CL	
Y	001
n	
CL	
Z	081cfc71c44c83faf3c5242fa81ae2e533fc945f3bef30bc13323ea4a55b3debc11301c6a9ecb8f7ef92fa169b157435af728a145497f2cdf75b3007b9732da4c47d67683f09ae1edc8f698f5ec7549593d9f1d06adafae4ad09514928bf0367a2719f7c1715803
CL	
A	
CL	
B	0
LCM	
Rem.	
GCD	6867616D657B7273615F31735F737469316C5F653473795F6E6F77217D
Ans	
CL	
TO X	

X-Y	X+Y	X*Y	X/Y	A*X+B*Y	X^A+Y^B+Z	X^Y MOD Z	Ans =Y/X MOD Z
X !	Prime(X)	X^n	X^(1/n)	GCD(X,Y)	X*Y*Z+A*B	X^A*Y^B MOD Z	

Base ☐ 2 ☐ 8 ☐ 10 ☒ 16 ☐ 36 ☐ 60 ☐ 64 ☐ 256

Bits: x- 231 y- 1 z- 1024 a- 0 b- 0 ans- 231

6867616D657B7273615F31735F737469316C5F653473795F6E6F77217D

16进制转字符

字符转16进制

清空结果

hgame{rsa_1s_still_e4sy_now!}

14-----题目 ID: 26 不可能拿到的 flag

观察 php 代码发现要拿到这个 flag 必须满足 name 和 password 不相等，但是又要保证 sha1 () 相等，想到利用 sha1 函数存在的漏洞绕过。改 name 为 name[]=a，password 为 password[]=b，第一个处理条件时，两个数组并不相同，但 sha1 函数并不能处理数组，

都返回 false，此时两个条件同时满足，拿到 flag。

Load URL	http://115.28.78.16:13333/web/web3/
Split URL	
Execute	
<input checked="" type="checkbox"/> Enable Post data <input type="checkbox"/> Enable Referrer	
Post data	name[]=a&password[]=b

禁用 Cookies CSS 表单 图片 网页信息

Flag: hctf{o0k!!g3t_f14g_s0_ez}