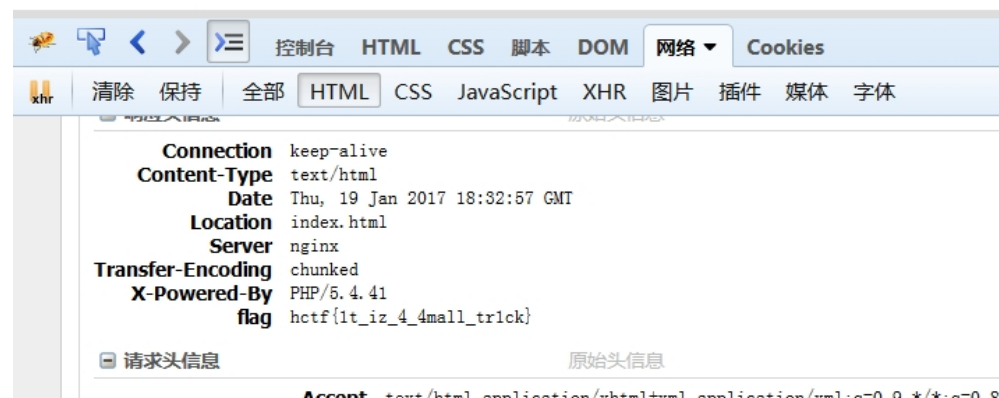


Hgame 第一周 write up

21 这 TM 是啥

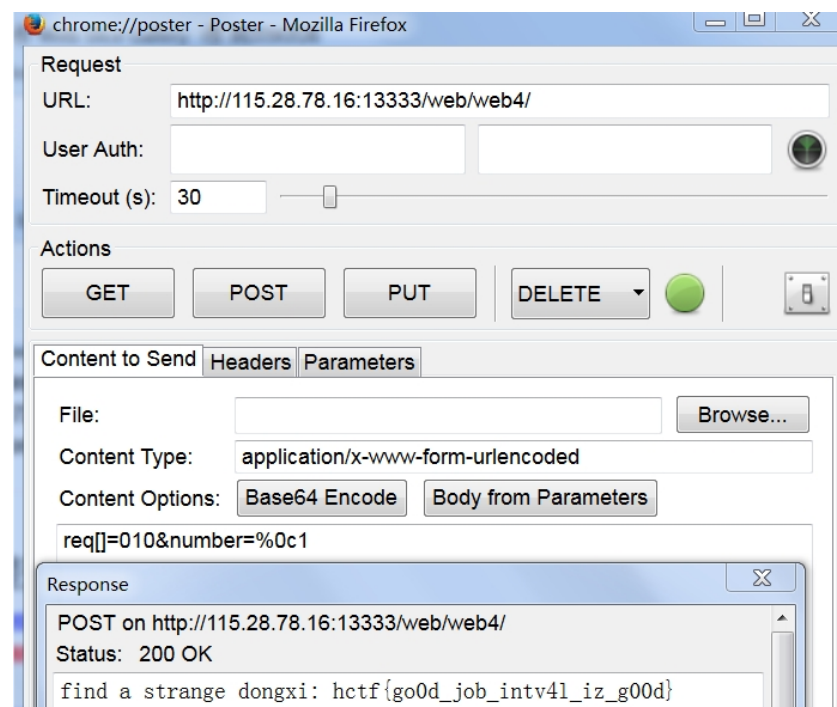


22 我是谁我在哪???



25. 神奇的数字

Ps:想了好久怎么绕最后试各种空白字符 233333 神奇的题目



26、不可能拿到的 flag



```
<?php
if(empty($_POST)){
    highlight_file(__FILE__);
    exit;
}
include_once("flag.php");

if (isset($_POST['name']) and isset($_POST['password'])) {

    if ($_POST['name'] == $_POST['password']){

        print 'Your password can not be your name.';

    }else if (sha1($_POST['name']) == sha1($_POST['password'])){

        die('Flag: '.$flag);

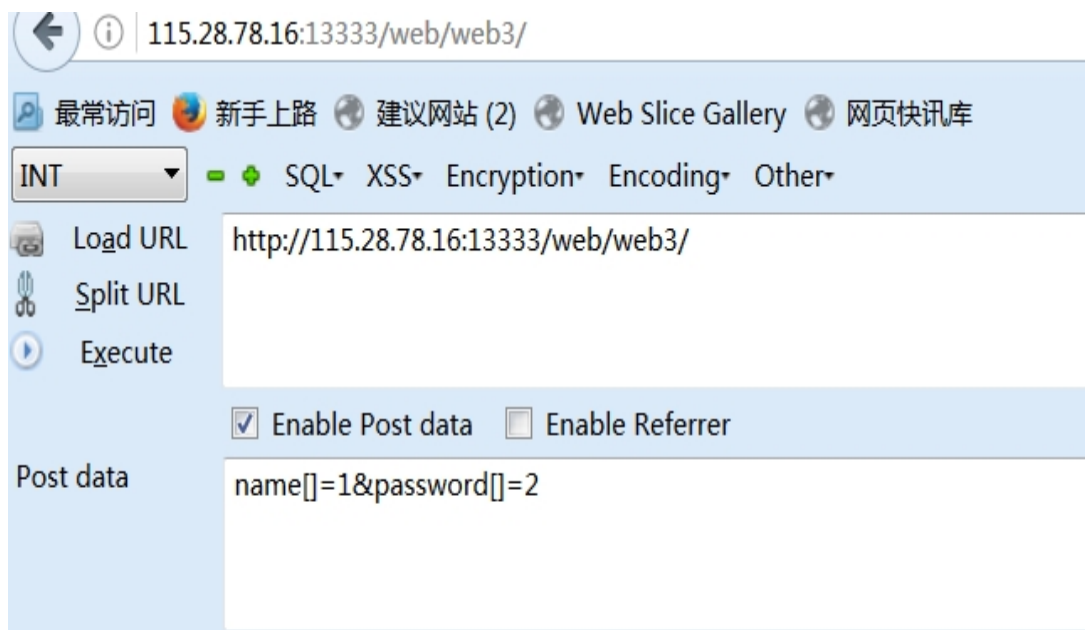
    }else{

        print 'Invalid password';

    }

}
?>
```

利用 php 弱类型，构造 POST 请求 name[]=1&password[]=1  
得到 flag



115.28.78.16:13333/web/web3/

最常用访问 新手上路 建议网站 (2) Web Slice Gallery 网页快讯库

INT SQL XSS Encryption Encoding Other

Load URL http://115.28.78.16:13333/web/web3/

Split URL

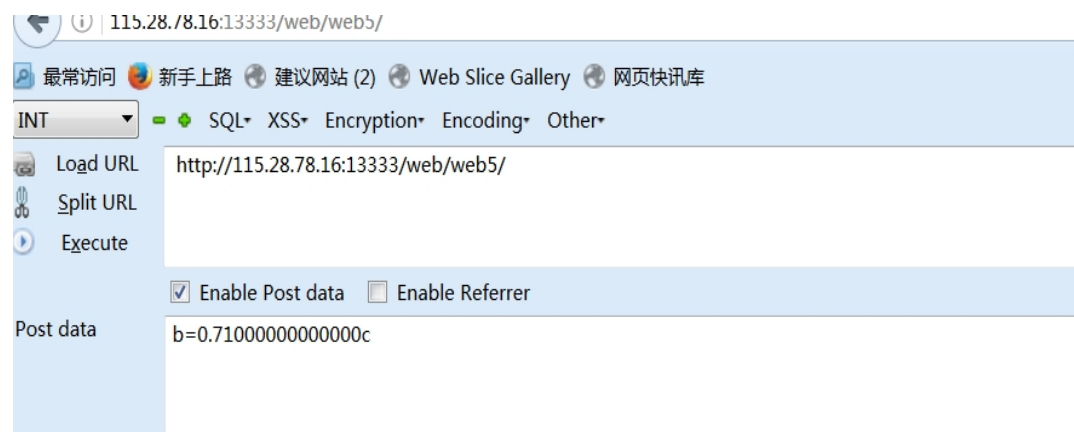
Execute

☒ Enable Post data ☐ Enable Referrer

Post data name[]=1&password[]=2

Flag: hctf{o0k!!g3t\_f14g\_s0\_ez}

27.php 真可怕我要回农村  
依然是构造 post 请求 b=0.710000000000000a 得到 flag



hctf{wochubuxiaque\_over}

Misc 一张图片中寻找 3 个 flag

1. 习惯性直接拖 winhex

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
)	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	00	00	01	ÿØÿà JFIF
)	00	01	00	00	FF	E1	00	A6	45	78	69	66	00	00	49	49	ÿá  Exif II
)	2A	00	08	68	63	74	66	7B	32	65	33	65	33	7D	00	00	* hctf{2e3e3}
)	00	00	00	00	0F	01	02	00	01	00	00	00	00	00	00	00	
)	10	01	02	00	01	00	00	00	00	00	00	00	1A	01	05	00	
)	01	00	00	00	7A	00	00	00	1B	01	05	00	01	00	00	00	z
)	82	00	00	00	28	01	03	00	01	00	00	00	02	00	00	00	, (
)	32	01	02	00	14	00	00	00	8A	00	00	00	3B	01	02	00	2 Š ;
)	01	00	00	00	00	00	00	00	98	82	02	00	01	00	00	00	~,
)	00	00	00	00	00	00	00	00	48	00	00	00	01	00	00	00	H
)	48	00	00	00	01	00	00	00	32	30	31	37	3A	30	31	3A	H 2017:01:
)	31	35	20	31	38	3A	35	35	3A	33	35	00	FF	DB	00	43	15 18:55:35 ŸÛ C
)	00	0A	07	07	08	07	06	0A	08	08	08	0B	0A	0A	0B	0E	
)	18	10	0E	0D	0D	0E	1D	15	16	11	18	23	1F	25	24	22	# % \$"
)	1F	22	21	26	2B	37	2F	26	29	34	29	21	22	30	41	31	"!&+7/&)4)!"0A1
)	34	39	3B	3E	3E	3E	25	2E	44	49	43	3C	48	37	3D	3E	49;>>>% .DIC<H7=>
)	3B	FF	DB	00	43	01	0A	0B	0B	0E	0D	0E	1C	10	10	1C	;ÿÛ C
)	3B	28	22	28	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	; (" (::::::::::::
)	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	::::::::::::
)	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	3B	::::::::::::
)	3B	3B	3B	3B	3B	3B	FF	C0	00	11	08	02	4A	02	98	03	;;;;;;;;ÿÀ J ~
)	01	22	00	02	11	01	03	11	01	FF	C4	00	1F	00	00	01	" ŸÄ

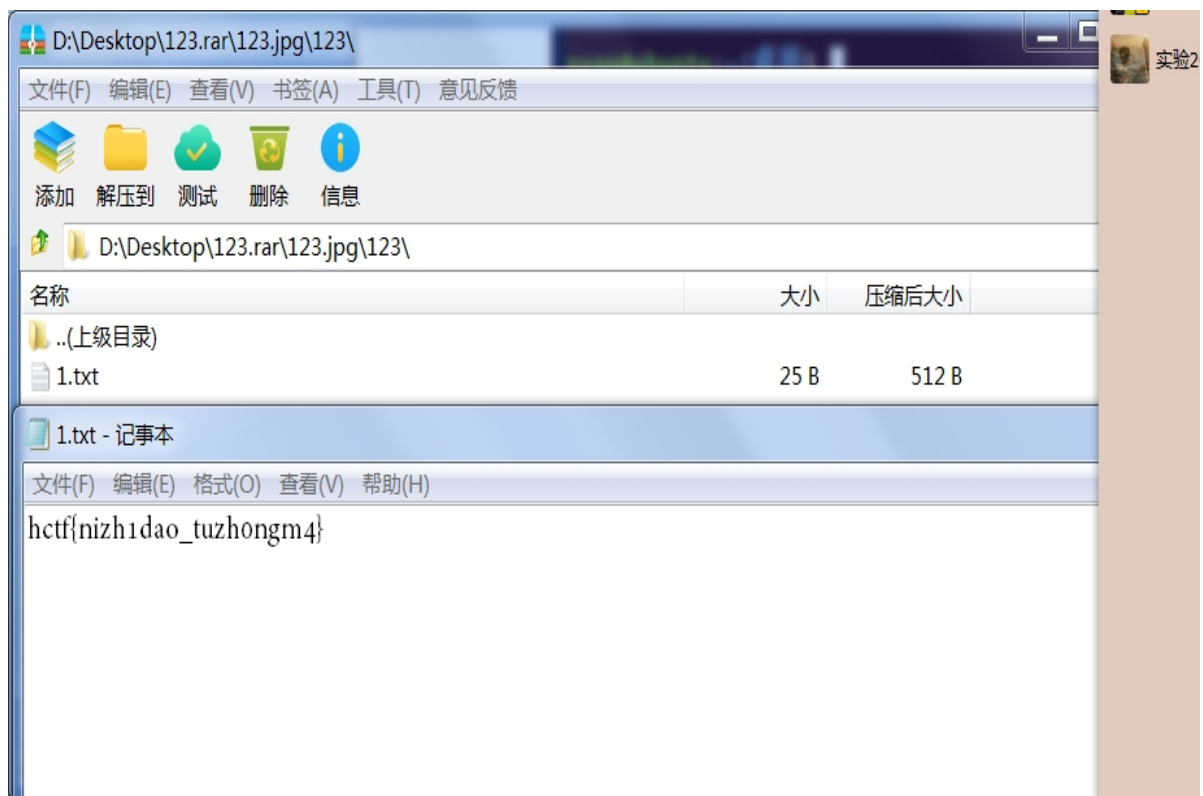
发现文件格式及 flag，改文件名. jpg

Binwalk 一下

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

xuan@ubuntu: ~/桌面\$

发现 gZip 文件，然后我压缩解压一下，最里面有个文件，打开后得到 flag



最后一个比较坑，我以为是加密了什么文件，结果看了一天才看出来在最低位提出文件中有一串 base64 加密 2333333333

Base64: aGN0Znsx c2JfYWFiYmJfaXpf ZXp6enp6en0=

hctf{1sb\_aabbb\_iz\_ezzzzzz}

## crypto

凯撒密码：换位（数字第一次试的是 3，不对，决定从一开始试）

Hgame{Caesar\_cipher\_1s\_just\_for\_fun}

同样凯撒，试一下就知道 ps: 数字我从偏移量一开始试的

hgame{dgfdyhcry42287235413//+/%}

Rsa 加密

通过 RSA 解密公式解得

RSA 是第一个既能用于数据加密也能用于数字签名的算法，易于理解和操作，应用十分广泛。名字以发明者的名字命名：Ron Rivest、Adi Shamir 和 Leonard Adleman。密码分析者既不能证明也定 RSA 的安全性，但这恰恰说明该算法有一定的可信度。

1. 算法原理

① 选取两个大素数： $p$  和  $q$ ，为了获得最大程度的安全性，两数的长度一样。（注：以下所涉理论知识不做过多说明，感兴趣的读者请进一步参阅相关书籍。）

② 计算  $n=p \times q$ ， $n$  称为模。

③ 计算欧拉（Euler）函数： $\phi(n)=(p-1) \times (q-1)$ 。

④ 选取加密密钥  $e$ ，其与  $\phi(n)$  互素。如果选择合适的  $e$  值，RSA 加解密的速度将快得多，常为 3、17 和 65537 ( $2^{16}+1$ )。

⑤ 使用扩展欧几里德算法（Extended Euclid）求出  $e$  模  $\phi(n)$  的逆元  $d$ ，即

$$ed \equiv 1 \pmod{\phi(n)}$$

⑥ 公钥为  $e$  和  $n$ ，私钥为  $d$ ， $p$  和  $q$  可以丢弃，但是必须保密。

⑦ 加密消息  $m$  时，将其看成一个大整数，把它分成比  $n$  小的数据分组，按下面的式子进行加密：

$$c_i \equiv m_i^e \pmod{n}$$

⑧ 对密文  $c$  解密时，取每一个加密后的分组  $c_i$  并计算：

$$m_i \equiv c_i^d \pmod{n}$$

RSA 加解密总结见表 6-3。

- MulInv.c
- Powmod
- CE.EXE
- Factor:
- RSAToo
- 制时，
- Bigcalc:
- 下面举一个
- 设  $p = 37$ ,

选取  $e=17$ .

息:

首先将其

SA 加解密总结见表 6-3。

表 6-3 RSA 加解密	
公钥	$n$ : 两素数 $p$ 和 $q$ 的乘积 ( $q$ 和 $p$ 必须保密) $e$ : 与 $(p-1)(q-1)$ 互素
私钥	$d$ : $e^{-1} \pmod{(p-1)(q-1)}$
加密	$c = m^e \pmod{n}$
解密	$m = c^d \pmod{n}$

SA 的安全性依赖于大整数因子分解，但是否等同于大整

4. 没有证明要解密 RSA 就一定要进行因

$n=p \times q$

结果为 6867616D657B7273615F31735F763372795F65347379217D

显然是十六进制转 ASCII 码