web buuctf [ACTF2020 新生赛]Include1

原创

半杯雨水敬过客 于 2021-08-04 08:57:20 发布 122 收藏 1

1.线索：查看tips页面源代码，只有一个标签传参链接?file=flag.php;

点击查看传参后几面源代码，网页上只有“Can you find out the flag?”一句话

2.分析：没有sql注入点

提示can you find out the flag，且传递的file是flag.php

flag是否在flag.php中

尝试查看一下flag.php代码

3.知识储备： php伪协议；base64

相关链接：https://segmentfault.com/a/1190000018991087

这里用的是查看源代码的：php://filter

//这里单独拿出来了，大家可以自行学习

php:// 协议

条件：

- `allow_url_fopen`:off/on
- `allow_url_include` :仅`php://input` `php://stdin` `php://memory` `php://temp` 需要on

- **作用：**
`php://` 访问各个输入/输出流（I/O streams），在CTF中经常使用的
是`php://filter`和`php://input`，`php://filter`用于**读取源码**，`php://input`用于**执行php代码**。

说明：
PHP 提供了一些杂项输入/输出（IO）流，允许访问 PHP 的输入输出流、标准输入输出和错误描述符，内存中、磁盘备份的临时文件流以及可以操作其他读取写入文件资源的过滤器。

协议	作用
php://input	可以访问请求的原始数据的只读流，在POST请求中访问POST的`data`部分，在`enctype="multipart/form-data"`的时候`php://input` 是无效的。
php://output	只写的数据流，允许以 print 和 echo 一样的方式写入到输出缓冲区。
php://fd	(>=5.3.6)允许直接访问指定的文件描述符。例如 `php://fd/3` 引用了文件描述符 3。
php://memory php://temp	(>=5.1.0)一个类似文件包装器的数据流，允许读写临时数据。两者的唯一区别是 `php://memory` 总是把数据储存在内存中，而 `php://temp` 会在内存量达到预定义的限制后（默认是 `2MB`）存入临时文件中。临时文件位置的决定和 `sys_get_temp_dir()` 的方式一致。
php://filter	(>=5.0.0)一种元封装器，设计用于数据流打开时的筛选过滤应用。对于一体式（`all-in-one`）的文件函数非常有用，类似 `readfile()`、`file()` 和 `file_get_contents()`，在数据流内容读取之前没有机会应用其他过滤器。

php://filter参数详解

该协议的参数会在该协议路径上进行传递，多个参数都可以在一个路径上传递。具体参考如下：

php://filter 参数	描述	
resource=<要过滤的数据流>	必须项。它指定了你要筛选过滤的数据流。	
read=<读链的过滤器>	可选项。可以设定一个或多个过滤器名称，以管道符（*\	*）分隔。
write=<写链的过滤器>	可选项。可以设定一个或多个过滤器名称，以管道符（\	）分隔。
<; 两个链的过滤器>	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。	

可用的过滤器列表（4类）

此处列举主要的过滤器类型，详细内容请参考：https://www.php.net/manual/zh/filters.php

字符串过滤器	作用
string.rot13	等同于`str_rot13()`，rot13变换
string.toupper	等同于`strtoupper()`，转大写字母
string.tolower	等同于`strtolower()`，转小写字母
string.strip_tags	等同于`strip_tags()`，去除html、PHP语言标签

转换过滤器	作用
convert.base64-encode & convert.base64-decode	等同于`base64_encode()`和`base64_decode()`，base64编码解码
convert.quoted-printable-encode & convert.quoted-printable-decode	quoted-printable 字符串与 8-bit 字符串编码解码

压缩过滤器	作用
zlib.deflate & zlib.inflate	在本地文件系统中创建 gzip 兼容文件的方法，但不产生命令行工具如 gzip的头和尾信息。只是压缩和解压数据流中的有效载荷部分。
bzip2.compress & bzip2.decompress	同上，在本地文件系统中创建 bz2 兼容文件的方法。

加密过滤器	作用
mcrypt.*	libmcrypt 对称加密算法
mdecrypt.*	libmcrypt 对称解密算法 `php://filter/read=convert.base64-encode/resource=[文件名]`读取文件源码（针对php文件需要base64编码）

`php://filter/read=convert.base64-encode/resource=[文件名]`读取文件源码（针对php文件需要base64编码）

`php://input + [POST DATA]`执行php代码

4.构造payload

```
?file=php://filter/read=convert.base64-encode/resource=flag.php
```

打开界面后得到：

PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL2ZsYWd7OTI4Zjc5N2UtOTY4MC

进行base64解码，得到：

flag{928f797e-9680-4928-9111-59ede16e84bf}