ctf中怎样获取php文件源码,ctf中关于php伪协议的考查

转载

原创： Archerx 合天智汇

1

php://input协议

第一个例子

flag.php

test1.php

include('flag.php');

$a= $_GET["a"];

if(isset($a)&&(file_get_contents($a,'r'))=== 'this is test'){

echo"success\n";

echo$flag;

}

else{

echo"error";

}

看上面php代码可知当读取文件的内容是thisis test时才显示flag，我们并不知道那个文件有这个内容，我们可以使用php://这个协议php://input可以得到原始的post数据,访问请求的原始数据的只读流,将post请求中的数据作为PHP代码执行,如下操作来绕过：

使用条件：

allow_url_fopen：off/on

allow_url_include：on

第二个例子

php://input实现代码执行

test1.php改为如下

$a= $_GET["a"];

include($a);

注：只在php5.2.17 下测试成功，其他均出现报错，原因未知。

php://filter/convert.base64-encode/resource=

看另外一个代码：

$a= $_GET['a'];

include($a);

如何显示flag.php的内容呢?直接包含是不会显示的，这时就要用到这个php://filter/convert.base64-encode/resource=取源代码并进行base64编码输出，不然会直接当做php代码执行就看不到源代码内容了。

php://filter在双off的情况下也可以正常使用；

allow_url_fopen：off/on

allow_url_include：off/on

利用反序列化读取文件

借鉴2016xctf 一道题的思路，代码被我简化了：

classflag{

public$file;

publicfunction __tostring(){

echofile_get_contents($this->file);

return'yes';

}

}

$a= new flag();

$a->file= 'php://filter/convert.base64-encode/resource=flag.php';

$data= serialize($a);

echo$data.'
';

echounserialize($data);

定义一个flag类，并重写了tostring()，我们先new一个新对象，并给变量赋值，最后序列化一下。

假设在某个题目中序列化后变量是可控的而且我们知道类内容，那我们就可以通过可控变量实现任意文件读取，如上代码中，反序列化过程中实现了flag.php文件的读取

2

file://协议

file://协议在双off的情况下也可以正常使用；

allow_url_fopen：off/on

allow_url_include：off/on

file://用于访问本地文件系统，在CTF中通常用来读取本地文件的且不受allow_url_fopen与allow_url_include的影响

前几天某比赛web第二道题就是利用注入控制反序列化，file://协议读取本地文件

注：file://协议必须是绝对路径

zip://,bzip2://, zlib://协议

双off情况下正常使用

allow_url_fopen：off/on

allow_url_include：off/on

payload：

先将要执行的PHP代码写好文件名为phpcode.txt，将phpcode.txt进行zip压缩,压缩文件名为file.zip,如果可以上传zip文件便直接上传，若不能便将file.zip重命名为file.jpg后在上传，其他几种压缩格式也可以这样操作。

由于#在get请求中会将后面的参数忽略所以使用get请求时候应进行url编码为%23，且此处经过测试相对路径是不可行，所以只能用绝对路径。

3

phar协议

1.jpg是一个里面含有1.php的压缩包，改了后缀名，包含方法如下。

include.php?f=phar://./images/1.jpg/1.php

4

zlib://协议

使用方法：

compress.zlib://file.gz

绝对路径

相对路径

5

总结

上面只是最基础的例子，在ctf中要会活用，正所谓再难的题也离不开基础。

题外话：近来国内ctf比赛越来越趋向于国际化，pwn、re题目占了绝大部分，web题很少或者直接没有，作为一个web狗要坚强的走下去。

(如需转载，请注明出处)