计科 234 徐潘第 10 周周报

```
#**Upload-Labs-Pass15**



**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
```

```
查看网站源码：
```

```
```shell
function isImage($filename){
    $types = '.jpeg|.png|.gif';
    if(file_exists($filename)){
        $info = getimagesize($filename);
        $ext = image_type_to_extension($info[2]);
        if(stripos($types,$ext)>=0){
            return $ext;
        }else{
            return false;
        }
    }else{
        return false;
    }
}
```

```
$is_upload = false;
$msg = null;
if(isset($_POST['submit'])){
    $temp_file = $_FILES['upload_file']['tmp_name'];
    $res = isImage($temp_file);
    if(!$res){
        $msg = "文件未知，上传失败！";
    }else{
        $img_path = UPLOAD_PATH."/".rand(10, 99).date("YmdHis").$res;
        if(move_uploaded_file($temp_file,$img_path)){
            $is_upload = true;
        } else {
            $msg = "上传出错！";
```
```

```
        }
    }
}
```

可以看到这次网站使用了 getimagesize()函数，getimagesize（）函数返回一个包含图像信息的数组。该数组的索引如下所示：

索引 0：图像的宽度（单位：像素）

索引 1：图像的高度（单位：像素）

索引 2：图像类型的常量值（可以使用 image type to mime type（）函数将其转换为 MIME 类型），1 代表 gif，2 代表 jpeg，3 代表 png

索引 3：包含图像属性的字符串，以逗号分隔（如“width=500,height=300"）

如果 getimagesize（）函数无法读取图像信息，则返回 false 否则，返回一个包含上述索引的数组。

然后使用 image_type_to_extension()函数根据得到的图像类型的常量值将上传的文件转化为对应的类型

本关不能直接修改一句话木马前两个字符了，需要生成图片马来绕过

在 cmd 命令区输入：

```shell
C:\Users\xupan\Desktop\网安实验室\CTF\upload-labs 资料>copy Ghost.jpg/b+trojan.php 1Ghost.jpg
```

这样就将一个 jpg 图片和一句话木马 php 文件合成了一个 jpg 类型的图片马，其中一句话木马代码在图片代码的后面，否则头字节可能会被检测出错

将图片马成功上传，然后利用文件包含漏洞，在网址上使用 file 访问，使得图片马以 php 的方式运行：

```shell
http://localhost/upload-labs/index.php?file=./upload/1220250329082130.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，可以通过在新签页中打开图片马得到

使用蚁剑连接后门也能连接成功，png 和 gif 图片马同理

#**Upload-Labs-Pass16**

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

查看网站源码：

```shell
function isImage($filename){
    //需要开启 php_exif 模块
    $image_type = exif_imagetype($filename);
    switch ($image_type) {
        case IMAGETYPE_GIF:
            return "gif";
            break;
        case IMAGETYPE_JPEG:
            return "jpg";
            break;
        case IMAGETYPE_PNG:
            return "png";
            break;
        default:
            return false;
            break;
    }
}
```

```shell
$is_upload = false;
$msg = null;
if(isset($_POST['submit'])){
    $temp_file = $_FILES['upload_file']['tmp_name'];
    $res = isImage($temp_file);
    if(!$res){
        $msg = "文件未知，上传失败！";
    }else{
        $img_path = UPLOAD_PATH."/".rand(10, 99).date("YmdHis").".".$res;
        if(move_uploaded_file($temp_file,$img_path)){
            $is_upload = true;
        } else {
            $msg = "上传出错！";
        }
    }
}
```

可以看到本关依旧是会先判断图片类型，输出图像类型的常量值（1 代表 gif，2 代表 jpeg，3 代表 png）来判断

绕过检测的思路依旧是上传图片马然后以 php 文件的方式解析：

在 cmd 命令区输入：

```shell
C:\Users\xupan\Desktop\网安实验室\CTF\upload-labs 资料>copy Ghost.jpg/b+trojan.php 1Ghost.jpg
```

这样就将一个 jpg 图片和一句话木马 php 文件合成了一个 jpg 类型的图片马，其中一句话木马代码在图片代码的后面，否则头字节可能会被检测出错

将图片马成功上传，然后利用文件包含漏洞，在网址上使用 file 访问，使得图片马以 php 的方式运行：

```shell
http://localhost/upload-labs/index.php?file=./upload/2720250329102707.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，可以通过在新签页中打开图片马得到

使用蚁剑连接后门也能连接成功，png 和 gif 图片马同理

#**Upload-Labs-Pass17**

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

查看网站源码：

```shell
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])){
    // 获得上传文件的基本信息，文件名，类型，大小，临时文件路径
    $filename = $_FILES['upload_file']['name'];
    $filetype = $_FILES['upload_file']['type'];
    $tmpname = $_FILES['upload_file']['tmp_name'];
```

```
    $target_path=UPLOAD_PATH.'/'.basename($filename);
```

```
    // 获得上传文件的扩展名
    $fileext= substr(strrchr($filename,"."),1);
```

```
    //判断文件后缀与类型，合法才进行上传操作
    if(($fileext == "jpg") && ($filetype=="image/jpeg")){
        if(move_uploaded_file($tmpname,$target_path)){
```

```
        //使用上传的图片生成新的图片
        $im = imagecreatefromjpeg($target_path);
```

```
        if($im == false){
            $msg = "该文件不是 jpg 格式的图片！";
            @unlink($target_path);
        }else{
            //给新图片指定文件名
            srand(time());
            $newfilename = strval(rand()).".jpg";
            //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;
            imagejpeg($im,$img_path);
            @unlink($target_path);
            $is_upload = true;
        }
    } else {
        $msg = "上传出错！";
    }
```

```
}else if(($fileext == "png") && ($filetype=="image/png")){
    if(move_uploaded_file($tmpname,$target_path)){
        //使用上传的图片生成新的图片
        $im = imagecreatefrompng($target_path);
```

```
        if($im == false){
            $msg = "该文件不是 png 格式的图片！";
            @unlink($target_path);
        }else{
             //给新图片指定文件名
            srand(time());
            $newfilename = strval(rand()).".png";
            //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;
            imagepng($im,$img_path);
```

```
            @unlink($target_path);
            $is_upload = true;
        }
    } else {
        $msg = "上传出错！";
    }
```

```
}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;
            }
        } else {
            $msg = "上传出错！";
        }
    }else{
        $msg = "只允许上传后缀为.jpg|.png|.gif 的图片文件！";
    }
}
```

可以看到本关在接收上传的文件时，先在网站前端获取文件名及其后缀名，在后端判断名称合法后，再将文件移动至上传路径

之后关键的步骤是：使用 `imagecreatefromjpeg()`函数对文件进行重写，重写时对其代码内部的非法内容（如一句话木马）进行删除和修改，即二次渲染

需要注意的是，二次渲染是会保留系统认为与图片文件相关的代码的，因此我们可以尝试寻找二次渲染的规律

首先上传一个正常的图片，上传成功后从网站上再将这个图片保存回本地上，使用 `010Editor` 中的比较工具比较上传前的该图片和刚才从网站上保存下来的图片的十六进制代码

可以看到对比后二者有标识为蓝色的共有的部分，这就是在二次渲染中保留的代码，可以将一句话木马写到其中一个保留的部分（尽量在靠后的位置）

上传带有一句话木马的文件，上传成功后从网站上再将这个图片保存回本地上，可以看到一句话木马在二次渲染后保留了下来

利用文件包含漏洞，在网址上使用 `file` 访问，使得图片马以 `php` 的方式运行：

```shell
http://localhost/upload-labs/index.php?file=./upload/750255727.gif
```

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，可以通过在新签页中打开图片马得到

使用蚁剑连接后门也能连接成功，png 和 jpg 图片马同理

注意，有些图片可能在二次渲染后保留的部分很少，不方便插入一句话木马，可以直接用上传成功后从网站上再将这个图片保存回本地上的二次渲染后后的图片做图片马