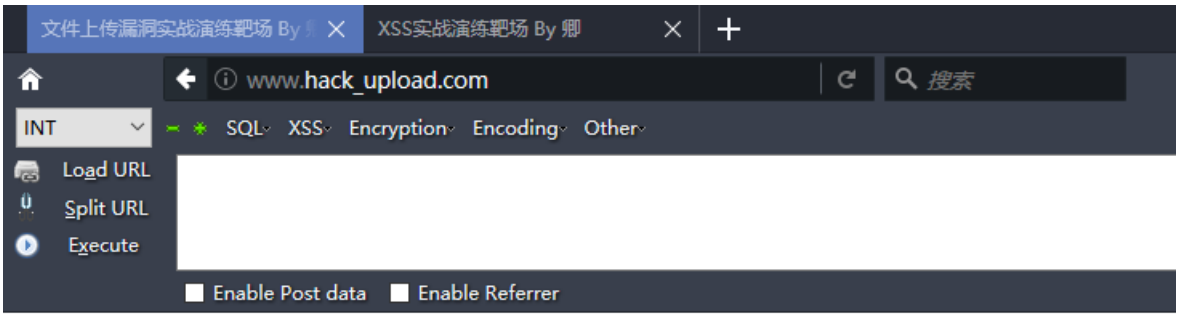
文件上传漏洞实战靶场笔记

转载

weixin_30544657 于 2019-06-11 12:39:00 发布 984 收藏 8
文章标签： php
原文链接：http://www.cnblogs.com/-qing-/p/11002959.html

记录下自己写的文件上传漏洞靶场的write up，包括了大部分的文件上传实战场景，做个笔记。

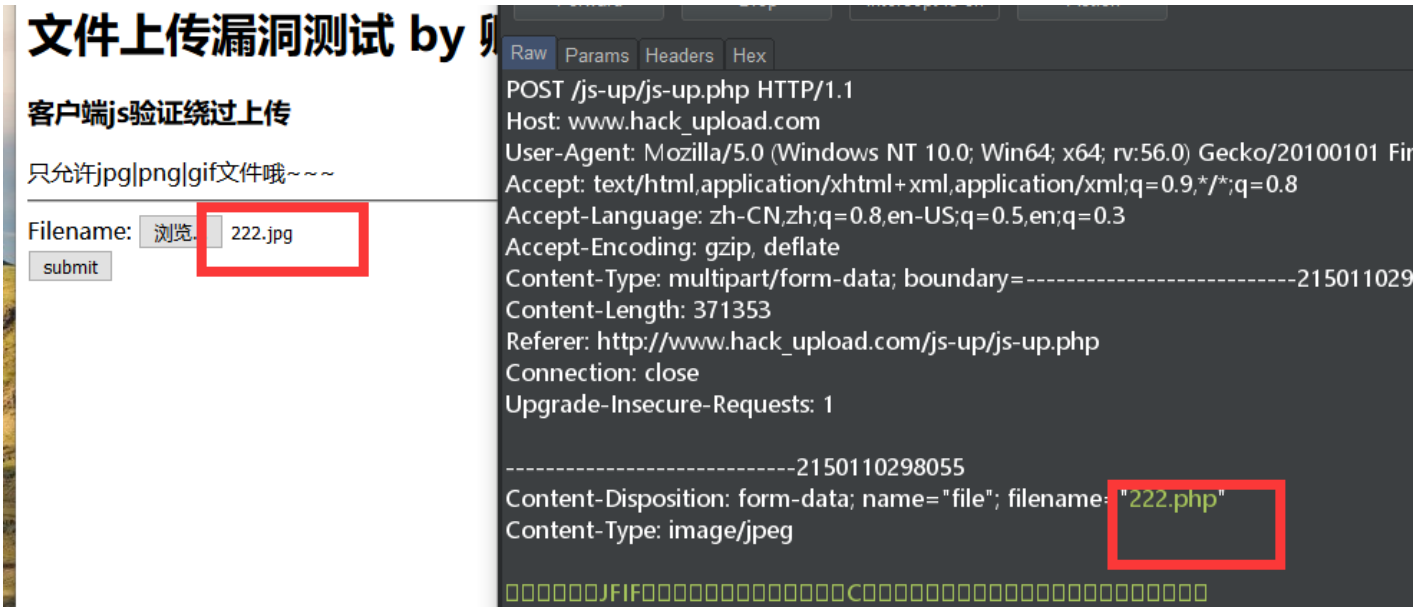


文件上传漏洞实战演练靶场 By 卿

- 客户端js验证绕过上传
- MIME类型验证绕过上传
- 黑名单后缀名(php3、phtml)绕过上传
- .htaccess配置文件绕过上传
- 黑名单后缀名(大小写)绕过上传
- 黑名单后缀名(空格)绕过上传
- 黑名单后缀名(点)绕过上传
- 黑名单后缀名(::$DATA)绕过上传
- 黑名单后缀名(后缀点+空格+点)绕过上传
- 黑名单后缀名(双写后缀)绕过上传
- get类型%00截断绕过上传
- post类型%00截断绕过上传
- 图片内容伪造绕过上传
- 竞争条件绕过上传
- CVE-2015-2348绕过上传

0x01 客户端js验证绕过上传

只是客户端验证 关闭js或者抓包上传即可。



0x02 MIME类型验证绕过上传

验证了Content-Type 改成图片的MIME类型即可

0x03 黑名单后缀名(php3、phtml)绕过上传

这关检测了文件后缀名，禁止了常见动态脚本后缀

不过php3 和pthml没有禁止，在apache配置文件中当php解析

```
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
AddType application/x-httpd-php .php .phtml .php3
</IfModule>
```

文件上传漏洞测试 by 卿

黑名单文件后缀名绕过上传

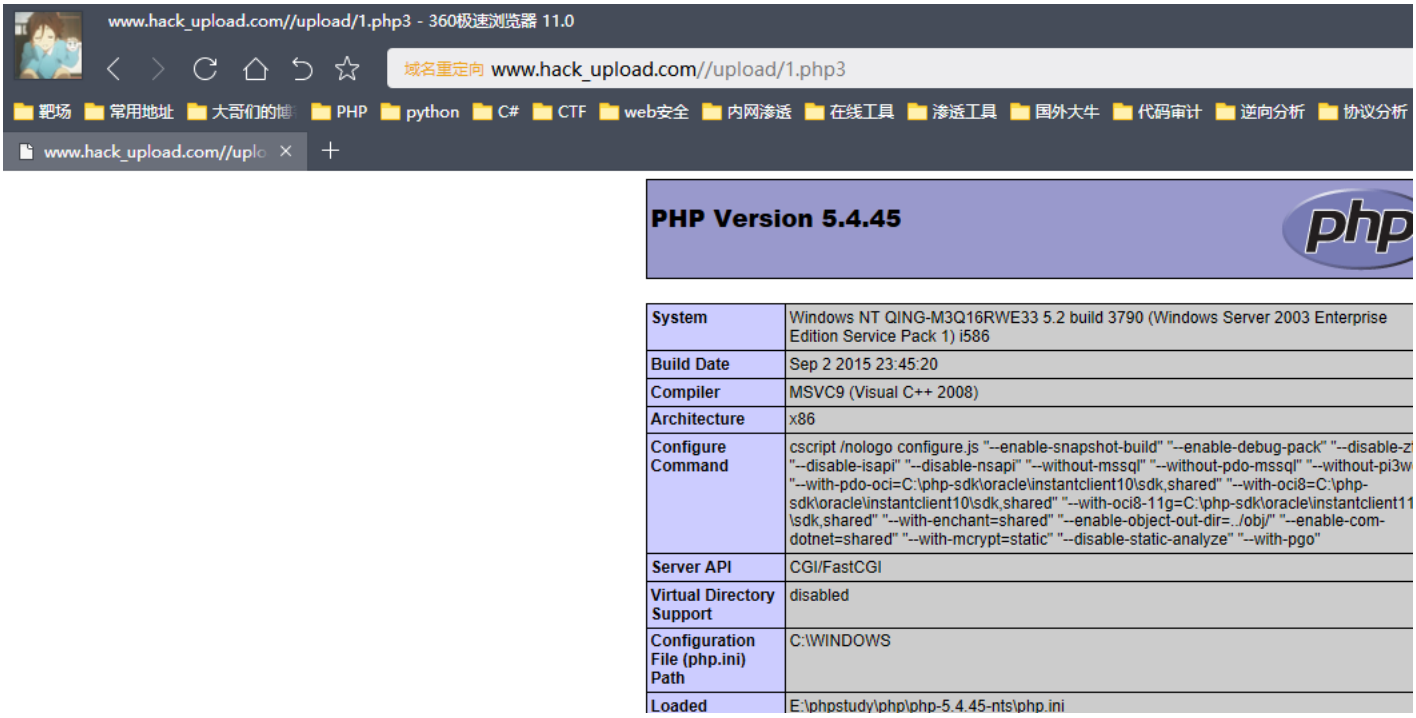
不允许.asp.aspx.php.jsp动态脚本后缀的文件上传！！！

Filename: 浏览... 未选择文件。

Submit

Stored in: ../upload/20190614172624656.php3

Upload: 20190614172624656.php3
Type: application/octet-stream
Size: 0.0224609375 Kb
Temp file: C:\WINDOWS\php41.tmp

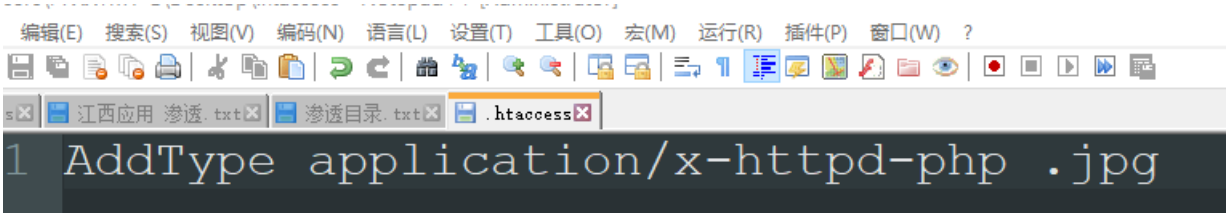


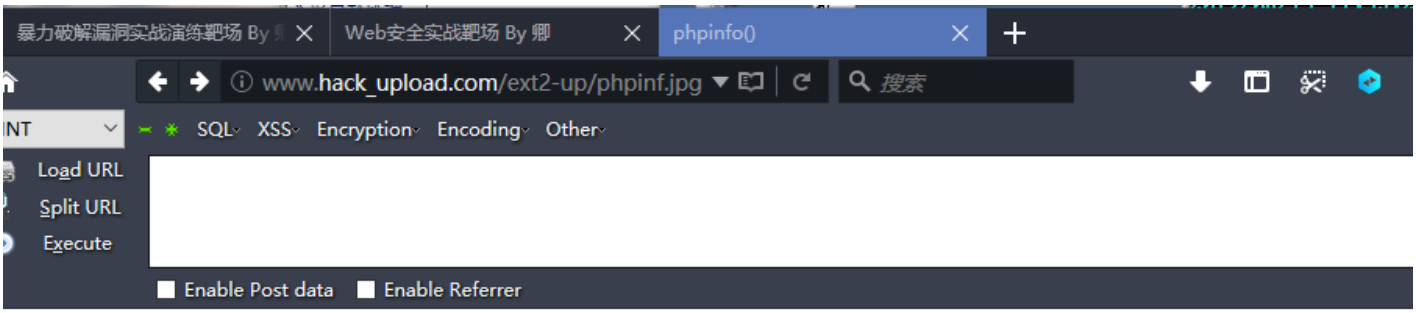
注意php的版本 非线程安全版本的php是解析不了的

0x04 .htaccess配置文件绕过上传

这关把所有动态后缀都禁用了，包括php3、phtml，但是目录下没有**.htaccess配置文件**

我们可以通过上传这个配置文件达到以php解析的目的





PHP Version 5.4.45

System	Windows NT QING-M3Q16RWE33 5.2 build 3790 (Windows Server 2003 Enterprise Edition Service Pack 1) i586
Build Date	Sep 2 2015 23:45:53
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory	enabled

0x05 黑名单后缀名(大小写)绕过上传

这关已经有**.htaccess配置文件，**不过可以通过文件后缀大小写来绕过。例如PHp，测试尝试即可。

0x06

黑名单后缀名(空格)绕过上传

黑名单后缀名(点)绕过上传

黑名单后缀名(::$DATA)绕过上传

这三关都是通过在文件后缀后面加相应的字符，绕过服务端对于文件后缀的检测，执行的时候又以php执行，达到上传木马的目的。

这里利用的都是windows系统的特性，文件后缀有空格、点、::$DATA、<,·>,>>>,0x81-0xff等但不限于这些字符时，windows会自动去除这些字符。

例如1.php::$DATA　保存的时候就是1.php

0x07 黑名单后缀名(后缀点+空格+点)绕过上传

这关单纯去除了末尾的点和空格，所以 点+空格+点 第一次过滤后剩 点，又达到我们前面几关类似的情况，造成上传。

```
echo $_FILES["file"]["name"] . " already exists. ";
}else{
  $deny_ext = array(".php",".php5",".php4",".php3",".php2","php1",".html",".
  $file_name = trim($_FILES['file']['name']);
  $file_name = deldot($file_name);//过滤末尾的.
  $file_ext = strrchr($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)) {
 //非黑名单后缀 可以上传
  $img_path = "../upload/".date("YmdHis").rand(1000,9999).$file_name;
```

0x08 黑名单后缀名(双写后缀)绕过上传

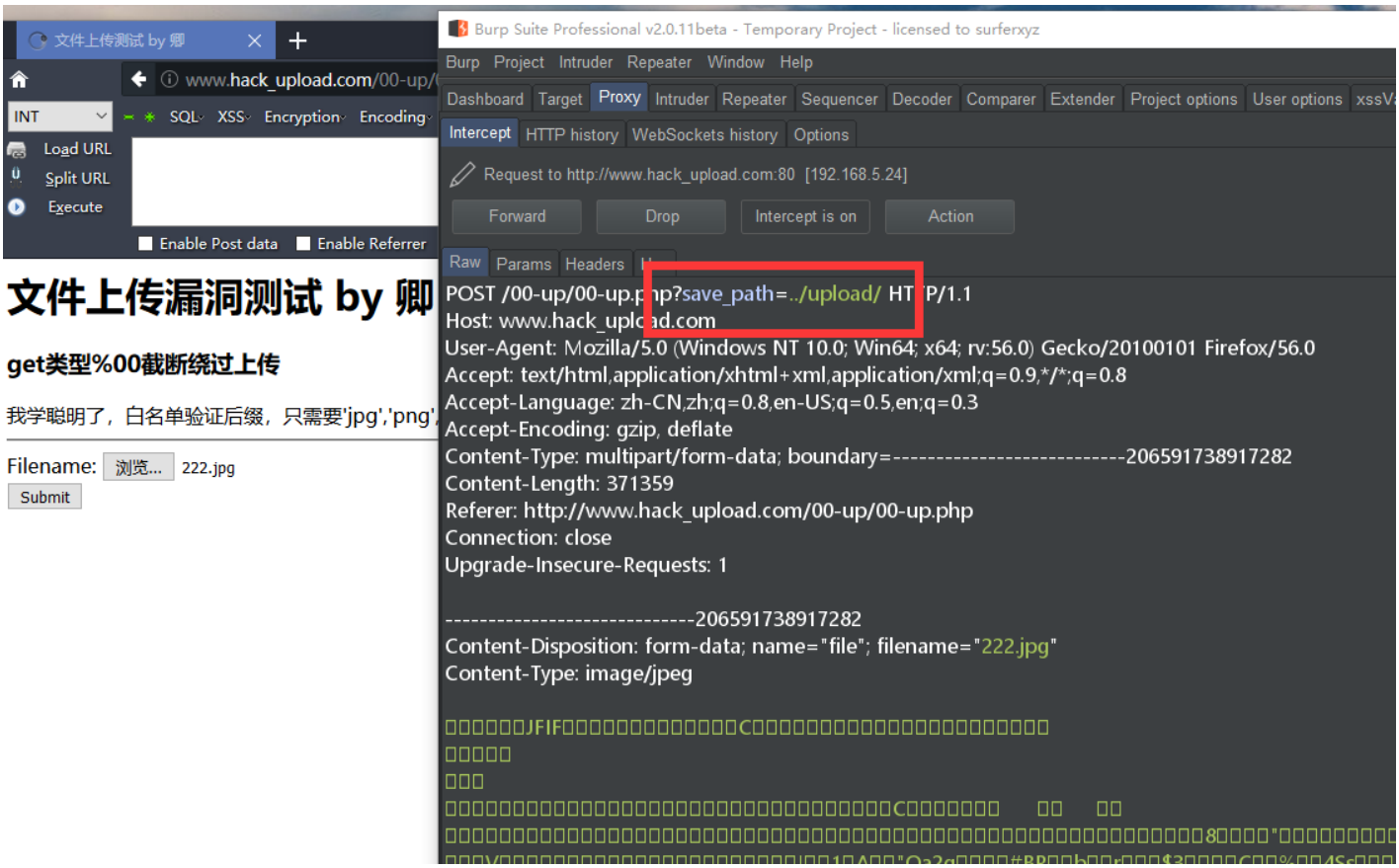
这关是个逻辑代码问题，使用str_replace函数对文件中动态脚本名替换为空，我们可以双写后缀

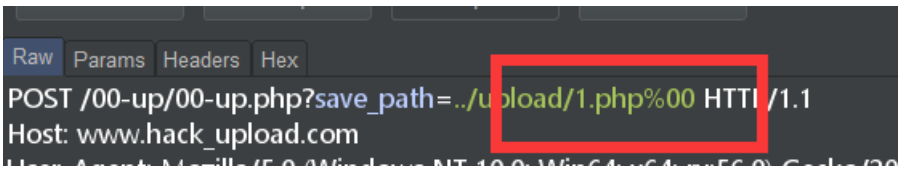
例如1.phphpp 去除了中间的php变成1.php，同样达到我们的目的。

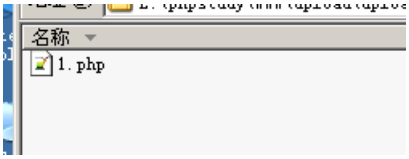
```
  if (file_exists("../pload/" . $_FILES["file"]["name"]))
{
echo $_FILES["file"]["name"] . " already exists. ";
}else{
  $deny_ext = array(".php",".php5",".php4",".php3",".php2","php1",".html",
  $file_name = str_replace($deny_ext,"", $file_name);
    if(!in_array($file_ext, $deny_ext)) {
 //非黑名单后缀 可以上传
  $img_path = "../upload/".date("YmdHis").rand(1000,9999).$file_name;
    move_uploaded_file($_FILES["file"]["tmp_name"],
```

0x09 get类型%00截断绕过上传

这关报错的路径在get参数里 可以直接%00绕过 ， 00截断版本只限于5.3之前哦~







0x0A POST类型%00截断绕过上传

这关报错的路径在post参数里 可以直接00截断绕过 ， 唯一不同的是post中内容不会给你直接url编码， 所以我们需要改一下hex,00截断版本只限于5.3之前哦~

```
POST /00_2-up/00_2-up.php HTTP/1.1
Host: www.hack_upload.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------39132358925690
Content-Length: 371465
Referer: http://www.hack_upload.com/00_2-up/00_2-up.php
Connection: close
Upgrade-Insecure-Requests: 1

-----------------------------39132358925690
Content-Disposition: form-data; name="save_path"

../upload/hac .php
-----------------------------39132358925690
Content-Disposition: form-data; name="file"; filename="222.jpg"
Content-Type: image/jpeg
```

在后面加上hack.php+空格 然后切换hex选项卡，找到hack.php后面这个空格的hex，对应是20，我们改成00

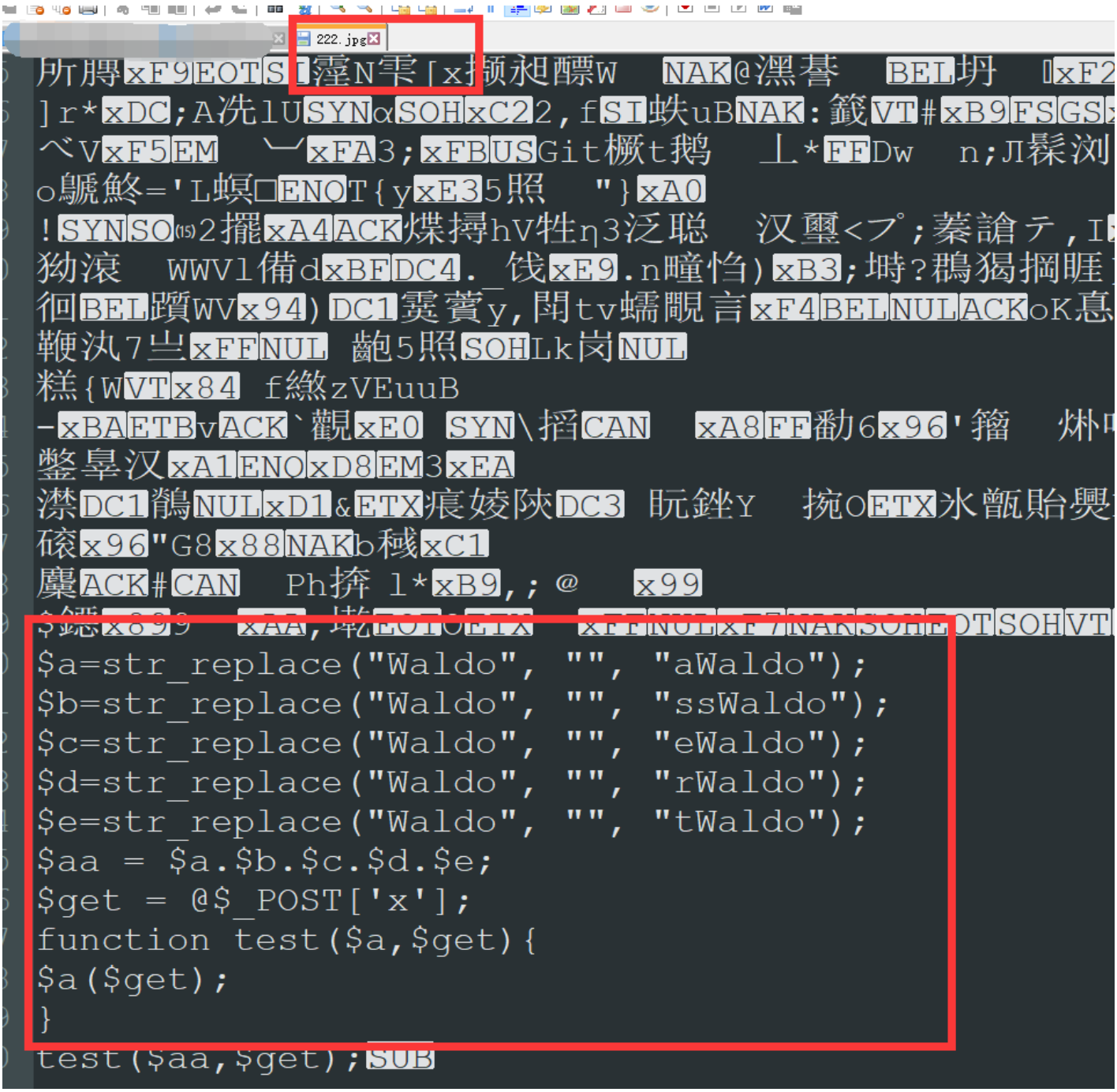
2f	75	70	6c	6f	61	64	2f	h" /upload/
20	0d	0a	2d	2d	2d	2d	2d	hack.php -----
2d	2d	2d	2d	2d	2d	2d	2d	

```
../upload/hack.php
```

即可上传

0x0B 图片内容伪造绕过上传

这关检测了内容 我们使用cmd合成图片马或者加上图片文件头(列如GIF89A)进行欺骗即可



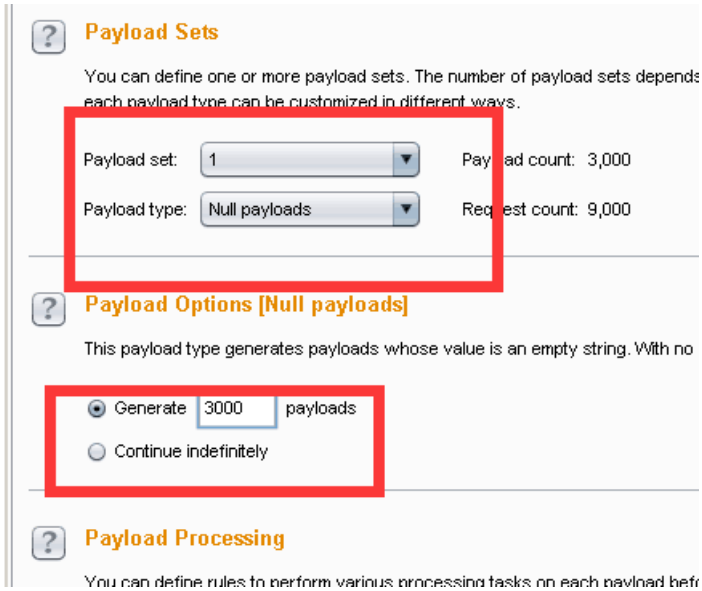
0x0C 竞争条件绕过上传

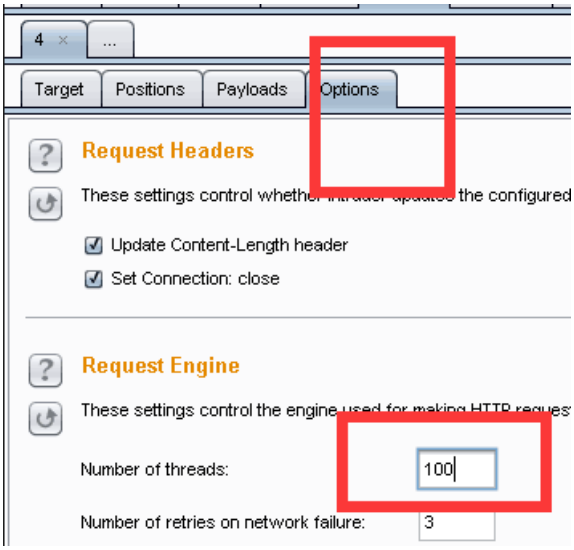
这关利用的是竞争条件，服务器先允许你上传文件，然后检测是否合法，不合法再删除，我们要利用的就是在服务器删除前，访问到我们上传的php。

例如这里我准备一个tj.php，内容为

```
<?php
$myfile = fopen("qing.php", "w");
$txt = "<?php phpinfo();?>";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

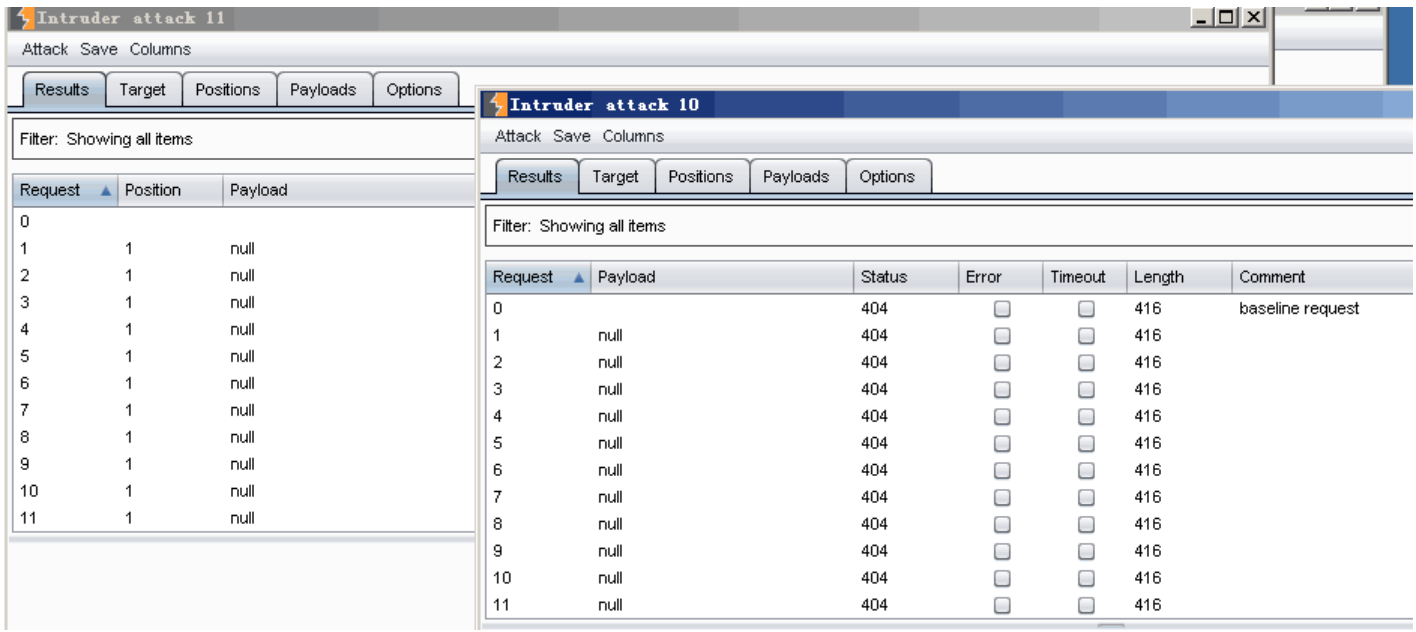
条件竞争中burp需要线程设置偏大



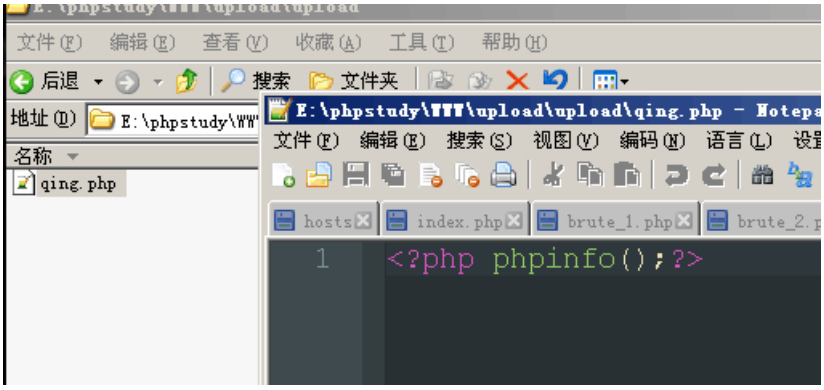


这里我上传我的tj.php，然后不停的访问tj.php上传后的地址，即http://www.hack_upload.com/upload/tj.php

这里使用两个发包器，一个包是上传我们tj.php的包，一个是访问我们上传tj.php后的地址

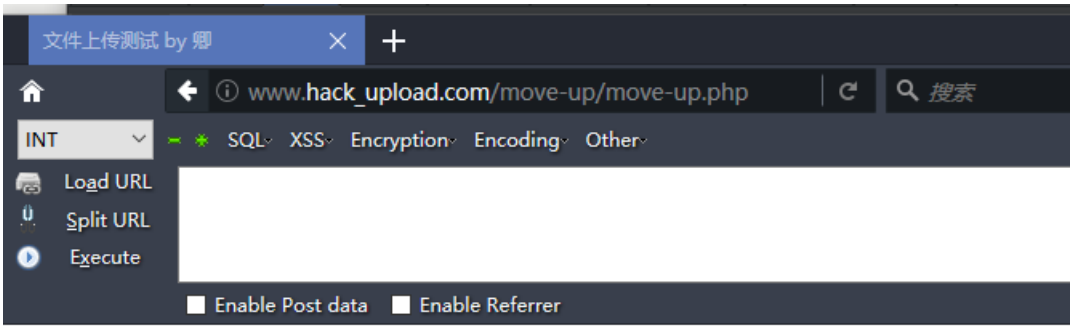


利用条件竞争，访问tj.php成功，所以新建了一个qing.php



0x0D CVE-2015-2348绕过上传

这关利用的是CVE-2015-2348 move_uploaded_file() 00截断，这里需要提交一个文件名作为保存文件的名称



文件上传漏洞测试 by 卿

CVE-2015-2348绕过上传

move_uploaded_file()函数有问题？？我可不信，php是最好的语言！！

Filename:

保存名称:

填写保存文件的名称

浏览... 未选择文件。

Submit

看源码发现move_uploaded_file()函数中的img_path是由post参数save_name控制的，因此可以在save_name利用00截断绕过：

```
}else{
  $deny_ext = array('jpg','png','gif');
  $file_name = $_POST['save_name'];
  $file_ext = substr($file_name,strrpos($file_name,".")+1);//后缀
  $img_path = "../upload/".date("YmdHis").rand(1000,9999).$file_name;
    if(in_array($file_ext, $deny_ext)) {
    move_uploaded_file($_FILES["file"]["tmp_name"], $img_path);
    echo "Stored in: " . $img_path."<br/><hr>";
```

转载于:https://www.cnblogs.com/-qing-/p/11002959.html