目录

**[一、题目来源 - 3 -](#_Toc6036)**

**[二、解题工具 - 3 -](#_Toc19600)**

**[三、小类目录 - 4 -](#_Toc10065)**

**[四、基础知识 - 4 -](#_Toc29860)**

**[五、知识点详解 - 5 -](#_Toc21463)**

**[①JS验证绕过 - 5 -](#_Toc238)**

**[②MIME类型绕过 - 10 -](#_Toc30982)**

**[③黑名单绕过 - 11 -](#_Toc13400)**

**[④文件头检查绕过 - 13 -](#_Toc29314)**

**[⑤.htaccess文件 - 14 -](#_Toc5838)**

**[⑥.user.ini文件 - 15 -](#_Toc16381)**

**[⑦windows服务器下的绕过 - 16 -](#_Toc10984)**

**[1) 大小写绕过 - 16 -](#_Toc31918)**

**[2) 后缀（空）绕过 - 17 -](#_Toc15866)**

**[3) 后缀（点）绕过 - 17 -](#_Toc19372)**

**[4) ::$DATA(windows文件流绕过) - 17 -](#_Toc3463)**

**[⑧00截断 - 17 -](#_Toc10252)**

**[⑨图片马绕过 - 20 -](#_Toc5486)**

**[⑩二次渲染 - 22 -](#_Toc16105)**

**[1) GIF - 23 -](#_Toc13378)**

**[2) PNG - 23 -](#_Toc27336)**

**[3) JPG - 23 -](#_Toc32690)**

**[⑩①条件竞争 - 24 -](#_Toc25330)**

**[⑩②apache解析漏洞 - 24 -](#_Toc12526)**

**[⑩③nginx解析漏洞 - 25 -](#_Toc1716)**

**[⑩④IIS解析漏洞 - 25 -](#_Toc4454)**

**[1) 目录解析 - 25 -](#_Toc11184)**

**[2) 文件解析 - 25 -](#_Toc21086)**

**[3) 默认可执行文件 - 25 -](#_Toc20386)**

**[⑩⑤压缩包解压漏洞 - 25 -](#_Toc19998)**

**本篇旨在引导小白入门，并不会对文件上传中的难点进行大篇幅介绍，介绍的内容也没有深入，不过模块所列出的都是常见的绕过套路，读者需要好好理解。**

**目录使用方法：按住ctrl键再点击即可跳转对应内容**

**一、题目来源**

全来自于upload-labs (下载地址：github.com/c0ny1/upload-labs)

(有些小类的题目链接我会放在小类说完之后，方便大家练习，网上都有WP)

**二、解题工具**

|  |  |  |  |
| --- | --- | --- | --- |
| 工具类型 | 工具名称 | 版本 | 备注 |
| 基础环境 | **Php环境变量** | **随意** |  |
| **攻防世界账号**  **CTFHub账号**  **Github账号** | **i春秋账号**  **BUU、bugku账号** |  |
| 工具 | **Burp** | **要会抓包**  **要会intruder模块** | **要会用基础的repeater模块** |
|  |  |  |
| 脚本 | **Jpg\_payload** | **文中涉及到时会稍微** | **介绍一下的** |
|  |  |  |
|  |  |  |

**三、小类目录**

（基础）

·JS验证绕过

·MIME类型绕过

·黑名单绕过

·文件头检查绕过

·.htaccess文件(看清楚最前面有个点，下同)

·.user.ini文件(有练习题的简单WP)

·有关windows的绕过

·00截断

（进阶）

·图片马绕过

·二次渲染

·条件竞争

（解析漏洞）

·apache解析漏洞

·nginx解析漏洞

·IIS6.0漏洞

（）

·有关上传压缩包的漏洞

**四、基础知识**

* 先找到上传点。
* 网站一般对你所上传的文件都会有判断，例如判断你上传的文件后缀是否为jpg、png等，或者对你上传文件的文件头进行检查，又或者是其他的一些检查，使得你无法随意的上传你想传的文件(例如一句话木马)，这时，我们就需要利用一些方法来绕过这些检查。
* 上传后，题目会返回给你上传文件所保存的路径(也有些题目不告诉保存路径)，这时我们就可以用菜刀or蚁剑连接上shell从而拿到更多信息。
* 一句话木马：<?php @eval($\_POST['123']); ?>

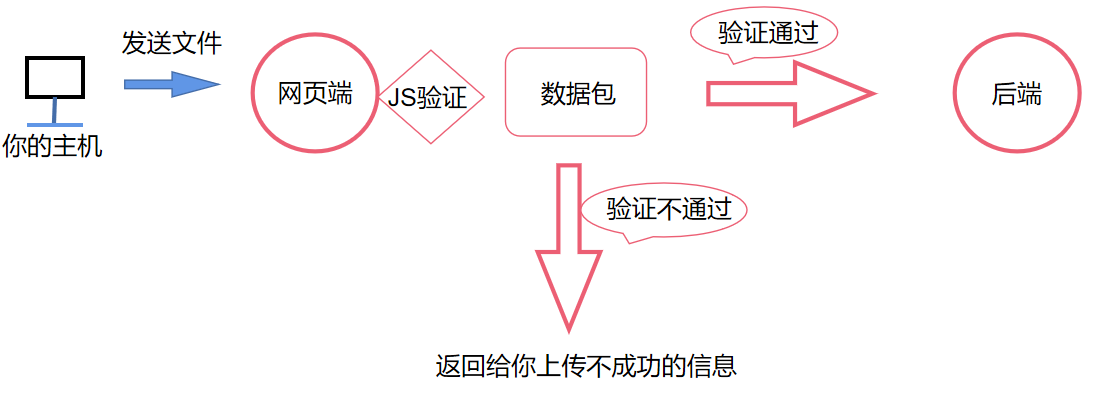
*## POST中的123为菜刀or蚁剑连接shell时需要的密码，可改为其他你喜欢的密码*

**五、知识点详解**

因为JS验证＝没有验证，所以我就在JS验证这块中详细说明了文件上传最最最最最最基本的操作，主要是给0基础看的，后续遇到的步骤将会简要概括。

**①JS验证绕过**

* 原理剖析：



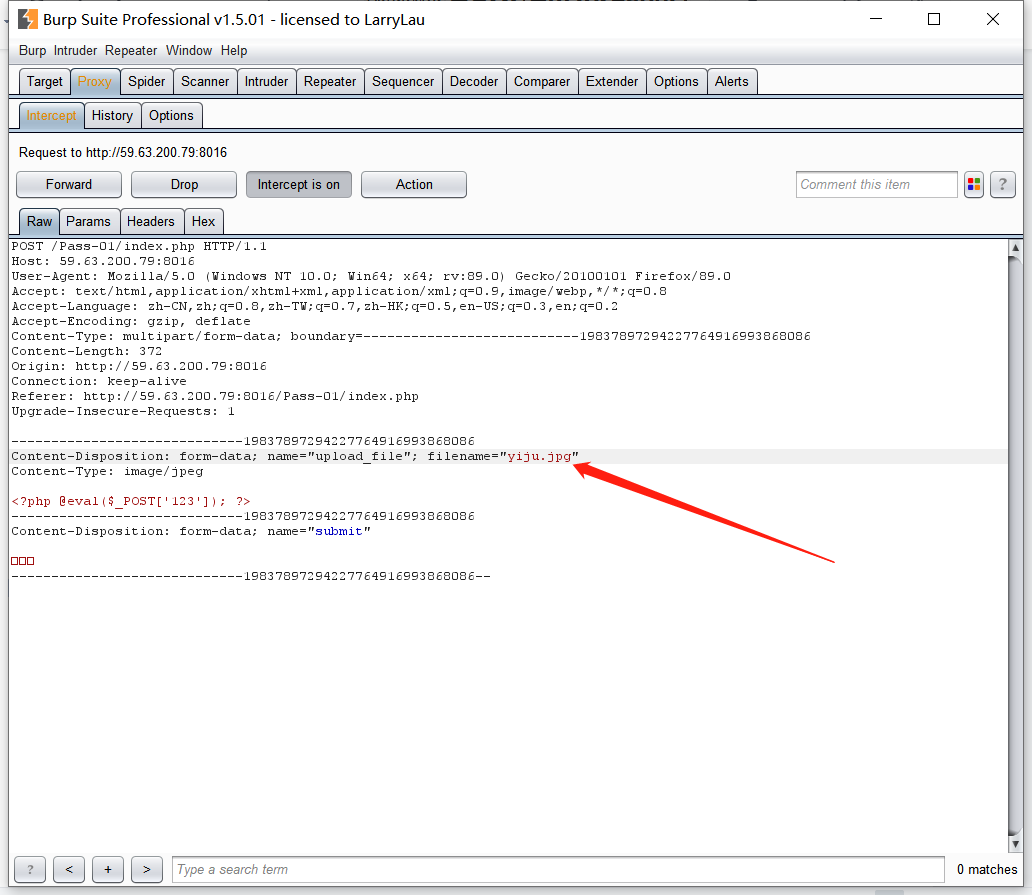
绕过方法：很明显，我们可以先上传JS允许的文件后缀，再通过截获数据包修改回php后缀发送至后端

* ↓下面↓实战操作演示：



*## 右侧为JS验证的源码*

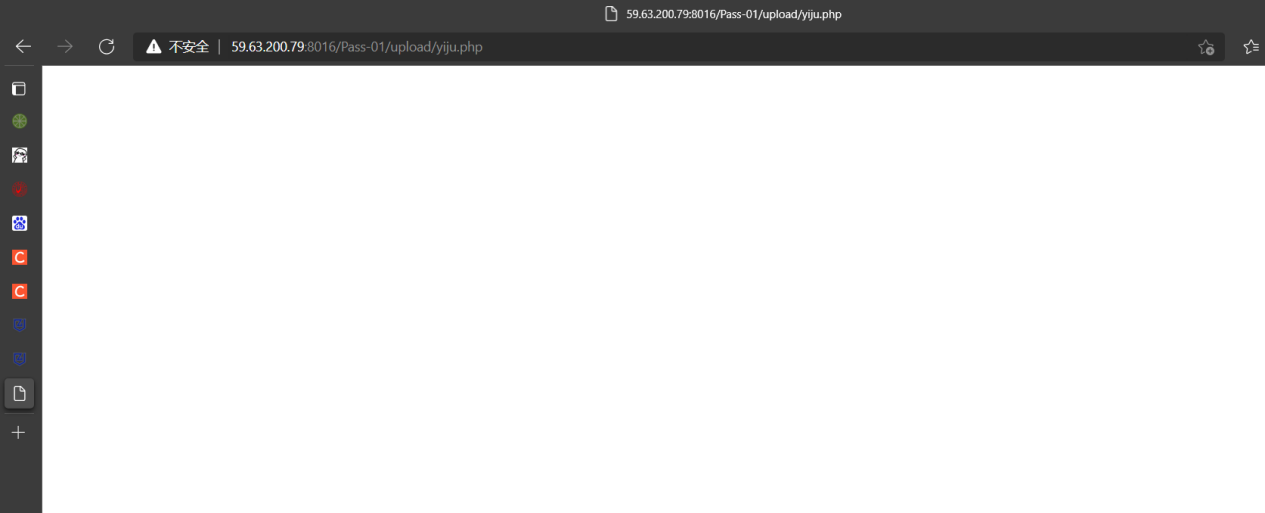
好，我们先在桌面创建txt文件，写入一句话木马，保存退出，修改后缀为jpg（阅读源码可知允许的后缀为jpg、png、gif），开启burp，上传刚刚保存的jpg，抓包得到↓



此时将jpg修改成php，再forward下去（也可以直接关闭抓包）



可以发现这题是没有返回地址的，我们右键这个图片转到图片链接（新标签页打开）

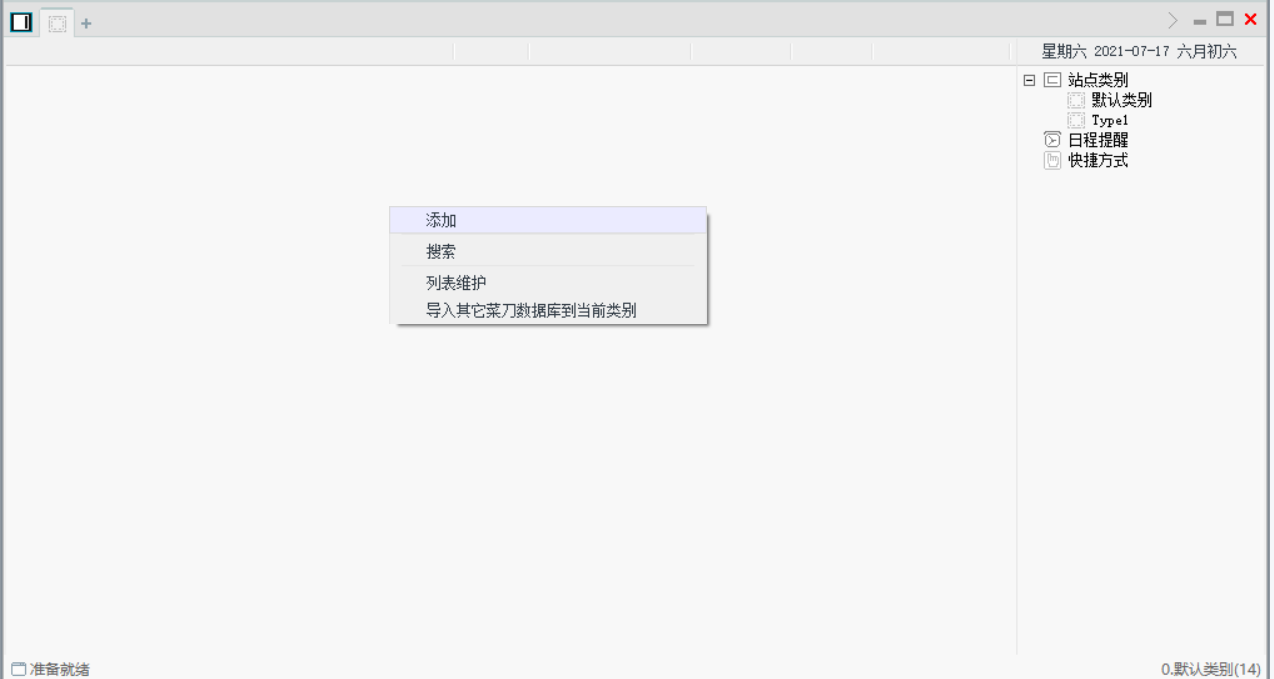
**

就可以看到图片所保存的位置了（最上面的URL栏）

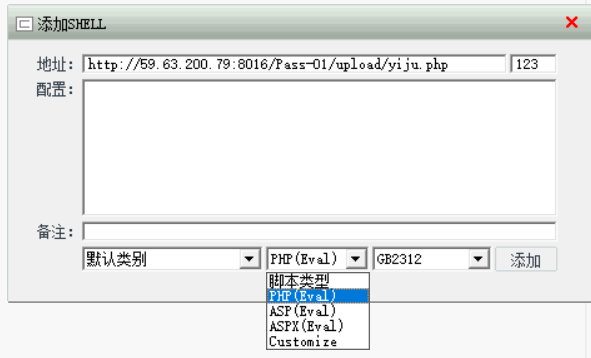
现在可以打开菜刀or蚁剑连接（我用的菜刀，接下来都直接说菜刀了）

* 菜刀教程：

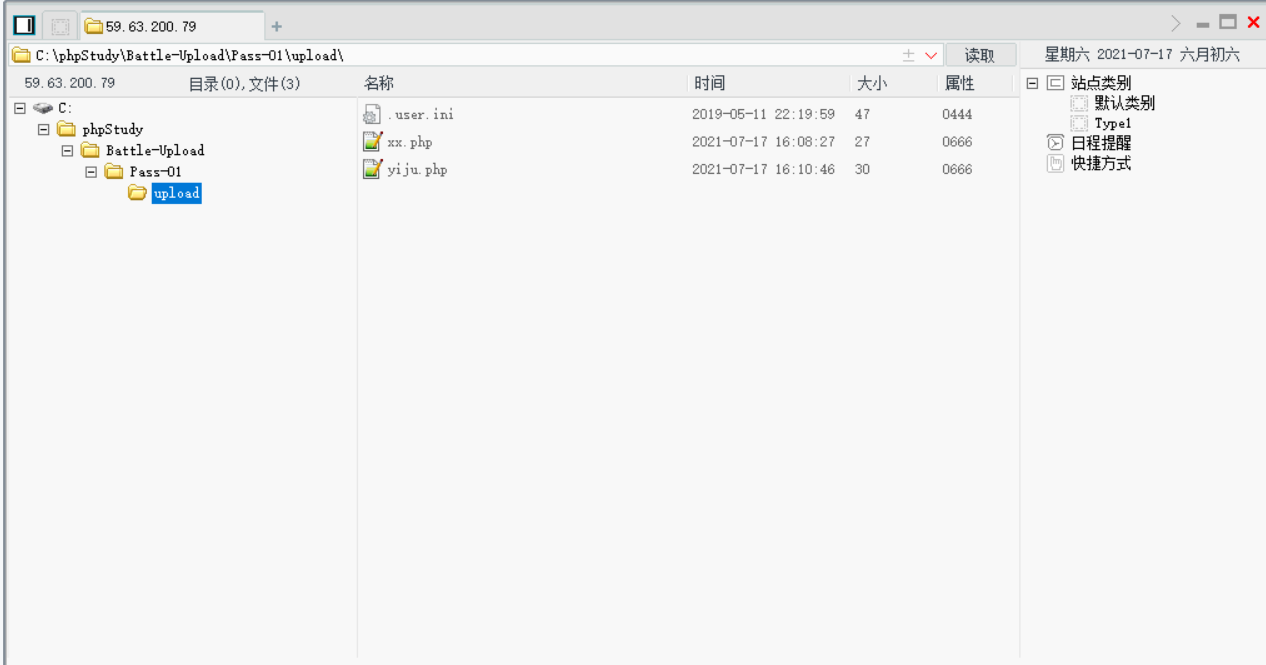
点开后右键点击添加



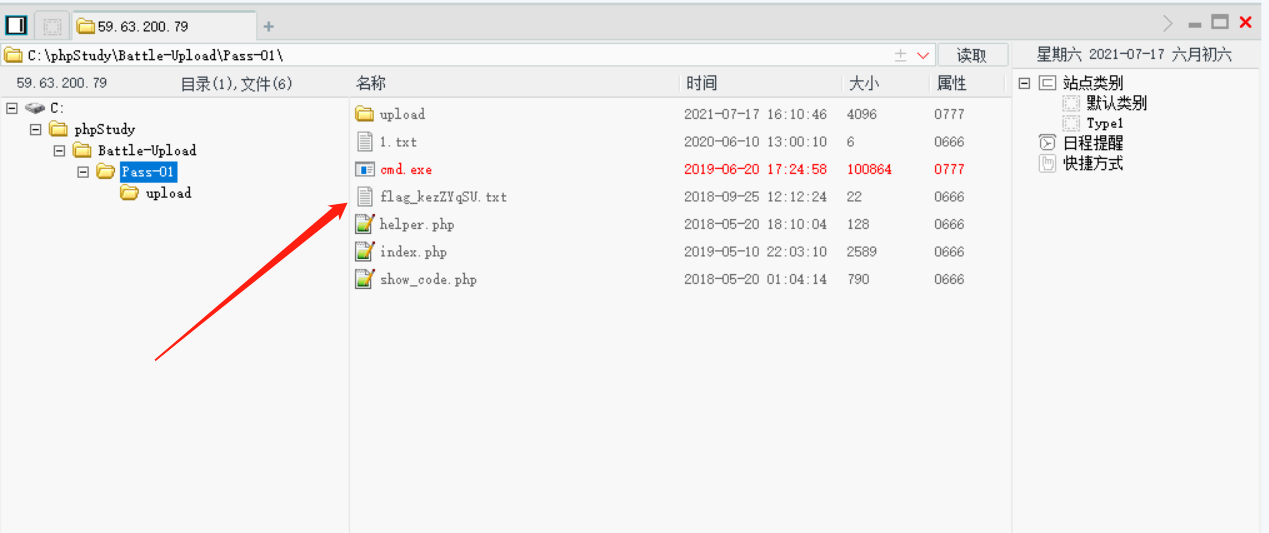
↓地址为你上传的文件的路径，密码为一句话里的密码↓



双击连进去后长这样



这就是网站的全部文件，然后我们随便找一下就能找到flag



*## 菜刀当然不止这些内容，这些只是基操，后续碰到新的姿势还会说的*

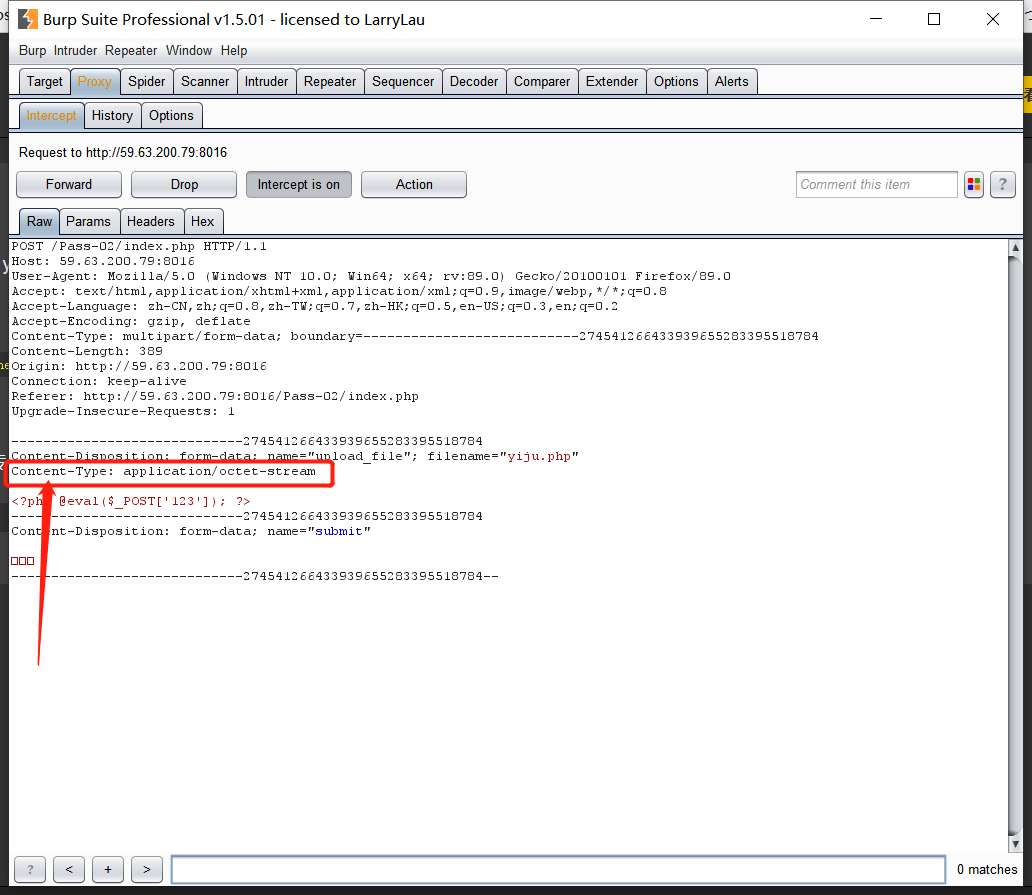
- - 题目练习 --> 攻防世界web高手进阶区 upload 1 ： adworld.xctf.org.cn/task/answer?type=web&number=3&grade=1&id=4822&page=1

②MIME类型绕过

既然要绕过MIME，首先要了解MIME是啥（具体自行百度或看下面文章）

可戳 --> [www.cnblogs.com/tjudzj/p/6528008.html](http://www.cnblogs.com/tjudzj/p/6528008.html)

看不懂也没事，就是每个文件后缀对应着一个MIME类型，在数据包中长这样↓



*## 也可以回看上个模块①中的数据包，jpg对应着image/jpeg*

一些题目中会对MIME类型进行检测，只需抓包将上图框框处的content-type后面改为image/jpeg或image/gif 等 ，然后就能上传成功，再菜刀连接找到flag

*## 其实，按照模块①的操作就可以完成MIME类型的绕过*

- - 题目练习 -- > CTFHub ：

[www.ctfhub.com/#/skilltree](http://www.ctfhub.com/#/skilltree)

③黑名单绕过

黑名单就是不允许名单中的后缀名传入

同理，白名单就是只允许名单中的后缀名传入（此处先介绍黑名单绕过）

当我们用了模块①②中的方法还是上传不成功时，我们就应该先考虑是不是有着黑名单限制了（当然也有白名单可能，此处先不考虑）

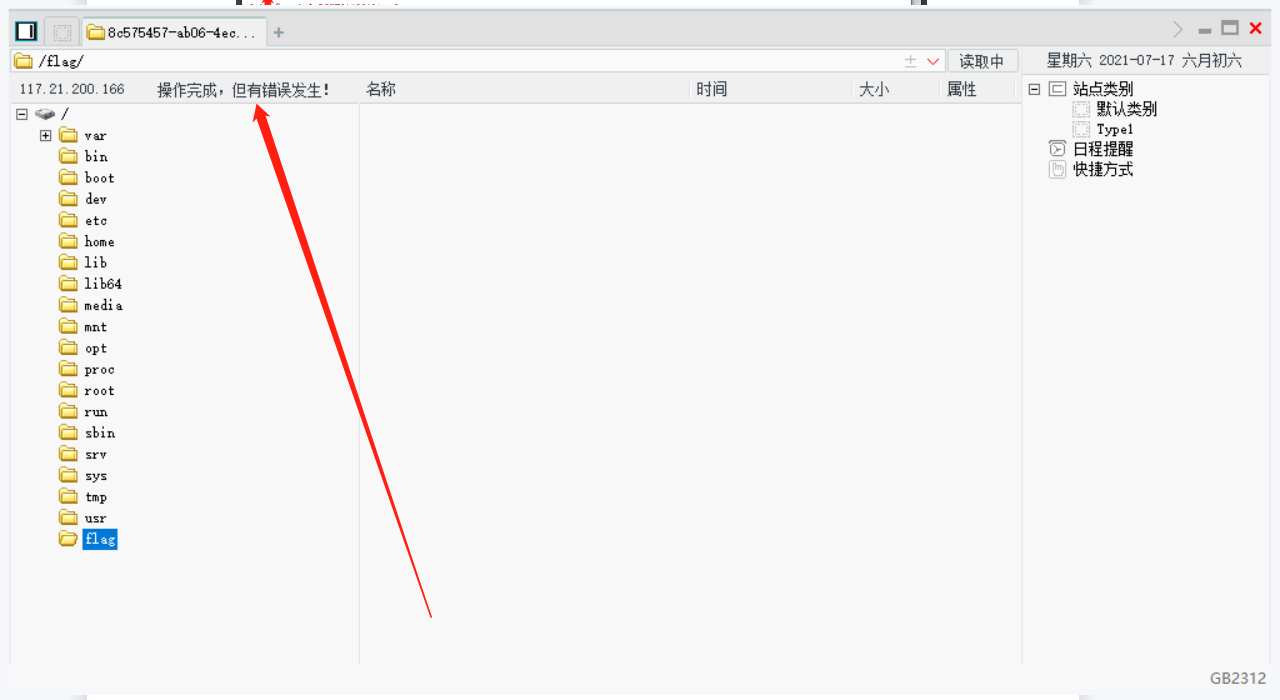
上传不成功显然就是黑名单里有你上传的后缀名，比如php，那我们就需要找到后缀不为php，但是文件仍会被当作php解析的后缀名。（就好比你的钢笔坏了，但是你可以找来一支铅笔继续写字）例如phtml、php2、php3、php4、php5等，常用的是phtml

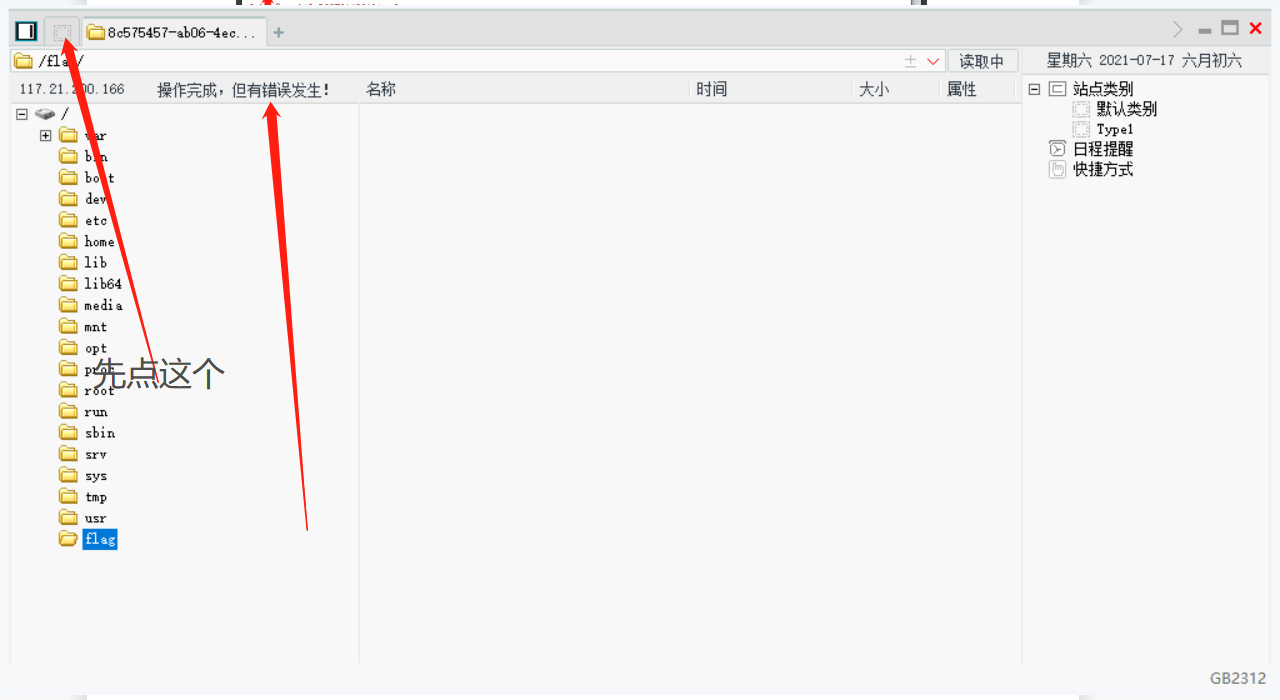
- - 题目练习 -- > BUU :

buuoj.cn/challenges#[ACTF2020%20新生赛]Upload

* 此处再介绍一点菜刀的基操（针对0基础）

当你遇到 ↓下图这种情况时↓

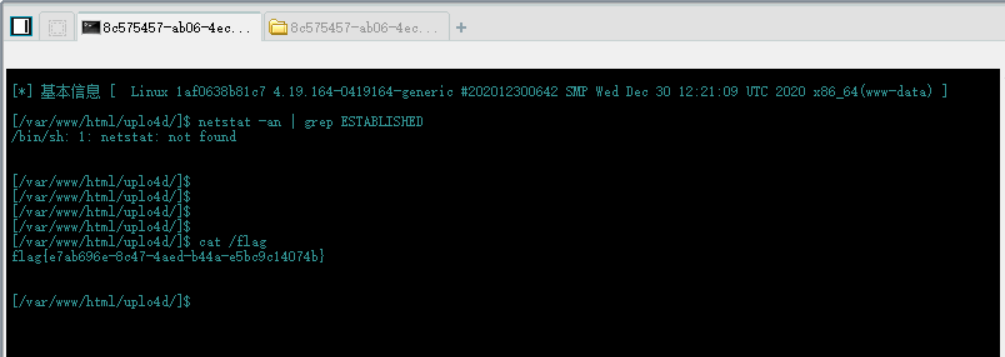
发现打不开flag文件夹，这时你可以



右键打开虚拟终端



打开后敲几下回车，然后输入cat /flag



*## 其实这就是服务器操作系统的终端，所有终端命令都是适用滴*

*## 这里有/var/www，一看就是linux系统；如果是模块一中的C盘，则为windows*

④文件头检查绕过

首先我们要知道每个特定类型的文件都有不一样的开头或者标志位，例如

JPG ：FF D8 FF E0 00 10 4A 46 49 46

GIF ：47 49 46 38 39 61

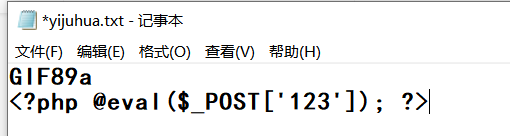
PNG：89 50 4E 47

*## 将文件放在010editor下或者winhex下都可以进行查看*

* Php中的exif\_imagetype()函数可以判断你所传入文件的文件头是否规范

绕过方法很简单，手动构造一个文件头即可，GIF的文件头转换为文本正好是GIF89a

所以你的一句话木马内容变成这样↓

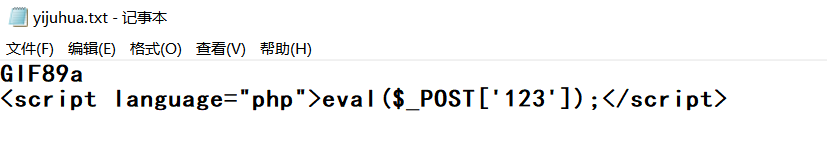


* 拓展：有些题目会检查是否存在 < ? 字样，所以上面这个木马还不够格，这时我们需要用到php5的语法：↓

GIF89a

<script language="php">eval($\_POST['123']);</script>

这时你的一句话木马应该变成这样↓



- - 题目练习 -- > BUU ：

buuoj.cn/challenges#[极客大挑战%202019]Upload

***!!! ---- > 针对这题，如果你对文件上传只是刚了解，十分建议你从第一个模块的思路开始一步一步往下操作，不要直接跳到用最后的一句话木马。初期务必要循序渐进着来！ < ---- !!!***

⑤.htaccess文件

好，现在我是开发，我把php、phtml、php5等所有能解析php的后缀全部过滤掉

显然，以我们前四个模块所学是没办法绕过的，这里我们要了解一下.htaccess文件

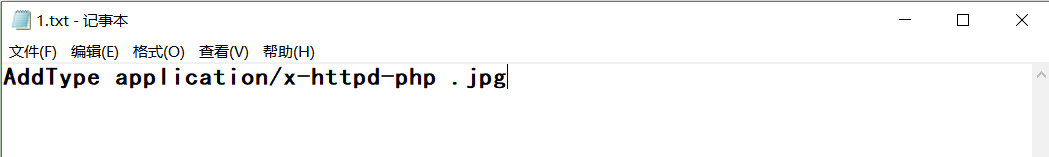
.htaccess文件：分布式配置文件，只在apache中有

.htaccess文件功能：在当前文件夹内定义一个新规则（文件夹密码保护、用户自定义重定向、自定义404页面、扩展名伪静态化、禁止特定IP地址的用户、只允许特定IP地址的用户、禁止目录列表等）

*## 为什么.php会被当成php解析，这是网站自带的规则；而.htaccess文件可以在他所在的文件夹中自定义规则，比如你可以添加进“将jpg后缀的文件当作php解析”，很显然，这个功能很强大。不过很可惜，.htaccess在apache中是默认不开启的，不过当你无路可走时，这也不失为一种选择。*

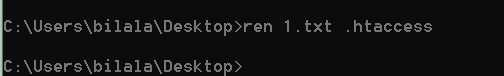
* ↓实际演示↓

桌面创建1.txt，内容写入AddType application/x-httpd-php .jpg //*将jpg后缀的文件当作php解析*



保存退出后，你想修改文件名为 .htaccess，但发现这是行不通的，因为windows必须要有文件名，不能只有后缀

我们可以打开cmd，进入文件所在目录，输入下方命令↓



→ ren 原文件名 修改后文件名

然后再将.htaccess文件上传至网站，再上传写了一句话木马的yijuhua.jpg

- - 题目练习 --> BUU :

buuoj.cn/challenges#[MRCTF2020]你传你🐎呢

buuoj.cn/challenges#[GXYCTF2019]BabyUpload

###############################################################################

# #

# ! @ # $ % ^ & \* 学 习 新 姿 势 \* & ^ % $ # @ ! #

# 当你用菜刀连上时，先看看同文件夹目录下有什么文件，如果有config.php等一些 #

# 文件，或许你要点开看看，flag没准不在整个目录里，他可能在数据库，菜刀也是可 # # 以连接数据库的 #

# #

# 又或者你找到了flag文件夹，但系统禁用了cat等函数，这里就需要读者自行去查 #

# 阅相关的代替函数或相关的代替方法 #

# #

###############################################################################

- - 题目练习 -- > i春秋 ：

[www.ichunqiu.com/battalion?t=1&r=61025](http://www.ichunqiu.com/battalion?t=1&r=61025)

⑥.user.ini文件

除了.htaccess可以配置新规则，还有.user.ini文件(apache/nginx/IIS等服务器都可以配置)

.user.ini文件的详细介绍：

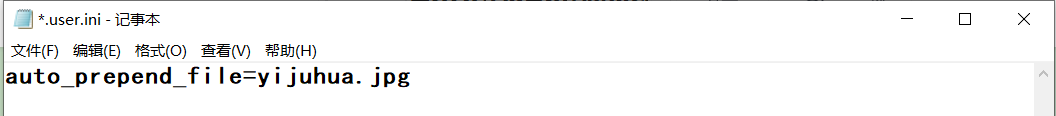
戳 -- > : wooyun.js.org/drops/user.ini文件构成的PHP后门.html

那我通俗点说就是php配置项中的两个函数可以被利用

auto\_append\_file和auto\_prepend\_file

Auto\_prepend\_file函数的作用是指定一个文件，包含在要执行的文件前

语法：auto\_prepend\_file=yijuhua.jpg



*## 举个栗子：网站要执行index.php文件，然后你通过这个函数指定了一个含有一句话木马的jpg，然后网站就会先运行这个木马，再执行php文件*

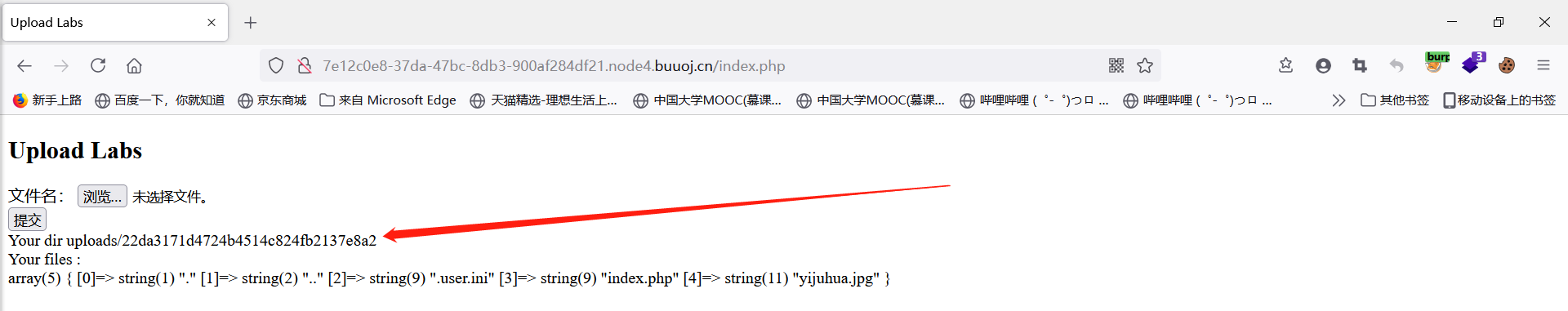
上传了.user.ini文件之后，你再去访问一个php文件，网站就会运行你上传的木马了

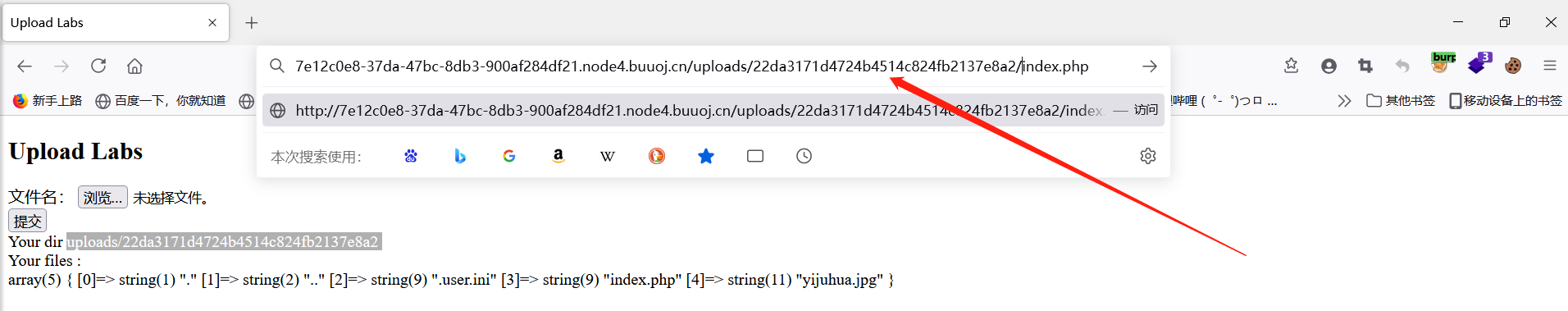
所以使用前提就是题目中有提示哪里哪里有php文件

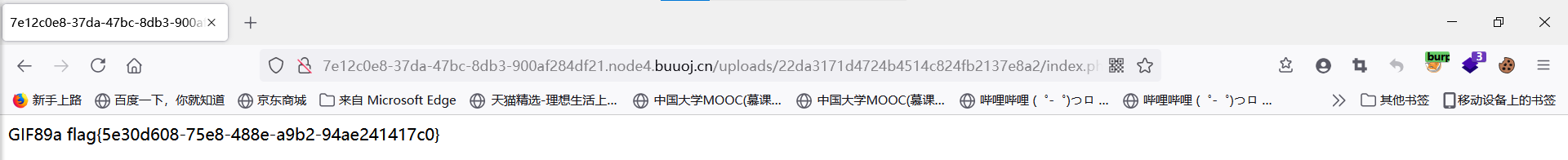
- - 题目练习 -- > BUU :

buuoj.cn/challenges#[SUCTF%202019]CheckIn

* ↑对这题的简单WP：先上传.user.ini，再上传个一句话木马，返回路径后，访问路径里的index.php文件，跳出flag







⑦windows服务器下的绕过

上面所说的都是在linux服务器下的绕过，那因为Windows的一些特性，windows有对应的独特的绕过方法（不过笔者刷题少，还没做到过windows为服务器的题目，这里大家了解即可）在以下的讲解中，大家可以去桌面实际操作一下↓

1. 大小写绕过

在windows中，系统不会区分大小写，比如你在桌面创建一个1.pHp，创建完成后还是会变成1.php，同样的，Php、PHp这些后缀都是一样的，那我们在上传时就可以利用这个特性，比如将后缀改为1.PHp从而绕过黑名单检测

操作：只需抓包后改一下后缀名即可

1. 后缀（空）绕过

与上面类似，当你创建时在php后缀后面加个空格，也是等同于php，也是抓包修改即可

1. 后缀（点）绕过

同理，后缀后面加个点

1. ::$DATA(windows文件流绕过)

这个特性的话，是因为windows的文件系统为NTFS。

当你创建1.php后，系统会在后边添加上::$DATA,不过这个是你看不见的，而且这个你也不可以手动去敲(系统会自动过滤这些敏感符号)

操作：抓包改后缀，后缀加上::$DATA来绕过黑名单验证

- - 题目练习 -- > 无 ：

Zhezhishi.weile.meiguan.com

⑧00截断

前面说了这么多，都是黑名单验证，接下来来说说白名单的绕过方法

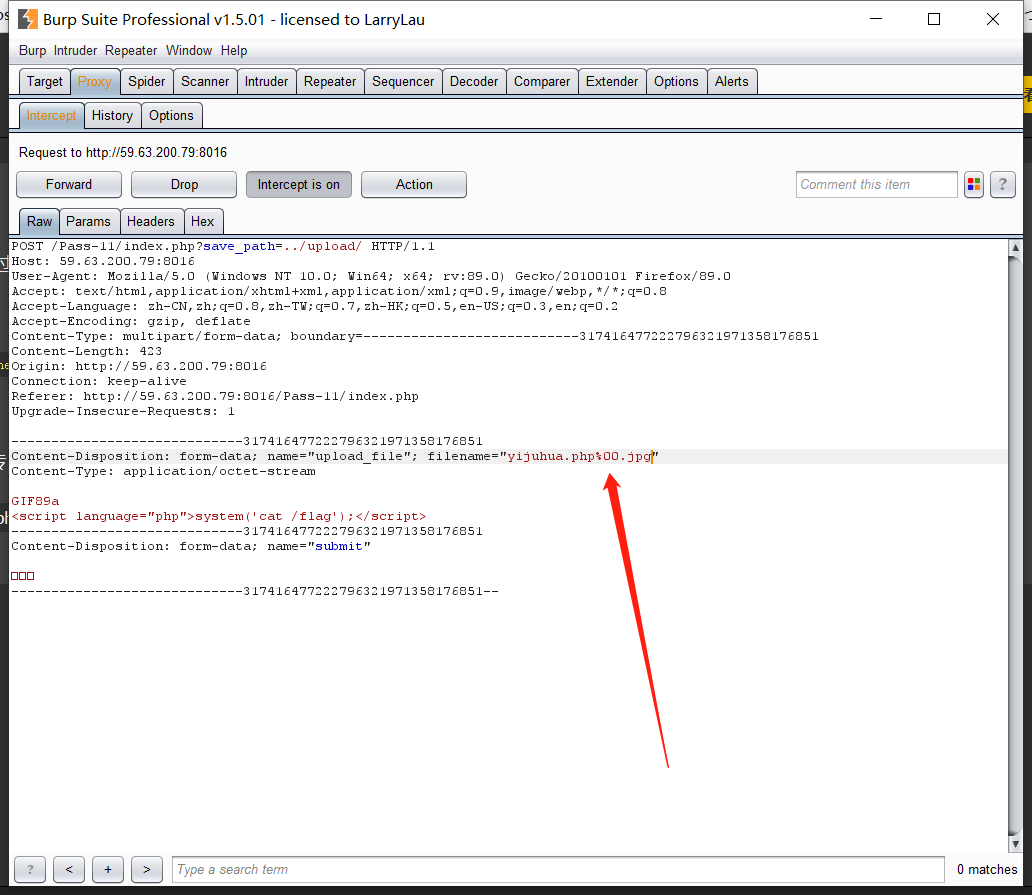
*## 首先先解决大家的一个疑问，白名单安全性比黑名单高，那为什么开发人员们还要用黑名单呢。那其实现实的环境中呢，开发者已经不局限于用名单来检测了，所以我们对于黑白名单的验证绕过都要有所了解。*

*## 00字符就好比终止符，结束语*

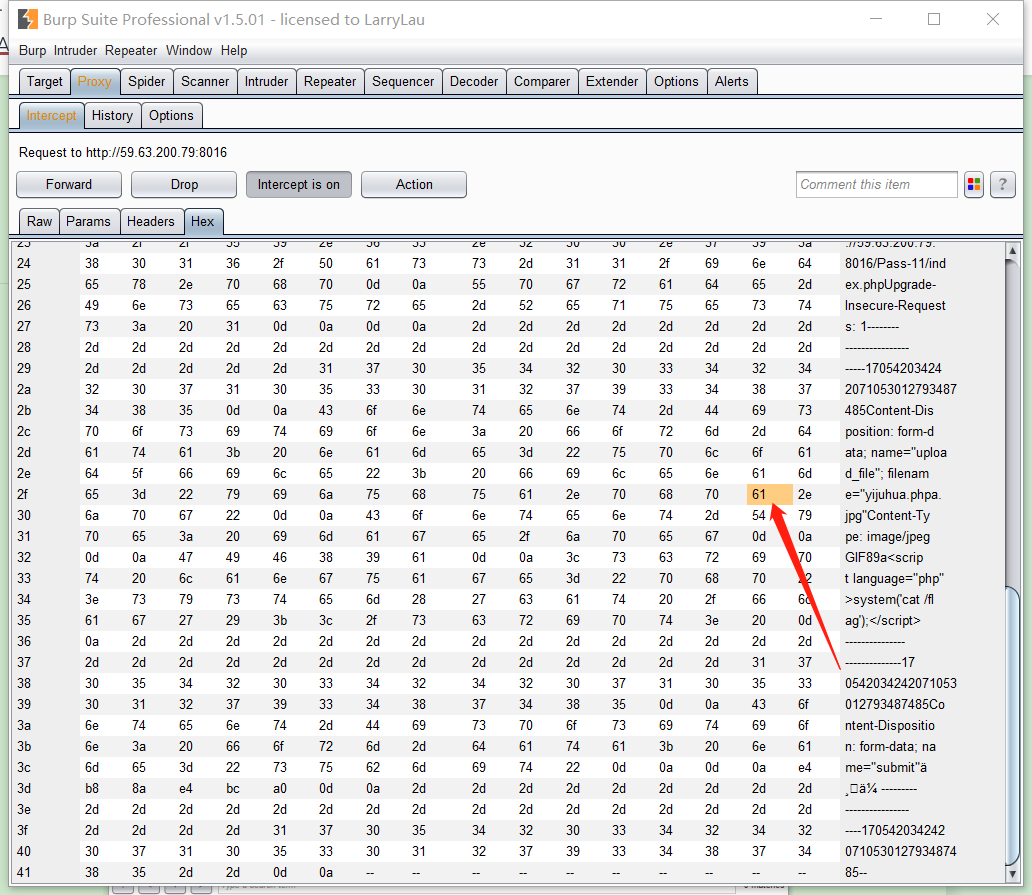
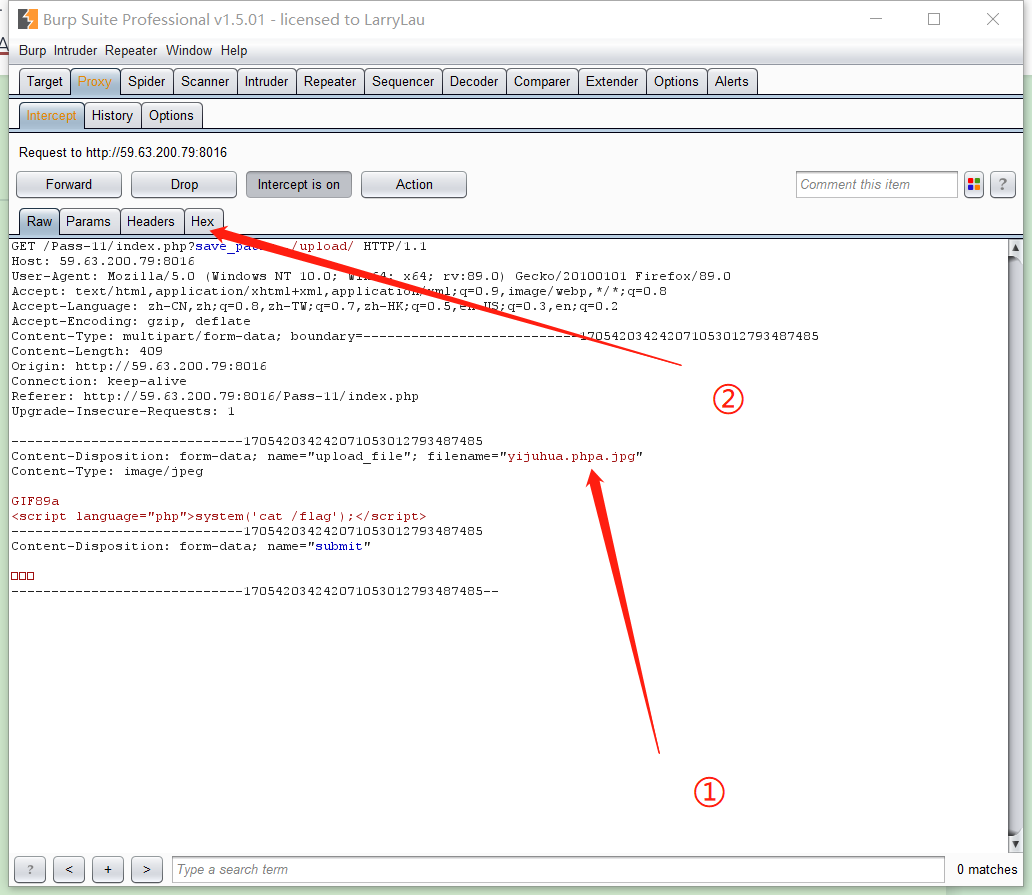
Php的底层语言为C语言，所以当服务器读取到0x00字符时也会自动丢弃后面的数据，比如你上传的是1.php%00.jpg，那服务器读取到%00时就不再读取了，也就变成了1.php。

%00是0x00字符的url编码，系统在读取这个后缀时，还是会将文件的后缀判定为jpg，正因如此，我们可以靠这个方法来绕过白名单的验证。

姿势①:



姿势②：



找到这个a，将其改为00（a为61）

*## 姿势②也可以直接先写php%00.jpg，然后将%00选中按ctrl+shift+u直接改成00*

显示为这样



*## GET请求头可以接受URL编码，POST不行，所以两种姿势都要会*

- - 题目练习 ： CTFHub ：

[www.ctfhub.com/#/skilltree](http://www.ctfhub.com/#/skilltree)

那介绍了这么多模块，读者会发现，其实最后一种00截断直接通杀，只要配上文件头，再改个一句话木马格式，最后加上00截断，不论黑白名单直接过掉。

⑨图片马绕过

既然你总是想尽办法的改后缀绕过我，那我选择直接对你上传的图片进行判断，判断你是不是真的图片，不是我就不允许上传。这里就涉及到了图片马

*## 上面的④模块文件头绕过也算在这里边*

Php中的getimagesize和exif\_imagetype()等函数都可以对上传的文件进行判断，只需上传图片马就可以绕过这些判断（如④中的添加文件头）

###############################################################################

# #

# 学习新姿势 #

# #

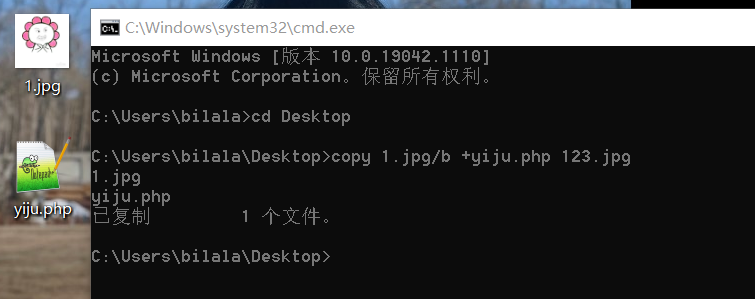
###############################################################################

* 这里介绍一种制作图片马的方式↓

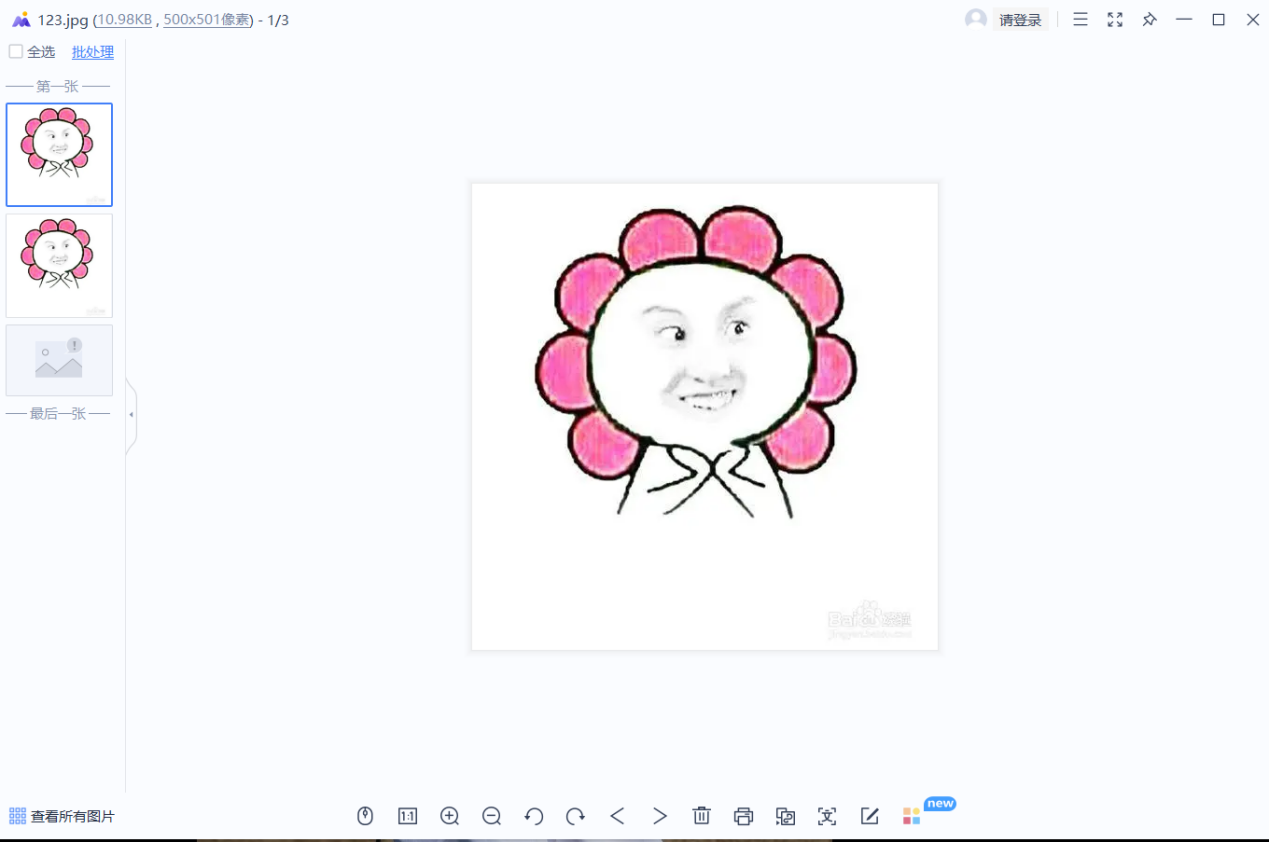
先去网上随便找张图片(越小越好)（一定要看保存时的后缀，不是你命名了jpg他就是jpg），然后保存到本地

敲下面命令，让两个文件合并生成123.jpg

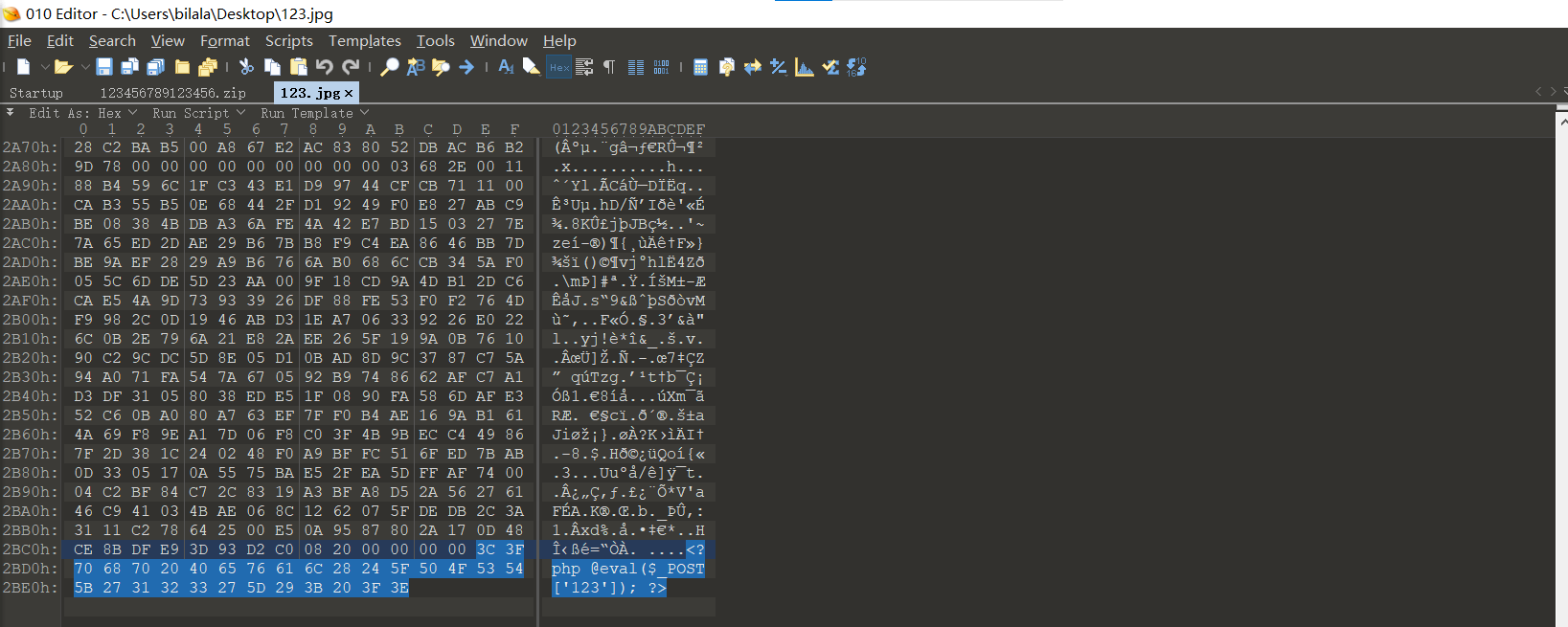
命令：copy 1.jpg/b +yiju.php 123.jpg



我们可以发现生成的123.jpg与原图一样，没有任何差别



但当将他放进010editor下时，拉到最下面会发现，多了我们合并进去的一句话木马



然后再将这个图片马上传至网站

*## 通常图片马都不是单独来使用，毕竟你上传的还只是一个jpg文件，我们要让网站在读取这个jpg文件时执行这个一句话木马就要搭配上别的漏洞，例如解析漏洞、文件包含漏洞等*

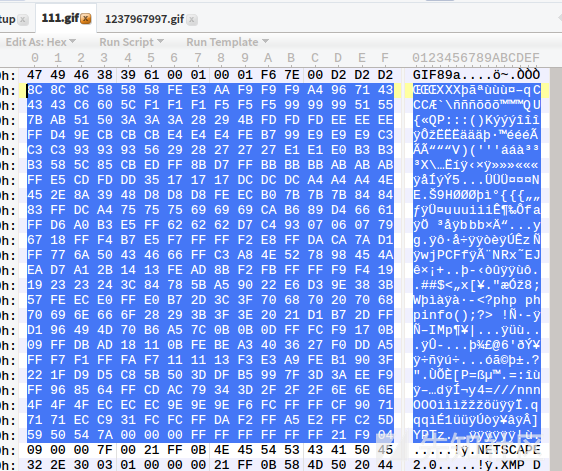
⑩二次渲染

那么针对上述情况呢，开发人员的应对情况是，将你上传的图片进行处理（渲染），处理后的图片也是外表看起来一样，不过其中的数据都已经全部重整了，也就是你加入的一句话木马会被渲染掉（就是放在010下看不到一句话木马了，自然你也就无法利用木马了）。

对于这种情况，我们可以分三类后缀去绕过（会一种就行）

1. GIF

GIF的图片在渲染后，还是会有一大部分数据块没变，大概在前二十行↓（网图）



标出的都是没有变的，然后我们在这其中（最好在三、四行的位置）写入我们的一句话，再上传即可

1. PNG

PNG的二次渲染很麻烦，我也不太会（读者想了解的可以百度）

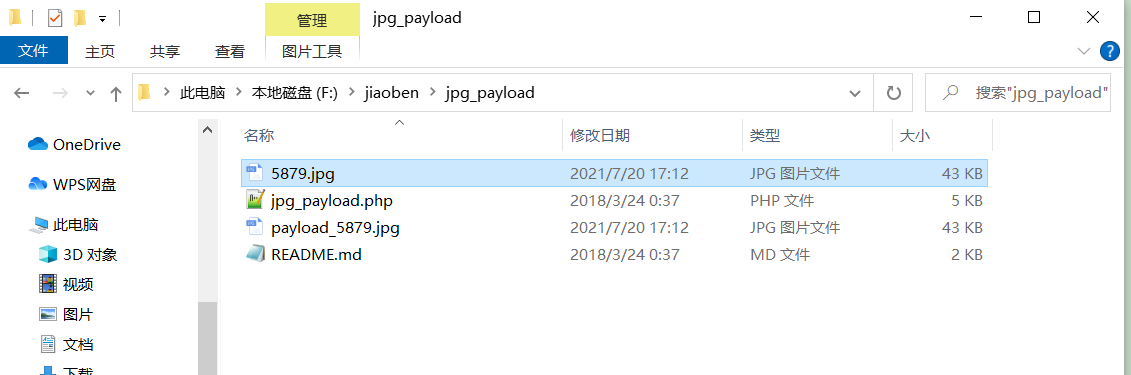
1. JPG

这里介绍一个脚本：github.com/BlackFan/jpg\_payload

先随便找一张jpg图片，再将图片上传到服务器，再把渲染后的图片下载到本地，然后运行脚本：php jpg\_payload.php 5879.jpg



目录中就会多出对应的pay\_load jpg



用010打开会发现其中写入了php代码，再将payload图片上传

上传后再下载回来可以发现图片还是存在php代码的

*## 不是每个jpg图片都可以被脚本处理的，读者可以多试几张jpg图片（我还没找到，所以没有截图给大家看，大家自己找找）*

⑩①条件竞争

网站对文件上传有两种思路：上传→检测→删除（不合规定的话删，合就保留）

检测→上传（合规定的话传，不合就不给上传）

那很显然第一种思路有利用空间↓：

我们先上传一个违规文件比如这个文件的功能是在当前文件夹写入一个文件，然后趁网站还没删除这个文件之前，我们访问这个文件，这个文件就会执行，也就是在当前文件夹写入一个文件，写入的这个文件是不会被删除的，因为他不是被上传上去的。然后我们就可以用菜刀连接这个写入的文件，从而达到控制网站。这就是条件竞争。

操作思路：众所周知，计算机的计算是很快的，他的检测到删除的时间极其的短，所以我们只能靠burp中的intruder模块来和他拼速度。让burp一边疯狂的上传违规文件一边疯狂的访问违规文件，总会有一次访问到，访问到后你的违规文件就会执行，网站的对应目录下就会被写入你想写入的文件，目的就能达成。

*## 一般如果跑五分钟还访问不到，大概率就是网站对条件竞争进行了防御，或者他用的是第二种上传思路。*

*## 那因为这种疯狂的上传疯狂的访问会对服务器造成资源消耗，如果操作的狠一点，可以当作DDOS攻击了，所以题目还是很少见的，大家懂原理即可，以后碰到再深入了解*

⑩②apache解析漏洞

Apache对于文件的解析规则是从右往左执行，如果后缀名为不可执行文件，就再往左判断。例如你传入1.php.qwe ，qwe为不可执行文件后缀，所以往左判断为php，所以这个文件会被当作php文件执行。

操作：抓包改后缀

⑩③nginx解析漏洞

（1）在Fast-CGI关闭的情况下，Nginx <=0.8.37 存在解析漏洞，在一个文件路径(/xx.jpg)后面加上%00.php会将 /xx.jpg%00.php 解析为 php 文件。

（2）在Fast-CGI运行模式下, Nginx <=0.8.37，在一个文件路径(/xx.jpg)后面加上/xx.php会将/xx.jpg/xx.php 解析为 php 文件。

也就是你上传了一个图片马1.jpg，然后你去访问图片所在的路径，在后面加上/1.php或者%00.php，这个图片马就会被当作php执行，就可以菜刀连接

⑩④IIS6.0解析漏洞

一句话木马当然不止php有，asp、jsp等都有，IIS服务器一般用的是asp语言

1. 目录解析

服务器默认会把.asp目录下的文件都当作asp执行，也就是你上传个1.asp/1.jpg，这个图片马就可以当作asp执行

1. 文件解析

服务器默认会把/1.asp;.jpg、/1.asp:.jpg这类文件解析成asp文件，服务器默认不解析；号后的内容，所以文件变为1.asp

1. 默认可执行文件

Asp的其他后缀：asa、cer、cdx（类似php的phtml等）

* 有关IIS7.0/7.5的漏洞

同nginx一样，在Fast-CGI运行模式下，在路径后加上/xx.php会将文件当作php执行

⑩⑤压缩包解压漏洞

有时，为了方便用户一次性上传多个文件，网站会对zip后缀开放（或者其他压缩包后缀）

网站收到压缩包后，会将里面的文件解压到相应位置，我们可以利用网站的解压来传入我们的木马，例如压缩包内的文件使用了目录穿越。

这里涉及文件上传更深的内容，不再细讲，写了一道例题的WP在WP.docx中，有兴趣的可以看看。

- - 题目链接 -- > BUU :

buuoj.cn/challenges#[第二章%20web进阶]文件上传

有关文件上传的常见套路总结到这里就结束了，实际中呢文件上传的骚姿势还是很多的（例如双层url编码绕过黑名单等），只不过需要大家在平时刷题中去积累，这里只是介绍了些基础概念和操作。另外，upload-labs靶场中的题目WP网上都可以查得到，并且对于我讲到的只有文字的一些模块，大家可以自行去靶场实践操作一下。