《安全程序设计与实践》
实验报告

实验名称：文件上传漏洞

实验类型：实操型

指导教师：孙海峰

专业班级：

姓　　名：

学　　号：

实验地点：东 11C227

实验日期：2025/6/4

一、 实验目的

1. 建立基于文件类型白名单过滤的文件上传功能网站。
2. 使用 Fiddler 进行 MIME 上传漏洞攻击和 0x00 截断上传漏洞攻击.
3. 通过多种方式实现文件上传漏洞的防护。
4. 可以解释文件上传漏洞产生的原理和危害，掌握多种防护文件上传漏洞的方法。

二、 实验过程

任务一：平台搭建

将 php5.3.3 和 apache2.2 安装好后，在 apache 安装路径的 bin 目录下打开 cmd，输入 httpd -k install 安装，出现以下“Service is already installed”说明安装完成。然后输入“httpd -k start”启动 apache2.2 服务。

```
C:\Users\Administrator\Desktop\Web\apache2.2\httpd-2.2.34-win32\Apache2\bin>httpd -k install
[Fri Jun 06 21:07:13 2025] [error] Apache2.2: Service is already installed.
```

访问 http://localhost/info.php，出现如下图所示“PHP Version 5.3.3”以及 GD、MYSQLi 相关信息模块则服务运行正常。

PHP Version 5.3.3

System	Windows NT WIN-DEM9HKQN6U1 6.2 build 9200 (Windows Server 2008 R2 Standard Edition) i586
Build Date	Jul 21 2010 20:25:38
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--disable-isapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=D:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet" "--with-mcrypt=static"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\Users\Administrator\Desktop\Web\php5.3.3\php-5.3.3-Win32-VC9-x86\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,TS,VC9
PHP Extension Build	API20090626,TS,VC9
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled

gd

GD Support	enabled
GD Version	bundled (2.0.34 compatible)
FreeType Support	enabled
FreeType Linkage	with freetype
FreeType Version	2.3.9
GIF Read Support	enabled
GIF Create Support	enabled
JPEG Support	enabled
libJPEG Version	6b
PNG Support	enabled
libPNG Version	1.2.37
WBMP Support	enabled
XBM Support	enabled

mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 5.0.7-dev - 091210 - $Revision: 300533 $
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

任务二：建立基于白名单过滤的上传网站

2-1 任务实现

将代码回滚到初始版本 final code

Git reset – hard a9fb79bc24ad97db9bb38d317a7b69697f7d161f

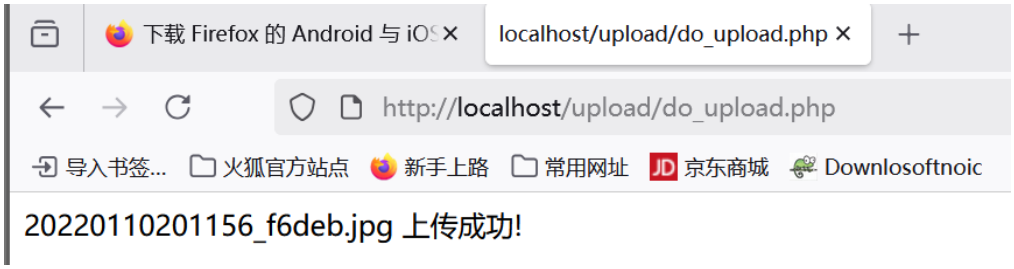
```
pyp@LAPTOP-6NGOAOTJ MINGW64 /e/作业/web/代码-git/代码-git/项目13/upload (master)
$ git log
commit a9fb79bc24ad97db9bb38d317a7b69697f7d161f (HEAD -> master)
Author: Dr.Sun <dr_hfsun@163.com>
Date:   Fri Jun 18 11:04:02 2021 +0800

    final code

commit b7610863b1863824a2f76f9d7cae30dd9a7c5cb3
Author: Dr.Sun <dr_hfsun@163.com>
Date:   Fri Jun 18 10:37:58 2021 +0800

    fix MIME

commit 249e89d7dc6a83a8de59f22650ba51b2144bb74b
Author: Dr.Sun <dr_hfsun@163.com>
Date:   Fri Jun 18 10:25:50 2021 +0800

    basic code
```

放置在 apache2.2 下的 htdocs 目录下，即网站建立成功。

2-2 功能测试

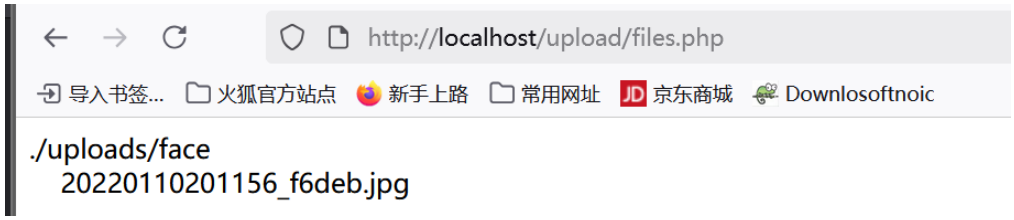
1. 建立文件上传表单页面

访问 http://localhost/upload/login.html，使用 admin/admin123 进行登录，登陆后上传一张图片。

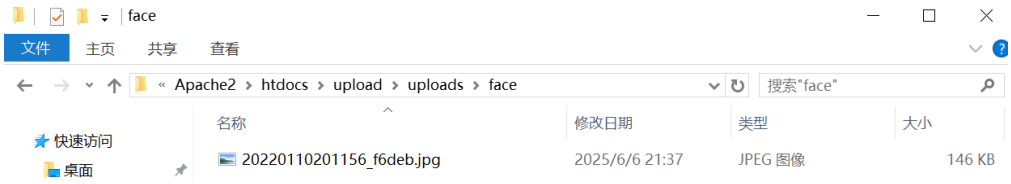


上传成功。

返回打开文件浏览，发现新上传的文件已经存在。



服务器目录 uploads 目录下出现 face/20220110201156_f6deb.jpg 文件。



本网站满足相关功能需求。

任务三：文件上传漏洞攻击测试

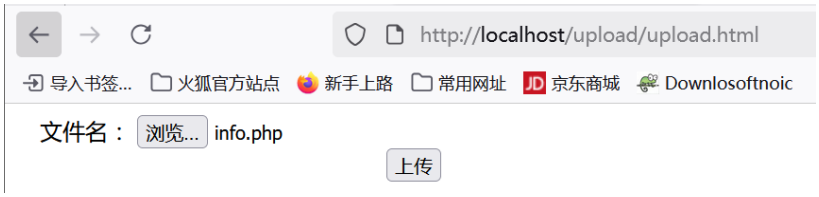
3-1 Fiddler 和浏览器的设置

由于电脑上安装了 BurpSuite，因此使用 BurpSuite 进行实验。

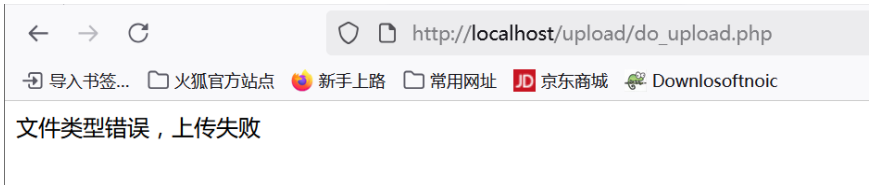
BP 配置与 Fiddler 相似，需要配置浏览器代理，这里就不详细讲解。

3-2 MIME 上传漏洞攻击

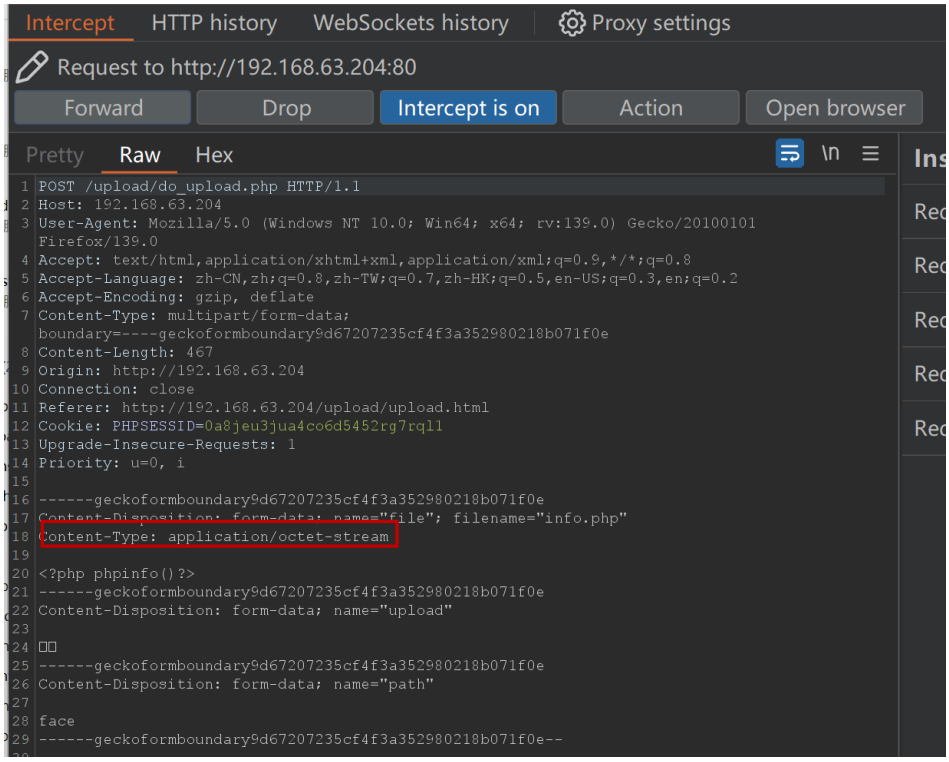
新建一个 info.php，内容为<?php phpinfo()?>，将 info.php 上传。



发现报错“文件类型错误，上传失败”。



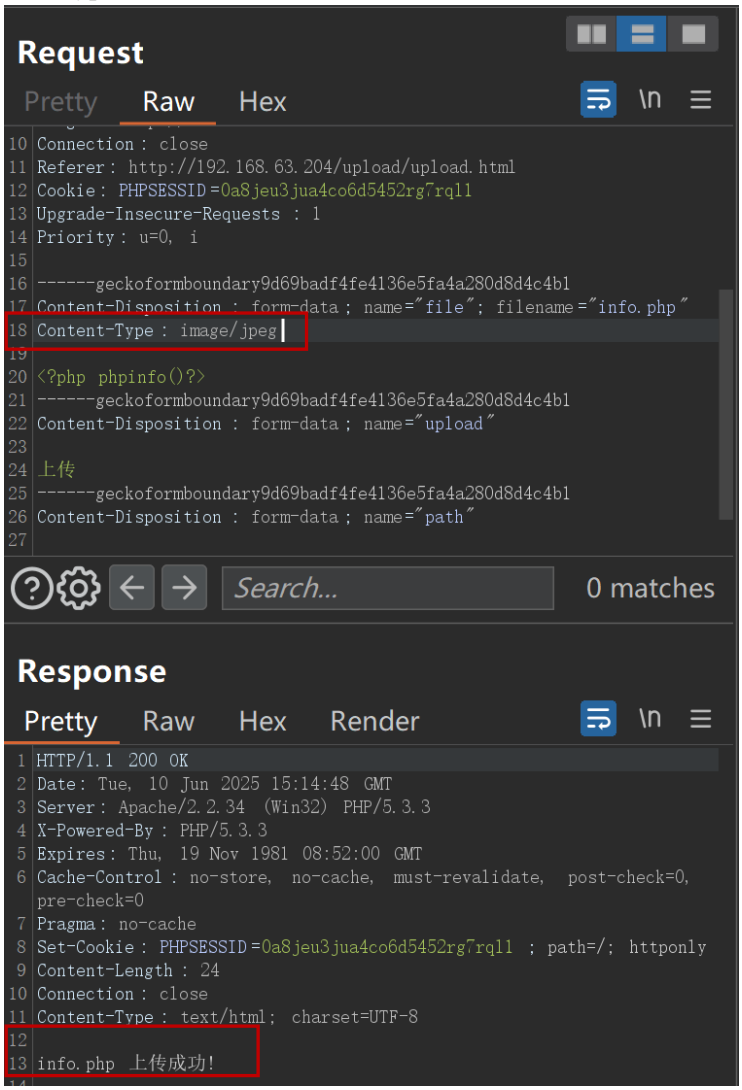
我们抓包查看，发现文件类型 Content-Type 是 application/octet-stream。



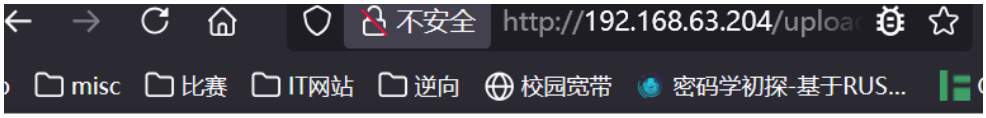
而网站代码上传文件（do_upload.php）的代码中存在一个 if 语句，如果匹配到 type 不是 image/gif 或 image/jpeg 或 image/jpg，则上传失败。

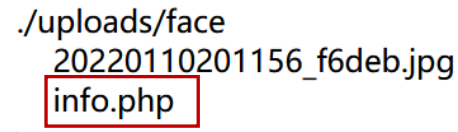
```
37  if($uploaded_type != "image/gif" && $uploaded_type != "image/jpeg" &&$uploaded_type != "image/jpg" )
38  {
39      exit('文件类型错误，上传失败');
40  }
```

而且 type 类型是从浏览器读取文件 MIME 信息获得，我们可以修改 Content-Type 从而控制 type 类型。我们改包发送请求。



上传成功。查看文件浏览，发现新上传的 info.php 已经存在于服务器。

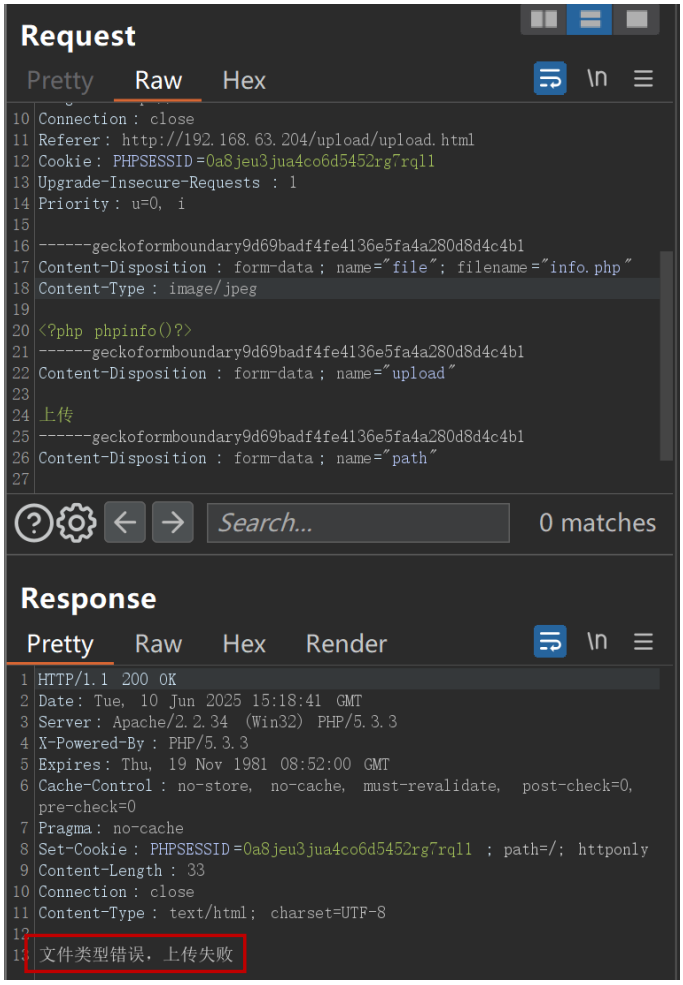




存在 MIME 上传漏洞。

3-3 0x00 截断路径上传漏洞

修改 do_upload.php 相关代码，不从浏览器读取文件 MIME 信息获得文件类型，而是使用从文件名中截取扩展名的方式判断上传文件类型。



发现上传失败，这是由于此时代码中存在

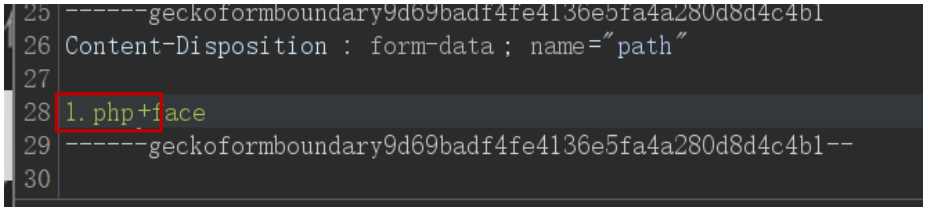
```
31  if(strtolower($uploaded_type) != "gif" &&
32      strtolower($uploaded_type) != "jpeg" &&
33      strtolower($uploaded_type) != "jpg" )
34  {
35      exit('文件类型错误，上传失败');
36  }
```

此时除了 gif、jpeg、jpg 后缀文件都不能上传成功。将 info.php 改后缀为 info.jpg，进行上传，发现上传成功，但是 php 代码不能被当作 php 类型执行。

由于使用 explode 函数获取文件扩展名，而该函数会忽略 0x00 截断后的内容，我们可以进行文件检验绕过，从而上传 php 文件。

```
21  $temp = explode(".", $uploaded_name);
```

进行 0x00 截断：首先将 info.jpg 上传，抓包，在下图所示位置中输入 1.php+



然后选择“+”号，找到对应 16 进制的位置，将“2b”修改为“00”。

```
000003d0   35 66 61 34 61 32 38 30   64 38 64 34 63 34 62 31
000003e0   0d 0a 43 6f 6e 74 65 6e   74 2d 44 69 73 70 6f 73
000003f0   69 74 69 6f 6e 3a 20 66   6f 72 6d 2d 64 61 74 61
00000400   3b 20 6e 61 6d 65 3d 22   70 61 74 68 22 0d 0a 0d
00000410   0a 31 2e 70 68 70 00 66   61 63 65 0d 0a 2d 2d 2d
00000420   2d 2d 2d 67 65 63 6b 6f   66 6f 72 6d 62 6f 75 6e
00000430   64 61 72 79 39 64 36 39   62 61 64 66 34 66 65 34
00000440   31 33 36 65 35 66 61 34   61 32 38 30 64 38 64 34
00000450   63 34 62 31 2d 2d 0d 0a   -- -- -- -- -- -- -- --
```

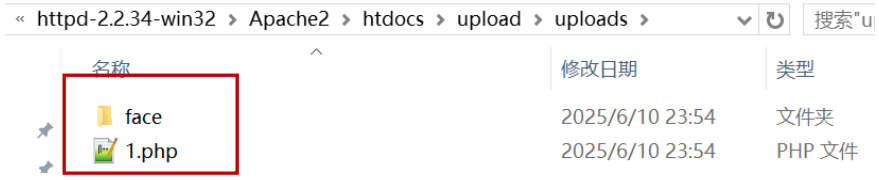
修改后此时文件保存路径“path”是“./uploads/1.php0x00face/info.jpg”，保存在 uploads 目录下，进行发包，发现上传成功。

```
Pretty  Raw  Hex                                                    \n
13 Upgrade-Insecure-Requests : 1
14 Priority : u=0, i
15
16 ------geckoformboundary16c5adcc5ec6afd67b7872ce8b7c31b3
17 Content-Disposition : form-data ; name="file"; filename="info.jpg"
18 Content-Type : image/jpeg
19
20 <?php phpinfo()?>
21 ------geckoformboundary16c5adcc5ec6afd67b7872ce8b7c31b3
22 Content-Disposition : form-data ; name="upload"
23
24 上传
25 ------geckoformboundary16c5adcc5ec6afd67b7872ce8b7c31b3
26 Content-Disposition : form-data ; name="path"
27
28 1.phpface
29 ------geckoformboundary16c5adcc5ec6afd67b7872ce8b7c31b3--
30

Search...                                                  0 matches

Response
Pretty  Raw  Hex  Render                                            \n
1 HTTP/1.1 200 OK
2 Date: Tue, 10 Jun 2025 15:30:03 GMT
3 Server: Apache/2.2.34 (Win32) PHP/5.3.3
4 X-Powered-By: PHP/5.3.3
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
  pre-check=0
7 Pragma: no-cache
8 Set-Cookie: PHPSESSID=0a8jeu3jua4co6d5452rg7rql1 ; path=/; httponly
9 Content-Length: 24
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12
13 info.jpg 上传成功!
```

文件浏览，发现 face 目录下只存在 info.jpg

```
  1.php
./uploads/face
  20220110201156_f6deb.jpg
  info.jpg
  info.php
```

Uploads 目录下出现 1.php，0x00 截断绕过成功。



任务四：文件上传漏洞防护

4-1 判断路径变量

测试一下 apache2.4+php 抓包效果将 upload 目录放置在 apache2.4 中的 htdocs 目录下，运行 apache2.4，其对应 php 版本为 7.1，为高版本。高版本中已经修复了 0x00 路径截断漏洞。

跟之前一样修改 path 中的值如下图所示，

```
Content-Disposition : form-data ; name="path"

1.phpface
------geckoformboundary7c4dc2de43f76763b650b73a6605101--
```

然后 send 发送，出现报错，上传失败，这是因为高版本 php 已经修复了 0x00 漏洞。

```
------geckoformboundary7c4dc2de43f76763b650b73a6605101
Content-Disposition : form-data ; name="path"

1.phpface
------geckoformboundary7c4dc2de43f76763b650b73a6605101
--

Search...  0 matches

Response
Pretty  Raw  Hex  Render  \n

Content-Type : text/html; charset=UTF-8

<br />
<b>
  Warning
</b>
:  move_uploaded_file()  expects parameter 2 to be a valid path, string given in <b>
  C:\Users\Administrator\Desktop\Web\apache2.4\httpd-2.4.33-o102o-x86-vc14-r2\Apache24\htdocs\upload\do_upload.php
</b>
 on line <b>
  43
</b>
<br />
内部错误，上传失败
```

4-2 路径白名单

在 do_upload.php 文件下的第 14 行添加如下代码

```
13
14      if($_POST['path'] != 'uploads' && $_POST['path'] != 'face')
15      {
16          exit('路径错误');
17      }
```

对路径 path 变量进行白名单过滤，可以防止 0x00 路径截断过滤。在 apache2.2 中进行测试。报错路径错误，实现防护。

```
26 Content-Disposition : form-data ; name="path"
27
28 1.phpface
29 ------geckoformboundary512802480814b13065e83314bf5d9a2--
30
```

Search... 0 matches

Response

Pretty Raw Hex Render \n

```
1  HTTP/1.1 200 OK
2  Date: Tue, 10 Jun 2025 16:56:53 GMT
3  Server: Apache/2.2.34 (Win32) PHP/5.3.3
4  X-Powered-By: PHP/5.3.3
5  Expires: Thu, 19 Nov 1981 08:52:00 GMT
6  Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
   pre-check=0
7  Pragma: no-cache
8  Set-Cookie: PHPSESSID=95vm33vj8ddjfsv5aftmngpa7s ; path=/; httponly
9  Content-Length: 12
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12
13 路径错误
```

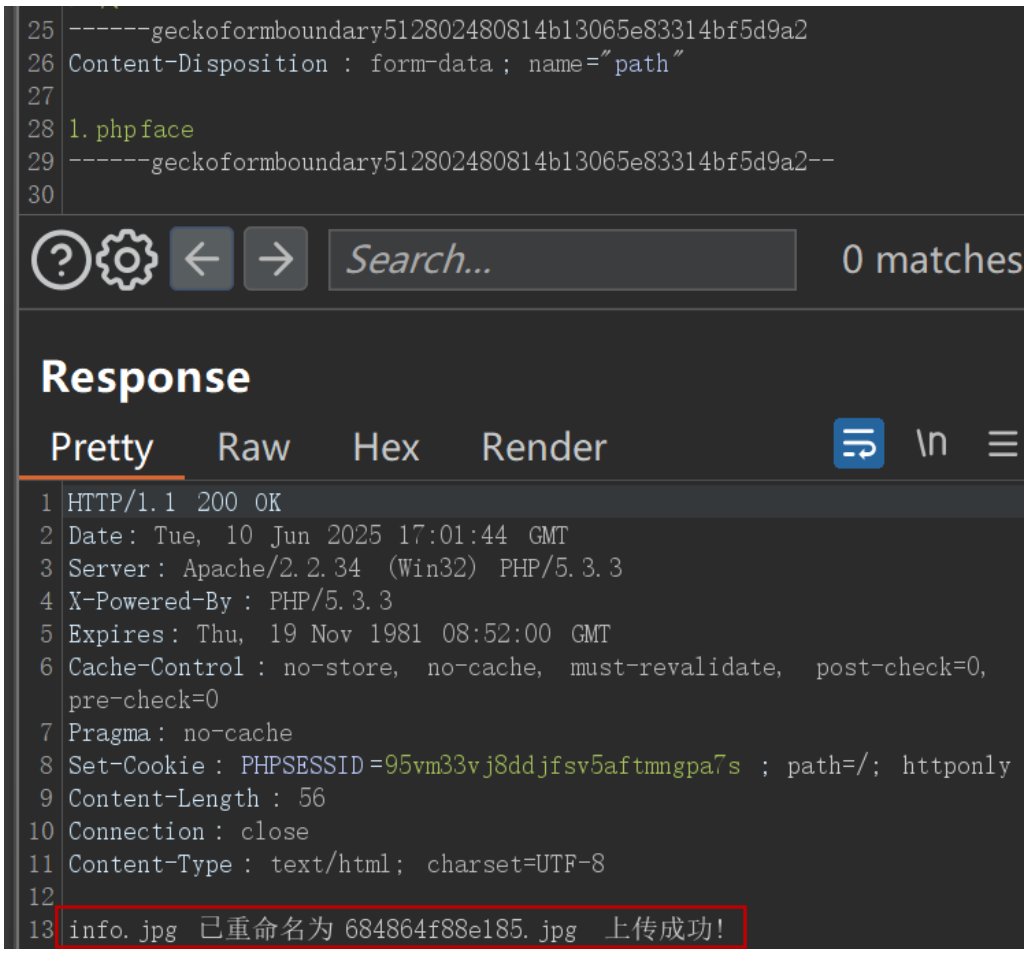
4-3 文件重命名

在 do_upload.php 中添加文件重命名代码，大概就是生成一个唯一文件名更新原本的文件名。

```
33      // 文件重命名防护 ======
34      $originalFilename = $uploaded_name;
35      $fileExtension = strtolower($uploaded_type);
36      $uniquePrefix = uniqid(); // 生成唯一前缀
37      $renamedFilename = $uniquePrefix . '.' . $fileExtension; // 组合唯一文件名
38      $target_path = dirname($target_path) . '/' . $renamedFilename; // 更新目标路径
39
40      if(!move_uploaded_file($_FILES['file']['tmp_name'], $target_path))
41      {
42          echo '内部错误，上传失败';
43      } else
```

不能解决 0x00 路径截断问题，但是攻击者不知道文件名就不能连接 webshell。能起到

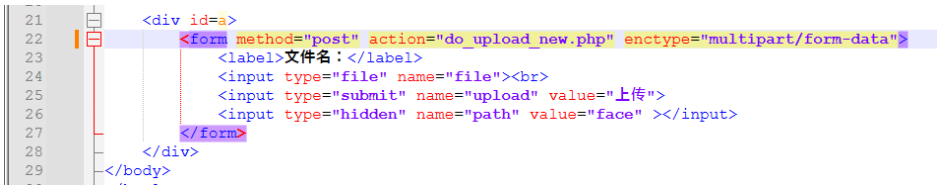
一定的防护效果。



这里我是输出了重命名的文件名，方便查看结果，实际上实际应用中不应该输出。

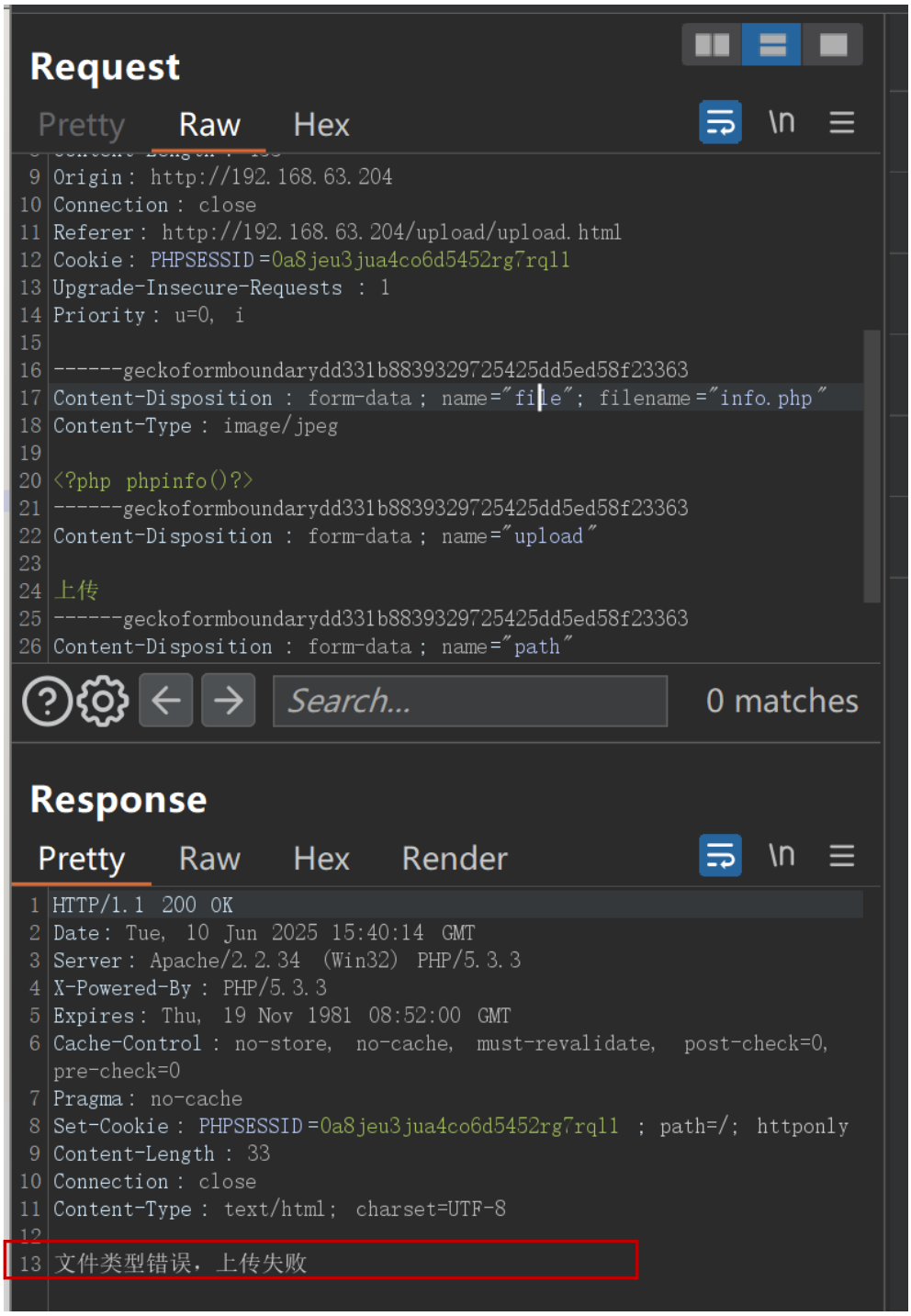
4-4 设置非 Web 目录保存文件

我们将上传按钮对应的 action 修改为 do_upload_new.php，里面的内容是具有一定文件上传漏洞防护功能的代码。



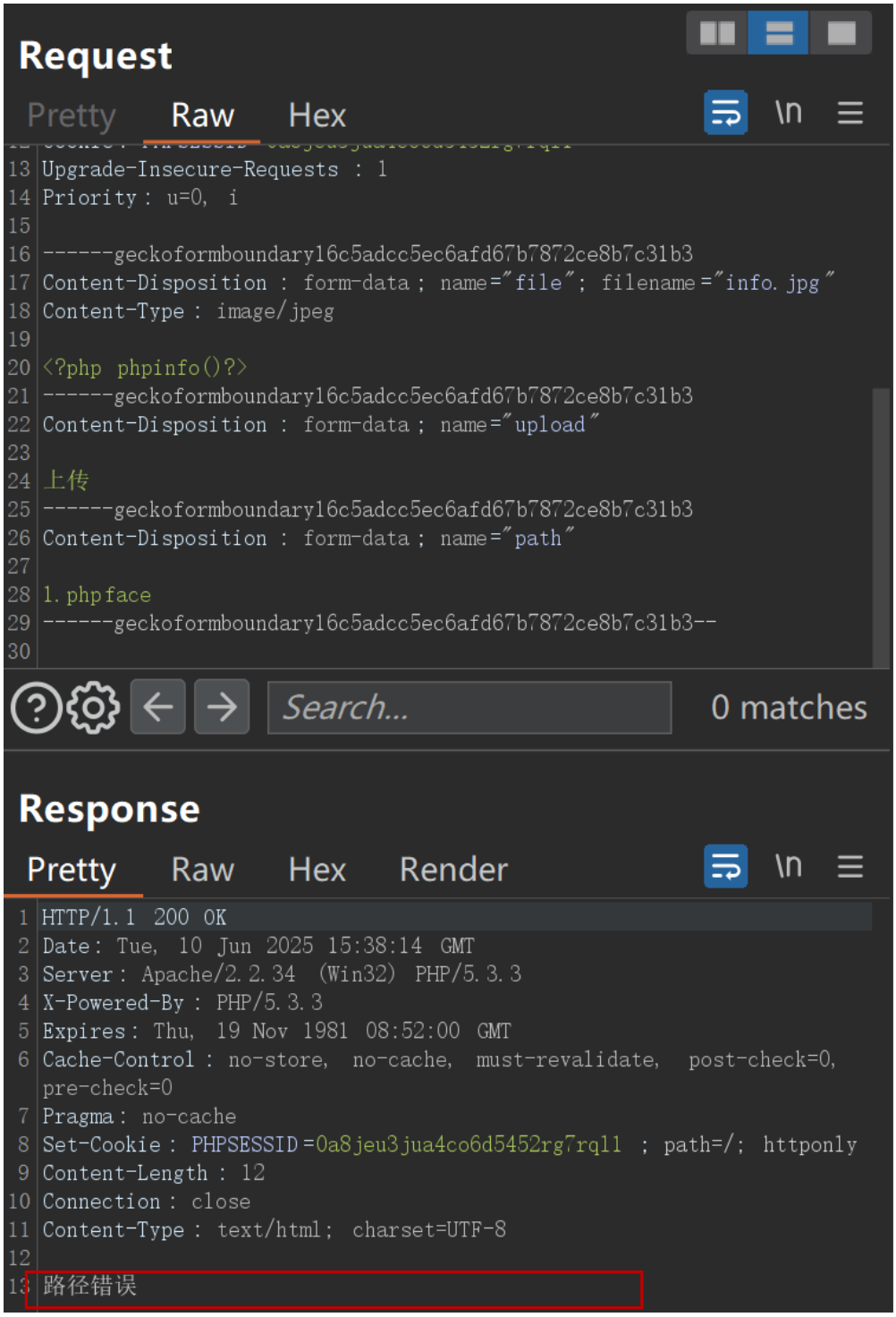
Files.php 也要修改为 file_traverse("c:/uploads"); 更改文件浏览目录位置。

配置完成后我们进行 MIME 绕过测试。

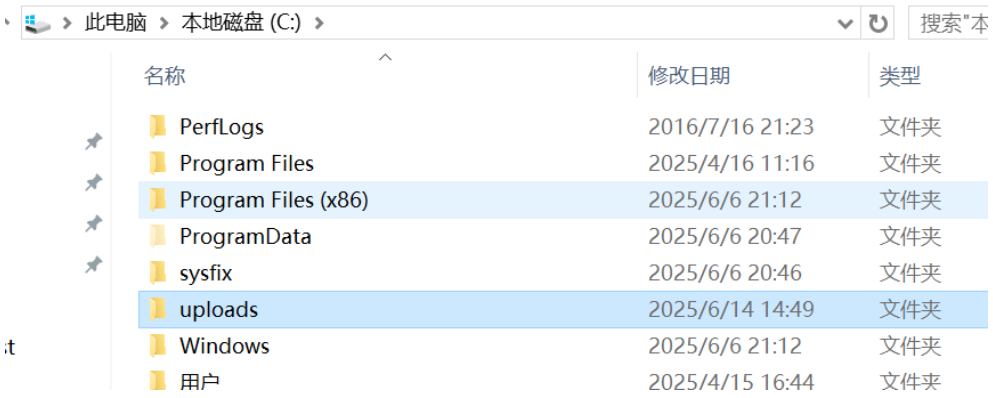


由于 do_upload_new.php 中添加了文件类型白名单检验，因此只能上传 gif/jpeg/jpg 后缀文件，没法利用。

继续进行 0x00 路径截断漏洞利用。



发现电脑 c:/目录下存在 uploads 目录，里面是上传内容。



三、 实验结论

通过实验掌握了文件上传漏洞原理与防护方法。搭建基于白名单过滤的上传网站，利用 BurpSuite 实现 MIME 与 0x00 截断攻击，验证漏洞危害。通过高版本 PHP、路径白名单、文件重命名及非 Web 目录存储等方式实现防护。可知文件上传漏洞因类型校验不严、路径处理不当产生，可致恶意代码执行，需结合多种防护措施提升安全性。

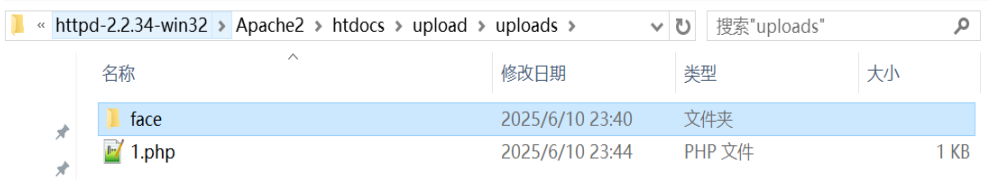
四、 思考题

1. 问题：启动 httpd.exe 时提示丢失 MSVCR100.dll，因该动态链接库是 Visual C++ 运行时组件，丢失会致程序无法启动，多是 C++ 运行环境损坏或缺失。

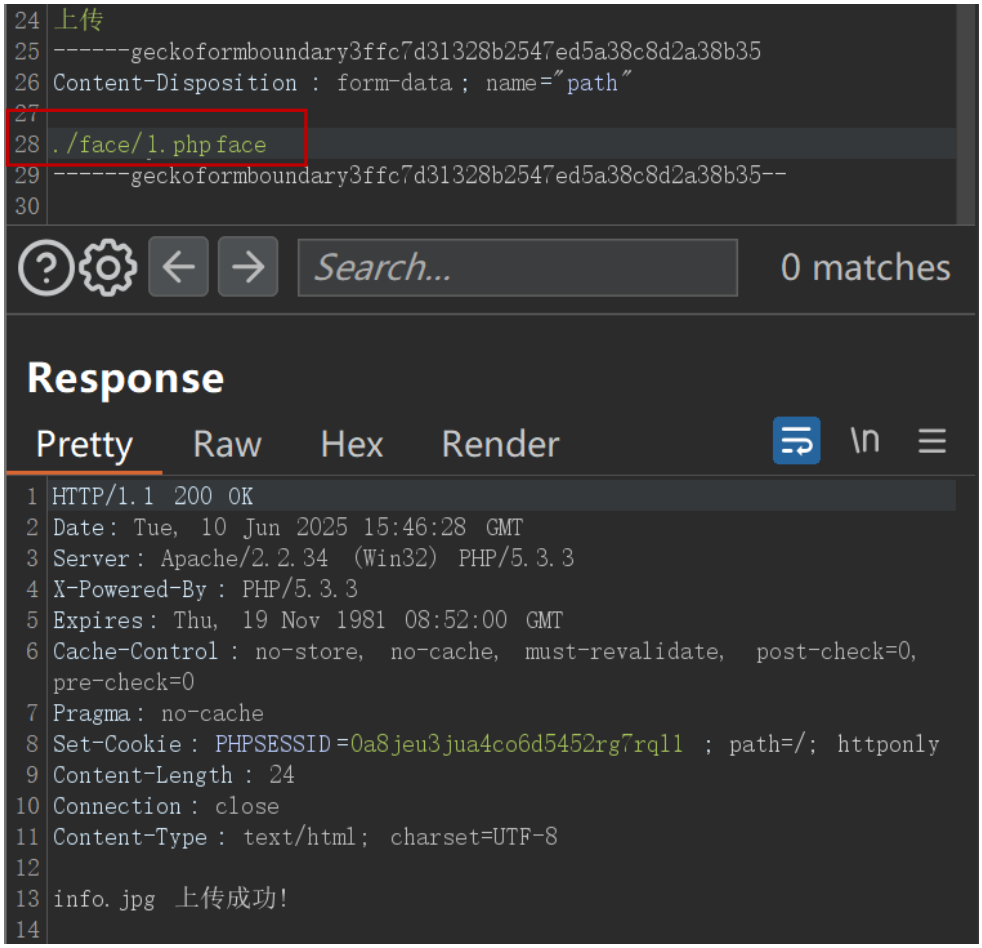
解决方法：安装 dll 修复工具，修复 C++环境。

2. 如果要使用 0x00 截断把 php 文件上传到 face 目录下，应该在 Fiddler 怎么修改路径？

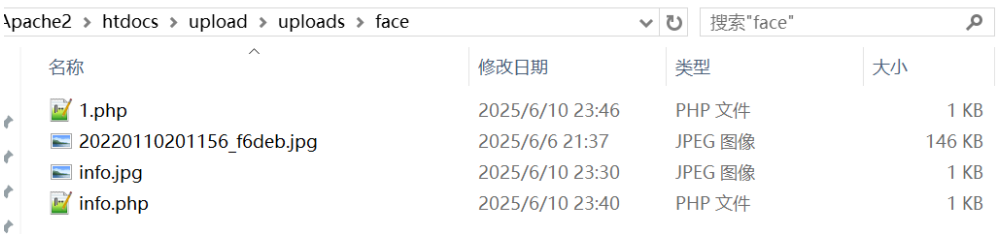
原本上传 1.php 在 upload 上



我们修改 path 变量中的内容为“./face/1.php0x00face”，0x00 后面的内容忽略，即文件保存在./face/目录下



查看目录 face，发现确实存在 1.php 文件。



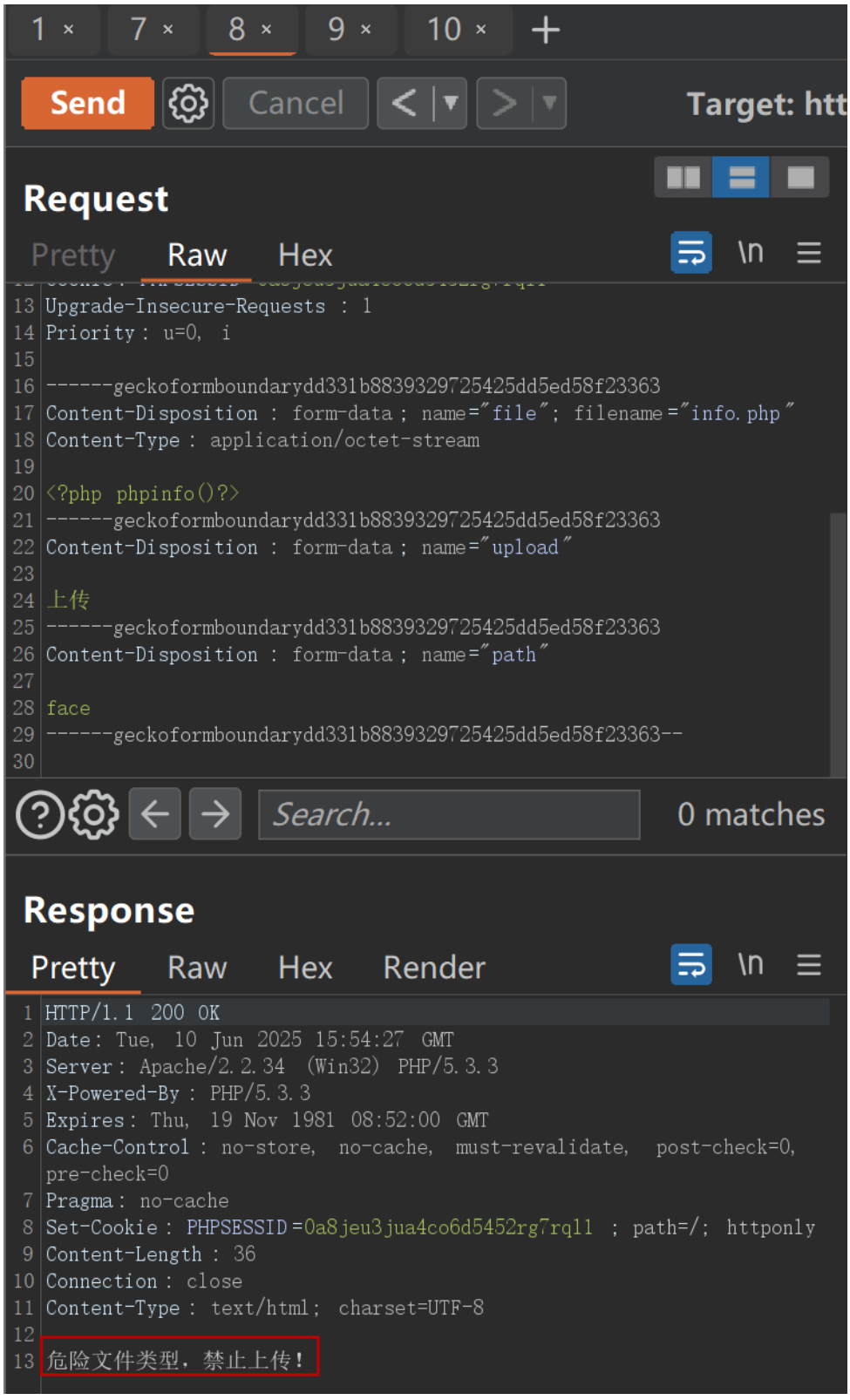
3. 怎么使用黑名单的方式实现上传文件过滤？

创建新 do_upload_black.php，写入代码

```
<?php
include_once "functions.php";
if(!isset($_SESSION)) start_session($expires);

if(!isset($_SESSION['username'])) {
    exit('您没有权限访问此页面');
}

if (!isset($_POST['upload'])) {
    exit('请选择需要上传的文件');
}
```

```
    $target_path = './uploads/' . $_POST['path'];
    $target_path = $target_path . '/' . $_FILES['file']['name'];
    $uploaded_name = $_FILES['file']['name'];
    $uploaded_type = $_FILES['file']['type'];
    $uploaded_size = $_FILES['file']['size'];

    // 黑名单
    $blacklist_ext = array(
        'php', 'php3', 'php4', 'php5', 'phtml', 'phps', 'html', 'htm', 'js','asp', 'aspx',
'jsp','exe', 'bat', 'sh', 'py','htaccess'
    );

    $file_ext = strtolower(pathinfo($uploaded_name, PATHINFO_EXTENSION));

    // 黑名单验证
    if (in_array($file_ext, $blacklist_ext)) {
        exit('危险文件类型，禁止上传！');
    }

    // 文件大小限制（1MB）
    if($uploaded_size > 1000000) {
        exit('文件超过 1M 字节，上传失败');
    }

    $allowed_mime = array('image/gif', 'image/jpeg', 'image/jpg', 'image/png');
    if (!in_array($uploaded_type, $allowed_mime)) {
        exit('文件类型错误，上传失败');
    }

    if(!move_uploaded_file($_FILES['file']['tmp_name'], $target_path)) {
        echo '内部错误，上传失败';
    } else {
        echo htmlspecialchars($uploaded_name) . ' 上传成功!';
    }
?>
```

写入之后进行测试，上传 php 文件，发现报错“危险文件类型，禁止上传”，这是因为识别到了黑名单后缀 php 触发了警报。


1 × 7 × 8 × 9 × 10 ×
Send
Cancel
Target: htt
Request
Pretty Raw Hex
\n
13 Upgrade-Insecure-Requests : 1
14 Priority : u=0, i
15
16 ------geckoformboundarydd331b8839329725425dd5ed58f23363
17 Content-Disposition : form-data ; name="file"; filename="info.php"
18 Content-Type : application/octet-stream
19
20 <?php phpinfo()?>
21 ------geckoformboundarydd331b8839329725425dd5ed58f23363
22 Content-Disposition : form-data ; name="upload"
23
24 上传
25 ------geckoformboundarydd331b8839329725425dd5ed58f23363
26 Content-Disposition : form-data ; name="path"
27
28 face
29 ------geckoformboundarydd331b8839329725425dd5ed58f23363--
30
Search...
0 matches
Response
Pretty Raw Hex Render
\n
1 HTTP/1.1 200 OK
2 Date: Tue, 10 Jun 2025 15:54:27 GMT
3 Server: Apache/2.2.34 (Win32) PHP/5.3.3
4 X-Powered-By : PHP/5.3.3
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control : no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
7 Pragma: no-cache
8 Set-Cookie : PHPSESSID=0a8jeu3jua4co6d5452rg7rql1 ; path=/; httponly
9 Content-Length : 36
10 Connection : close
11 Content-Type : text/html; charset=UTF-8
12
13 危险文件类型，禁止上传！