**存储型xss（上传文件类）笔记**

| | | | |
|---|---|---|---|
| **笔记本:** | SRC实战 | | |
| **创建时间:** | 2021/6/6 13:10 | **更新时间:** | 2023/2/23 20:06 |
| **作者:** | Winding | | |

# 【文件上传基础】

---

- 前端检查
    - 绕过方式：通过浏览器F12很简单的修改文件后缀名就可以完成绕过检查，或者是讲木马修改后缀名后上传，通过改包工具修改上传。如果是JS脚本检测，在本地浏览器客户端禁用JS即可。可使用火狐浏览器的NoScript插件、IE中禁用掉JS等方式实现绕过。

---

- 检查扩展名
    - 黑名单 （最危险的过滤策略）大小写、双写、特殊（后缀加空格）
    - 白名单
    - 绕过方式：
        - 老版本的IIS6中的目录解析漏洞，如果网站目录中有一个 /.asp/目录，那么此目录下面的一切内容都会被当作asp脚本来解析
        - 老版本的IIS6中的分号漏洞：IIS在解析文件名的时候可能将分号后面的内容丢弃，那么我们可以在上传的时候给后面加入分号内容来避免黑名单过滤，如 a.asp;jpg
        - 旧版Windows Server中存在空格和dot漏洞类似于 a.php. 和 a.php[空格] 这样的文件名存储后会被windows去掉点和空格，从而使得加上这两个东西可以突破过滤，成功上传，并且被当作php代码来执行
        - nginx(0.5.x, 0.6.x, 0.7 <= 0.7.65, 0.8 <= 0.8.37)空字节漏洞 xxx.jpg%00.php 这样的文件名会被解析为php代码运行（fastcgi会把这个文件当php看，不受空字节影响，但是检查文件后缀的那个功能会把空字节后面的东西抛弃，所以识别为jpg）
        - apache1.x,2.x的解析漏洞，上传如a.php.rar a.php.gif 类型的文件名，可以避免对于php文件的过滤机制，但是由于apache在解析文件名的时候是从右向左读，如果遇到不能识别的扩展名则跳过，rar等扩展名是apache不能识别的，因此就会直接将类型识别为php，从而达到了注入php代码的目的

---

- 检查Content-Type【见以下表格】https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types

| | | |
|---|---|---|
| .txt | text/plain | 普通文本 |
| .mrp | application/octet-stream | MRP文件（国内普遍的手机） |
| .ipa | application/iphone-package-archive | IPA文件(IPHONE) |
| .deb | application/x-debian-package-archive | DED文件(IPHONE) |
| .apk | application/vnd.android.package-archive | APK文件(安卓系统) |
| .cab | application/vnd.cab-com-archive | CAB文件(Windows Mobile) |
| .xap | application/x-silverlight-app | XAP文件(Windows Phone 7) |
| .sis | application/vnd.symbian.install-archive | SIS文件(symbian平台) |
| .jar | application/java-archive | JAR文件(JAVA平台手机通用格式) |
| .jad | text/vnd.sun.j2me.app-descriptor | JAD文件(JAVA平台手机通用格式) |
| .sisx | application/vnd.symbian.epoc/x-sisx-app | SISX文件(symbian平台) |

- 绕过方式：使用各种各样的工具（如burpsuite）强行篡改Header就可以，将Content-Type: application/php改为其他web程序允许的类型

---

- 检测文件头　　https://en.wikipedia.org/wiki/Magic_number_(programming)
  https://en.wikipedia.org/wiki/List_of_file_signatures

| | |
|---|---|
| TIFF (tif) | 49492A00 |
| Windows Bitmap (bmp) | 424D |
| CAD (dwg) | 41433130 |
| Adobe Photoshop (psd) | 38425053 |
| JPEG (jpg) | FFD8FF |
| PNG (png) | 89504E47 |
| GIF (gif) | 47494638 |
| XML (xml) | 3C3F786D6C |
| HTML (html) | 68746D6C3E |
| MS Word/Excel (xls.or.doc) | D0CF11E0 |
| MS Access (mdb) | 5374616E64617264204A |
| ZIP Archive (zip), | 504B0304 |
| RAR Archive (rar), | 52617221 |
| Wave (wav), | 57415645 |
| AVI (avi), | 41564920 |
| Adobe Acrobat (pdf), | 255044462D312E |

- 绕过方式：给上传脚本加上相应的幻数头字节就可以，php引擎会将 <?之前的内容当作html文本，不解释而跳过之，后面的代码仍然能够得到执行比如下面：（一般不限制图片文件格式的时候使用GIF的头比较方便，因为全都是文本可打印字符。）

---

- 文件00截断

- 0x00或%00放在最后的.前面或者直接替代最后的.

- 条件竞争
    - 绕过方式：同文件包含漏洞所用木马
- 检测文件内容
    - 绕过方式：除了文件头幻数，在一句话木马前后加上无关数据

---

# 【文件上传型存储xss】

## 搜集文件上传点+构造含有payload的文件+成功上传+找到保存路径+成功解析

【可以插入xss代码的文件格式】svg html pdf doc/docx

svg文件内容：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
x="0px" y="0px" width="100px" height="100px" viewBox="0 0 751 751" enable-background="new 0 0 751 751"
xml:space="preserve">  <image id="image0" width="751" height="751" x="0" y="0"
   href=""
/>
<script>alert(1)</script>
</svg>
```

html文件内容：

```
<html>
<body>
<script>alert(1)</script>
</body>
</html>
```

pdf文件内容：

使用迅捷PDF编辑器打开，见https://www.shuzhiduo.com/A/WpdKyvvAJV/
app.alert(1);

doc/docx文件内容：https://bestestredteam.com/2018/07/27/learning-to-pop-xss-with-docx-files/

【待更新】

---

Payload构造点： https://brutelogic.com.br/blog/file-upload-xss/

1. 文件名【任意文件格式都可以】
2. 元数据【EXIFTOOL】
3. 内容【见上】
4. 来源【尚不太懂】

---

文件上传导致的存储xss漏洞存在的功能点：【文件上传点总结】

意见反馈 编辑器 建议 提交工单 在线客服 评论区 留言处 头像图片自定义 / 直接搜在线课程或网上商城等需要客服的网站，找在线客服

也可以直接谷歌搜索 inurl:目标域名 上传 | inurl:feedback 上传 | intitle:意见反馈 | intitle:提交工单 | inurl:chat 在线客服

---

在线客服上传点的漏洞挖掘经验：

- 看看有没有图片上传或文件上传功能
- 看看图片上传是否接受xss.svg.jpg或xss.svg，（注意：上传文件的时候，会设定后缀要求，但是如果没有加以验证，可以提交某些其他后缀的文件）成功则看是否有保存路径（开发者工具，burp抓包，直接右键复制保存路径）且保存路径是否可用【保存路径是否开头有blob：/ 域名是否不是文件上传点的域名 /是不是先自动下载文件而不是解析文件/ 是不是post提交的保存地址（几乎没救）】（注意:保存路径有时候会分为临时和永久，一般提交完毕后看到的是永久，而一些漏洞可能在临时中，也可以造成不小的影响）失败看下面
- 看看是否接受xhtml html xml，成功就同上，失败看下面
- 用pdf或swf格式的，以上的可以随时注意有没有通用型漏洞，找寻相同特征（查看js或者抓包，用fofa进行搜索）

---

存储xss via 文件上传挖掘分析思路：【黄色是可以绕过，绿色是完全不检查，红色是无法绕过，白色是不确定】

【三大要素】Filename Coontent-Type Payload

| 检查部分 | 不检查 | Filename | Content-Type | Payload | F+C | C+P | F+P | 全部 |
|---|---|---|---|---|---|---|---|---|
| 不检查 | | | | | | | | |
| Filename | | | | | | | | |
| Content-Type | | | | | | | | |
| Payload | | | | | | | | |
| F+C | | | | | | | | |
| C+P | | | | | | | | |
| F+P | | | | | | | | |
| 全部 | | | | | | | | |

说明如下：

- 第一个阶段是检查（纵向），第二个阶段是解析文件的类型（横向），顺序可变，依照代码顺序来
- 单个或组合要素进行文件验证或解析

---

- 案例：https://hackerone.com/reports/880099