In the security world, you might have heard of the exploit used by hackers to reveal passwords from their hashed counterparts. We call this technique password cracking or in practicality ‘password guessing’. Even with the complexity of password controls put in by organizations today, this threat is very much real. This tutorial is intended for any individual with a mindset of security who wants to learn more about how hackers are able to crack Windows stored user passwords.

**Introduction to hashing, rainbow tables**

Hashing is a software process of generating fixed character length hash values for a text file. This is a *one-way* function meaning the original text file cannot be generated back from the hash value. This hash value is used to verify the integrity of original text when it is sent over a communication medium. For example, when A sends a text message to B, it first creates a SHA-2 (*popular hashing algorithm)*hash of the message and sends it along with the message. When B receives the message, it also creates a hash of the text message using same SHA-2 algorithm and compares it with the hash provided by A. If the hashes match, B can be rest assured that the original message has not been corrupted on the way.

Application engineers also use this technique for securing passwords of users logging into their systems. Instead of storing passwords in the back-end database in clear text, **password hashes** are used. This protects clear-text passwords from internal application developers and also from hackers in case they are able to breach the database. Hackers are cognizant of this process and have lot of tools in their arsenal to efficiently *guess*the passwords from the hashes. I use the word ‘guess’ because remember hashes are one-way functions, you cant decode them like you can do to an encrypted string. You would need to create a hash of a guessed password and compare to the extracted hash to determine if you have guessed correct.

Free online tables are available which store password hashes of common passwords which can make a hackers job lot easier if people are not serious about password complexities. These tables are called **rainbow tables**or hash tables. In case of complex passwords, there are free tools which use a brute-force approach of comparing hashes of multiple combinations of text. Regardless of the approach being used, it is appropriate to state that password hashes are NOT SAFE if in the hands of an ill-will hacker.

Windows hashing basics

You really need to know only the following three basic concepts before extracting Windows hashes:

**LM hash**

LAN Manager (LM) hash is an old and weak Windows technique for creating hashed passwords, which has been disabled by default in current Windows environments. But this can still be enabled manually on current systems — See Microsoft documentation on how to protect your systems from using it:

**[Network security Do not store LAN Manager hash value on next password change (Windows 10)](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-security-do-not-store-lan-manager-hash-value-on-next-password-change" \t "_blank)**

[Describes the best practices, location, values, policy management and security considerations for the Network security…](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-security-do-not-store-lan-manager-hash-value-on-next-password-change" \t "_blank)

[docs.microsoft.com](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-security-do-not-store-lan-manager-hash-value-on-next-password-change" \t "_blank)

The reason why LM hash is easier to break is because passwords are not case sensitive, password length is maximum 14 characters and more importantly because it breaks the text in two halves of seven characters before hashing them separately and concatenating. So if your password is less than seven characters, it should be a breeze for a hacker to guess the password. [1]

**NT hash or NTLM hash**

New Technology (NT) LAN Manager hash is the new and more secure way of hashing passwords used by current Windows operating systems. It first encodes the password using UTF-16-LE and then hashes with MD-4 hashing algorithm.

If you need to know more about Windows hashes, the following article makes it easy to understand [2]

**SAM database file**

Security Account Manager (SAM) is the database file that stores the user’s password in the hashed format. You would need access to this file in order to retrieve hashes from your local or remote Windows machine [3]

Extracting local hashes from Windows Server 2016

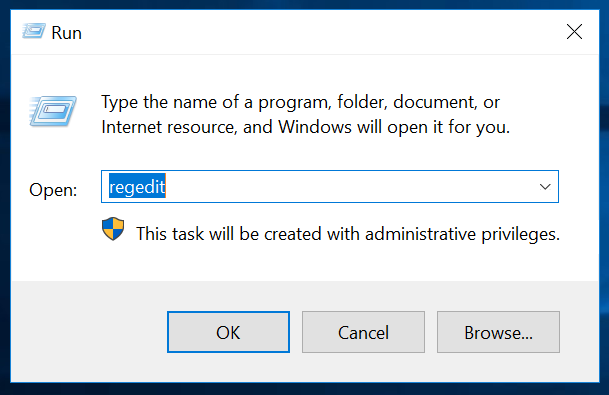
In this section, I will show you how to extract hashed passwords from your Windows desktops using a very popular and powerful tool — mimikatz. The screenshots are from Windows Server 2016.

*Step 1: Download mimikatz*

Binaries are available at — <https://github.com/gentilkiwi/mimikatz/releases>

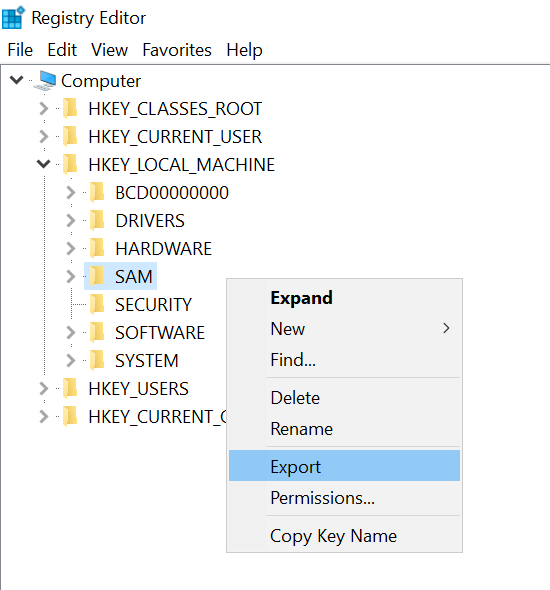
*Step 2: Run (regedit)*

Image for post



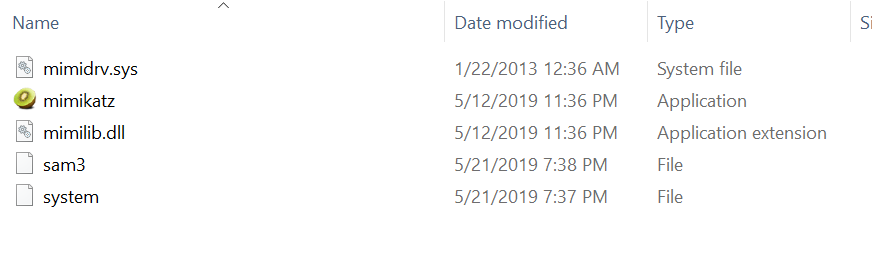
*Step 3: Navigate to HKEY\_LOCAL\_MACHINE and export SAM registry file and SYSTEM registry file to the same directory as the mimikatz installation. Save the files as “Registry hive files”*

Image for post



Your mimikatz directory should look as below:

Image for post

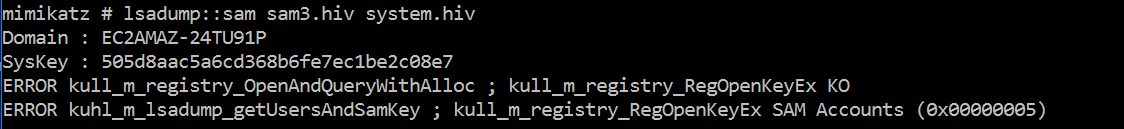


*Step 4: Run mimikatz.exe and type “lasdump::sam” command followed by the file paths of sam and system file:*

*lsadump::sam sam3.hiv system.hiv*

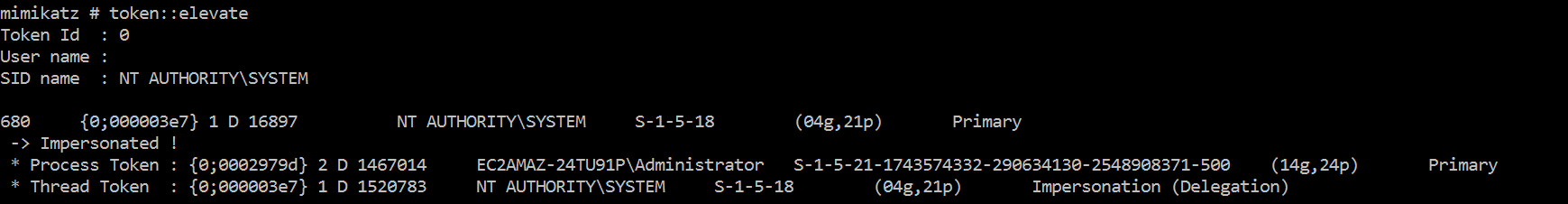
If you get an error as below, you will need to elevate permissions of mimkatz

Image for post



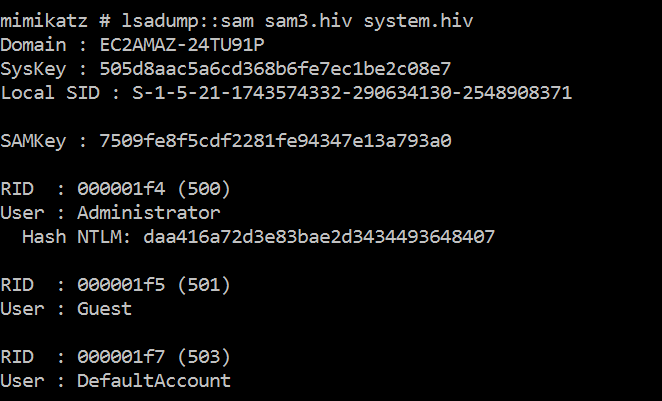
*Step 5: Type “token::elevate” to elevate the permissions*

Image for post



*Step 6: Type the lsadump command again and you should now see the hash values of local users*

Image for post

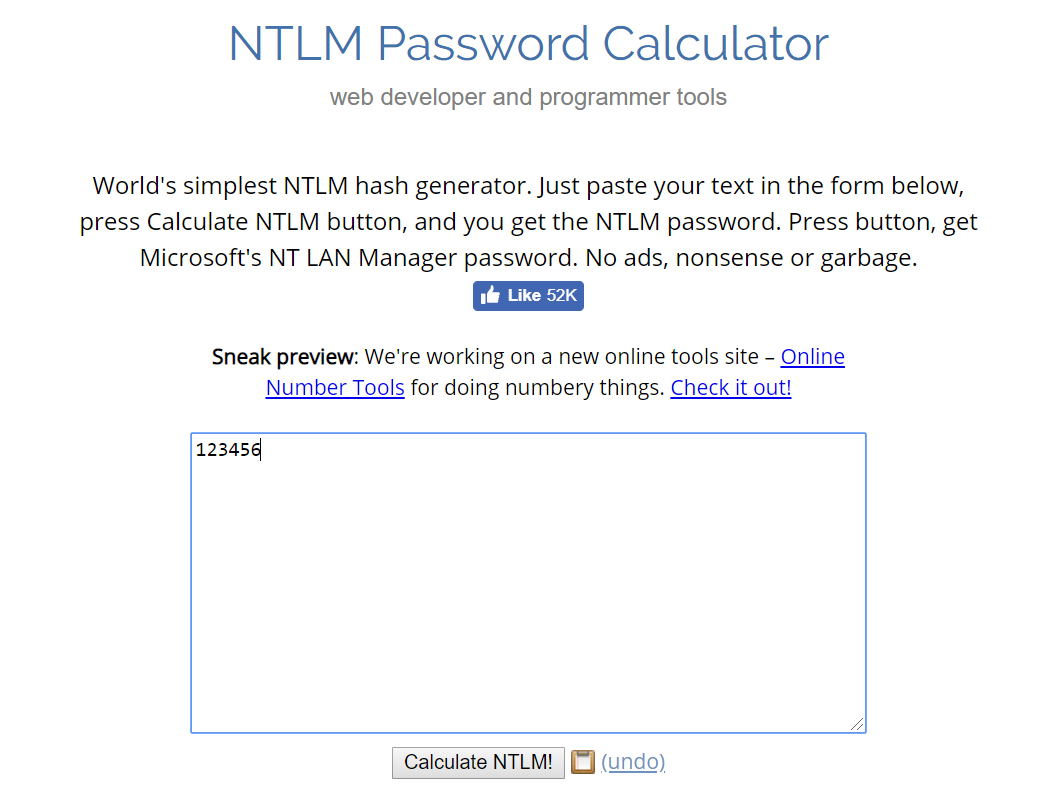


Confirm if you got the right hash

Use Windows commands to create local users and extract the generated NTLM hash using the above process. Once you have the hash, use the below online utility to generate hashes by yourself and confirm if it matches.

<https://www.browserling.com/tools/ntlm-hash> [4]

Image for post



**Windows commands for user and password modifications:**

List of all users → net user

Add user → net user /add username -key=”password”

Update password of user → net user username newpassword

**Other tools that can be used in place of mimikatz:**

HashSuite, fqdump, pwdump2

**Password cracking/guessing tools**:

L0phtCrack, Cain and Abel, John the Ripper

A quick note on **Salting**

Salting is a quick way of increasing the security of your hashed passwords. Passwords first are concatenated with a randomly generated set of bits (salt) and then the hash is calculated. Even if users have same password, they will have different hashes since the salt is randomly generated for each user. Salting also protects against rainbow tables since the table now must contain “salt.password” hashes which is unlikely for a long and random salt value. [5]

**Sources**

1. <https://en.wikipedia.org/wiki/LAN_Manager>
2. <https://medium.com/@petergombos/lm-ntlm-net-ntlmv2-oh-my-a9b235c58ed4>

3. <https://en.wikipedia.org/wiki/Security_Account_Manager>

4. <https://www.browserling.com/tools/ntlm-hash>

5. <https://en.wikipedia.org/wiki/Salt_(cryptography)>