Passwords technical overview

Article • 07/29/2021 • 11 minutes to read

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows 10, Windows Server 2012 R2, Windows 8.1, Windows Server 2012, Windows 8, Windows 7, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Vista

This topic for the IT professional explains how Windows implements passwords in versions of Windows beginning with Windows Server 2012 and Windows 8.1. It also discusses strong passwords, passphrases, and password policies.

How passwords are stored in Windows

This article provides information about the storage of passwords "at rest".

Windows represents passwords in 256-character UNICODE strings, but the logon dialog box is limited to 127 characters. Therefore, the longest possible password has 127 characters. Programs such as services can use longer passwords, but they must be set programmatically.

The Windows operating system stores passwords many different ways for different purposes.

Passwords stored as OWF

For use in Windows networking, including Active Directory domains, the password is stored two different ways by default: as the LAN Manager one-way function (LM OWF) and as the NT OWF. "One-way function" is a term that denotes a one-way mathematical transformation of data. The data that is being transformed can only be converted through encryption one way and cannot be reversed. The most common type of one-way function in use is a cryptographic hash. A hash is a small set of data that is mathematically tied to some larger set of data from which the hash is calculated. If the larger set of data is changed, the hash also changes. Hashes are useful, for example, as a checksum to verify that data has not been modified in transmission. A cryptographic hash is a hash that fulfills certain properties. A cryptographic hash must, for instance, be created in such a way that it is mathematically infeasible in a reasonable amount of time to infer the larger set of data from only the hash. Likewise, it is mathematically infeasible to find two sets of large data that generate the same hash.

There are many different types of one-way functions. All hash functions are, by definition, one-way functions. However, ordinary cryptographic functions that are typically reversible can also be used to create a one-way function. This can be done by swapping the data and the key in a cryptographic function and encrypting the fixed value (the key) by using the data as the key. This is how the LM hash is computed. The LM hash is computed as follows:

1. The password is padded with NULL bytes to exactly 14 characters. If the password is longer than 14 characters, it is replaced with 14 NULL bytes for the remaining operations.
2. The password is converted to all uppercase.
3. The password is split into two 7-byte (56-bit) keys.
4. Each key is used to encrypt a fixed string.
5. The two results from step 4 are concatenated and stored as the LM hash.

The LM OWF algorithm is included in Windows for backward compatibility with software and hardware that cannot use newer algorithms.

The NT hash is simply a hash. The password is hashed by using the MD4 algorithm and stored. The NT OWF is used for authentication by domain members in both Windows NT 4.0 and earlier domains and in Active Directory domains.

Neither the NT hash nor the LM hash is salted. Salting is a process that combines the password with a random numeric value (the salt) before computing the one-way function.

Passwords stored in Active Directory

Passwords at rest are stored in several attributes of the Active Directory database (NTDS.DIT file). These attributes are listed in the following table:

Active Directory Attribute	Content
unicodePwd	Encrypted NT Hash
dbcsPwd	Encrypted LM Hash
ntPwdHistory	Encrypted NT Hashes - Password History
lmPwdHistory	Encrypted LM Hashes - Password History
supplementalCredentials	Kerberos Keys, WDigest, etc.

> **Note**

The storage of LM hashes is disabled by default since Windows Vista and Windows Server 2008.

When stored in the DIT file, the NT hash is protected by two layers of encryption. In Windows Server 2016/Windows 10 and later versions, it is first encrypted with DES for backwards compatibility and then with CNG BCrypt AES-256 (see CNG BCRYPT_AES_ALGORITHM). Previous Windows versions encrypt NT hashes using two layers of DES + RC4 encryption.

For more information about Supplemental Credentials, see MS-SAMR: supplementalCredentials and Supplemental Credentials Structures.

Passwords stored in the local SAM

On domain members and workstations, local user account password hashes are stored in a local Security Account Manager (SAM) Database located in the registry. They are encrypted using the same encryption and hashing algorithms as Active Directory. The passwords in the supplementalCredentials attribute for local user accounts are also stored in the local SAM Database since Windows Server 2016.

Cached credentials

Windows also stores a password verifier on domain members when a domain user logs on to that domain member. This verifier can be used to authenticate a domain user if the computer is not able to access the domain controller. The password verifier is also commonly called a cached credential. It is computed by taking the NT hash, concatenating the user name to it, and then hashing the result by using the MD4 hash function.

How passwords work in Windows

In Windows and many other operating systems, one method for authenticating a user's identity is to use a secret passphrase or password.

We recommend using secure multi-factor authentication such as Smart Card, FIDO, and Windows Hello for Business. However, password authentication is still required in some scenarios.

Securing your network environment requires that strong passwords be used by all users. This helps avoid the threat of a malicious user guessing a weak password, whether through manual methods or by using tools, to acquire the credentials of a compromised

user account. This is especially true for administrative accounts. When you change a complex password regularly, it reduces the likelihood of a successful password attack.

Password policy settings control the complexity and lifetime of passwords. Password policies affect Windows passwords, not necessarily feature passwords.

Users' ability to modify their passwords is governed by the password policies and the available interfaces. For example, through the Secure Desktop, users can change their password at any time based upon the password policies administered by the system administrator or domain administrator. Features such as Windows Vault, BitLocker, and Encrypting File System allow users to modify passwords specific to that feature.

How passwords are used in Windows

When a user logs on, the password the user types is converted into both types of one-way functions and held in memory by the Local Security Authority Subsystem Service (LSASS) process. If the user using a local account for authentication, the NT OWF is compared against the locally stored NT hash, and if the two match, the user is logged on. If the user is authenticating against an Active Directory domain by using a host name to access a resource, the NT hash is used in a Kerberos logon against the Key Distribution Center (KDC), which is typically the domain controller.

Kerberos cannot be used in the following situations:

- Authenticating against a domain running only Windows NT 4.0 or earlier
- Accessing a resource on an Active Directory domain member by using an IP address rather than a host name
- Accessing a resource on a computer that is not a member of an Active Directory domain
- Accessing a resource on a computer that is a member of an Active Directory domain but not trusted by your domain
- Accessing any resource on a computer running that does not support Kerberos

In these situations, the authentication process uses two different protocols, called LAN Manager and NTLM. The process starts with the client requesting a challenge from the authentication server. After the challenge is received, the client computes a response to this challenge. This is done by first padding the two hashes of the password with null values to 168 bits. The 168 bits of each hash are then split into three 56-bit DES keys. The six DES keys are then used to encrypt the challenge. The three cipher texts produced by using the LM hash are concatenated and become the LAN Manager response. The three cipher texts produced by using the NT hash are concatenated and become the NTLM response.

The functions used to compute the response may be modified by the **LM Compatibility Level** setting in the **Network security: LAN Manager authentication level** Group Policy setting. If that value is set to 1 or lower, the client will send the original LAN Manager and NTLM responses. If it is set to 2, only the NTLM response is sent. If it is set to 3 or higher, a new version of both protocols is used. The NTLM version is called NTLMv2. The LAN Manager version is often referred to as LMv2. Both protocols use the NT hash to compute the response, and both use a client-side challenge, either instead of or in addition to the server challenge. In addition, if the **LM Compatibility Level** setting is set to 1 or higher, the NTLM response is time-stamped to help prevent replay attacks. For information about the **LM Compatibility Level** setting, see Network security: LAN Manager authentication level.

Strong passwords

Passwords provide the first line of defense against unauthorized access to your organization. Beginning with Windows Server 2003, Windows checks the complexity of the password for the Administrator account during setup of the operating system. If the password is blank or does not meet complexity requirements, the **Windows Setup** dialog box prompts you to create a strong password for the Administrator account. If you leave this password blank, you will not be able to access this account over the network.

Weak passwords provide attackers with easy access to your computers and network, while strong passwords are considerably more difficult to crack. The following table compares weak and strong passwords.

Weak password	Strong password
Blank	Is at least seven characters long
Contains easily discoverable or known information, such as user name or domain name	Contains "secret" or random information
Is similar to previous passwords	Is significantly different from previous passwords
Contains a complete dictionary word	Contains a mix of the following characters: - Uppercase letters - Lowercase letters - Numerals

Weak password	Strong password
	- Symbols including spaces

An example of a strong password is J*p2leO4>F.

A password can meet most of the criteria of a strong password but still be rather weak. For example, Hello2U! is a relatively weak password even though it meets most of the criteria for a strong password and also meets the complexity requirements of password policy. H!elZl2o is a strong password because the dictionary word is interspersed with symbols, numbers, and other letters. It is important to educate users about the benefits of using strong passwords and to teach them how to create passwords that are actually strong.

You can create passwords that contain characters from the extended ANSI character set. Using extended ANSI characters increases the number of characters that you can choose when you create a password. As a result, it might take more time for password-cracking software to crack passwords that contain these extended ANSI characters than it does to crack other passwords. Before using extended ANSI characters in your password, test them thoroughly to make sure that passwords containing extended ANSI characters are compatible with the applications that your organization uses. Be especially cautious about using extended ANSI characters in passwords if your organization uses several different operating systems. For example, these systems may standardize in ISO-8859-15. The actual protocol implementation on Windows often use UNICODE or UTF8 rather than actual ANSI encoding.

Examples of passwords that contain characters from the extended ANSI character set are kUµ!¶0o and Wf©$0k#»g¤5ªrd.

Passphrases in Windows

A passphrase is a different form of token-based password in which the tokens are words instead of symbols from a character set. An example of a passphrase is a sentence that contains special characters, numerals, uppercase letters, and lowercase letters. The key differences between passphrases and passwords are:

- A passphrase usually has spaces; passwords do not.
- A passphrase is much longer than the vast majority of words, and, more important, longer than any random string of letters that an ordinary person could remember.

Passphrases that conform to the character limit as set in the policy are generally, more difficult to crack than passwords because they contain more characters. It is the LM and

NT hash that stores the password or passphrase, and the LM hash is the weaker of the two.

There are several ways to ensure the LM hash is not stored; one of them is to use passwords or passphrases longer than 14 characters. You can also use the **Network security: Do not store LAN Manager hash value on next password change** Group Policy setting. Using this policy setting globally turns off storage LM hashes for all accounts. The change will take effect the next time the password is changed. Because the policy's effect is not immediate, you will not immediately notice any potential interoperability problems caused by not storing LM hashes.

Local password policies available in Windows

You can implement a password policy setting that enforces password complexity requirements. For more information about this policy setting, see Password must meet complexity requirements. For information about how to apply a password policy, see Apply or Modify a Password Policy . For information about all available password policy settings, see Password Policy.

Fine-grained password policy available through Active Directory Domain Services (AD DS)

Beginning with Windows Server 2008, you can use fine-grained password policies to specify multiple password policies and apply different password restrictions and account lockout policies to different sets of users within a single domain. For example, to increase the security of privileged accounts, you can apply stricter settings to the privileged accounts and then apply less strict settings to the accounts of other users. Or in some cases, you may want to apply a special password policy for accounts whose passwords are synchronized with other data sources.

To store fine-grained password policies, two new object classes exist in the AD DS schema:

- Password Settings Container
- Password Settings

For more information about these policies, see AD DS: Fine-Grained Password Policies.