MANAGING LOCAL AUTHENTICATION IN WINDOWS

Credentials Manager

Windows OS has a set of tools that help remedy some of the authentication challenges. For example, the Credential Manager in Windows 7 and newer Windows versions (it was called Saved Credentials on older Windows versions) enables users to store login usernames and passwords for network resources and websites they use. The username and password information is stored in a protected area called the Windows Vault. Logon information is kept and automatically reused when the user visits the network resource or website. The Windows Vault information can be backed up and restored to another computer. This way we can also transfer our saved credentials. However, we can't backup and restore certificates that are used by Encrypted File System (EFS) in this way.

There are two ways to save credentials to the Credential Manager. For example, when we try to access some resource, we can check the box "Remember My Credentials" in the Windows security dialog. The other way is to manually enter the credentials for the resource by going to the Credential Manager tool in the Control Panel. When manually entering the credential we must not only specify the username password, but also the Internet or network address of the resource we want these credentials to be saved and used for.

We can also use the Credential Manager to edit or delete saved credentials. We can change the password of an existing resource in the Credential Manager as well. Have in mind that we won't be able to see the existing password, but we can change it to a new value. Also, Credentials Manager is not necessary in an Active Directory environment because of domain-level authentication.

Run As Command

After users authenticate and log on into workstation, any task or action they perform execute in the context of the logged in users rights and permissions. Administrator or user can take advantage of the Run As GUI prompt or the *runas* command line utility to run or perform a task as a different user. The *runas* command utility has some interesting options. The */user:<username>* option enables us to specify the username we want to run the application as. The */profile* or the */noprofile* option will load or not load the users profile when the runas command is running. The default is to load the profile which will allow us to access files encrypted by the user, as the encryption certificate is stored within the users profile. The */savecreed* option can be used to save the password and the username used for the runas command in the Windows Vault. The *"<paht\applicationname.exe>"* specifies the path and the name of the application. Have in mind when using the runas command, the user is prompted for the password after running the command. There is no option to specify the password in the command line. There are times when we need to run an application as

administrator only to fix something that the user does not have access to. We can take advantage of the run as option to fix that issue without having to logoff and log back on as an administrator. Runas cannot execute an application that requires elevation if the target user account's UAC settings include prompt for consent or prompt for credentials. To access the GUI version of runas, press Shift and right-click an application.

Account Policies

When it comes to managing and maintaining how passwords are created and how they work, Windows enables us to take advantage of the Password and Lockout Policies. The account policies and lockout policy can be managed either through Group Policies or Local Policies, which reside under the Security Settings of the Computer Configuration node. These policies enable us to configure settings such as acceptable length of password, or the number of times we can incorrectly enter our password before account is locked out, etc. Important policies in this context are:

- Enforce Password History policy - this policy prevents or reduces password rotation. When enabled, Windows remembers the number of passwords you specify and does not allow a user to use one of those previously used passwords. For example, we can configure the policy to three passwords, and the user can not repeat a password until he changed at least three different passwords.

- Maximum Password Age - this policy is the maximum number of days a person can keep the same password. Once a number of days are reached, a user must change the password. This policy is ignored if we enable the password never expires setting on the user account.
- Minimum Password Age - another policy which specifies the number of minimum days that a person must use their password, and cannot change their password. This prevents users from constantly changing passwords, which would also enable them to go trough the existing password history and reusing existing password.
- Minimum Password Length - identifies the minimum number of characters the password must have. If we set this value to zero, which is not a recommended, users will be able to use blind passwords.
- Password Complexity - policy which requires passwords to include uppercase letters, lowercase letters, numbers and alphanumeric symbols. In addition, passwords cannot contain part of the users first name, last name or even username.
- Store Password Using Reversible Encryption - if enabled, passwords are stored in a less secure manner for use with other applications that have older authentication technologies. This essentially means that passwords will be stored as plain text, and is not recommended be enabled.

There are three account lockout policies:

- Account Lockout Duration - enables us to configure the length of time an account is locked out before the user can attempt to login again. If we set this value to zero, the account will be locked out indefinitely, until the administrator unlocks the account manually by going to the account tab in the user properties.
- Account Lockout Threshold - specifies the number of incorrect logins before account is locked out. If we specify number two low, users will get locked up quickly as they make mistakes when entering passwords. If we enter number to high, we increase the chance of brute force in guessing password attacks.
- Reset Account Lockout Counter - amount of time in which Windows records invalid login attempts. After this time is past the users number of invalid logins reset. Additionally when a user logs in, the counters also reset to zero.

When we configure account lockout policies through local policies, they also apply to the administrator account.

Smart Cards

Smart cards can be used for authentication as they store the user's digital certificate. Smart cards provide the most secure method of authentication over usernames and passwords. The main benefit of smart cards is that a persons username and password can be stolen, hacked

or even guessed. The chance of someone losing their smart card and not being aware of it missing is a lot less. For smartcards reported missing, the administrators can quickly revoke the certificate stored on smartcard and basically render the card useless. Smartcards can be used in conjunction with username and password to perform multifactor authentication, which is a process in which a user uses two or more separate forms of authentication to identify themselves. Windows has a set of policies that allow us to manage the use of smartcards for authentication. For example, we can require smartcards for authentication, or we can specify what to do when the smartcard is removed. For example, if the user removes a smartcard from the computer, Windows can lock the workstation, logoff the user, or even shut down the computer.

Starting from Windows 7, Windows has a support for PIV or Personal Identity Verification. This enables Windows to download drivers for smartcards from Windows update, and pick and use PIV compliant drivers. The main advantage of this is the ability for Windows to use smartcards without requiring vendor specific software installed. Stronger authentication protocols and methods such as smart cards and biometrics in multifactor authentication will help maintain a secure and protected environment.

Source: http://www.utilizewindows.com/security/authentication/471-managing-local-authentication-in-windows