Automatické přihlašování windows serveru

Edit: od Windows Server 2019 platí následujícíc postup (ale Autolog platí stále):

https://learn.microsoft.com/en-us/troubleshoot/windows-server/user-profiles-and-logon/turn-on-automatic-logon

EA Infoport a některé další tooly běží pouze pokud je na serveru přihlášen uživatel (tj. nelze je provozovat jako windows service). To je způsobeno tím, že tyto tooly pracují s grafickou vrstvou a ta není, systémovým službám, dostupná.

Tento problém lze vyřešit nastavením automatického přihlašování, vybraného uživatele, na server a to několika způsoby viz:

Varianta User Management Applet

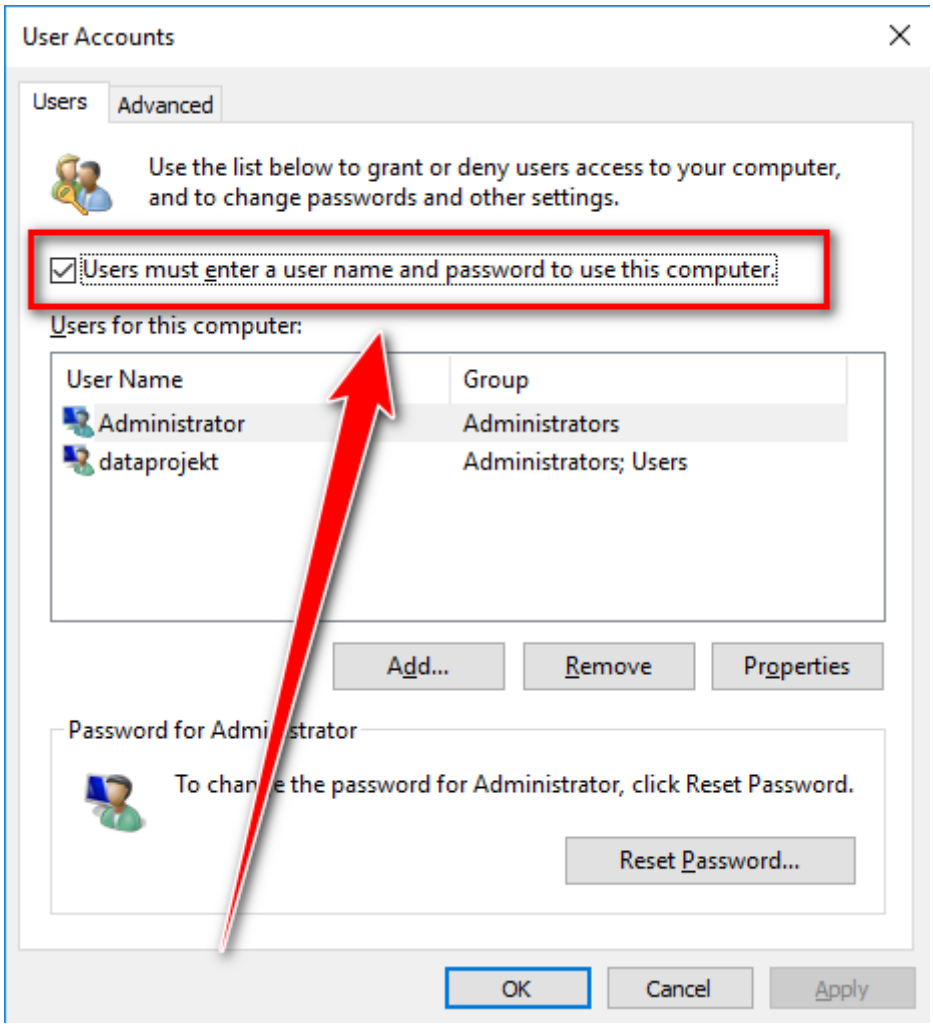
Full steps:

Disabling CTRL+ALT+DEL logon requirement

- Start
- Type 'Local Security Policy' (no quotes) and click on item matching this name
- Open 'Local Policies'
- Open 'Security Options'
- Locate Policy called 'Interactive logon: Do not require CTRL+ALT+DEL'
- Double click on it
- Set to 'Enabled'

To set auto-login account

- Start > Run > netplwiz (as explained in other answers)
- Un-tick "Users must enter a user name and password to use this computer."
- Provide login credentials to be used for auto-logon



After applying this change and rebooting the server it managed to successfully auto-login to the account I had provided.

Varianta Autologon

https://docs.microsoft.com/en-us/sysinternals/downloads/autologon

https://docs.microsoft.com/en-us/windows/win32/secauthn/protecting-the-automatic-logon-password

---

Autologon v3.10

- Article
- 07/27/2021
- 2 minutes to read

-

By Mark Russinovich

Published: August 29, 2016

Download

Image not found or type unknown **Download Autologon (495 KB)**

Run now from Sysinternals Live.

Introduction

Autologon enables you to easily configure Windows' built-in autologon mechanism. Instead of waiting for a user to enter their name and password, Windows uses the credentials you enter with Autologon, which are encrypted in the Registry, to log on the specified user automatically.

[!WARNING] Although the password is encrypted in the registry as an *LSA secret*, a user with administrative rights can easily retrieve and decrypt it. (For more information see Protecting the Automatic Logon Password)

Autologon is easy enough to use. Just run autologon.exe, fill in the dialog, and hit Enable. The next time the system starts, Windows will try to use the entered credentials to log on the user at the console. Note that Autologon does not verify the submitted credentials, nor does it verify that the specified user account is allowed to log on to the computer.

To turn off auto-logon, hit *Disable*. Also, if the shift key is held down before the system performs an autologon, the autologon will be disabled for that logon. You can also pass the username, domain and password as command-line arguments:

autologon user domain password

Note: When Exchange Activesync password restrictions are in place, Windows will not process the autologon configuration.

Download

Image not found or type unknown **Download Autologon (495 KB)**

Run now from Sysinternals Live.

```
#include <windows.h>
#include <stdio.h>
```

```
DWORD UpdateDefaultPassword(WCHAR * pwszSecret)
{

    LSA_OBJECT_ATTRIBUTES ObjectAttributes;
    LSA_HANDLE LsaPolicyHandle = NULL;

    LSA_UNICODE_STRING lusSecretName;
    LSA_UNICODE_STRING lusSecretData;
    USHORT SecretNameLength;
    USHORT SecretDataLength;

    NTSTATUS ntsResult = STATUS_SUCCESS;
    DWORD dwRetCode = ERROR_SUCCESS;

    //  Object attributes are reserved, so initialize to zeros.
    ZeroMemory(&ObjectAttributes, sizeof(ObjectAttributes));

    //  Get a handle to the Policy object.
    ntsResult = LsaOpenPolicy(
        NULL,    // local machine
        &ObjectAttributes,
        POLICY_CREATE_SECRET,
        &LsaPolicyHandle);

    if( STATUS_SUCCESS != ntsResult )
    {
        //  An error occurred. Display it as a win32 error code.
        dwRetCode = LsaNtStatusToWinError(ntsResult);
        wprintf(L"Failed call to LsaOpenPolicy %lu\n", dwRetCode);
        return dwRetCode;
    }

    //  Initialize an LSA_UNICODE_STRING for the name of the
    //  private data ("DefaultPassword").
    SecretNameLength = (USHORT)wcslen(L"DefaultPassword");
    lusSecretName.Buffer = L"DefaultPassword";
    lusSecretName.Length = SecretNameLength * sizeof(WCHAR);
    lusSecretName.MaximumLength =
        (SecretNameLength+1) * sizeof(WCHAR);
```

```
    //  If the pwszSecret parameter is NULL, then clear the secret.
    if( NULL == pwszSecret )
    {
        wprintf(L"Clearing the secret...\n");
        ntsResult = LsaStorePrivateData(
            LsaPolicyHandle,
            &lusSecretName,
            NULL);
        dwRetCode = LsaNtStatusToWinError(ntsResult);
    }
    else
    {
        wprintf(L"Setting the secret...\n");
        //  Initialize an LSA_UNICODE_STRING for the value
        //  of the private data.
        SecretDataLength = (USHORT)wcslen(pwszSecret);
        lusSecretData.Buffer = pwszSecret;
        lusSecretData.Length = SecretDataLength * sizeof(WCHAR);
        lusSecretData.MaximumLength =
            (SecretDataLength+1) * sizeof(WCHAR);
        ntsResult = LsaStorePrivateData(
            LsaPolicyHandle,
            &lusSecretName,
            &lusSecretData);
        dwRetCode = LsaNtStatusToWinError(ntsResult);
    }

    LsaClose(LsaPolicyHandle);

    if (dwRetCode != ERROR_SUCCESS)
        wprintf(L"Failed call to LsaStorePrivateData %lu\n",
            dwRetCode);

    return dwRetCode;

}
```

Revision #5
Created 4 August 2022 12:24:12 by Petr Kalíš
Updated 24 February 2025 14:31:43 by Petr Kalíš