

Impersonate not working from command line

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2005-04/0352.html>

From: Divya (*Divya_at_discussions.microsoft.com*)

Date: 04/27/05

Date: Wed, 27 Apr 2005 14:00:19 -0700

Hello,

I am developing a simple move file utility to move files from one domain to another in the same internal network. My program runs fine when it reads the parameters (user, domain, password etc) from a XML file to move files across the internal network to a different domain. This is a console application that should have the option of specifying the parameters at the commandline prompt. When I specify the parameters at cmd line, it gives me a "Access denied" (access to the file at the destination) error message.

I debugged the code in both the above cases (1-reading params from Xml file and 2-reading params from commandline). The values of username, password, destination domain, source file and destination file were the same in both the cases. But I am getting the Access denied error only when I do this from cmd line.

Do you know how I can fix this? Any help in this regard will be appreciated.

Thanks,
-Divya

I am using the following Impersonator class to login as the admin of the destination domain -

```
public class Impersonator
{
    public WindowsImpersonationContext impersonationContext;

    [DllImport("advapi32.dll")]
    public static extern int LogonUser(String lpszUsername, String lpszDomain,
        String lpszPassword,
        int dwLogonType, int dwLogonProvider, ref IntPtr phToken);

    [DllImport("kernel32.dll")]
    public extern static bool CloseHandle(IntPtr hToken);
}
```

microsoft.public.dotnet.framework.aspnet.security: Impersonate not working from command line

```
public bool Impersonate(string userName, string domain, string password)
{
    WindowsIdentity tempWindowsIdentity;
    IntPtr token = IntPtr.Zero;
    IntPtr tokenDuplicate = IntPtr.Zero;
    // request default security provider a logon token with
LOGON32_LOGON_NEW_CREDENTIALS,
    // token returned is impersonation token, no need to duplicate
    if(LogonUser(userName, domain, password, 9, 0, ref token) != 0)
    {
        tempWindowsIdentity = new WindowsIdentity(token);
        impersonationContext = tempWindowsIdentity.Impersonate();
        // close impersonation token, no longer needed
        CloseHandle(token);
        if (impersonationContext != null)
            return true;
    }
    return false; // Failed to impersonate.
}
}
```