

Re: Using EFS with Network Shares and SFU 3.5

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.windows.server.security/2006-11/msg00126.html>

- *From:* "dln" <dnadon_nospm@xxxxxxxxxxx>
 - *Date:* Wed, 22 Nov 2006 12:00:35 -0600
-

"Roger Abell [MVP]" <mvpNoSpam@xxxxxxx> wrote in message
news:ed2EKfKDHHA.4680@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

"dln" <dnadon_nospm@xxxxxxxxxxx> wrote in message
news:eySgT7aDHHA.4832@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hello all,

Our site is running in an environment that is required to support both Windows and *nix clients. To help support our clients, we have a central Windows 2K3 SP1 file server that also has the NFS server component from Services For Unix 3.5 installed and running on it. The idea is that our users can access their home directory, regardless of the OS they are using. This setup hasn't presented any problems, but today I was doing some testing with EFS on the file server and I found some inconsistencies when I access an encrypted file over a network share via Windows Explorer versus accessing the same file from a Linux client that has my home directory mounted from the Windows file server via NFS.

On my Windows XP client, I can access my home directory on the file server and encrypt a file. This file is then inaccessible to other network users via Windows Explorer as I would expect. However, if I log into a Linux client that has my home directory mounted via NFS, "su" to another user (same user that couldn't access the file via Windows Explorer – not the root user), this user can then open that same encrypted file (using vi) that was previously inaccessible when going through Windows Explorer. If this file was actually encrypted, I would have expected to see a bunch of gooblygook.

I have read that EFS encrypted files are transmitted over the network in the clear and maybe this is a result of that behavior, but I would have expected that file server check the requesting user's credentials before allowing access to the file? Along those lines, it may be a result of the file server being delegated, a topic that I must admit I don't understand that well. In any event, I'm hoping someone can tell me whether or not I have a (mis)configuration problem or if this is expected

behavior?

I am not familiar with the specifics of your scenario. However, I did want to point out that, if the network access is not being made with the credentials to which you have su'd, but the base account, then what you experience would be pretty much what one would expect.

As tests, can you access this if logged in directly as the account that you had su'd to ? What are the NTFS permissions allowing on the file (both accounts?, only the base account?). Can you place an audit on the files used for testing in order to see what credentials the server is seeing sent? The delegation only allows use of the credentials that are received, so it comes down to how the network access is done.

Roger,

Thanks for the response. While testing your suggestions, I think I discovered the root of the problem. I attempted to log into the Linux system as a non-me user (who shouldn't have access to the file). When I attempted to access the encrypted file with this user account, I received an I/O error. I took this to mean that the encryption was working. I then switched to my user account and opened the file without any problems. I then switched back to the user for which the open previously failed and he could access the file without any issues. I even had several other users attempt to access the file and they didn't experience any problems opening it. Additionally, I turned on auditing for the file and it does show successful access by the users who should otherwise not be allowed to access the file.

I used the same process to test access to the file from another server that mounts the same file system and I was able to duplicate the behavior; the non-me user couldn't access the file until I accessed it and then the non-me user could then read the file. I'm thinking the NFS client is caching a copy of the file after it is successfully opened. There's probably an NFS option I can specify somewhere to disable this behavior on the client, but it could impact performance if caching is turned off on the client. I'll need to investigate this further.

Thanks again for the suggestions.

.