

# Re: EFS Decryption Problem

---

*Source:*

[http://www.derkeiler.com/Newsgroups/microsoft.public.windowsxp.security\\_admin/2008-07/msg00170.html](http://www.derkeiler.com/Newsgroups/microsoft.public.windowsxp.security_admin/2008-07/msg00170.html)

---

- *From:* VanguardLH <V@xxxxxxxxxx>
  - *Date:* Sat, 19 Jul 2008 15:35:35 -0500
- 

Ziah wrote:

The solution in the end was to use efsinfo to view the encryption

certificate "thumbprint" for the file in question, then find the

profile

from the backups which had the corresponding certificate intact.

The OP never mentioned having backups. So many users don't do backups that asking them about them usually results in the "deer caught in headlights" reaction. Having the backups of the userprofile comes back to having somewhere that the EFS cert got exported.

The File Encryption Key (FEK) is encrypted using the EFS public key. To decrypt the FEK, you need the EFS private key. After the FEK is decrypted, it is used to decrypt the file.

So what was the actual cure? After you got the thumbprint, what did you do with it? Was it only used to match up to the backed up userprofile, and then restore that userprofile under the new instance of Windows? That would've brought along the user hive (ntuser.dat) for the registry.

I thought the account's SID and password was involved in generating the EFS certificate (and probably the RSA seed, or master key, assigned when a new account is created). There are Microsoft KB articles about how to regain access to EFS-protected files if the password changes, so the password is involved. The username is irrelevant in identifying an account. The SID is used to identify an account, not the username (because you can change the username but the account retains access control rights for that account's SID). A new account under a new instance of Windows would have a different SID even after restoring the userprofile atop the account's current userprofile. So once a new account (same or different username) is created in the new instance of Windows, how do you change the SID for the account in the SAM database

## Re: EFS Decryption Problem

to match up with the SID for the old account under which the EFS cert was generated?

From what I've read, the same SID for an old account can be had for a new one by doing a fresh install of Windows, changing the new Windows instance's computer SID to match up with the old one (by using NewSID from SysInternals) because all account SIDs are based off the computer SID, and then creating the accounts in the same order they were created before. So, in the case where there was only 1 user account, it probably would get the same SID. The user should remember what was their old password.

The certificate & public key are in:

```
%appdata%\Microsoft\SystemCertificates\My\Certificates\
```

The private key is under:

```
%appdata%\Microsoft\Crypto\RSA\
```

%appdata% is under %userprofile%. So restoring the userprofile from backups regained access to these binary values because of the inclusion of the ntuser.dat registry file for the user hive. efsinfo let you figure out which old userprofile to restore (but I'm not sure to where you restored the old userprofile). NewSID lets you reuse the old computer's SID under the new instance of Windows. I'm just leery that recreating accounts actually gets them the same account SID as before so it matches in the SAM database.

Under Windows 2000, the Administrator account was automatically assigned as a recovery agent. The Administrator account gets the same SID under every install of NT-based version of Windows, so you don't have to get its SID changed in a new install of Windows. However, under Windows XP, the Administrator is not automatically included as a recovery agent. The user would have to remember to do that (and I doubt it got done). So the Administrator account in the new Windows install still wouldn't be a recovery agent for the user's EFS-protected files.

So while the user should know their old password and you can recover the old userprofile to get at the keys in the restored ntuser.dat file, how did you get the account SID to match up? The account SID in the recovered ntuser.dat registry file for the user's hive [probably] doesn't match up with the SID for the new account under the new instance of Windows, so even restoring the user hive into the new account would specify the wrong SID everywhere specified in that registry hive and everywhere it was incorporated in generating the encryption keys.

How do you edit the SAM to change the SID for an account? I know that the HKLM\SAM registry key and %systemroot%\system32\config are locked

## Re: EFS Decryption Problem

against changes, even by the Administrator, while that instance of Windows is running, so you have to boot and edit these from a different OS instance. Maybe you restored the system32\config folder from backups, too, to get the old SAM database file (and other system registry hive data). There are various ways to crack into the SAM database but I've never gotten into those, most of which seem to insert a new password hash rather than change the account's SID.

.