



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Securing Windows and PowerShell Automation (Security 505)"  
at <http://www.giac.org/registration/gcwn>

# **Keys to EFS Data Recovery**

**GIAC Certified Windows Security Administrator (GCWN)**

Practical Assignment Version 5.0

Option 2

**Valdo Araiza**

October 11, 2004

© SANS Institute 2004. Author retains full rights.

# Table of Contents

## Abstract:

### Chapter 1

#### An introduction to EFS and PKI

- I. What is Encrypting File System?
- II. How has Microsoft changed or improved EFS?
- III. Public Key Infrastructure and it's relation to EFS

### Chapter 2

#### The Encrypted Data

- I. "My data is encrypted and secure, that is all that matters, right?"
- II. Reasons encrypted data would need to be recovered

### Chapter 3

#### The Recovery Types

- I. Data recovery pros and cons
- II. Key recovery Pros and cons

### Chapter 4

#### The Recovery Process

- I. Data Recovery Step-by-Step
- II. Key Recovery Step-by-Step

## Conclusion:

## Resources:

## **Abstract:**

The growing concern for data confidentiality, integrity, and availability spurred-on by the likes of HIPAA, the Gramm-Leach-Bliley Act, California's Database Security Breach Notification Act (SB1386), and the more recent Sarbanes-Oxley Act has raised a definite interest in encryption of data. Today people are also relying on mobile computing in order to conduct daily business if not due to commuting requirements definitely for the convenience. Microsoft's Windows 2000 Operating System brought on the ability to encrypt data with EFS (Encrypting File System). With just about any new functionality there were some questions and concerns with the use of EFS. An area of question is recovery of encrypted data that has become inaccessible for one reason or another. In this paper I will illustrate what may cause encrypted data to become unavailable and the different methods used for recovery as well as some key notes for using EFS.

## **Chapter 1**

### **An introduction to EFS and PKI**

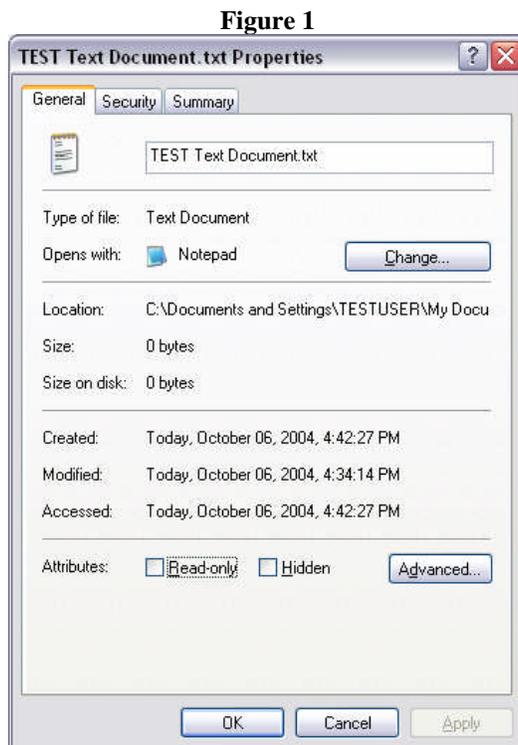
#### **IV. What is Encrypting File System?**

EFS, or Encrypting File System is a feature that is built into Microsoft Windows 2000, XP Professional (not Home Edition), and 2003 NTFS based operating systems. With EFS one can encrypt a file so that only the account that encrypted the file can access it or at least this is the intention. With the release of Windows XP a feature that allows the encrypted file to be shared to selected accounts is available. This will be reviewed in more detail in section II.

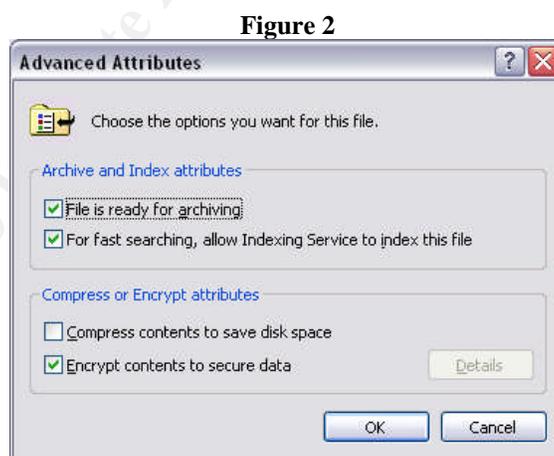
By encrypting files the data becomes secure from attacks that may involve gaining physical access to the hard drive, changing permissions to take ownership of files, or accessing via a different operating system. Even using a local administrator account the encrypted file cannot be opened unless the administrator account is also designated a recovery agent.

EFS employs certificate-based access control structured around the use of both public and private key pairs. A certificate is a digital document that binds a public key to the user's private key. When a file is selected to be encrypted EFS will encrypt the file and generate a File Encryption Key, or FEK, that is based on the encryption algorithm established for the operating system. The FEK is then encrypted with the user's public key and stored as an attribute called Data Decryption Field or DDF. The public key is obtained from the user's X.509 version 3 file encryption certificate if it available. If EFS is not able to locate an encryption certificate it will simply create a self-signed encryption certificate. The process to decrypt the file is basically reversed. The DDF is decrypted by the user's private key then the FEK is used to decrypt the actual data. All this is transparent to the user and requires no additional tools, utilities, or applications. Encryption is an attribute just as compression is. Make note that a compressed file will decompress when encrypted as well as an encrypted file will decrypt when compressed.

To encrypt a file simply right-click the file and select **Properties** then select **Advanced** (see **Figure 1**).



In the **Advanced Attributes** dialog box select **Encrypt contents to secure data** so that a check mark appears and select **OK** (see **Figure 2**).

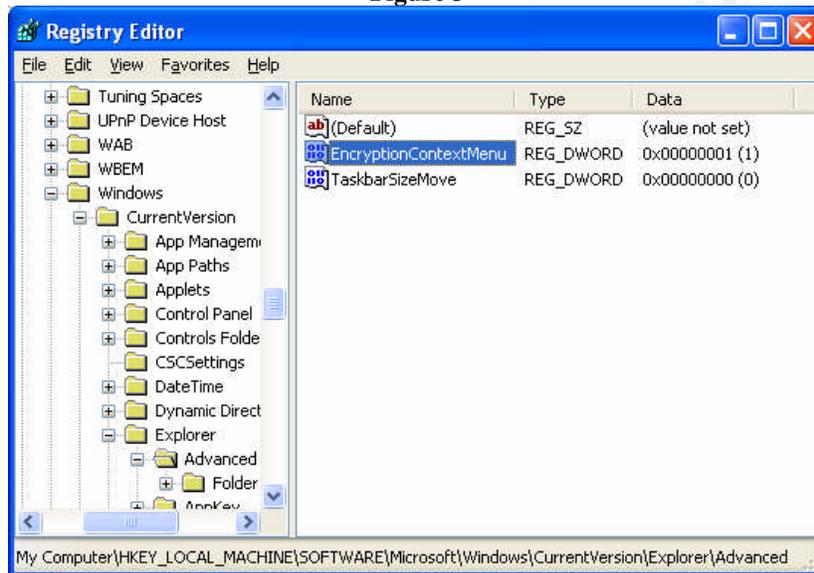


That is all that is required to encrypt a file. You can also have the Encrypt and Decrypt options available on the **Shortcut Menu**. This does require editing the Registry so only do this if you are experienced with Registry edits and as always make sure you backup the Registry before making any changes.

- Run: **regedit.exe**
- Expand to:  
**HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced**
- From the menu select: **Edit > New > DWORD Value**
- Type: **EncryptionContextMenu** as the value name
- Set the value data as **(1)**

See **Figure 3, 4, and 5:**

**Figure 3**



**Figure 4**

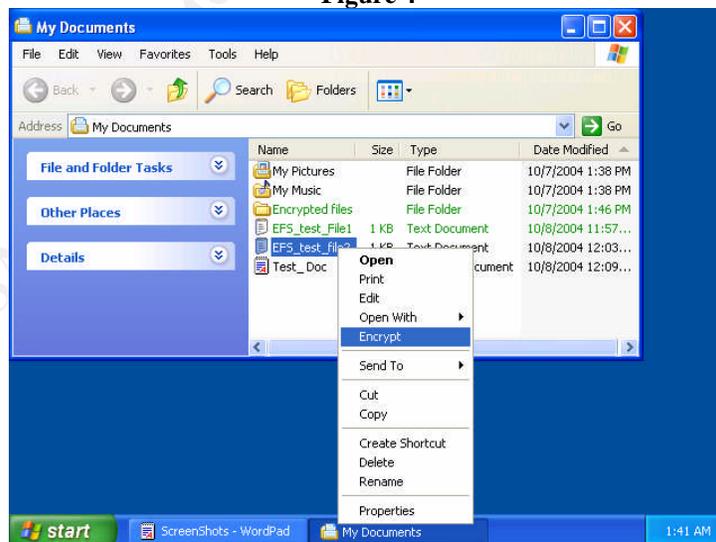
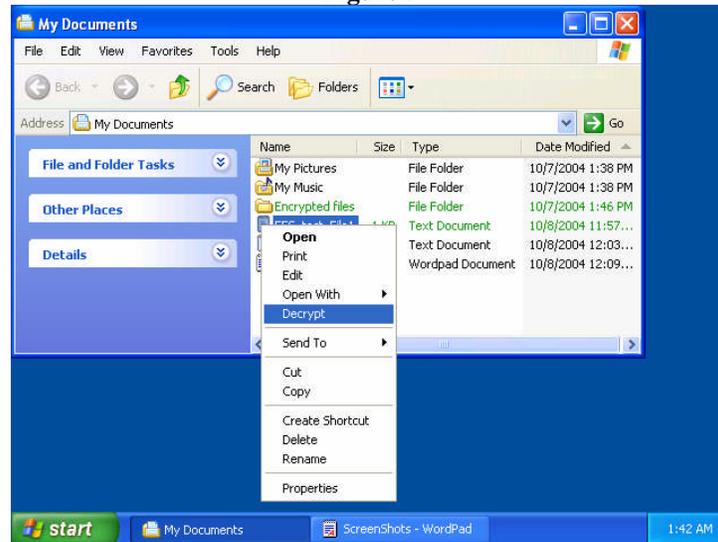


Figure 5



If you do not wish for EFS to be used on a particular computer, you can set this with either a registry edit or by the local security policy.

#### Registry:

- Run: **regedit.exe**
- Expand to:  
**HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\EFS**
- From the menu select: **Edit > New > DWORD Value**
- Type: **EfsConfiguration** as the value name
- Set the value data as **(1)**

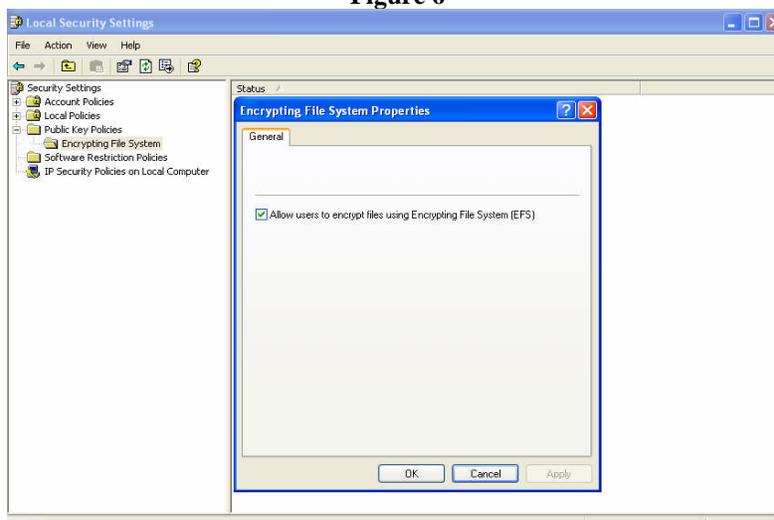
#### Local Security Policy:

(This can also be applied via Group Policy for domain computers either within a domain or particular OU.)

- Go to: **Control Panel > Administrative Tools** (You may need to switch to Classic View) > **Local Security Policy**
- Expand: **Public Key Policies**
- Right click on **Encrypting File System** and select **Properties**
- Remove the checkmark on: **Allow users to encrypt files using Encrypting File System (EFS)**

See **Figure 6**:

Figure 6



#### V. How has Microsoft changed or improved EFS?

With different releases of Windows Operating Systems and Service Packs Microsoft has employed various methods of encryption which are referred to as encryption algorithms. Although this paper primarily focuses on Windows XP and 2003 it is important to know the different algorithms EFS uses.

The different encryption algorithms to date include:

- DESX in Windows 2000
- DESX and 3DES in the initial XP release
- AES with XP SP1 or later as well as 2003

Which OS version can encrypt and which OS version can decrypt a file can get a bit confusing so you should review **Microsoft Knowledge Base Article – 329741**. It is important to be aware of this because if you attempt to decrypt a file with an incorrect algorithm not only will it be inaccessible, it may also become corrupt for even the correct algorithm to open.

A few of the additional functionalities added to EFS in XP and 2003 that were not available in 2000 have been listed below.

- **Encrypted files are green**

Encrypted files and folders can be set to appear as a different color (See **Figure 7 and 8**). Having the encrypted files and folders appear as a different color will help with managing the location of files. This is especially important when applying different permissions to files and folders. It is important to remember that EFS is only a file encryption method and that file permissions are still in place. If a file or folder does not have the proper NTFS permissions applied it can be moved or deleted by another user without every needing to decrypt it.

Figure 7

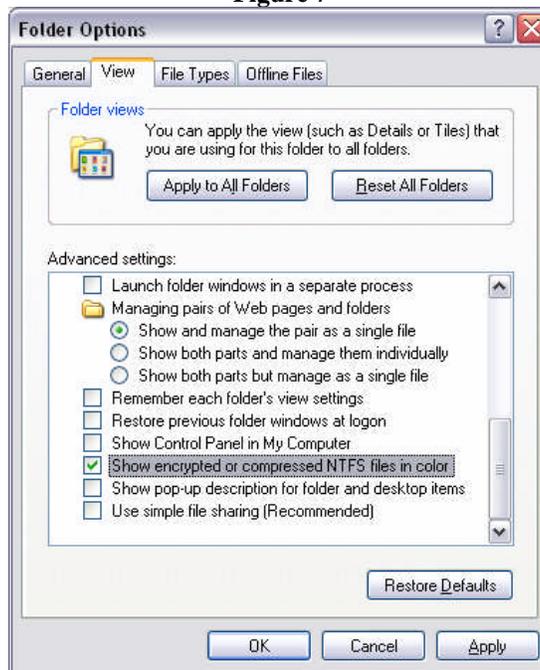
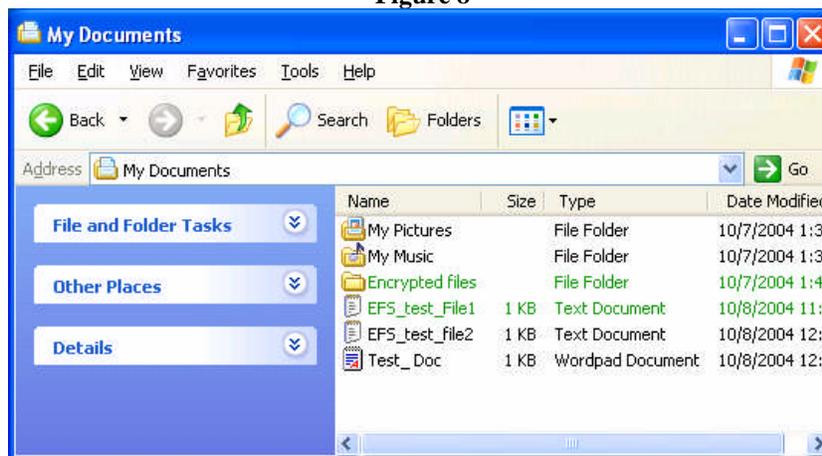


Figure 8



- **Encrypted files can be shared**

Encrypted files can be shared with other accounts that have a certificate present on the same computer. This feature is only for individual accounts and can only be applied to individual files, not folders. Be aware of the fact that any account(s) added to share an encrypted file can also add and REMOVE which accounts share the file. This means that the original account that created and shared the encrypted file can actually be removed therefore denied access to the file.

- **Offline files can be encrypted**

The added security of EFS can be used with the Offline Files option for mobile users to encrypt files that have been cached on their local drive. This option can be allowed or denied through either the local security policy or as a Group Policy Object in Active Directory.

- **3DES can be used instead of DESX**

The encryption algorithm can be set to 3DES instead of the default DESX. 3DES is FIPS (Federal Information Processing Standard) compliant which is required for some U.S. government agencies.

- **Data Recovery Agent not required**

XP does not require a Data Recovery Agent for EFS to be used. This can have some adverse effects but let's first look at the subject of Data Recovery Agents in Windows 2000 to understand this change to XP.

A Windows 2000 stand alone system has an inherent security flaw associated with EFS and the Data Recovery Agent or DRA. A DRA is an account that is used if an encrypted file is unable to be decrypted by the file owner. On a Windows 2000 stand alone computer a user cannot encrypt data unless a recovery agent exists. Once EFS is initiated the Recovery Agent certificate and private key are created and assigned to the local Administrator account. The security threat that lies with this is that the Administrator account, if ever compromised, would then be able to decrypt your files. Remember that in Windows 2000 as well as XP you can install the operating system with a blank Administrator password. Unfortunately this is a huge security hole that is often overlooked. Microsoft has corrected the blank administrator password issue in Windows 2003 Server. There are many attacks that focus on accessing the local administrator account so you want to make sure that you do not increase your chances for data vulnerability. The way to avoid this type of risk is to immediately export the EFS Recovery certificate and private key of the Administrator/DRA to a floppy disk, make a copy and store them away in a secured location. You must also make sure that certificate and key are deleted off the computer.

Windows XP when used as a stand alone computer does not require a DRA certificate to be assigned in order to use EFS. You can immediately start to encrypt data and not worry about the vulnerability associated with Windows 2000 and the local administrator account having the recovery certificate and private key. This does create an entirely new set of problems with EFS. If there is ever an issue where an encrypted file cannot be accessed by the account that encrypted it and a DRA assigned to the computer, the encrypted data will be considered useless since it cannot be decrypted. We will go into more detail on data recovery and what may prompt the need to recover encrypted data in Chapter 2.

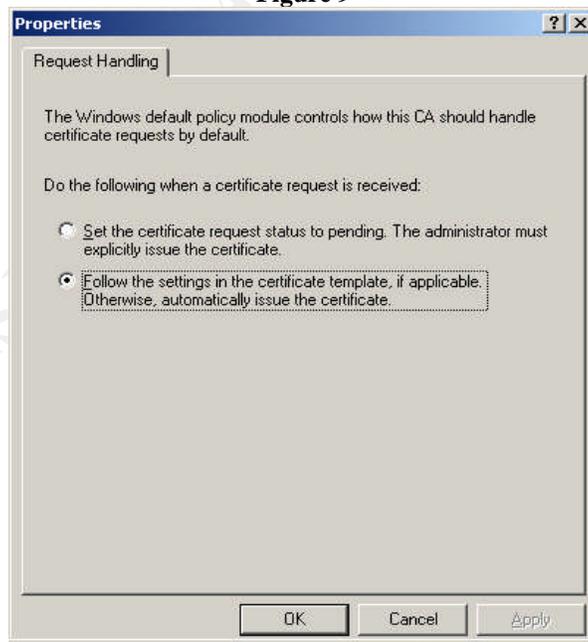
### III. Public Key Infrastructure and it's relation to EFS

A Public Key Infrastructure or PKI is a system that uses digital certificates as an authentication mechanism by managing certificates and the public keys associated to them. The subject of deploying and managing a PKI is highly documented with entire books dedicated to the subject. For our purposes we will look at PKI's use with EFS.

PKI uses a Certificate Authority (CA) to issue, manage, and revoke certificates. A CA will confirm that a public key corresponds to the user based on their certificate. EFS uses certificates to confirm who has encrypted a file and who can decrypt it. When there is no CA available to assign a file encryption certificate Windows 2000 and XP will issue a self-signed certificate for file encryption. This works fine as long as the certificate and/or user profile never becomes corrupt. Without PKI or some other third-party CA in place, key recovery is not an option.

In a domain environment an organization may not want everyone to use EFS. You may also not want every computer to use EFS. This can be controlled with CA permissions established in a PKI. In order for a certificate to be requested the object, user or computer, must have the **Enroll** and **Read** permission. A PKI can use one of two types of CAs. There is a Stand-alone CA and an Enterprise CA. Stand-alone CAs will hold all certificate requests until an administrator approves it. This can be an efficient way to manage who is granted an EFS certificate as long as the volume of requests can be handled. An Enterprise CA will rely on certificate templates and the permissions set on the different templates (see **Figure 9**).

Figure 9



These permissions are managed by Active Directory so there can be groups designated to have access to request an EFS certificate. If additional EFS users are requested they can simply be added to the authorized group in AD.

## Chapter 2

### The Encrypted Data

III. “My data is encrypted and secure, that is all that matters, right?”

Let us start with an illustration of what encryption of data can be compared to:

#### ***Bill and his journal***

*Bill has a journal that contains very personal information so Bill puts it in a safe that uses a combination. Bill writes down this combination on a piece of paper then locks the paper in a safe that has a key. Bill then take this key down to his local bank and locks it in a safe deposit box.*

This is similar to how encryption can be viewed except that when everything is working the way it should be “the bank” is open anytime Bill needs the key. So what does Bill do when the bank is not open? This is the situation you can be placed into with a stand alone XP computer with no Data Recovery Agent. The DRA for the above example would be Bill’s unmentioned best friend Steve that was also supplied with the combination on a piece of paper. Keep this example in mind because it will be revisited when we review different data recovery methods.

Data can be considered very valuable and the need to encrypt it for security may be necessary but that value will only be retained if the data is accessible whenever it is needed. Without the proper precautions in place encrypted data may not be able to be decrypted.

#### II. Reasons encrypted data would need to be recovered

The use of EFS can be used in various environments such as desktop workstations, laptops and servers. All of these can either be on a domain or as a stand alone computer. We will look at a few examples of why data would need to be recovered.

In a stand alone environment a user on a Windows XP computer forgets their local account password. Using the local Administrator account the user’s password is reset. EFS on a stand alone XP computer will use the Data Protection API to store the private key. The encryption of this key is based on the user’s password. If the password is changed the key is lost. Once the DPAPI key is lost the data cannot be decrypted. If a DRA is present the files can be recovered and the user can then re-encrypt them with the current password. An interesting occurrence is that if the user happens to remember exactly what their previous password was, they can change back to that password and the file(s) will be accessible once again. This is encountered only in a stand alone

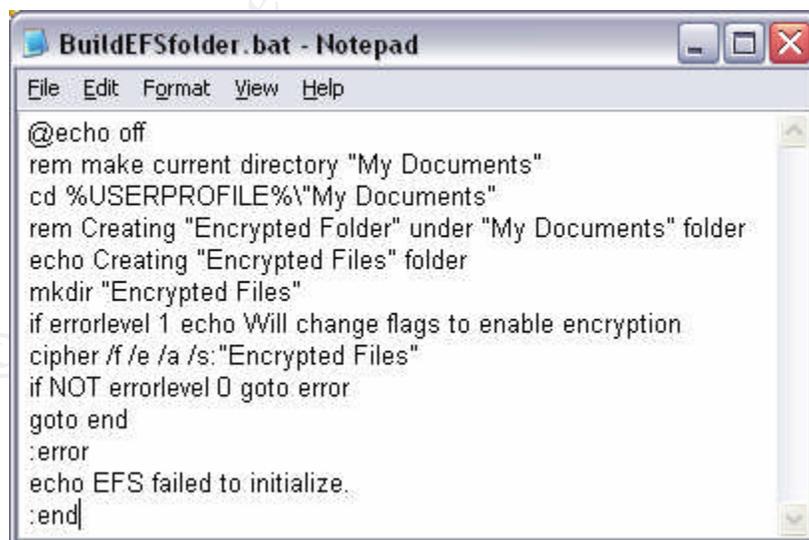
environment; domain accounts do not use the local account password for EFS. For more detail see *Microsoft Knowledge Base Article – 290260*.

An example that can affect both stand alone computers as well as computers in a domain that does not have an EFS policy in place can be if an employee is no longer with the organization. This user had access to important data that unfortunately has been encrypted and resides on their computer. In a domain this would only be a problem if the domain account was deleted, which is a practice that is not recommended. Do not delete domain accounts immediately but rather disable the account. On a stand alone computer you run into the same issue as the first example. If there is no DRA and the local account password is unknown the data may never be decrypted.

Some other examples that can require the use of a DRA is if the profile on the computer changes, becomes corrupted, or is deleted. Remember that if the private key that is located in the user's profile is no longer accessible the encrypted data will need to be recovered. Another option that may be available is key recovery.

### EFS Configuration Tip:

Before exploring the different recovery types and methods I want to share a simple configuration tip that will help cut down time spent and the number of headaches you have when troubleshooting or recovering EFS. This is a batch file that our organization requires anyone that is authorized to use EFS, run before using it. We also state that the only folder that will be supported for recovery will be **Encrypted Files** under **My Documents**.



```
BuildEFSfolder.bat - Notepad
File Edit Format View Help
@echo off
rem make current directory "My Documents"
cd %USERPROFILE%\My Documents
rem Creating "Encrypted Folder" under "My Documents" folder
echo Creating "Encrypted Files" folder
mkdir "Encrypted Files"
if errorlevel 1 echo Will change flags to enable encryption
cipher /f /e /a /s:"Encrypted Files"
if NOT errorlevel 0 goto error
goto end
:error
echo EFS failed to initialize.
:end
```

## Chapter 3

### The Recovery Types

#### III. Data recovery pros and cons

The first method for recovering encrypted data is data recovery. A simple way to view data recovery is that there is more than one account that has access to retrieve the encrypted data. When a file is encrypted and there is an existing Data Recovery Agent, EFS will have the DRA's public key assigned to a copy of the FEK as an attribute called DRF or Data Recovery Field. If there is multiple recovery agents assigned, each will have their public key assigned to a copy of the FEK. The important thing to understand about Data Recovery is that the DRA has the ability to decrypt the data to plaintext and it will be up to the owner to re-encrypt the data.

In reviewing the pros and cons of data recovery, on the plus side data recovery is simple to manage. There is no need for a Certificate Authority or PKI to be implemented and the user does not need to manage certificates or private keys for DRA accounts. The user can have the DRA account and password and recover the data themselves. Data recovery policies can also be managed by Active Directory for computers on the domain.

On the down side the process to decrypt data can be labor intensive. The DRA account must logon to proceed with recovery. Data recovery is file by file unless an entire folder is encrypted. The data is recovered to a plaintext format. It will be the user's responsibility to re-enroll for a new EFS certificate then re-encrypt the data. What data can and cannot be decrypted by the DRA cannot be defined. The DRA will be able to decrypt all encrypted data.

Let's refer back to the example in Chapter 2 section I. *Bill and his journal*. In this example Bill's pal Steve is acting as the Data Recovery Agent. Steve has the combination to open the safe and get to the journal (the data). This is the only way Bill can get to his journal if he cannot get to the bank. Now if Steve just happens to be house-sitting while Bill is away he has the ability to access Bill's entire journal.

#### IV. Key recovery Pros and cons

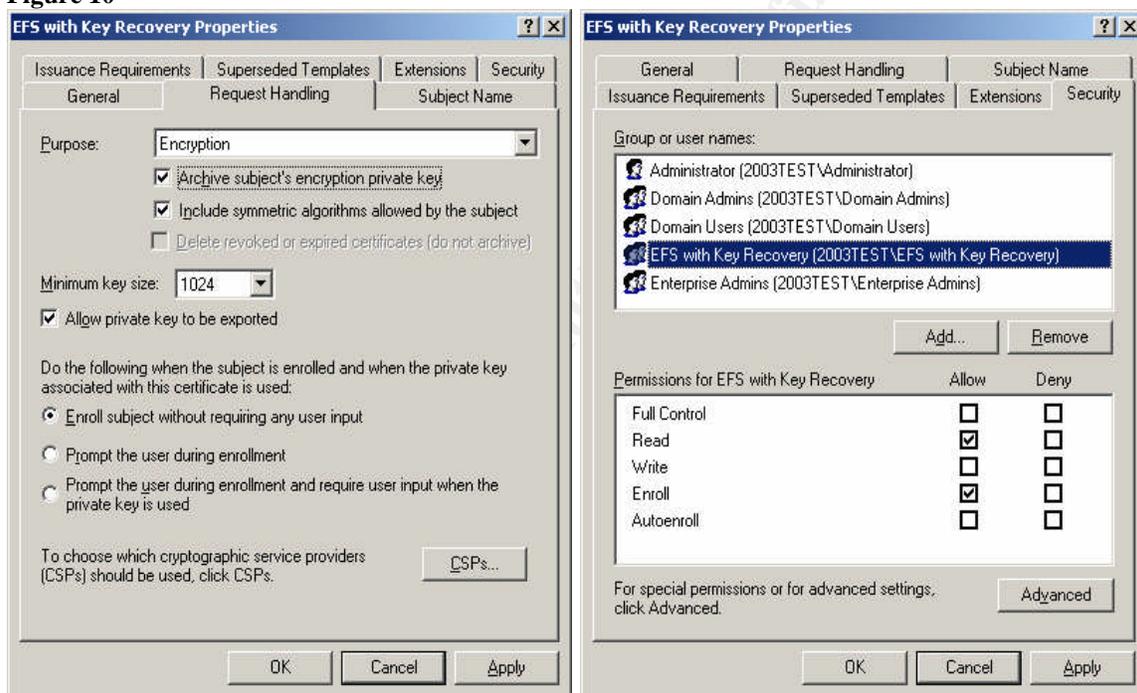
Key recovery is another method used to recover encrypted data. Key recovery agents are designated to have access to users' private keys rather than the data itself. Key recovery is simply recovering an archived copy of a user's private key and distributing the key to the user. Having the ability to access a copy of someone's private key must be clearly understood in an organization due to issues of non-repudiation.

The advantages of key recovery are that the files do not need to be decrypted to a plaintext state and the user does not need to re-enroll for a certificate. There is no need for Key Recovery Agent to log onto the user's workstation.

The disadvantages of using key recovery are that this must be performed by a designated administrator and cannot be performed by the user. The user's private keys are in control of another account therefore non-repudiation is not assured. Key archival is only available with Windows Server 2003 Enterprise and Datacenter editions running an enterprise CA.

Only version 2 certificate templates can be configured for key archival and recovery. **Figure 10** illustrates a version 2 certificate template labeled **EFS with Key Recovery**. Note that under the *Request Handling* tab **Archive subject's encryption private key** has been checked and under the *Security* tab the group that has agreed to allow key recovery is added with the Read and Enroll permission.

**Figure 10**



## Chapter 4 The Recovery Process

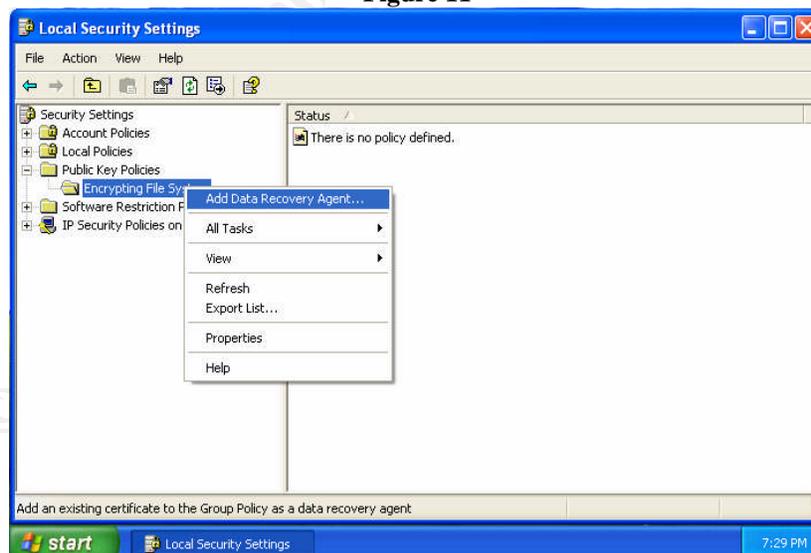
### III. Data Recovery Step-by-Step

#### XP data recovery

Since XP stand-alone computers do not create a DRA we will start with the DRA creation and export then proceed to data recovery.

- Logon as the designated DRA account
- Insert a blank formatted floppy diskette
- Go to **Start > Run** then type: **cipher.exe /r:a:efskey**
- Enter a complex password
- Re-enter password
- Under **Administrative tools** open **Local Security Policy**
- Expand **Public Key Policies** > right click and select **Add Data Recovery Agent** (see **Figure 11**)
- Select **Next > Browse Folders...** (select the A: drive)
- Select the **efskey.cer** > **Next > Finish**
- Make a copy of the disk and store each in a separate secure location

Figure 11



## Importing the File Recovery Certificate

- Insert the floppy diskette that has the exported certificates
- Open **My Computer** and open the A: drive then open **efskey.pfx** file
- Select **Next** on the **Certificate Import Wizard**
- The selected file will appear in the **File name:** field > select **Next**
- Enter the password created during the Export and select **Enable strong private key protection...** (See **Figure 12**) *This will require the password to be entered each time the key is used. This will prevent unauthorized use if the certificate is not removed from the computer.*
- Select **Next >Next > Finished**

Figure 12



Now that the recovery certificate has been imported the DRA account can decrypt any encrypted files on the computer in the same manner that a user decrypts a file.

Once the decryption process is completed make sure to delete the file recovery certificate from the local computer.  
(See **Figure 13 and 14**)

Figure 13

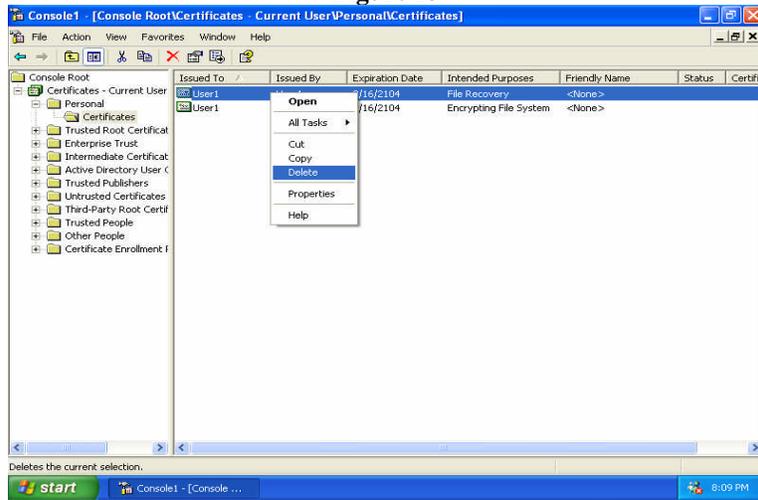


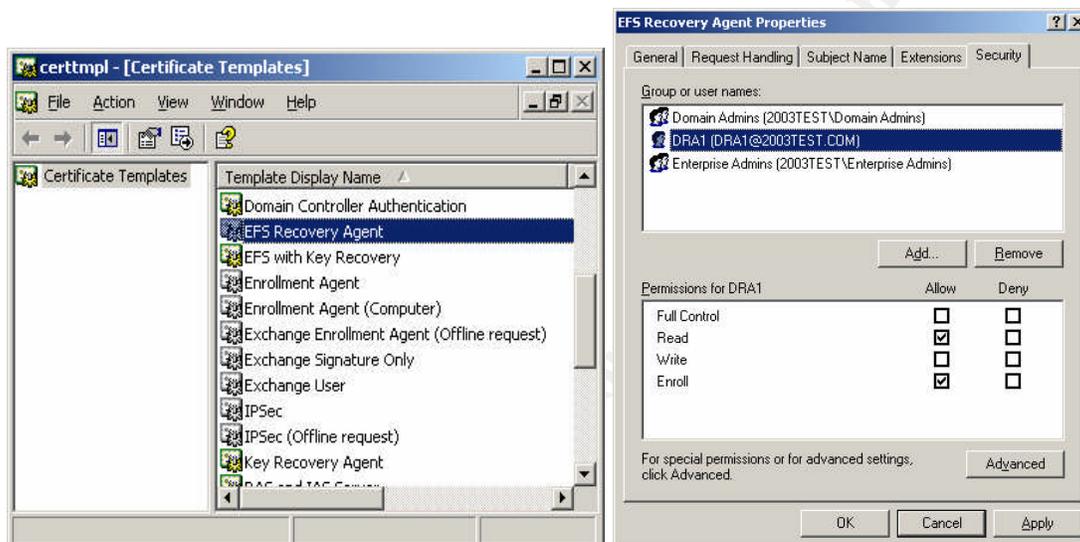
Figure 14



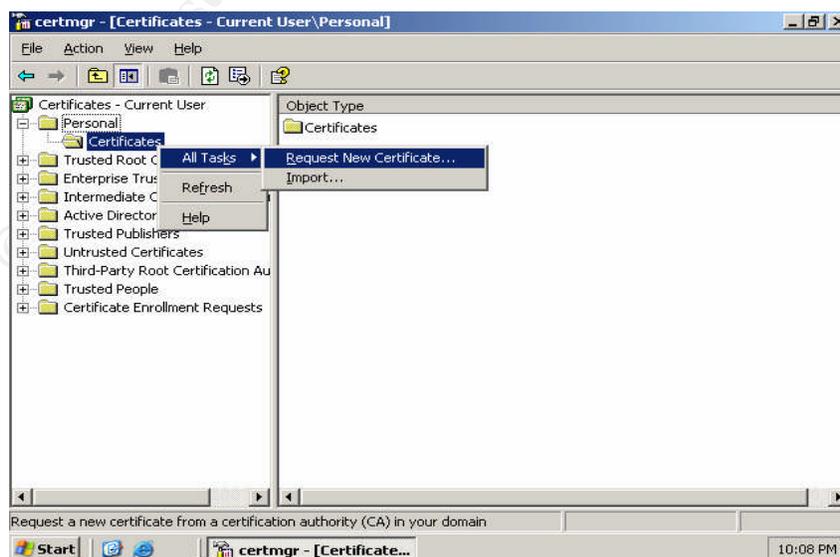
## DRA in a domain using PKI

The actual recovery of encrypted data in a domain environment is no different than in a stand-alone environment. The difference lies in the initial architecture for EFS, Group Policy, and the certificates. The CA has a built in EFS Recovery Agent template which we will use for this purpose.

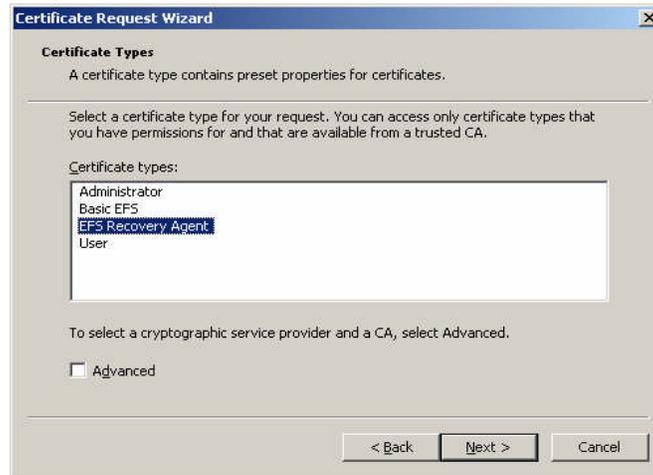
- Using Certtmpl add the DRA account to the EFS Recovery Agent template with the Read and Enroll permission



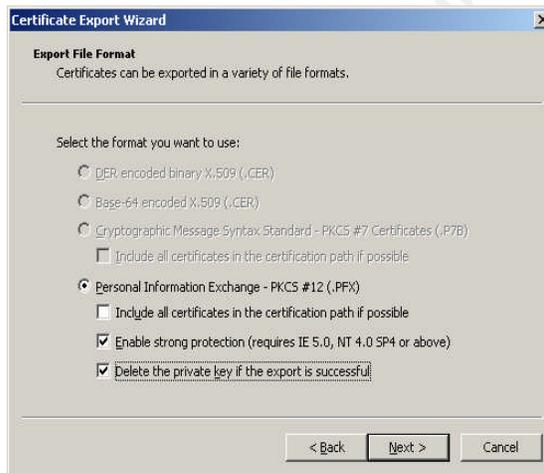
- Have the DRA account (**DRA1**) logon to a Domain Controller
- Select **Start > Run** and type **certmgr.msc**
- Expand **Personal >** right click **Certificates >** select **All Tasks >** select **Request New Certificate**



- Select **EFS Recovery Agent** > Select **Next**



- For **Friendly name**: type **EFS Recovery Agent** > select **Finish**
- In certmgr under **Personal** > **Certificates** > right click the file recovery certificate
- Select **All Tasks** > **Export** > Select **Next**
- Select **Yes, export the private key** > select **Next**
- Select **Delete the private key...** select **Next**
- Enter a complex password and select **Next**



- For **File name**: select the location and name for the export file (**A:\DRA**)



- Select **Finish**

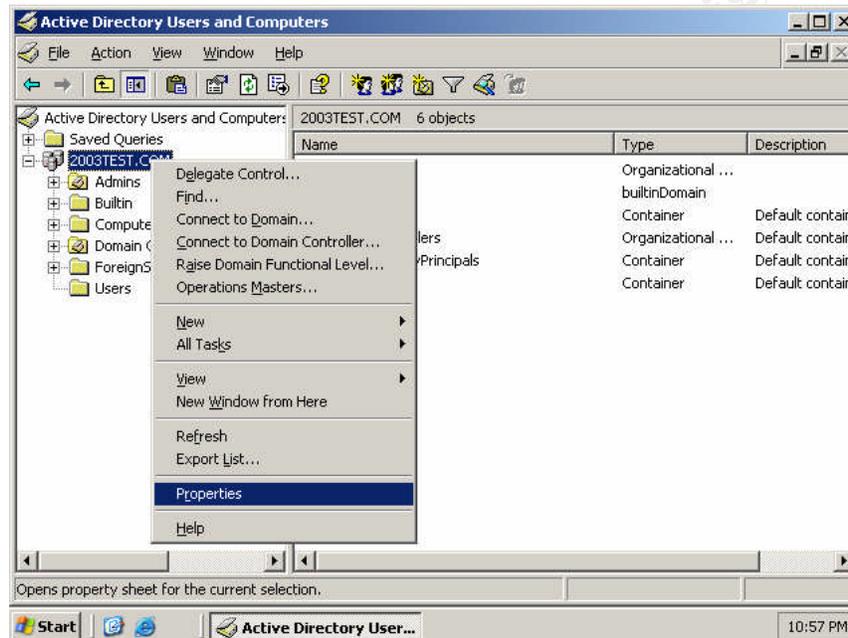


- Once the media is removed confirm the certificate is deleted from the DC
- The original and a copy of the removed media should then be stored in separate secure location

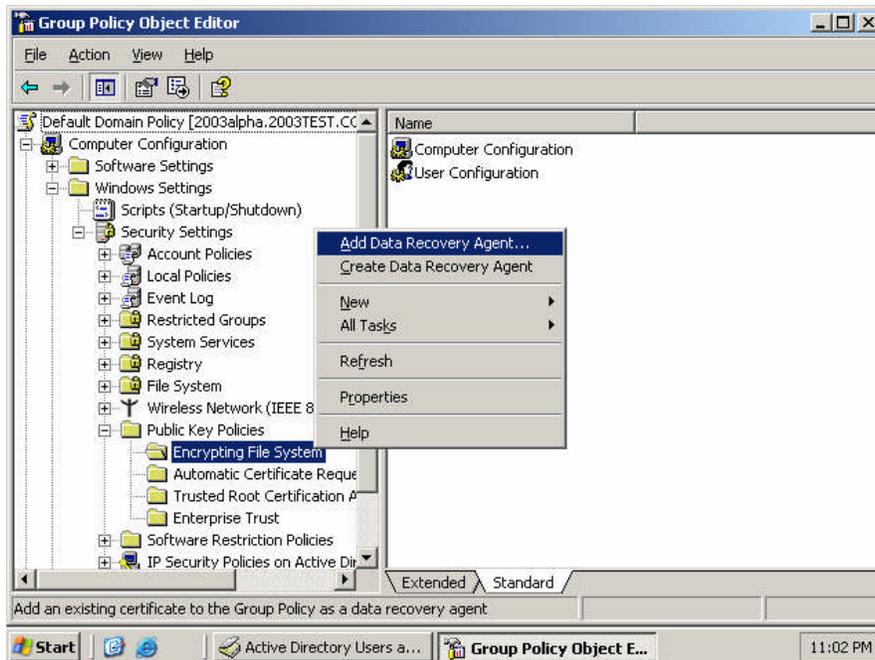
## Creating a domain DRA Policy

In order for the DRA agent to have access to recover data on a domain computer the public key must populate to the computers in the domain. This is carried out by configuring the Group Policy Object for the domain.

- Run **Active Directory Users and Computers** as a Domain Admin
- Right click the domain name and select **Properties**



- Select the **Group Policy** tab > select the highest priority GPO and select **Edit**
- Expand **Computer Configuration > Windows Settings > Security Settings > Public Key Policies** > right click **Encrypting File System**



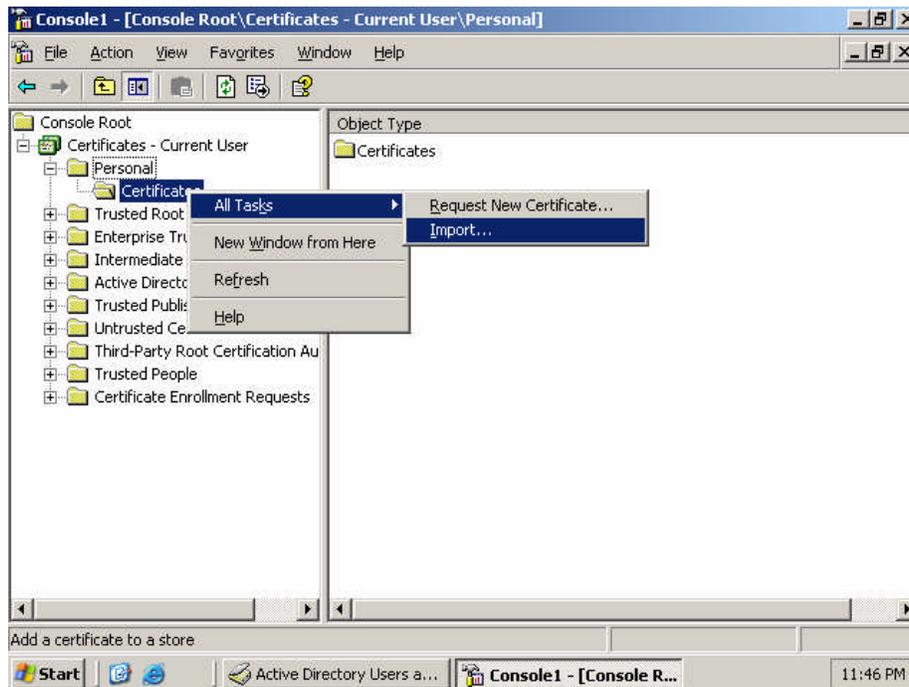
- On the **Add Recovery Agent Wizard** select **Next**
- Select **Browse Directory** and enter the recovery agent account in the name field > select **Find Now**
- Once the designated DRA account is located and entered select **Next** > **Finish**
- The DRA account will appear in the right pane
- Open the certificate to verify that this account is intended for File Recovery
- Close the certificate and the Group Policy console > select **OK** for the Domain properties and close AD

The DRA account will now apply its public key to all new EFS Certificate requests.

### Performing the Data Recovery process

The administrator that will be performing the recovery will need to have the exported DRA key.

- Logon the computer physically or use a remote access tool such as **Remote Desktop** or **SMS** with the DRA account
- Import the DRA key to the computer needing the file recovery
- Open a **MMC** console and select the **Certificates** snap-in
- Select My User Account
- Expand **Certificates – Current User > Personal > Certificates > All Tasks** > select **Import...**



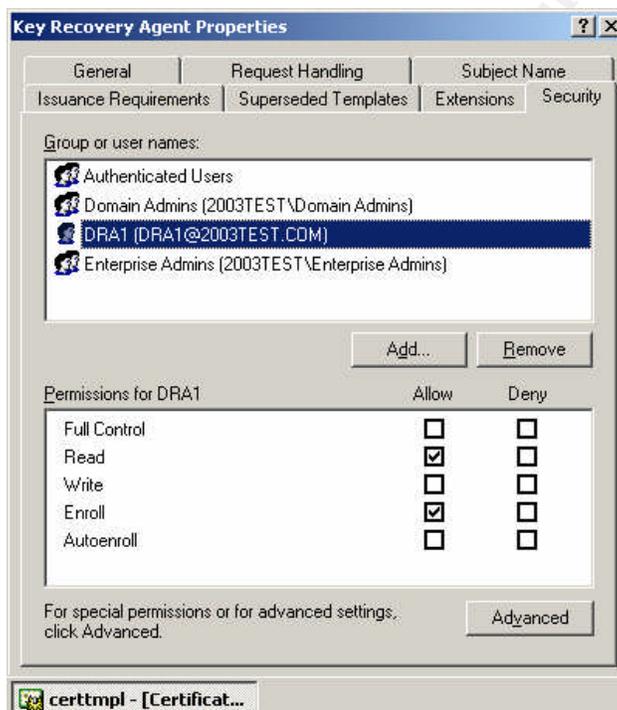
- In the **Certificate Import Wizard** select **Next**
- Select **Browse...** and locate the DRA's **.pfx** file and select **Next**
- Type the password set when the file was exported and select **Next**
- Select **Place all certificates in the following store**
- **Certificate store:** should display **Personal**
- Select **Finish**

The DRA key is now imported and you can begin to decrypt the required documents. File and folder decryption is performed in the same manner that a user would encrypt or decrypt their files.

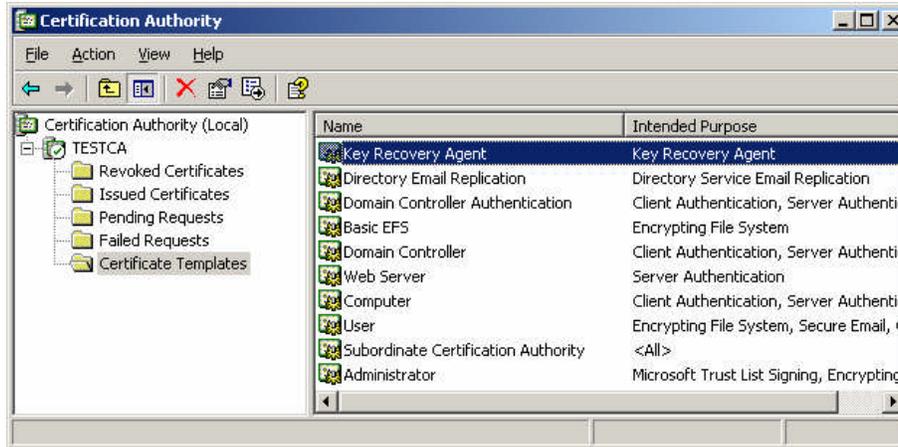
## II. Key Recovery Step-by-Step

Key recovery similar to data recovery for a domain requires that certain functions and features are in place before the recovery process can be implemented.

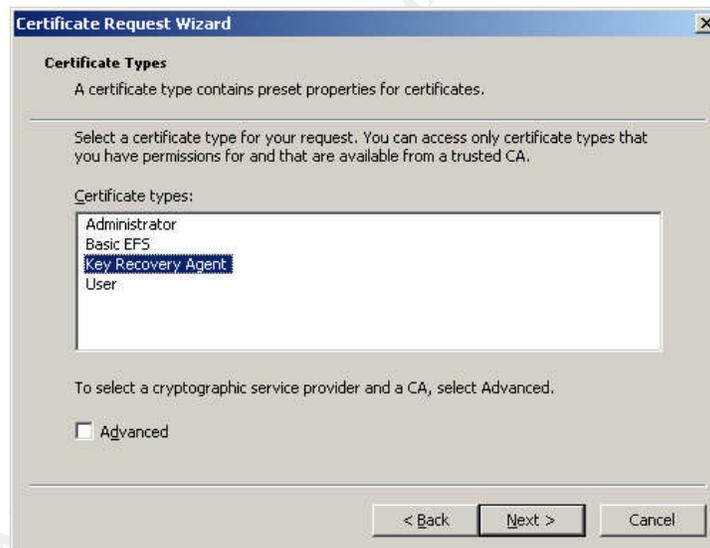
- Logon to a DC as an administrator
- Select **Start > Run** type **certtmpl.msc**
- Select **Certificate Templates** > in the details pane scroll to **Key Recovery Agent** > right click and select **Properties**
- Select the **Security** tab and add any individual account or group that will be authorized to process key recovery (**Read** and **Enroll** permissions required)



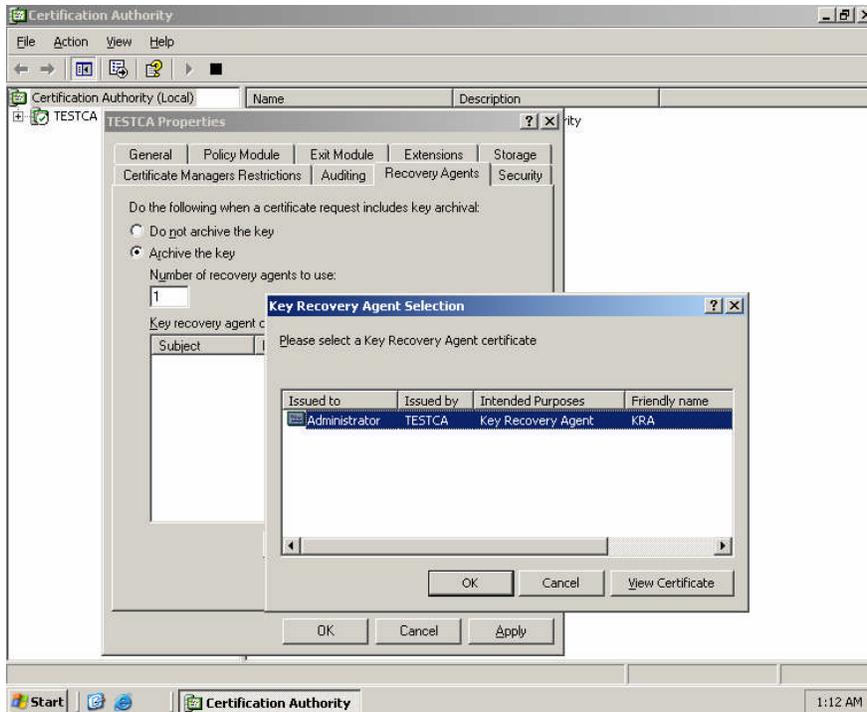
- Under the **Issuance Requirements** tab remove the check from **CA certificate manger approval**
- Open **Certification Authority** under **Administrative Tools**
- Expand the **CA** > right click **Certificate Templates** > select **New > Certificate to Issue**
- Scroll down and double-click **Key Recovery Agent** and close the open windows



- Open a **MMC** console and select the **Certificates** snap-in
- Select **My User Account**
- Expand **Certificates – Current User > Personal > Certificates > All Tasks > select Request New Certificate**
- In the **Certificate Request Wizard** select **Next**
- Select **Key Recovery Agent** and select **Next**



- For a **Friendly Name** simply put **Key Recovery** select **Next** then **Finish**
- Open **Certification Authority** under **Administrative Tools**
- Right click the **CA** > select **Properties** > select **Recovery Agents** > select **Archive the key** > select **Add**
- In the **Key Recovery Agent Selection** make sure the **KRA** is highlighted and select **OK**

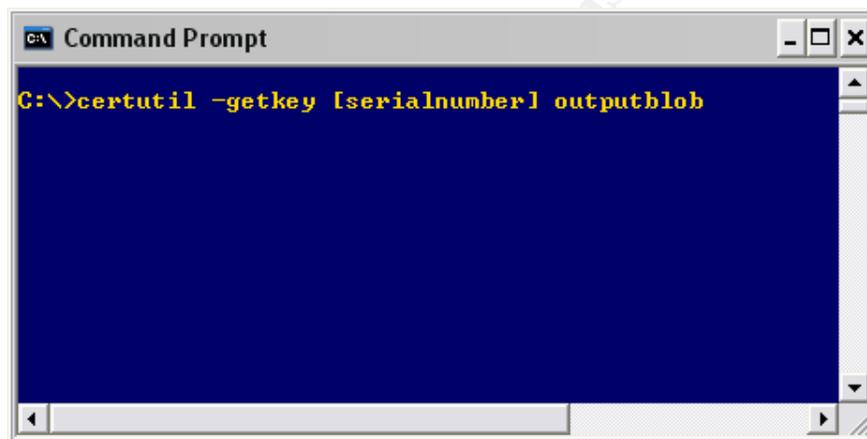


- Select **OK** again and you will be prompted to restart



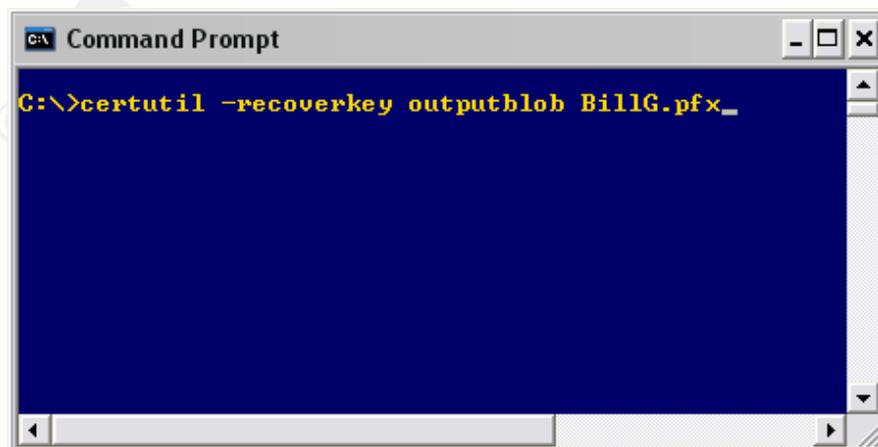
## Key Recovery Process

- Logon to the Domain Controller
  - Open **Certification Authority** under **Administrative Tools**
  - From the Menu select View and select Add/Remove Columns
  - Add Archived Key to the Displayed Columns and select OK
  - Locate the user needing their key recovered
  - Open the Archive User certificate and select the Detail tab
  - Write down the 20 character hexadecimal serial number
  - Select OK and close the CA window
- Open a Command Prompt and type the following:  
**Certutil -getkey [serialnumber] outputblob**  
note: [serialnumber retrieved from the certificate]



```
C:\>certutil -getkey [serialnumber] outputblob
```

- Type: **dir outputblob** (*BLOB is an acronym for Binary Large Object. This serves as the temporary store for extracted data and remains encrypted*)
- In the Command Prompt type:  
**Certutil -recoverkey outputblob [User].pfx**

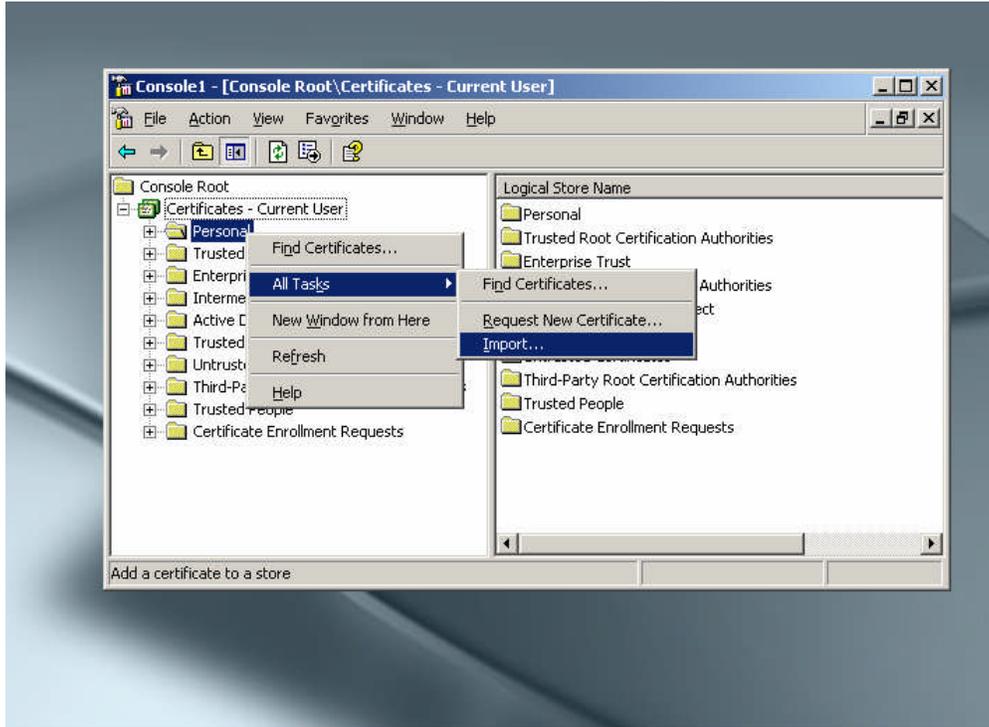


```
C:\>certutil -recoverkey outputblob BillG.pfx_
```

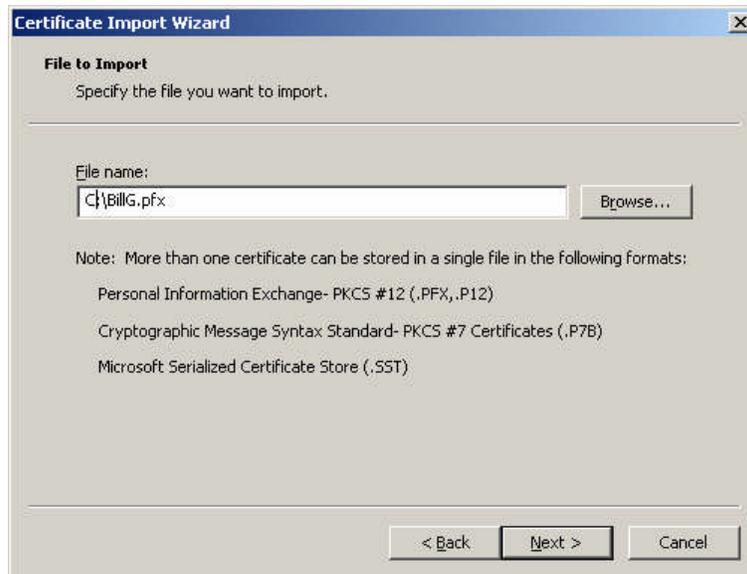
- You will be prompted for a password > type a new password > confirm password
- Type **Exit**

## Recovered Key Import Process

- Logon to the DC as the user that will need their key restored
- Open a **MMC** console and select the **Certificates** snap-in
- Select **My User Account**
- Expand **Certificates – Current User** > right-click **Personal** > select **All Tasks** > select > **Import**



- In the **Certificate Import Wizard** select **Next**
- In **File Name:** type the path to \*.pfx file (C:\BillG.pfx for our example) select **Next**



- Type the password entered during the recovery > select **Next**
- In **Certificate Store** select **Automatically select the certificate store based on the type of certificate** > select **Next** > **Finish**
- Select **Yes** > **OK**
- Expand **Personal** > select **Certificates**
- Locate users' certificate > open the **Certificate** and select the **Details** tab
- Confirm that the 20 character hexadecimal serial number matches the that which was written down in the beginning of the key recovery process
- Close all windows and logoff

In the **Windows Server 2003 Resource Kit Tools** there is a utility called:

### **Krt.exe: Certification Authority Key Recovery**

This is basically a GUI version of CERTUTIL which allows you to select options such as **Show KRA**; **Retrieve Blob**; **Decrypt Blob**; and **Recover**.

## Conclusion:

EFS is a very useful tool to secure documents and can be implanted with different operating systems as well as different environments. To successfully implement EFS requires plenty of planning and a number of well thought-out decisions to be made. XP, when configured correctly, does perform well with EFS and has a number of beneficial features over 2000. A domain environment with a 2003 CA utilizing PKI, when thoroughly planned, can prove to manage EFS efficiently.

The type of recovery selected will be based on the policies and politics of your organization. The issue with non-repudiation in some organizations may hold higher precedence than in others. Document the entire process of both data recovery and key recovery before deciding which method best suits your organizations needs as well as future growth. Make sure to only select one type of recovery method for EFS. This will eliminate and conflict or confusion as to who has recovery access.

© SANS Institute 2004, Author retains full rights.

## Resources:

Microsoft® Windows® XP Professional Resource Kit Documentation:  
[http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prork\\_overview.asp](http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prork_overview.asp)

Windows Server 2003 Resource Kit Tools  
<http://www.microsoft.com/downloads/details.aspx?FamilyID=9d467a69-57ff-4ae7-96ee-b18c4790cffd&DisplayLang=en>

The Windows Server 2003 Family Encrypting File System  
<http://www.msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/WinNETSrvr-EncryptedFileSystem.asp>

Best practices for the Encrypting File System  
<http://support.microsoft.com/default.aspx?scid=kb;en-us;223316&sd=tech>

Microsoft Knowledge Base Article – 329741  
EFS Files Appear Corrupted When You Open Them  
<http://support.microsoft.com/default.aspx?scid=kb;en-us;329741>

Microsoft Knowledge Base Article – 290260  
EFS, Credentials, and Private Keys from Certificates Are Unavailable After a Password Is Reset  
<http://support.microsoft.com/default.aspx?scid=kb;en-us;290260>

Public Key Infrastructure for Windows Server 2003  
<http://www.microsoft.com/windowsserver2003/technologies/pki/default.mspx>

Mastering Windows Server 2003  
Copyright 2003 SYBEX Inc  
Mark Minasi, Christa Anderson, Michele Beveridge, C.A. Callaghan, Lisa Justice  
ISBN: 0-7821-4130-7