Report Number: C4-006R-01

Guide to Securing Microsoft Windows 2000® Encrypting File System

Systems and Network Attack Center (SNAC)

Authors:
Graham Bucholz
Harley Parkes

Updated: January 2001
Version 1.0

National Security Agency
9800 Savage Rd. Suite 6704
Ft. Meade, MD 20755-6704

410-854-6015
securew2k@dewnet.ncsc.mil

This Page Intentionally Left Blank

Warnings

- **Do not attempt to implement any of the settings in this guide without first testing in a non-operational environment.**
- This document is only a guide containing recommended security settings. It is not meant to replace well-structured policy or sound judgment. Furthermore this guide does not address site-specific configuration issues. Care must be taken when implementing this guide to address local operational and policy concerns.
- The security changes described in this document only apply to Microsoft Windows 2000 systems and should not be applied to any other Windows 2000 versions or operating systems.
- SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN NO EVENT SHALL THE CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- This document is current as of January 2001. See Microsoft's web page http://www.microsoft.com/ for the latest changes or modifications to the Windows 2000 operating system.

This Page Intentionally Left Blank

Acknowledgements

The authors would like to acknowledge Paul Bartock and Julie Haney for their help reviewing the document.

Some parts of this document were drawn from Microsoft copyright materials with their permission.

Trademark Information

Table of Contents

Warnings ... **iii**

Acknowledgements ... **v**

Trademark Information ... **vi**

Table of Contents ... **vii**

Introduction ... **1**

Getting the Most from this Guide ... *1*

About Windows 2000 Encrypting File System Configuration Guide ... *1*

Chapter 1 Overview of Windows 2000 Encrypting File System ... **3**

Features of Windows 2000 Encrypting File System ... *3*

Cautions ... *4*

Chapter 2 EFS Security Recommendations ... **5**

General Usage Recommendations ... *5*

Application Usage Recommendations ... *6*

Temporary Application Files ... 6
User Profile Directories ... 6
Print Spooler Documents ... 6

Network Usage Recommendations ... *7*

Recovery Agent and Administrative Recommendations ... *7*

Miscellaneous Recommendations ... *10*

Chapter 3 EFS Limitations ... **11**

Deletion of Encrypted Files ... *11*

Physical Access Boot Attacks ... *11*

Appendix A References ... **13**

13

Table of Figures

Figure 1 – Setting a Data Recovery Agent via Group Policy 8
Figure 2 – EFS Certificate 9

Introduction

The Windows 2000 Encrypting File System (EFS) is an enhancement available only to the Windows 2000 version of the NTFS v.5 file system. It allows users to encrypt/decrypt files on the fly to protect sensitive data from unauthorized access, while still providing for data recovery by the administrator if a user loses/deletes their keys or leaves the organization and their profile is removed before the files are decrypted.

WARNING: This guide does not address security issues for the Microsoft Windows 2000 operating system that are not specifically related to the Encrypting File System.

This document is intended for Windows 2000 network administrators, but should be read by anyone involved or interested in Windows 2000 or network security. It is assumed that the reader has administrative knowledge of EFS. See the list of references, especially the *Step-by-Step Guide to Encrypting File System (EFS)* Microsoft white paper at http://www.microsoft.com/windows2000/techinfo/planning/security/efssteps.asp for further information on EFS basics.

Getting the Most from this Guide

The following list contains suggestions to successfully secure the Windows 2000 Encrypting File System according to this guide:

WARNING: This list does not address site-specific issues and every setting in this book should be tested on a non-operational network.

- Read the guide in its entirety. Omitting or deleting steps can potentially lead to an unstable system and/or network that will require reconfiguration and reinstallation of software.
- Perform pre-configuration recommendations:
  - Perform a complete backup of your system before implementing any of the recommendations in this guide
- Follow the security settings that are appropriate for your environment.

About Windows 2000 Encrypting File System Configuration Guide

This document consists of the following chapters:

Chapter 1, "Overview of Windows 2000 Encrypting File System," contains a brief description of the Encrypting File System, its features, and some precautions.

Chapter 2, "Guide to Windows 2000 Encrypting File System," contains security recommendations on how to use the Encrypting File System.

Appendix A, "References," contains a list of resources cited.

This Page Intentionally Left Blank

Chapter

1

Overview of Windows 2000 Encrypting File System

The Encrypting File System (EFS) is an integral part of the new NTFS file system.Support for EFS is built directly into the kernel of Windows 2000, so is automatically installed with every Windows 2000 system. EFS uses a combination of symmetric key and public key encryption. A random file encryption key (FEK) is generated for each file when it is selected for encryption. Using the FEK, the file is encrypted using DESX (128 bit strength for US versions of Windows 2000, 40 bits for international versions, although this might have changed since publication). Then the FEK is encrypted with the user's public key from their file encryption certificate (automatically created when EFS is first used) and stored with the file. Also, the FEK is encrypted with the public key of the assigned recovery agent(s) and also stored with the file. Decryption uses the user's or recovery agent's private key to get the FEK and then to restore the file to a decrypted state.

Features of Windows 2000 Encrypting File System

EFS provides many features including:

- Allows for protection of sensitive files and folders.
- Encryption and decryption is done on the fly with no need for the user to enter a password (only the encrypting user can decrypt).
- Encrypting a directory/folder encrypts all subsequent files created in that location.
- Moving an encrypted file out of the encrypted folder it was created in retains its encryption (on Windows 2000 NTFS volumes only!)
- EFS will encrypt temporary files created by applications if those applications are EFS-aware (or those which copy the permissions of the original file to the temporary file).
- EFS does not cache any of the keys onto the hard disk.
- Prevents the encryption of required system files and folders (anything in the %SystemRoot% path or that has the System attribute set).
- A command line utility, CIPHER, is available for batch file use.

Cautions

Due to both the newness of EFS, as well as the way it was designed, there are a number of issues with regards to what EFS can and cannot accomplish that should be understood before the decision to use the Encrypting File System is made. These limitations are listed below.

- EFS requires at least one recovery agent. Without one, EFS is turned off and cannot be used.
- Only works on the Windows 2000 version of NTFS.
- Copying encrypted files to any other file system type (FAT, FAT32, earlier versions of NTFS) using normal commands (copy, move, etc.) will save the file in decrypted form.
- EFS does not support encrypting for multiple users (except for multiple recovery agents).
- Opening encrypted files over the network decrypts the file on the remote side and sends decrypted data over the network.
- Remote encryption is not enabled by default.
- Folder encryption does not prevent the listing of files contained within, nor does encryption prevent deletion of folders or files.
- Does not provide for a secure boot/fully encrypted system.
- Due to the way NTFS does compression, compression and encryption are mutually exclusive for the same file.
- Encryption cannot be used on files on a cluster of servers.

Chapter

2

EFS Security Recommendations

General Usage Recommendations

The following suggestions deal with the general usage of EFS.

Computer Usage

EFS is recommended for use on any portable computer that contains sensitive data and has the possibility to be removed from a secured area to prevent loss if it is stolen or lost.

EFS should also be used on any public computers that allow unrestricted access, preventing each user from accessing other users' data.

Passwords and EFS

EFS encryption and decryption does not require a separate password from the user's normal logon under the assumption that only the user can log on as himself and use his certificate to encrypt/decrypt their data. Therefore, the security of EFS is dependent on the strength of the site password scheme and the user's particular password. A strong password policy is required if EFS will be used.

Refer to the *Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Toolset* for suggested password policies.

Disabling EFS

Some sites may decide not to use the Encrypting File System. To prevent loss from unintended or malicious encryption of important data, EFS should be disabled when not in use. This can be done at any level of Group Policy, allowing fine-grained control over which organization, department, or individual computer is allowed or not allowed to use EFS.

NOTE: Changes to Group Policy take 90 minutes by default. Unless the time period is changed or the policy is forced down to all the client computers, users will be able to encrypt new files and folders until the policy takes effect.

Application Usage Recommendations

Because the Encrypted File System is a new feature of Windows 2000, not all Windows applications support it. Many applications do not realize they are dealing with encrypted files and therefore perform a number of insecure operations on them. Below are recommendations on using EFS securely with current applications.

Avoiding Plaintext Copies of Encrypted Files

In the event of a system crash or failure, EFS implements a crash recovery scheme that creates a plaintext backup of a file in the process of being encrypted or decrypted. Once a file is successfully encrypted or decrypted, the backup file is deleted. However, remnants of the plaintext version may remain on the hard drive until those disk blocks are overwritten. This issue is known by Microsoft and currently is being investigated further for possible improvement. In the interim, it is recommended that all encryption be set at the directory level instead of individual files. In other words, start with an empty, encrypted directory and create files in that directory as needed. The files will automatically be encrypted and no plaintext copy will be created.

Temporary Application Files

Many application programs create temporary files in the directory of the opened file. Unless the application specifically states that it is EFS aware and encrypts temporary files, these temporary files are not encrypted by default, even if the original document was. Therefore, to prevent unencrypted temporary files, encryption should be set at the directory level instead of individual files. Then, any file created in that directory (both original and temporary) will be encrypted by default.

User Profile Directories

New Microsoft Office documents are created by default in the %UserProfile%\MyDocuments directory.

It is recommended to encrypt the %UserProfile%\MyDocuments directory path for each user in order to automatically encrypt any new Microsoft Office documents created.

In addition to this directory, any user temp directories should be encrypted as well, to encrypt any sensitive files created with applications that use them. These directories vary by system, application, and user, but common locations are c:\Temp and %UserProfile%\LocalSettings\Temp, among others.

NOTE: %UserProfile% is usually C:\Documents and Settings*user_name* where *user_name* is the user's account logon name.

Print Spooler Documents

Documents waiting to be printed in the system print spooler are stored in the clear. Since EFS will not allow the encryption of any system directories (anything in the %SystemRoot% path, usually C:\Winnt and below), either the use of print spooling must be evaluated for the risk of compromise, or (if possible) print spooling should be set up outside of the %SystemRoot% path where EFS can be set on the spool directory.

Data Backups

Most data backup programs are not yet aware of EFS encrypted files back files up in the clear. Ensure that the data backup tool being used is verified to work with EFS.

Currently only Microsoft's BACKUP utility included with Windows 2000 is able to back up EFS encrypted files without decrypting them in the process.

Copying and Moving Encrypted Files

Copying or moving a file with either the Windows graphical interface or command line utilities will decrypt a file if it is transferred to a file system that does not support EFS (such as FAT16, the standard floppy disk file system). It is recommended to not use these commands to transfer files to disks to transport them to other machines. Instead use the BACKUP utility to back up the file (which will keep it encrypted) if floppies or other non-NTFS formatted media need to be used.

Network Usage Recommendations

The Encrypting File System is only meant to secure files on the local machine. It does not provide network support.

Transferring files across a network connection transparently decrypts the files and sends the unencrypted file over the to the remote computer. If network transfer capabilities are required, a secure networking protocol such as SSL, TLS, PPTP, or L2TP/IPSec should be used to encode the connection.

By default, the Encrypting File System does not allow a user to encrypt remote files. Therefore a transferred file is in unencrypted format on the remote system. If it is required to have complete end-to-end encryption of a file, the domain administrator can enable users to remotely encrypt their files on a computer-by-computer basis by making the remote computer "Trusted for Delegation." Combined with one of the secure protocols above, this allows end-to-end encrypted transport of the file for that user.

NOTE: Making a computer trusted for delegation may have other, unintended effects which need to be looked into before assigning this right.

Recovery Agent and Administrative Recommendations

The Encrypting File System requires there to be some account that is the Data Recovery Agent for all encrypted files at any particular level of organization. This can be set at any level where group policy can be set. Since the Data Recovery agent has the ability to decrypt any file it is responsible for, this account should be subject to a number of recommendations.

Delegating a New Recovery Agent

By default, the Recovery Agent is the Domain Administrator account for a domain modeled network, or the local Administrator account for workgroup and standalone machines. For a larger segregation of power and better accountability, the Recovery Agent right should be delegated to a special account created for just this purpose.

Assignment of a recovery agent can be done at any Group Policy level (Site, Domain, OU), allowing localized control over the emergency decryption of files. **Figure 1** shows

the location of Data Recovery Agents in a Group Policy Object. See the *Guide to Securing Windows 2000 Group Policy* for more information on accessing Group Policy.

NOTE: The Data Recovery right is not cumulative. The nearest Group Policy to the user object sets that user's Data Recovery Agent.

Adding a new recovery agent is a multi-step procedure that needs to be completed by both the Administrator and the Recovery Agent. See the *Step-by-step Guide to Encrypting File System* Microsoft white paper available at http://www.microsoft.com/windows2000/techinfo/planning/security/efssteps.asp for detailed information on setting a Recovery Agent.

NOTE: Before adding a new agent, there must be an Enterprise Certificate Authority to request new recovery agent certificates.

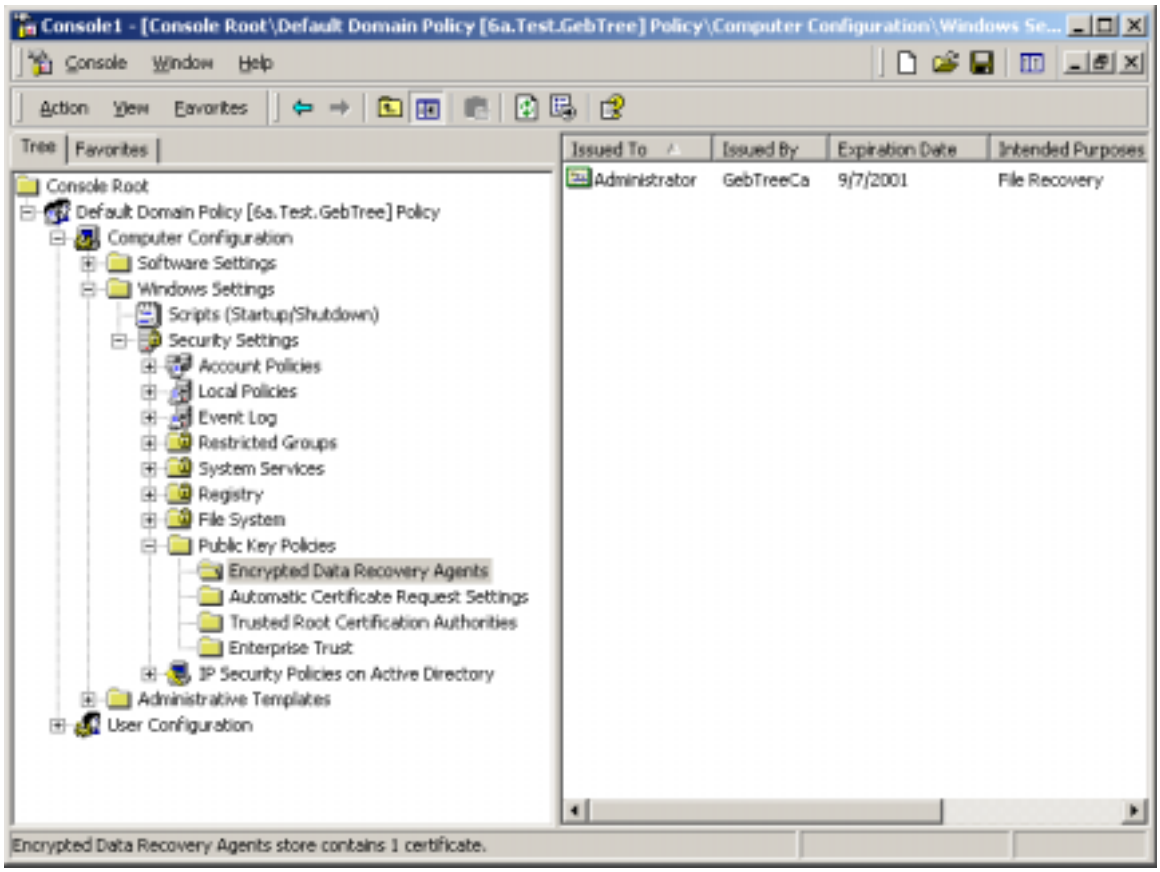

Figure 1 – Setting a Data Recovery Agent via Group Policy

Exporting the EFS Certificate and Private Key

Compromise of the data recovery account could be serious. Therefore, in addition to creating a separate data recovery agent account, It is recommended that the recovery agent's decryption certificate and private key should be exported to a floppy disk and stored in a secure place.

The certificate and private key are not needed in day-to-day usage of EFS, only for emergency recovery. Also, when exporting the certificate and key, there is an option to

protect them with a strong password, preventing unauthorized reinstallation of them. Use of the strong password feature for certificate and key protection is recommended.

To export the EFS certificate and key, log on as the data recovery agent account and perform the following steps:

- **Start→Run→mmc.exe**
- In the **Console** menu, select **Add/Remove Snap-in**
- Click **Add**
- Select the **Certificates** snap-in
- When prompted, select the **My user account** radio button
- Click **Add**
- Click **Close**
- Click **OK**
- Expand the list of certificates by clicking on the plus sign to the left of the **Certificates** node
- Expand the **Personal** entry by clicking on the plus to the left of it
- Click on the **Certificates** folder under **Personal**
- Scroll the right panel over until the **Intended Purpose** column can be seen
- Select the certificate whose intended purpose is **Encrypting File System** as shown in **Figure 2**

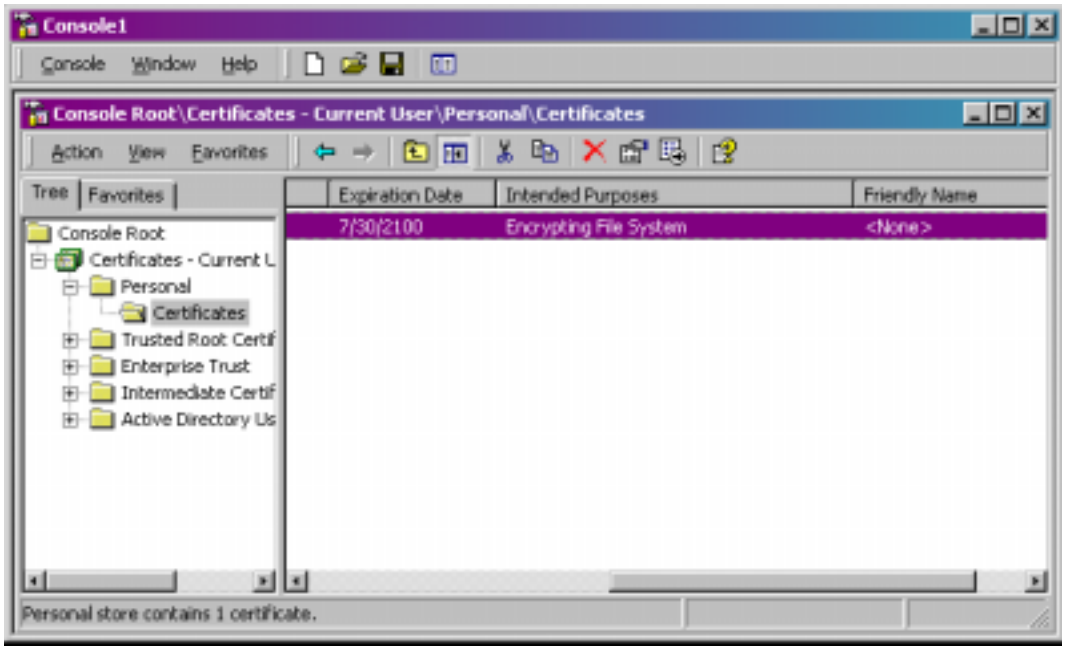

Figure 2 – EFS Certificate

- Right click on that certificate and choose **All Tasks→Export**. This will launch the **Certificate Export Wizard**
- Click **Next**

- On the **Export Private Key** screen, select **Yes, export the private key**
- Click **Next**
- In the **Export File Format** screen, check the **Enable Strong Protection** box. If you are completely transferring this key to another location or storing it off-site, also check the **Delete the private key if export is successful** checkbox.
- Click **Next**
- Enter a strong password and confirm it. This will protect the certificate from unauthorized users installing it and then being able to decrypt any EFS-protected files encrypted with it.
- Click **Next**
- In the **File to Export** screen, choose a descriptive name and location so that there won't be any mistake as to what or where this file is. A suggestion would be to use the user's name, the date it is exported, and EFS in the name somehow to label the file.
- Click **Next**
- A summary page comes up next. Review it and make sure all the info is correct. If it isn't click back until you get to the right page and fix it. Once you are sure it is correct, click **Finish**.
- When you are done, you will get a message saying it was completed successfully. Click **OK** to return to the Microsoft Management console.

There are situations that do not allow for the above procedure of exporting the certificate and private key or require additional security measures. In those situations, setting up a dedicated data recovery terminal in a secure location and restricting the ability of the recovery agent to log on to just that terminal can be another security precaution taken.

Miscellaneous Recommendations

The certificates and keys used for encryption and decryption in EFS are located on the local machine of the user. Special care must be taken if the users must change machines (such as for repair of their original computer).

A user must transfer their certificate and private key to any new computer that they are going to use to open already encrypted files. If the original machine is to be given to another user or otherwise reused, the certificate should be deleted from it before going to anyone else. This will prevent anyone from acquiring the former user's decryption key.

NOTE: In addition to deleting the key, the machine must be rebooted to clear the key stored in memory.

If the user's machine is to be upgraded, such as adding it to a domain from workgroup mode or switching domains, it is recommended to decrypt all data first or make a backup of the user's private key and certificate as a security precaution. Changing the user's account could result in a loss of their decryption certificate, preventing the decryption of any files encrypted by that user previously.

Chapter

3

EFS Limitations

The purpose of this chapter is to address common issues and concerns over limitations in the Encrypting File System (EFS) within Windows 2000. The issues relate to the ability to access encrypted files on a local drive by booting the operating system from a floppy disk and the ability to delete files that another user has encrypted. It is worth noting that while limitations do exist within the EFS, it remains another means of providing a greater level of protection for critical data files and can be used as part of a layered approach toward system security.

Deletion of Encrypted Files

Issue: A Windows 2000 user can delete files encrypted by another user on the system.

Discussion: EFS does not provide any protection against the renaming or deletion of files in a folder. **If a user has full access (Full Control) permissions on a directory or files within the directory, they can delete or rename files encrypted by another user.** However, they cannot copy or move files encrypted by another user to a different directory. This protects against the possibility of a user copying an encrypted file to a file system, such as FAT, that does not support EFS.

Mitigation Technique: Use of proper NTFS file and directory permissions will prevent a user from modifying or deleting files created by another user. This includes files encrypted by other users.

Physical Access Boot Attacks

Issue: With physical access to a system, a user can boot the system from floppy disks and use O&O BlueCon or another similar tool to access files encrypted by users.

Discussion: If as attacker has physical access to the computer and can boot the computer from a floppy disk, it is possible to use tools such as O&O BlueCon or other similar password editing tools to modify the administrator password and logon as the administrator. Once logged on as administrator the attacker can then read files encrypted by the administrator. As the administrator, the attacker can also change the password for local user accounts and access files encrypted by those users as well.

Mitigation Techniques: Several alternatives exist that can help mitigate the risk from this type of attack however, if an individual has physical access to the computer for an extended period of time, it may still be possible to recover encrypted files from the system. The following are methods that can be used to mitigate the above risk.

- Enable a BIOS boot password and disable booting from floppy or CD within the BIOS. This can help prevent an individual from booting into another operating system and accessing NTFS files.
- Consider using the SYSKEY utility to further protect the SAM database. Although some password editing tools are available that may even overwrite SYSKEY protected databases, these tools are not completely reliable and the use of SYSKEY greatly increases the complexity of the attack.
- Export the recovery certificates and keys and store them on removable media such as a floppy disk or CD ROM. Once exported, delete the recovery keys from the local system. The exported keys should be physically protected in a separate area from the actual system. This procedure is recommended in Microsoft's EFS documentation and will help prevent an attacker from using the recovery keys to access other local users encrypted files.
- Consider making the machine part of a domain and, if required for portable computers, allow a limited number of cached credentials. As part of a domain the recovery agent is, by default, the domain administrator and the recovery keys will be located on the domain controller. This will prevent an attacker logged in as a local administrator from recovering files encrypted by domain users. Also as a local administrator, the attacker would not be able to change passwords for domain accounts and therefore could not log in as another user and access their encrypted files.

Appendix

A

References

Designing a Secure Microsoft Windows© 2000 Network. Microsoft© Official Curriculum, class 2150A. Microsoft Corporation, copyright 2000.

"Encrypting File System for Windows© 2000," Microsoft white paper. http://www.microsoft.com/windows2000/techinfo/howitworks/security/encrypt.asp

McLean, Ian. Windows 2000 Security Little Black Book, Coriolis Group, 2000.

"Step-by-Step Guide to Encrypting File System (EFS)," Microsoft, http://www.microsoft.com/windows2000/techinfo/planning/security/efssteps.asp

"Windows© 2000 Certificate Services," Microsoft white paper. http://www.microsoft.com/windows2000/techinfo/howitworks/security/windows2000csoverview.asp.