Global Information Assurance Certification Paper

Yu-Min (Phillip) Hsieh

Configure Encrypted File System on a Microsoft Windows Server Cluster for Secured Data Storage

Configure Encrypted File System on a Microsoft Windows Server Cluster for Secured Data Storage

Background - Every organization has sensitive data that should be protected against unauthorized access. Normally, this data is protected by logon authentications and file permissions. However, these mechanisms do not prevent unauthorized access if someone gains physical access of the workstation or when data is placed in a shared storage location.

In the first scenario, another operating system can be installed to bypass the existing system's security. In the second scenario, members in local administrator groups of the server hosting the shared storage can always take ownership of the protected files and alter file permissions.

When storing confidential data such as those distributed in the Human Resources department in a local computer, users may encrypt them using the Encrypted File System (EFS) feature provided by Microsoft Windows 2000/XP/2003. This is a relatively simple task. However, in an environment where users' data is stored on a central clustered file server, applying EFS can be tricky. This report discusses different considerations in designing EFS on a Microsoft Windows 2000 Server Cluster. It also provides descriptions and solutions on encountered difficulties.

The following technologies provided by Windows 2000/XP/2003 are utilized in this project:

- Public Key Infrastructure – to generate encryption keys for EFS.
- IPSEC – to provide secure communications between servers and workstations.
- Group Policy – to efficiently configure and deploy policy settings for the above.

Detailed descriptions of these technologies can be found in this book: "Windows 2000 Server Distributed System Guide" by Microsoft Corporation, chapters 13, 14, 15, 16, and 22.

Network Environment - The following lists the network environment this report is based on:

Logical structure

- A network built upon Microsoft Windows 2000 Active Directory Architecture in native mode with one Forest Root Domain (Microsoft, p.20-23).
- User and computer accounts for different departments are placed into separate Organization Units (Microsoft, p. 48, 1193) under the root domain node.

Physical structure

- One site (Microsoft, p. 162, 326-330) with many high-speed, switched IP subnets.
- Three domain controllers reside in two different subnets.
- Active Directory Integrated DNS service (Microsoft, p. 15-18) provided by each domain controller.
- The domain includes only Windows 2000 and Windows XP workstations.
- The domain includes only Windows 2000 and Windows 2003 servers.

Administrative model and general configurations

- Managerial responsibilities for the Organization Units are delegated to departmental administrators (Microsoft, p. 49-50, 656-661).
- Folder Redirection and Offline Files (Microsoft, p. 1196-1197, 1211-1214) are deployed via Group Policies.
- User data is stored on a Microsoft server cluster consisting of two cluster nodes (Microsoft, "Guide to Creating and Configuring a Server Cluster Under Windows Server 2003", Version 1.0, 11/21/2003).
- An Enterprise Root Certification Authority (Microsoft, p. 813-815) is configured on one of the domain controllers.
- All domain controllers and servers are configured via group policies so that only Administrators and Backup Operators groups have logon locally user right assignment.

Design Considerations

Basic EFS - Microsoft Windows 2000/2003/XP uses an asymmetrical key pair to encrypt (with the public key) and decrypt (with the private key) a bulk symmetrical encryption key. The bulk encryption key is actually used to encrypt and decrypt files and folders. Using a symmetrical encryption key provides enhanced performance, commonly 100 to 1,000 times faster compared to using an asymmetrical key (Microsoft, p. 843).

Normally, on a standalone workstation, a user can encrypt data simply by using the following three steps:

- Right-click the file/folder
- Select [Advanced…]
- Check [Encrypt contents to secure data] option.

When there is no existing public key infrastructure for an encryption key, the Operating System will automatically generate one for both the user and the local administrator to complete this task.

The local administrator's private key is used as the recovery agent key to recover encrypted data, in case the private key for the primary user is lost or damaged.

The recovery agent account (key) is required when using Windows 2000 Encrypted File System (Microsoft, p. 831).

In a Windows 2000 Active Directory domain environment, however, the default recovery agent account becomes the domain's administrator account when there is no EFS recovery policy defined. This recovery agent account can also be modified using EFS recovery policy, so multiple accounts can be assigned for this purpose. An EFS recovery policy configured with no recovery agent simply disables the EFS for the entire domain (Microsoft, p. 831).

The project described in this report has an EFS recovery policy with three recovery agent accounts defined. Each recovery agent account is configured on a specific domain controller by logging on to the domain controller and applying for an EFS Recovery Agent certificate via the certificate service website.

Afterwards, the certificate and keys were exported to a secured location and then destroyed at the domain controllers.

Remote EFS – In the case of encrypting files on a server, the following are essential settings (Microsoft, "Remote EFS Operations in a File Share Environment". See web link in Reference 5):

1. Correct share permission settings.
2. Correct NTFS permission settings.
3. User must either have a local profile on the server or a roaming profile. Otherwise, EFS creates a local profile for the user. If the remote computer is a server in a cluster, the user must have a roaming profile.
4. To encrypt a file, the user must have an EFS certificate or one will be obtained either from a certification authority or self-signed.
5. To decrypt a file, the user's profile must have the private key associated with the EFS certificate used during encryption.
6. EFS must impersonate the user to obtain the public or private keys. In order to do so, the following conditions should be met:
    a. The server must be a domain member that uses Kerberos authentication.
    b. The computer must be trusted for delegation.
    c. The user must log on with a domain account that can be delegated.

Among these requirements, item 3 and item 6 need special attentions:

User Profiles – In the network environment for this project, users are restricted from an interactive logon to all servers and domain controllers by configuring the group policies to assign only local [Administrators and Backup Operators] groups in the ***Log on locally*** User Rights policy (Group Policy | Computer Configuration | Windows Settings | Security Settings | Local Policies | User Rights | Log on locally).

In this configuration, a network connection to the server's share generates a temporary user profile, which cannot be granted an EFS certificate. The attempt to encrypt a file via such connection generates an Application Event error under source name Winlogon (EventID 1010).

EFS does not require a user to log on to the server cluster interactively, however, it does need a permanent user profile in the servers so the encryption keys can be stored. The minimum requirement for EFS to create a permanent user profile is to let the user have the *Log on locally* user right. After EFS creates the permanent user profile, the user or group can be removed from the *Log on locally* group policy.

When EFS generates a permanent user profile, it can be a local profile or a roaming profile. The reason that a server cluster requires a roaming profile is because the file share resource can be owned by any one of the cluster nodes. A roaming profile stores the encryption keys in the domain controllers (Microsoft, p. 853) so that wherever the profile is used, the encrypt/decrypt keys remain the same. By contrast, a local profile stores the encryption key in the local RSA folder, %userprofile%\Application Data\Microsoft\Crypto\RSA (Microsoft, p. 853) and each local profile has a unique set of encryption keys in using EFS. An interesting fact is that instead of storing the encryption keys in the server location of the roaming profile, Microsoft decided to store them in the domain controllers.

In the beginning stage of this project, roaming profiles were not used based on a managerial decision. An attempt to create a local profile at each server in the server cluster and import the certificate and private key from the other cluster node was a successful operation. In this case, each cluster node has two sets of EFS certificates and private keys.

The computer must be trusted for delegation – In order to perform remote encryption, the cluster server (EFS) must impersonate the user to request an EFS certificate from a Certification Authority. In Microsoft Windows environment, this can be achieved only when the domain environment supports Kerberos authentication such as Windows 2000 and Server 2003. This is because the Kerberos authentication protocol supports the ***delegation of authentication*** and allows a service ticket to be forwarded (Microsoft, p. 656-657, 661).

At first glance, this requirement can be easily fulfilled by opening the property page of the server's Active Directory object and then check [Trust computer for delegation] option. But in reality, with this option set, and the user's profile correctly configured; an attempt to encrypt a file after connecting to the cluster file share failed with a "keyset does not exist" error message. The failure resulted because the virtual server name does not have the option to be configured to use Kerberos authentication in a Windows 2000 server cluster.

There are several new enhancements added in Windows XP and Server 2003 for EFS. Among those, the following is worthy of special consideration for the above Kerberos delegation of authentication issue:

- Cluster Virtual Server names can be accessed via Kerberos authentication protocol in Windows Server 2003 (Microsoft, "Applying Kerberos Authentication in a Clustered Environment". See web link in Reference 4)

Although you may use the cluster administration tool in Windows Server 2003 to configure the virtual server name and enable Kerberos Authentication on a Windows 2000 server cluster, remote encryption/decryption still does not work and gives the *key set does not exist* error. All member nodes in the server cluster have to be Windows Server 2003 in order for EFS to work with a virtual server name.

Printing Encrypted Documents – Typically, a server cluster is also used to provide high-availability print services. The documents are saved as spool files and then printed based on their priority in the print queue. As they stay in the print server waiting to be printed, however, they are in plain text format. This is an insecure storage. There are four methods to protect these documents:

- Print directly to the printer – this method defeats the main purpose of using a print server.
- Save spool files to an encrypted folder – this presents technical difficulties because the spool folder can only be encrypted by one user account. This is true even with the new EFS features in Windows XP and Server 20003. See details of the related new feature later in the section, *Public Shares*.
- Print to a local printer – this method again takes away the advantages of using a print server.
- Use NTFS permission and audit [Take Ownership] object access – since the spool files are deleted right after they are printed; configuring the spool folder with a tight DACL should provide satisfactory result. The detail NTFS and audit settings are described in **Final Design | *NTFS Permissions*** section.

E-Mail – e-mails are just like other important data which should always be saved on a server so consistent backup can be carried out. Since e-mails may contain sensitive information as well, EFS should be used to protect them whenever possible. In an environment where there is no strict policy to enforce the type of e-mail client users may choose, providing confidentiality of e-mails in transit and also the authorized access of the e-mail folder becomes difficult. The following are general guidelines in handling e-mails containing confidential information:

- Microsoft Outlook PST files can be protected by EFS, but data and program files in other e-mail clients such as Eudora may not work properly when encrypted.

- Always use POP protocol to download e-mails from the mail server and do not use [Leave mail on server] option.
- Always configure e-mail client to use SSL encryption to communicate from/to the mail servers.
- Since the PST file cannot be made available offline, users who use notebook computers and travel to places where Internet access is limited will not be able to review existing e-mails. There are at least the following three methods to resolve this issue: (1) If the Outlook PST file is stored in an encrypted folder in the server cluster, then make a copy of it to a local folder and reconfigure outlook to point to that file before going on the trip; (2) Store the PST file locally, in encrypted form if necessary, and then create a scheduled task to copy the PST file to the server cluster on a nightly basis (see detail in **Final Design | *E-mail*** section); and (3) use Microsoft Exchange Server, which provides an Offline e-mail feature using a local OST file.
- When sending confidential documents, encrypt them and use digital signatures. The recipient should use the same e-mail client with correct security settings to avoid incompatibility issues. Microsoft Windows 2000 PKI and PGP both provide good results.
- If encrypting e-mail attachments is not practical because the document is too big or because the sender and the recipient are using incompatible e-mail clients, then deposit the encrypted documents into a public share for the recipient to retrieve. See detail in the next section.

Public Shares – In creating public shares for users who may use EFS on a server cluster, there are two basic points to consider:

- When the original document is encrypted, copying it to a public share in EFS enabled server cluster makes the document encrypted in the target folder, which leaves it unreadable for intended viewers.
- Asking users to decrypt the file before placing it into the public share creates three difficulties: (1) the NTFS security setting for the public share becomes multi-layered and complicated because it must be based on the recipients; (2) confidential documents are not encrypted and thus are subject to unauthorized access; and (3) the user may forget to decrypt the file before placing it into the public folder.

In the Windows 2000 environment, there is no way to resolve these difficulties. However, a new feature added in XP and Windows Server 2003 can be very useful in solving the above issue:

- Additional users can be authorized to access encrypted files (Microsoft, "Microsoft Windows XP – Overview of EFS". See web link in Reference 3). In essence, Microsoft improved EFS in XP and Server 2003 by allowing the owner of an encrypted document to add additional user accounts in the data Decryption Field, DDF (Microsoft, p. 844). This explains the limitation of this feature: one can only assign this privilege to an individual user account but

not to a group account. It also brings about another user support issue because any user account being added must meet the requirements described in the *Remote EFS* section. Another limitation of this feature is that this only applies at the file level and not at the folder level.

The following lists the steps to add other users to the DDF:

- Right-click the file/folder, select *Properties.*
- Click *Advanced* and then click *Details* button.
- Click *Add* and then select the user's certificate from list.
- If the user's certificate is not listed, then click *Find User…* button.
- Find the user from Active Directory. It may result in an error message: *No appropriate certificates correspond to the selected user*. In this case, make sure the user meets the perquisite described in the *Remote EFS* section.

When EFS needs to be applied to public shares on a server where Remote EFS is not desirable, Windows XP and Server 2003 offers another EFS enhancement:

- Encrypted files can be stored in Web folders (Microsoft, "Microsoft Windows XP – Overview of EFS". See web link in Reference 3). When using this feature, encryption is actually more convenient and more secure as far as EFS is concerned, but Web Folders requires IIS to be running. Encrypted files not only can be stored in the web folder but also remain encrypted in transit. Using this method, all difficulties and prerequisites discussed previously in this section (*Design Considerations*) no longer exist. The only drawback is that the http access of these web folders is not supported in IntelliMiror (Microsoft, p. 1189, 1211) technology which Microsoft offers in Windows 2000 and later operating systems. The IntelliMirror technology includes features such as Folder Redirections and Offline Files.

User Folder Security – All user folders in the organization are configured so that users have *Modify* privilege and administrators have *Full Control* privilege. This configuration benefits the administrators in routine management tasks. However, when handling confidential data, although they are protected by EFS, it is advisable to only allow the user *Full Control* privilege. It helps to keep the data more secure at the expense of remote administration.

Protecting Data in Transit – EFS provides security for data residing on client workstations and servers, but it does not provide security for data in transit. All encrypted files (except those in Web Folders for XP and Server 2003) are first decrypted, transmitted in plain text, and then encrypted at the target host. A network sniffer can easily intercept these communications and cause security breaches. Starting in Microsoft Windows 2000, IP security technology is available when end-to-end security is required. This project uses IPSEC to require encryptions between all hosts that contain encrypted data.

An alternative to using IPSEC is to use Web Folder. However, it cannot be easily deployed due to the lack of adaptability in the IntelliMirror technology.

Protecting Privileged Accounts – Although user data can be protected by EFS on both the local workstation and on the server cluster, it does not protect user passwords from being stolen or changed. When this happens, any one with the password can log on to any workstation in the enterprise and gain access to all encrypted data that this user owns.

Along with the Public Key Infrastructure (PKI), Microsoft offers the use of Smartcard (Microsoft, p. 767) technology to protect user passwords. Users need to have a Smartcard reader installed at the workstation and the system administrator issues a Smartcard with each user assigned a pin number. The detailed procedures in deploying and using a Smartcard can be found in Windows 2000 help.

Smartcard logon protects users against stolen passwords but it does not prevent user passwords from being intentionally changed by an administrator. There are multiple user and group accounts in an Enterprise which should be guarded from unauthorized changes. These accounts are (1) those protected by the security settings of AdminSDHolder; (2) EFS recovery agents; and (3) users using EFS. In order to protect these accounts, they should be configured with special Active Directory object permissions either by placing them a restricted OU or by removing the inherited object permissions and assign a custom set. The detail object security settings are described in **Final Design | *Active Directory Object Security*** section.

Data Backup and Restore – Storing user data on a file server provides an easier and more efficient method to back up data. Backup bypasses security restrictions and maintains the encryption status on files and folders. However, sensitive data can be restored to an alternate location and then deciphered using an EFS recovery agent account. The backup logs should be monitored for the restore of encrypted files to an alternate location.

Audit and Alert – No matter how well files and Active Directory objects are protected, without proper auditing procedures, no one knows when and where a security breach has happened. Although Windows has a security log, it contains too much information. It is confusing without a thorough understanding of the event details. Commercial log parser is available but a customized script may provide versatility and cost benefit.

In all network environments, the administrators normally have more access privileges then normal users. In order for them to be able to manage the network at all situations, administrators also carry the ultimate privilege, Take Ownership. As soon as the administrator becomes the owner of an object, all other security

settings can be altered. The auditing procedures are really aiming at the administrators' actions.

The auditing script should be at least configured to run on two protected servers. This server script should monitor the following six items: (1) password changes on protected accounts; (2) permission changes; (3) ownership changes; (4) successful READ on encrypted files by a non-owner account; (5) restoring encrypted data to an alternate location; (6) interactive logon using EFS recovery agent's account; and (7) successful READ on all exported .pfx file. If there is any security violation, the script sends an alert to both an administrator and a designated EFS user.

Another version of the script or a compiled audit application should also be configured to run on the client machine. This client level script monitors and records interactive logons, established network sessions at the workstation, and whether or not the server script is running properly. If there is any security violation or if the server script is not running, the script sends an alert to both the administrator and a designated EFS user.

Automation Scripts – The sample scripts listed in the Appendixes perform the auditing and alerting functions based on object access. The technique used in these scripts is the *Asynchronous Event Subscription,* which is available in Windows Management Instrumentation that Microsoft offers in Windows 2000 and later operating systems. Basically, when an audited security event happens, it gets recorded in the security log and then sent to the event subscriber, the script. The script verifies the event and sends an alert if it meets certain criteria.

Since the script should be run at a secured server, the best choice is running it on a domain controller, because it is generally configured in the most secured manner and only the domain administrators (not the Domain Admins global group) have full access to it. In this case, besides the audit and alert functions for EFS users, the script can also perform other domain level security monitoring and alerts such as the accesses of directory objects and excess logon/logoff events as a result of worm and virus activities from client workstations. It can also verify the executables running on all workstations in the enterprise so that any Malwares and Trojans can be spotted in a timely fashion.

Final Design

Server Cluster –

1. Upgrade Windows 2000 Advanced server to Windows Server 2003.
2. Check the [Trust computer for Delegation] option at the property page of both servers' Active Directory object.
3. Three resource groups: cluster quorum, cluster file share and cluster print share, were created on the cluster servers. They use virtual server names

ClusterQuorum, ClusterFile and ClusterPrint. Each virtual server has its own dedicated Physical Disk resource, which are logical disks created under a Disk Array.

4. In the Property window of each Network Name resource in the cluster group, select *Advanced* and check both options to configure the virtual server name to require DNS registration and Kerberos authentication.
5. The ClusterFile has two file share resources which point to subfolders named *Users* and *Groups* for the cluster user shares and group shares, respectively. The share names for these two resources are Users and Groups.
6. The ClusterPrint has one *Print Spooler* resource pointing to a spool folder.

Public Key Infrastructure –

1. Install Certificate Service and configure it as the Enterprise Root Certification Authority at one of the domain controllers, DC1. Assign a target for the configuration information during installation and it will be shared as \\DC1\CertConfig.
2. Configure NTFS permissions for this secure folder \\DC1\CertConfig to have only one Trustee *Administrator* and assign the *Full Control* permission to it.
3. Schlumberger Cryptoflex Smartcard reader was used in this project and it was installed on all domain controllers, cluster servers, and the EFS users' workstations.
4. All three domain controllers were configured as Smartcard Enrollment Agent by installing the Enrollment Agent certificate.
5. A Smartcard User certificate was issued and the card was prepared via the certificate service's website \\DC1\CertSrv for each member in the following groups: EFS users, EFS recovery agents, Domain Admins, Domain Administrators, and Enterprise Admins.

Group Policies –

- **EFS Recovery Policy** – The following steps were used to create a domain level EFS recovery policy:

    1. Create three accounts to be used as EFS Recovery Agents in an OU named *Restricted*.
    2. Log on to a workstation using one of the accounts and apply for an EFS certificate from the certification authority at \\DC1\Certsrv.
    3. Export this certificate and its private key to a .pfx file using the Certificate MMC snap-in and place it in a secured location, for example at one of the domain controllers: \\DC1\CertConfig\EFS.
    4. Delete the certificate in Certificate MMC snap-in.
    5. Repeat step 2 and 4 for the other two accounts in *Restricted* OU.
    6. Create an EFS Recovery policy at the domain node and add the three accounts in *Restricted* OU as EFS Recovery Agents. The policy is

configured under: *Group Policy* | *Computer Configuration* | *Security Settings* | *Public Key Policies* | *Encrypted Data Recovery Agents*.

- **IPSEC Policy** – The following steps were used to configure an IPSEC policy:

  1. Move all computer accounts for users who use EFS into *Restricted* OU.
  2. Move the computer accounts for the cluster servers into *Restricted* OU.
  3. Configure an IPSEC policy for *Restricted* OU at this location: *Group Policy* | *Computer Configuration* | *Security Settings* | *IP Security Policies on Active Directory* by assigning the *Server (Request Security)* policy. Name the group policy "IPSEC_EFS".
  4. Configure the properties of "IPSEC_EFS" policy:
  5. In *Server (Request Security) Properties* window, click *Rules* tab.
  6. In *Rules*, *All IP Traffic* should be checked and has Filter Action set to Request Security.
  7. Select *All IP Traffic* and then click *Edit*.
  8. In *Edit Rule Properties* window, select *IP filter List* and then click *Edit*.
  9. In *IP Filter List* window, remove the default filter and then add multiple filters so that each filter corresponds to each client workstation and the other server cluster node. These filters include all ports and all protocols and use My IP address as the source and the IP address of the client workstation or the other server cluster node as the destination. Allow the filter to be *Mirrored,* which is the default setting.
  10. This policy applies to all *Authenticated Users* within the OU. In this case, Authenticated Users means computer accounts.

- **Folder Redirection Policy** – The steps are configuration steps:

  1. Move all user accounts that use EFS into *Restricted* OU.
  2. Configure the folder redirection policy for *Restricted* OU so *My Documents* and *Desktop* point to the user folders under the cluster file share: \\ClusterFile\users\MyDocuments\%username% and \\ClusterFile\users\Desktops\%username%. The redirection policy settings can be found at this location: *Group Policy* | *User Configuration* | *Windows Settings* | *Folder Redirection* | *My Documents and Desktop*.
  3. Use default value for all other settings in this group policy, which grants the user exclusive rights to the redirected folders.
  4. This policy applies to all *Authenticated Users* within the OU.

- **Offline Files Policy** – Below are the configuration steps:

  1. Create a global group named *Notebook Computers* in *Restricted* OU.
  2. Add all notebook computers in *Restricted* OU into this group.
  3. Configure a group policy named *DisableOfflineFiles* for *Restricted* OU at this location: *Group Policy* | *Computer Configuration* | *Administrative Templates* | *Network* | *Offline Files.*

4. Under *Offline Files* policy folder, the first policy is called *Enabled*. Open this policy and select Disabled. This step disables Offline Files for all computers within the OU.
5. Open the Properties page for this *DisableOfflineFiles* group policy, click the Security tab.
6. In *DisableOfflineFiles* Properties window, add the global group *Notebook Computers* (created in step 1) as a trustee and check [Deny] for the [Apply Group Policy] access mask. This step lets only notebook computers have offline files capabilities. Since contents of Redirected folders are automatically available offline, this policy is sufficient to filter out all desktop computers from using offline files.

- **User Right and Audit Policy** – The following steps create an Audit Policy for the cluster servers and protected user accounts:

1. Create an audit policy for *Restricted* OU at this location: *Group Policy* | *Computer Configuration* | *Security Settings* | *Local Policies* | *Audit Policy*. Under the *Audit Policy* folder, enable these four policies for both Success and Failure audits:
    a. Audit account logon events.
    b. Audit account management
    c. Audit logon events
    d. Audit object access
2. In this location: *Group Policy* | *Computer Configuration* | *Security Settings* | *Local Policies* | *User Rights Assignment,* configure Logon locally to include only Administrators and Backup Operators.

Active Directory Object Security – The object security settings for *Restricted* OU is configured without inheritance from parent. The following three Access Control Entries (ACEs) are defined in the DACL: (1) Everyone with *Read* access mask; (2) Self with *Full Control* access mask; and (3) SYSTEM with *Full Control* access mask. All three ACEs apply to *This Object* and *All Child Objects*. Verify that the child objects in *Restricted* OU only have inherited ACEs. Remove non-inherited ACEs.

User Profiles – Use roaming profiles whenever possible. The roaming profile path is defined in the account Profile property window. It points to this location \\ClusterFile\users\profiles\%username%. For situations that roaming profiles are not desirable, the following steps configure local profiles on the cluster servers to use EFS:

1. At the user's workstation, log on as the user and then connect to the server cluster with the virtual server name. (Use the actual server name for a Windows 2000 server cluster. In this case, it should be considered as just a remote server encryption because a cluster fail-over results in service outage.)

2. Encrypt some files and disconnect from the server cluster. This step generates an EFS certificate at the cluster node.
3. Fail-over the cluster resource, and connect to server cluster with the same virtual server name at user's workstation.
4. Encrypt some other files. This step generates an EFS certificate at the second cluster node.
5. Log on as the user at one of the server cluster nodes, use Certificate management console to view user certificates. Export the EFS certificate and the private key that were just created in step 1 or 3 and place them in this secure location: \\DC1\CertConfig\EFS (created in previous EFS Recovery Policy step).
6. Import this EFS certificate and private key from \\DC1\CertConfig\EFS at the other cluster node using Certificate (User) MMC console.
7. Repeat step 5 and 6 at the second cluster node.

EFS Certificates and Private Keys – If local profiles are used, the certificates and private keys have been exported in step 5 in the last section. If roaming profiles are used, simply use the Certificate (User) MMC console to export the EFS certificate and associated private key. Save the .pfx file in this secure location: \\DC1\CertConfig\EFS.

Client Workstation – upgrade Windows 2000 Professional to Windows XP. In the Computer Property window, select *Remote* and configure the Remote Desktop to let users connect to this workstation via a Terminal Service session. This will help to resolve the user's offsite connection and encryption issues. See details in **Encountered Problems | *VPN Users Cannot Open Encrypted Files***.

Share Permissions – For all shared folders, change the settings to [Everyone – Full Control] only for Windows Server 2003. This setting is the default for Windows 2000 Advanced Server.

NTFS Permissions –

- For users' My Documents, Desktop, and Profile folders, change the settings to [UserID – Full Control] and remove all other trustee assignments in the Discretionary Access Control List (DACL). This is the default setting for redirected My Documents, Desktop and Profile folders.

- For public folders – The folder is configured at \\ClusterFile\public. Other subfolders are created based on user needs. The folders are not encrypted, but files deposited into these folders may be encrypted and configured for selected users to access.

    - Top level – \\ClusterFile\public
        - All users in the department – ***Read – All Objects***
        - Administrators - ***Full Control – All Objects***

    - Second Level – Other subfolders with *READ* access mask
        - Selected users – ***Modify – All Objects***
        - Administrators – ***Full Control – All Objects***

    - Second Level – Other subfolders with *Modify* access mask
        - Selected users – ***Modify – All Objects***
        - Administrators – ***Full Control – All Objects***

- Spool folder – Creator Owner – ***Full Control – Subfolders and Files***
  System – ***Full Control – All Objects***

- Audit settings for Folders – Successful READ, Take Ownership and Change Permissions on the following:
    - Exported .pfx files in \\DC1\CertConfig\EFS.
    - User folders in \\ClusterFile\users\MyDocuments for EFS users.
    - Public folders for EFS users.

Audit and Alert – The server script runs on both cluster servers. It records all audited events to log files in a local folder as well as to a shared folder on one of the domain controllers. The third domain controller also runs an audit script which tracks the same security events that are monitored at the cluster server.

Each client workstation has a simpler script (or a compiled executable) to track the interactive logons and network sessions. On the group representative's computer, a simple shell script is written to track the continuous running of the server scripts.

E-Mail – Since Eudora does not work properly when encrypted, a scheduled task is configured to copy Eudora folder on a nightly basis to the encrypted server location for backup purposes. This method is also utilized for users with notebook computers and use Microsoft Outlook. In the case of Microsoft Outlook PST file, they should be stored in the encrypted server location. In this project, it is stored in the home drive, which is configured to be the same as the redirected My Documents folder, \\ClusterFile\users\MyDocuments\%username%. All e-mail clients are configured to use POP protocol.

The following lists a simple shell script to copy the email folder at night. It should be scheduled using Windows Scheduled Task and the machine should be left on at night at lease several times a week. PsKill.exe (Mark Russinovich, Feb. 7, 2000, v1.03. Sysinternals, http://www.sysinternals.com/ntw2k/freeware/pskill.shtml) is used in this script to kill the E-mail client process.

```
@echo off
Pskill Eudora.exe or PsKill Outlook.exe
Xcopy EmailFolder \\ClusterFile\users\MyDocuments\%username% /S/D/R/Y
```

Notebook Computers – Notebook computers are configured using the following steps for extra protection:

- Use the hardware (BIOS) level password. The available password length and complexity level depend on the manufacturer of the system.
- Use Windows SYSKEY utility to guard the master key. Run SYSKEY program at the command prompt, select *Update*.
- In the next *Account Database Key* window, select Password Startup and then enter the password. A password phrase longer than 15 characters is used and in addition, a Unicode character is added in it.
- Since logon authentication uses Smartcard, users no longer use passwords; a 30-character long random password is assigned to user accounts.

User Training –

- How to encrypt and decrypt files and folders.
- How to add other users to the access list to read encrypted files in the public folder.
- How to VPN to the office network and establish a Terminal Service session to the office workstation.

Scripts – The sample scripts were written with the following goals in mind:

- Detect password attack or other denial of service attack as a result of virus and worm activities. It monitors the frequencies of creations of the following logon/logoff events in the security log:

    - EventID 529 – Failed network logon
    - EventID 540 – Successful network logon
    - EventID 681 – Failed account logon
    - EventID 673 – Successful Kerberos ticket issued, account logon
    - EventID 677 – Failed Kerberos ticket request, account logon

    All events are collected at the domain controllers and event 529 and 540 are collected at all other machines. Among these events, events 681 and 673 have been positively identified in the past to be evidence of password attack and virus/worm activities.

    The thresh-hold for a suspicious activity is set as 360 hits in 15 seconds, which is the sample period. Although, in the past, these type of activities generally create 70 to 150 hits per second for Event 681, and more than 45 hits per second for event 673.

- Detect Malwares and/or Trojans by verifying the MD5 sum of all processes running on all machines in the Enterprise. This is achieved by subscribing an asynchronous ProcessCreation event at the local computer, which gets the

MD5 sum of the process' executable by running the MD5sum.exe program (Forensic Acquisition Utilities, George M. Garner Jr., May 30, 2003, http://users.erols.com/gmgarner/forensics/). Then, compare it with a list of verified MD5 sums. If the MD5 sum does not exist in the database, the executable is copied to the central location for investigation.

- Detect *Take Ownership* and *Change Permissions* events on protected Active Directory objects (those provided by AdminSDHolder and those in the *Restricted* OU). This is accomplished by tracking those security events which have EventID 565, Object Server: DS and either DAC or OWNER in the access field.

- Detect *Take Ownership* and *Change Permissions* events on protected file objects, i.e., the exported .pfx files. Security EventID 560 is collected and logged when Object type is File and either DAC or OWNER is in the access field.

- Detect object open event (EventID 560) on encrypted user files where the Object Type is File and the user's samAccountName (embedded in Event User field) is not in the Object Name. The Object Name has the full path of the target file, which always has the userID embedded in it based on the design of the cluster server shares.

- Monitor and record password changes by tracking all security events where the Event Category (=7) is Account Management.

- Monitor and record interactive logons by tracking security EventID 528. Although the WMI class W32_LogonSession can be used to do the same, but it does not work with Windows 2000 workstations:
    - Logon Type 2 – console and Terminal Service logons for Windows 2000, console logon for Windows XP and Server 2003.
    - Logon Type 10 – Terminal Service logon for Windows XP and Server 2003.

- Monitor and record sessions established with the local computer by subscribing an asynchronous ServerSessionCreation event at the local computer.

- The scripts are broken down into the following components and they are incorporated into one .wsf file except the last one:

    - AD_Audit.wsf – Main script file. It includes all other procedures and runs the reporting procedures every 15 seconds.

    - ReadConfiguration.vbs – Reads the configuration information and determines what the script should do.

- CreateAsyncQueries.vbs – Generates asynchronous queries for audited security events on local computer.

- CreateRemoteAsyncQueries.vbs – Generates asynchronous queries for audited security events on remote computers.

- SecurityEventSinkHandler.vbs – Verifies that the security events arrived from the event subscription meets the audit criteria and saves them for further processing and reporting.

- ProcessCreationSinkHandler.vbs – Saves all ProcessCreation events arrived from the event subscription for further processing and reporting.

- EventLogSinkHandler.vbs – Saves all Event Log events (except those in the security log) arrived from the event subscription for further processing and reporting. This component is not related to EFS or security audit, but it is covered in the Enterprise audit and management script.

- SecurityEventSinkReport.vbs – Records audited security events into the logs and sends notification(s) to administrators and/or the affected user.

- ProcessCreationSinkReport.vbs – Compares the MD5sum with a central database and sends the executable to it if the verification is not successful.

- EventLogSinkHandler.vbs – Records all filtered Event Log events into a central log.

- Check_Save_SecurityLog.vbs – Verifies the size of security log and saves it into a file if it exceeds 50% of the maximum size. After saving it, clear the security log. This script runs by itself and it is called by a Wscript Shell "run" command in the main script so the backup uses another thread. This is because the backup may take several minutes to complete.

Encountered Problems

1. ***Cannot Set / Clear Object Attribute – Trusted for Delegation***

Background – Under certain situations, one of the computer properties [Trusted for Delegation] should be set so that the computer account can request services from other resources on behalf of the user. This situation occurs in Web related security account management, remote encryption (EFS), domain authentications via Kerberos protocol, etc.

Symptom – Around the time the domain controllers were updated to Service Pack 3, the computer property [Trusted for Delegation] can no longer be set or

cleared using the domain administrator account or any other accounts belonging to [Domain Admins] group. It gave an error message similar to this: ***The security does not have sufficient privilege to complete this task.***

This problem happened in a native mode Windows 2000 Active Directory domain. Because this attribute is not frequently modified, by the time it is apparent, the operating system could not be recovered to a previous working state. Since there is no Knowledgebase article from Microsoft relating to this issue and seeking information/assistance from newsgroups did not yield any satisfactory solution, this problem lasted for about six months. It caused some minor inconveniences during that time.

Solution – This problem was resolved incidentally. While applying service pack 4 to DC1 (owner of the PDC, Schema, Domain Naming FSMO roles), the process was interrupted due to a power loss and as a result, the operating system needed to be reinstalled from scratch (the restore of system state backup did not work, for some reason). After patching the new operating system and restoring NTDS database, the problem went away.

Cause – This problem has to be related to the OS or samAccount database, but exactly which file(s) or registry setting(s) caused this abnormal behavior remains unknown.

2. Cannot Open Encrypted Files – Access is denied

Background – This problem applies to both Windows 2000 and Windows XP users. The encrypted file system was configured for users in the Human Resources department and *My Documents* and *Desktop* were redirected to the server cluster. Users were using local profiles on a server cluster. The *My Documents* folder was encrypted and users were able to work on their documents properly.

Symptom – When users tried to open the same document the next morning, an error message ***"Access is denied"*** appeared.

Cause – Since *My Documents* was redirected and there was no policy to disable Offline Files at the workstations, files in the *My Documents* folder had an offline version by default. At the time, there were frequent network glitches that disconnected users' workstations from the network. The operating system sensed the network outage and marked the files in the offline files folder as OFFLINE. For some reason, the encrypted files could not be opened if the offline files were not synchronized.

Solution – Let users synchronize the files via Explorer | Tools | Synchronize. Remove Offline Files on desktop machines via Group Policy. See details in **Final Design | *Group Policy* | Offline Files** section.

Note: When attempting to verify this error again at this writing, it could not be reproduced on a Windows XP machine.

3. Cannot Encrypt Files – Keyset does not exist

Cause – There are several situations where this error message appears. It basically indicates that an EFS certificate or private key cannot be obtained to encrypt/decrypt files.

Solution – Follow the configuration settings in **Final Design** section.

4. Cluster Server Experienced 100% CPU Usage

Background – The cluster servers provide EFS support for Human Resources users. It is a Windows 2000 server cluster and has been configured properly to work with EFS. Users were accessing encrypted files transparently without problems.

Symptom – The cluster servers were sluggish and the performance log showed prolonged high CPU usage. The process consumed most of the CPU time (>85%) was WinLogon.

Cause – When users tried to access encrypted files on the cluster server, if the private key could not be found—whether corrupted or deleted—the cluster server tried continuously to retrieve the user's private key from the domain controller for every file access attempt.

Solution – Identify the EFS user who is experiencing the problem and turn off that workstation forcefully. Very likely, the workstation is hung at that time. Since the cluster server may be providing other services to other departments as well, restarting the server is not a good choice. Fail-over on the file service will not resolve this problem either as long as the user is still trying to access the encrypted files.

5. VPN Users Cannot Open Encrypted Files

Background – An EFS user connected to her office machine via VPN. Her home machine has Windows XP configured while the office machine is a Windows 2000 unit.

Symptom – When the user tried to access some encrypted files in the *My Documents* folder (redirected to the server cluster), they were read-only. When she tried to read her email from the Eudora folder in the office machine, it hung. In attempting to decrypt the *My Documents* folder remotely despite error messages, clicking [Ignore All] actually decrypted all files, but left the folders encrypted. In the attempt to decrypt locally encrypted Eudora files, the error message *Access is denied* appeared.

Cause – Files in the *My Documents* folder became read-only and were probably caused by the offline files issue described earlier (see **Encountered Problems** #2). Eudora does not work properly because some of the files were encrypted.

Solution – Upgrade office workstation to Windows XP operating system and configure Remote Desktop so user can connect to it from home via Terminal Service.

6. Cannot Open Encrypted Files Even After Synchronizing Offline Files

Symptom – EFS users cannot create new folder, existing documents become read-only.

Cause – The certificates for EFS Recovery Agents expired.

Solution – Typically, EFS automatically renews its certificates when they expire. However, in this case, it was unable to do so, and attempts to renew the certificates were not successful. New EFS recovery agents were added without deleting the old ones.

References

1. Microsoft Corporation – "Windows 200 Resource Kit, Microsoft Windows 2000 Server Distributed Systems Guide". Microsoft Press, 2000.

2. Microsoft Corporation – "Guide to Creating and Configuring a Server Cluster Under Windows Server 2003", Version 1.0, 11/21/2003. http://www.microsoft.com/downloads/details.aspx?FamilyID=96f76ed7-9634-4300-9159-89638f4b4ef7&displaylang=en).

3. Microsoft Corporation – "Microsoft Windows XP – Overview of EFS". http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/windows/xp/all/reskit/en-us/prnb_efs_awzg.asp

4. Microsoft Corporation – "Applying Kerberos Authentication in a Clustered Environment". http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/deployguide/en-us/Default.asp?url=/resources/documentation/WindowsServ/2003/all/deployguide/en-us/sdccf_cls_rtzh.asp).

5. Microsoft Corporation – "Remote EFS Operations in a File Share Environment". http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/windows/xp/all/reskit/en-us/prnb_efs_ijvx.asp

6. Microsoft Corporation – "Encrypting File System in Windows XP and Windows Server 2003". http://www.microsoft.com/technet/prodtechnol/winxppro/deploy/cryptfs.mspx

7. Microsoft Corporation – "Best Practices for the Encrypting File System". http://support.microsoft.com/default.aspx?scid=kb;en-us;Q223316

8. Microsoft Corporation – "Step-by-Step Guide to Encrypting File System (EFS)". http://www.microsoft.com/technet/prodtechnol/windows2000serv/howto/efsguide.mspx

9. Microsoft Corporation – "How EFS Works". http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/distrib/dsck_efs_duwf.asp