Easy Transparent Encryption File System

(User Mode Based)

ETEFS_User - Version 4.2

Product White Paper

0 Copyright notice

Please read the “license.doc” to get enough copyright information first. If you do not agree with this license term, please don’t use any software provided by ETEFS.COM.

1 Introduction

ETEFS_User is a transparent file encryption SDK based on API hooking technology. It runs in ring 3 mode. ETEFS_User can encrypt those files containing sensitive data generated by the selected applications, such as Microsoft Word, Excel and Power Point. All files are stored in encrypted form on the disk. The encryption progress is executed at background on-the-fly, and does not change the normal operation behavior of the end user. Because of developing by user mode API hooking technology, without any kernel mode drivers, it is easy to integrate ETEFS_User into your software product in a short development cycle. Besides of the core transparent file encryption functionality, ETEFS_User also provides some extension functionalities that are very useful in developing data security related software product, including a file access control module and an operation event monitor module. With the help of ETEFS_User, software developers can rapidly build various document security products from scratch, like DLP (Data Loss Prevent) system, document right management system and document operation auditing system, etc.

2 Product Benefits

Easy to use

The API set of ETEFS_User is comprised of several functions programming in “C” language exported by a DLL library. By using these functions, it’s very easy to build a basic transparent file encryption system. These API functions can also be called from other programming language, like “C#” and “VB”.

Easy to extend

To add file encryption support for a new application is also very easy. ETEFS_User provides two functions to support adding and deleting the file encryption policy. ETEFS_User can support most common applications by sending correct policy.

Provides useful extension modules

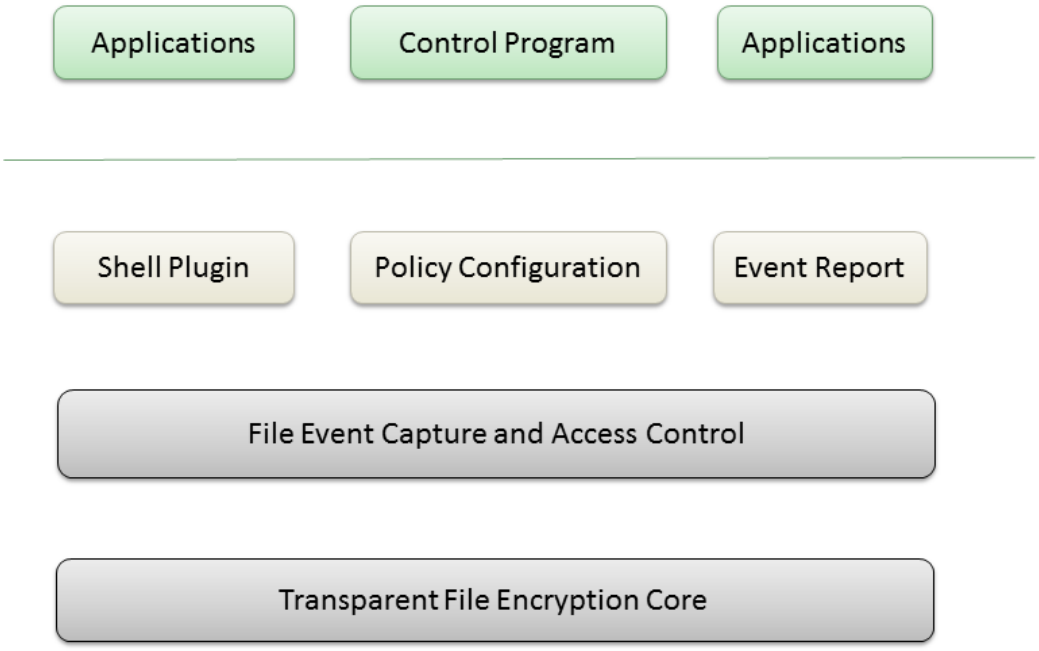
ETEFS_User includes a file access control module and an operation event monitor module. The file access control module is used to restrict the operation on the currently opened file issued by the user according to the access control policy. The operation event monitor is used to capture the user's action on the encrypted file, such as "file opened" and "file closed" event, etc.

Cut down the development cycle

Although developing a transparent file encryption engine in user mode is not as difficult as developing it in kernel mode, it still involves many debugging and testing works. If you want to create an engine from scratch, this may take a long development cycle and cause the loss of marketing.

3 System Architecture

The following diagram shows the architecture of ETEFS_User:



Architecture Diagram

Applications

The applications diagram represents the selected executable to be encrypted, such as Microsoft Word, Excel and Power Point. The document files generated by these applications will be encrypted automatically.

Control program

The control program diagram represents the process sending the policy to the transparent file encryption system core and receiving the file operation event log. If this program exits, all policy set by it will not take effect.

Policy configuration

The policy configuration diagram represents the module that receives the policy from control program. It stores the policy data to a file mapping object. Other modules can get the corresponding policy by reading the file mapping object.

Event report

The event report module receives the operation log data from the capture module and pushes these log data to the control program. It uses the named pipe and UDP protocol as the data transport mechanism.

File event capture and access control

The functionality of this module has been described in chapter 2.

Transparent file encryption core

This is the key component of ETEFS_User. It implements transparent file encryption by hooking the system file I/O functions. It decrypts data while an application loads data from disk and encrypts data while an application writes data to disk.

4 Product Features

Implementing in user mode

All modules of ETEFS_User are implemented in user mode. It is easy to use and integrate into your software product. Unlike kernel mode drivers, you don’t have to sign these executable files

with a digital certificate.

Random file encryption key

ETEFS_User supports the random file encryption key mechanism. After enable this feature, ETEFS_User will allocate a random file encryption key for each file. Using this method to encrypt file is much more safety than using fixed file encryption key.

“Custom data” binding

ETEFS_User adds an extended file header data to each encrypted file. A part of this header data is reservedly used by ETEFS_User itself to save some basic information about encryption. Another part of this header data called “custom data” is free for developer. Once this area of header data is set by the developer, it is firmly bind to the encrypted file forever by ETEFS_User. While editing the file, this custom data is still bonded with the encrypted file. The kind of data can be used to save the extended control information for the encrypted file.

Manually encryption

By default, once an unencrypted file is opened by a user, it will be encrypted by ETEFS_User automatically. If the developer enables “manually encryption” policy, ETEFS_User will not encrypt the file automatically. Users must use the encryption tools, for example a shell menu item, provided by the developer to convert the file into encrypted state.

Save-as encryption

When a user saves a currently opened file to a new file, the newly created file will be encrypted by ETEFS_User if the currently opened file is encrypted. This mechanism is used to keep the file containing sensitive data is always in encrypted state.

File event capture

ETEFS_User supports capturing common file operation event, such as “open”, “close “and “print” etc.

File access control

ETEFS_User supports controlling the access right of the file, such as “read-only”, “disable save-as” and “print” etc.

Faked exe checking

ETEFS_User identifies the process to be encrypted by the name of the process. This mechanism is not perfect. Because some skillful users may rename an executable file to the name of the target process, these faked executable are able to get the plain data of the encrypted file. ETEFS_User provides an extension feature to identify these faked executable files. To achieve this, ETEFS_User checks the MD5 of the faked executable files or the digital signature if any. Once a faked executable is detected, the transparent file encryption core will stop encryption and decryption services for this faked process.

Application support

ETEFS_User supports transparent file encryption feature for any type of process. ETETFS_User can support most common applications by sending correct policy. It may require some debugging works for these complicated applications.

File system support

ETEFS_User supports any type of file system only if the file system is available in windows. The supported file system list includes fastfat, ntfs, network file system, cdfs and udfs. ETEFS_User is compatible with the encryption and compression feature in ntfs.

Cipher support

ETEFS_User integrates the XTEA and AES encryption algorithm into the transparent file encryption core. ETEFS_User can support any type of block encryption algorithm by customized development.

OS support

32-bit (x86) and 64-bit (X64) architectures of Windows XP and later.

5 Support and Services

License type

Clients can evaluate ETEFS_User for 1 month. ETEFS_User supports two kinds of license term.

1. SDK license

The SDK package includes these items shown in the flowing list:

- ✓ Executable binary files for both x86 and x64 Windows OS
- ✓ Header and lib files for compiling and linking
- ✓ A full source code demo project that describes the usage of ETEFS_User API
- ✓ ETEFS_User SDK reference.pdf
- ✓ ETEFS_User user's guide.pdf

2. Full source code license

The full source code package includes these items shown in the flowing list:

- ✓ Executable binary files for both x86 and x64 Windows OS
- ✓ Full source code for all modules of ETEFS_User.
- ✓ ETEFS_User SDK reference.pdf
- ✓ ETEFS_User user's guide.pdf
- ✓ ETEFS_User source code guide.pdf

3. Both SDK and source code license are no limitation on number of copy installation.

Technical support

License of ETEFS_User includes one year of technical support, including questions, bug support, and access to framework maintenance updates. Licensees will also have options to secure major updates (functional enhancements) and OS upgrades as well.

Custom development

Some clients may want to customize core components of ETEFS_User to meet their product needs. In addition to providing full source license, we can be engaged to provide custom development services to modify ETEFS_User to client specifications.