DuD
Datenschutz und Datensicherheit

Encrypting File System unter Windows

Features, Lücken, Gefahren, Vorteile

Tim Fickert, Matthias Nau, Rainer W. Gerling

Die Verschlüsselung von gespeicherten Daten gewinnt mit der zunehmenden Verbreitung mobiler Computer zentrale Bedeutung. Eine naheliegende Methode ist die Verwendung des Encrypting File Systems (EFS) der Betriebssystemfamilie Microsoft Windows. Die Autoren stellen die Eigenschaften vor und diskutieren die Vor- und Nachteile.

Tim Fickert

studiert an der Fachhochschule München Informatik.

E-Mail: tim.fickert@thalion.de

Matthias Nau

studiert an der Fachhochschule München Informatik.

E-Mail: nau@muenchen.org

Dr.
Rainer W. Gerling

Datenschutzbeauftragter der Max-Planck-Gesellschaft, Lehrbeauftragter für Datensicherheit an der FH München.

E-Mail: rgerling@gmx.de

Einleitung

Im betrieblichen wie auch im privaten Umfeld stellt sich oft die Frage, wie schütze ich meine Dateien vor allzu neugierigen Blicken anderer? Dabei muss besonders im betrieblichen Umfeld auf einige Besonderheiten Rücksicht genommen werden: Fällt ein Mitarbeiter zum Beispiel für längere Zeit oder für immer aus, so muss gewährleistet sein, dass die Firma an seine Daten herankommt. Außerdem sind hier bestimmte Fälle, wie Gruppenarbeit oder große vernetzte Umgebungen, zu berücksichtigen.

Windows 2000 und XP bieten das *Encrypting File System (EFS)*. Das EFS ist ein tief im Kernel von Windows 2000 und XP verankertes System zur transparenten Ver- und Entschlüsselung von Dateien. Dabei wurde auf den Einsatz in betrieblichen Umgebungen Rücksicht genommen, so dass Funktionen wie Recovery oder transparente Verschlüsselung auf Netzwerklaufwerken möglich sind. EFS benutzt ein Hybridverfahren zur Verschlüsselung und zur Schlüsselverwaltung.

1 Features

Mit dem *EFS* können Dateien und Ordner zum Verschlüsseln markiert werden. Unter Windows XP können auch die so genannten *Offline Files* verschlüsselt werden. Wird ein Ordner verschlüsselt, wird nicht der Ordner selbst verschlüsselt, sondern durch das *EFS* alle neu im Ordner erzeugten bzw. in den Ordner kopierten Dateien. Wird ein Ordner als verschlüsselt markiert, bietet das System an, alle im Ordner befindlichen Dateien und Unterordner rekursiv zu verschlüsseln.

Systemdateien und Dateien unter *SystemRoot* können nicht verschlüsselt werden. Ebenso wenig können Dateien oder Ordner in einem *Roaming User Profile* verschlüsselt werden. Dateien oder Ordner können nicht sowohl verschlüsselt als auch komprimiert werden. Bei der Verschlüsselung der *Offline Files* werden nicht die Dateien verschlüsselt, sondern die gesamte *Offline Database*. Unter Windows XP kann das System so eingerichtet werden, dass eine Datei für mehrere Benutzer verschlüsselt wird, und diese so alle Zugriff auf die Datei erhalten. Dadurch lassen sich in betrieblichen Umgebungen Gruppenfunktionen verwirklichen.

Die Verschlüsselung von Dateien erfolgt vollautomatisch und vom Benutzer fast nicht zu bemerken. Ist einmal eine Datei als zu verschlüsseln gekennzeichnet, erfolgt die weitere Behandlung mit Ver- und Entschlüsseln vollständig transparent im Hintergrund. Der Benutzer muss sich nicht

darum kümmern, vor dem Bearbeiten einer Datei diese erst einmal zu entschlüsseln und sie nach Beendigung der Arbeit wieder zu verschlüsseln. So ist im Allgemeinen auch ausgeschlossen, dass der Benutzer irgendwann einmal vergisst, eine Datei zu verschlüsseln.

1.1 Recovery

Um den Bedürfnissen großer Firmen nach *Recovery*-Funktionen[31] entgegen zu kommen wurde in das *EFS* das sogenannte *Data Recovery* integriert. Durch den sogenannten *Recovery Agent* wird so auch ohne Kenntnis des privaten Schlüssels des ursprünglichen Benutzers eine Entschlüsselung der Dateien ermöglicht.

Der Systemadministrator, in Netzwerken der Domänenadministrator, legt einen Benutzer als *Recovery Agent* fest. Nur diesem ist es dann möglich, die Dateien ohne Kenntnis des privaten Schlüssels des ursprünglichen Benutzers zu entschlüsseln. Unter Windows 2000 ist automatisch der Systemadministrator der *Recovery Agent*, unter Windows XP muss dieser gesondert eingerichtet werden. Der *Recovery Agent* ist integraler Bestandteil des *EFS* und kann bei Bedarf aktiviert oder deaktiviert werden. Da für das *Recovery* ein eigener Schlüssel benutzt wird, wird so der Schlüssel des Benutzers geschützt.

1.2 Sicherheit

Das hybride Verfahren aus dem asymmetrischen *RSA-Public-Key*-Verfahren und dem symmetrischen *DESX* bzw. *3DES* kann auch für die nächste Zeit als sicher angesehen werden. Methoden der Kryptoanalyse, um eine unberechtigte Entschlüsselung vorzunehmen, sind bei Verwendung hinreichend langer RSA-Schlüssel bis heute eher theoretischer Natur.

Mit *EFS* wird auch die Möglichkeit gegeben, die Verschlüsselungsalgorithmen beliebig auszutauschen. Microsoft bietet bereits – und empfiehlt auch – die Möglichkeit, den symmetrischen Dateiverschlüsselungsalgorithmus *DESX* gegen *3DES* auszutauschen. Hier bietet sich ein breites Feld, auch für Drittanbieter, dieses Verfahren auch über einen längeren Zeitraum hinaus sicher zu halten.

EFS unterstützt *Backup* und *Restore* verschlüsselter Dateien. Das Programm *NTBackup* ermöglicht dies. Dies gilt unabhängig davon, ob die Sicherung auf ein Bandlaufwerk, CD-ROM oder ein anderes Medium erfolgt. Die Daten sind auch hier gegen unberechtigte Kenntnisnahme geschützt.

2 Technische Hintergründe

Unter Windows 2000 und XP wird die Bereitstellung der Zertifikate von der *CryptoAPI* übernommen, die im ganzen System für kryptographische Operationen zur Verfügung steht. Dadurch werden Anwendungen in die Lage versetzt, kryptographische Verfahren transparent zu benutzen. Die API bietet auch eigenen Programmen die Möglichkeit Eigenschaften des *EFS* zu nutzen. Die API ermöglicht zum Beispiel auch den unentschlüsselten Zugang zu Dateien. Dies ist wichtig, da das *NTFS* normalerweise bei Dateioperationen die verschlüsselten Dateien an das *EFS* weiterreicht und diese automatisch entschlüsselt werden. Bei Backupsystemen ist aber eine verschlüsselte Speicherung auf einem Medium erwünscht, das von sich aus *NTFS5*, und damit *EFS*, nicht unterstützt.

Durch die Verwendung der Zertifikate wird außerdem eine Identifizierung des Benutzers ermöglicht. Die öffentlichen Schlüssel sind dabei von einer vertrauenswürdigen Stelle (*Certification Authority*, *CA*) unterschrieben.

Das Kommandozeilen-Tool *cypher* ermöglicht auch die Verarbeitung verschlüsselter Dateien durch die Kommandozeile oder in Batch-Dateien. Dies ermöglicht zum Beispiel in größeren Umgebungen verschlüsselte Dateien mittels *Logon Script* zu behandeln.

2.1 Verschlüsseln

Zunächst generiert *EFS* einen sogenannten *File Encryption Key* (*FEK*). Dieser *FEK* ist ein vom System gelieferter 128 Bit langer *DESX*-Dateischlüssel (im Falle der Aktivierung von *3DES* analog). Der *FEK* wiederum wird nun mit Hilfe des öffentlichen Schlüssels des Benutzers verschlüsselt und im *Data Decryption Field* (*DDF*) abgelegt.

Sollte der Benutzer noch nicht im Besitz eines Schlüsselpaares sein, wird automatisch durch das *EFS* eines angelegt. Ist der Rechner Mitglied in einer Domäne, wird das Schlüsselpaar vom Schlüsselserver angefordert. Sollten im System einer oder mehrere *Recovery Agents* eingestellt sein, wird für jeden dieser Agenten ebenfalls der *FEK* mit dem zum Agenten gehörenden öffentlichen Schlüssel verschlüsselt und im *Data Recovery Field* (*DRF*) abgelegt. *DDF* und *DRF* bilden zusammen mit einer Prüfsumme den *Header* der verschlüsselten Datei.

Ab Windows XP ist die Einrichtung weiterer Benutzer, die Zugriff auf die Datei haben dürfen, möglich geworden. Der Vorgang erfolgt dabei analog zum Vorgang für den Ersteller der Datei.

2.2 Entschlüsseln

Stellt das System beim Öffnen einer Datei fest, dass diese verschlüsselt ist, wird die weitere Behandlung an das *EFS* abgetreten. Der *EFS*-Treiber extrahiert das *DDF* und ermittelt anhand des privaten Schlüssels des Benutzers den ursprünglichen *FEK*. Mittels des *FEK* werden durch das symmetrische Verfahren diejenigen Stellen entschlüsselt, die von der Anwendung benötigt werden. Da das *EFS Cypher Block Chaining* benutzt bleiben die anderen Bereiche einer Datei verschlüsselt.

2.3 Recovery

Dieser Vorgang verläuft im Wesentlichen genau so, wie die normale Entschlüsselung. Nur wird in diesem Fall der private Schlüssel des *Recovery Agent* benutzt, um aus dem *DRF* des jeweiligen *Recovery Agent* (bei Einrichtung mehrerer Agenten) den *FEK* zu gewinnen. Durch diese Vorgehensweise wird der private Schlüssel des Benutzers zu keiner Zeit von jemand anderem als dem Benutzer verwendet. Dies gewährleistet eine höchstmögliche Sicherheit.

3 Lücken

Um für ein durch ein LAN oder WAN vernetztes Unternehmen interessant zu sein, sollte ein Dateiverschlüsselungssystem verschiedene Eigenschaften mitbringen:

3.1 Verschlüsselung des gesamten Dateisystems

Aus Sicherheitsgründen wird gerade bei Laptops häufig gewünscht, dass gesamte Dateisystem eines Computers zu verschlüsseln. Aufgrund seiner Abhängigkeit von Benutzerschlüsseln ist es mit *EFS* nicht möglich, eine ganze Festplattenpartition

[31] R.W. Gerling Company Message Recovery DuD, **22** 38 (1998)

(vor allem nicht die Systempartition) zu verschlüsseln. Wichtige Systembereiche müssen unverschlüsselt bleiben, um überhaupt ein Starten des Betriebssystems und des *EFS* zu ermöglichen. Dazu gehören vor allem der *Bootloader* (normalerweise die Datei C:\ntldr), das *SystemRoot*-Verzeichnis (normalerweise C:\WINNT bei Windows 2000 oder C:\WINDOWS bei Windows XP) und der Teil der Benutzer Profile, in dem die Zertifikate (Schlüssel) abgelegt werden.

Eine Verschlüsselung von Systemverzeichnissen anhand eines Maschinenzertifikats ist derzeit nicht möglich.

3.2 Verschlüsselung und Datenübertragung im Lokalen Netzwerk

In vernetzten Umgebungen ist es allgemein üblich, Benutzerdaten, Gruppendaten oder sonstige, allgemein zugängliche oder beschränkt zugängliche, Daten auf einem File Server im lokalen Netz abzulegen und den Mitarbeitern zur Verfügung zu stellen. Verständlicherweise wird hier auch der Anspruch erhoben, dass die Daten auf dem File Server und während der Übertragung gegen unbefugte Einsichtnahme und Manipulation gesichert sind.

Allgemeines

Bei der Verwendung von *EFS* ist dazu aufgrund der Architektur zwingend erforderlich, dass der File Server als Dateisystem *NTFS5* verwendet. Dadurch kommen als Server-Betriebssysteme lediglich Windows 2000 Server beziehungsweise Windows .NET Server in Betracht, da diese momentan als einzige die benötigten Erweiterungen am Dateisystem mitbringen.

Zur Dateifreigabe bieten sich seit Windows 2000 grundsätzlich zwei verschiedene Möglichkeiten an: Die herkömmlichen *File Shares* und die von Microsoft im Rahmen ihrer .NET-Initiative entwickelten *Web Folder*.

- *File Shares*, wie sie auch schon in früheren Windows-Versionen verwendet wurden, benutzen zur Übertragung der Daten im Netzwerk das *NetBIOS*-Protokoll. Sie werden allgemein von Microsoft-Betriebssystemen und zum Beispiel auch von Samba unter Linux unterstützt.
- *Web Folder*, die zuerst mit Windows 2000 eingeführt wurden, verwenden zur Datenübertragung das bekannte *HTTP* und es wird ein HTTP-Server mit *WebDAV* benötigt. Im alltäglichen Gebrauch werden *Web Folder* genauso wie *File Shares* verwendet und verhalten sich auch so. Der Benutzer bemerkt keinen Unterschied zwischen den beiden Verfahren.

Der grundlegende Unterschied zwischen den beiden Freigabearten im Zusammenhang mit *EFS* ist der Zeitpunkt bzw. Ort, an dem die Verschlüsselung der Daten erfolgt.

File Shares

Bei der Verwendung von *File Shares* und *EFS* auf Netzwerklaufwerken findet die Ver- und Entschlüsselung der Daten auf dem File Server statt. Dazu muss das *EFS* des File Servers über die so genannte *Kerberos Delegation* die Identität des jeweiligen Benutzers annehmen, um Zugriff auf sein Verschlüsselungszertifikat zu erhalten. Im Anschluss darauf werden die Daten ver- bzw. entschlüsselt. Hierdurch entsteht eine größere Belastung des File Servers, was bei knapp proportionierten Geräten zu Engpässen führen kann. Außerdem sind die Daten bei der Übertragung ungesichert, das heißt es muss zusätzlich für einen sicheren Übertragungsweg, zum Beispiel über *IPsec*, gesorgt werden.

Web Folder

Im Fall der *Web Folder* in Verbindung mit *EFS* findet die Ver- und Entschlüsselung der Daten nur auf dem Computer des Benutzers statt. Das bedeutet unter anderem auch, dass keine besondere Vertrauensstellung zwischen dem Arbeitsplatzrechner und dem File Server bestehen muss. Es hat allerdings auch zur Folge, dass eine nachträgliche Verschlüsselung einer im Netzwerk abgelegten Datei zusätzliche Bandbreite beansprucht, da die betroffene Datei zunächst auf den lokalen PC kopiert werden muss, dort verschlüsselt wird und anschließend wieder zurückkopiert wird. Im laufenden Betrieb sollte dies aber eher selten passieren.

Eine zusätzliche Sicherung des Übertragungsweges ist theoretisch nicht notwendig, wenn auch trotzdem empfehlenswert, um zum Beispiel die Verbindungsdaten zu schützen.

3.3 Gruppenzugriff auf verschlüsselte Dateien

Bei größeren Projekten müssen meist mehrere Mitarbeiter Zugriffsrechte auf gemeinsame Dateien haben. Da beim *EFS* individuelle Benutzerzertifikate für die Verschlüsselung verwendet werden, kann normalerweise nur derjenige, der die Daten verschlüsselt hat, auf diese auch zugreifen. Erst mit der Einführung von Windows XP hat Microsoft einen Mechanismus entwickelt, durch den es möglich ist, mehreren Benutzern – zusätzlich zum *Data Recovery Agent*, den es schon bei Windows 2000 gab – den Zugriff auf eine verschlüsselte Datei zu ermöglichen. Dabei wird einfach für jeden autorisierten Benutzer der *FEK* mit dem jeweiligen Zertifikat verschlüsselt und in einem eigenen *DDF* abgelegt. Dieser Vorgang ist unabhängig davon, ob die zusätzlichen Benutzer Schreib- bzw. Leserechte auf der Datei besitzen.

Voraussetzung für diese Funktionalität ist allerdings, dass entweder eine *CA* (*Certification Authority*) in der Domäne existiert, oder dass die entsprechenden öffentlichen Schlüssel der zu autorisierenden Benutzer im Zertifikatspeicher des Dateieigentümers sind. (Letzteres ist in größeren Umgebungen nicht praktikabel.)

Bei dieser Form der Autorisierung ergibt sich eine völlig neue Problematik, da jeder autorisierte Benutzer neue Benutzer autorisieren kann. Die einzige Möglichkeit, dies zu verhindern, ist es, den Mitbenutzern keine Schreibrechte auf die Datei zu geben, was allerdings nicht immer sinnvoll ist. In der Praxis muss man sich also darauf verlassen, dass mit der nötigen Sorgfalt entsprechende Rechte vergeben werden.

Einer ganzen Gruppe Zugriff auf verschlüsselte Daten zu gewähren, ist nur möglich, indem man jedes einzelne Gruppenmitglied dazu berechtigt. Diese Problematik ergibt sich durch den Umstand, dass keine Gruppenzertifikate existieren, und jedem Benutzer nur ein Zertifikat zur Verwendung des *EFS* zugeordnet werden kann.

3.4 Roaming Profiles

Eine gängige Praxis in Firmen ist es, für die Benutzer sogenannte *Roaming Profiles* einzurichten. Auf diese Weise ist es möglich von einem Arbeitsplatz zum anderen zu wechseln und dennoch die gewohnten Programmeinstellungen und Arbeitsoberfläche vorzufinden.

Mit dem sogenannten *User Profile* wird unter anderem auch der Ordner *Eigene Dateien*, der die persönlichen Dateien des Benutzers enthält, übertragen. Wenn man regelmäßig an verschiedenen Rechnern arbeitet, befindet sich nach einiger Zeit auf all diesen Computern eine mehr oder weni-

ger aktuelle Version dieser Dateien. Diese sind zwar normalerweise durch die Rechtevergabe des Betriebssystems vor dem Zugriff durch andere Benutzer geschützt, jemand mit lokalen Administrator-Rechten unterliegt diesen Beschränkungen allerdings nicht. Es wäre also wünschenswert, dass die Benutzerprofile sowohl auf dem Server als auch auf der lokalen Festplatte verschlüsselt werden können.

Das zur Ver- und Entschlüsselung benötigte Zertifikat wird allerdings ebenfalls im jeweiligen Benutzerprofil gespeichert. Dadurch ist es nicht möglich, dieses durch das *EFS* zu sichern. Microsoft empfiehlt daher entweder von der Verwendung von *Roaming Profiles* abzusehen oder zumindest den Ordner *Eigene Dateien* stattdessen verschlüsselt auf einem *File Share* oder einem *Web Folder* abzulegen. Einige weitere Ordner lassen sich auf diese Weise ebenfalls aus dem Benutzerprofil „auslagern".

Für mobile Benutzer mit einem Laptop ergibt sich dadurch ein neues Problem. Normalerweise hätten auch sie ohne Verbindung zum Firmennetzwerk Zugang zu ihrem lokal gespeicherten Benutzerprofil und damit zum Ordner *Eigene Dateien*. Sobald sie ihr Gerät wieder im Firmennetz betreiben, wird bei der nächsten An- bzw. Abmeldung das Profil wieder mit dem servergestützten abgeglichen. Liegt das Verzeichnis *Eigene Dateien* allerdings auf einem *File Share* oder *Web Folder*, hat ein Benutzer ohne Netzwerkverbindung auch keinen Zugriff darauf.

Aus diesem Grund hat Microsoft mit Windows 2000 die Funktionalität *Offline Files* eingeführt. Dabei wird – wahlweise beim Abmelden oder durch manuelles Auslösen – in einer Art Datenbank eine lokale Kopie von frei wählbaren Netzwerkfreigaben gemacht. Diese Freigaben – seien es nun *File Shares* oder *Web Folder* – können zwar verschlüsselt sein, die lokale Kopie ist es jedoch nicht. Erst mit Windows XP ermöglichte es Microsoft, auch die lokalen *Offline Files* zu verschlüsseln.

3.5 Backup

In größeren Firmen und Institutionen werden normalerweise regelmäßige Backups gefahren. Dabei werden häufig auch Verzeichnisse erfasst, die verschlüsselt sein sollen. Diese Verschlüsselung soll auch auf dem Backup-Medium erhalten bleiben.

Durch die Transparenz, mit der *EFS* arbeitet, und die enge Bindung an *NTFS* werden Dateien, die auf ein anderes Dateisystem – wie FAT32 bei Windows 9x oder EXT2 bei Linux – kopiert werden, automatisch entschlüsselt. Dieser Effekt würde dementsprechend auch bei Backups, zum Beispiel auf ein Bandlaufwerk, auftreten.

EFS bringt aus diesem Grund auch einen Systemaufruf *RAW* mit, der es ermöglicht, die Daten verschlüsselt – sozusagen im „rohen" Zustand – auf ein entsprechendes Medium zu schreiben. Es ist allerdings an den Herstellern von Backup-Software, diese Funktionalität auch zu implementieren. Das hauseigene Backup-Tool von Microsoft (*NTBackup*) bringt diese Fähigkeit bereits mit.

3.6 Temporäre Dateien

Bei der Bearbeitung von verschlüsselten Dateien werden von den beteiligten Programmen häufig temporäre Kopien oder andere relevante Daten enthaltende Dateien angelegt. Gerade bei einem Programm- oder Systemabsturz kann es vorkommen, dass diese eventuell unverschlüsselten Kopien nicht gelöscht werden und vielleicht sogar frei zugänglich auf dem lokalen Rechner liegen.

Derartige Probleme zu lösen wird häufig dadurch erschwert, dass die Kopien nicht im selben Verzeichnis wie die Originale stehen. Werden sie von der Software im selben Verzeichnis wie die ursprüngliche Datei angelegt und das entsprechende Verzeichnis ist als verschlüsselt markiert, werden auch die temporären Dateien darin verschlüsselt. Werden sie aber an einem anderen Ort abgelegt, bleibt nur die Hoffnung, dass dies entweder konfigurierbar ist, oder dass dieser Speicherplatz im systemüblichen temporären Verzeichnis liegt. Daher ist es empfehlenswert, dieses Verzeichnis ebenfalls zu verschlüsseln.

Wird ein Verzeichnis verschlüsselt, auf das mehrere Benutzer Zugriff haben, werden die darin enthaltenen Dateien jeweils mit dem zugehörigen Benutzerzertifikat verschlüsselt. Einem anderen als dem Besitzer ist der Zugriff damit nicht möglich. Auf diese Weise ist ein gemeinsames verschlüsseltes Verzeichnis für temporäre Dateien also möglich.

3.7 Drucken

Ein besonderes Augenmerk verdient in diesem Zusammenhang auch das Drucken. Aus Performanzgründen wird eine zu druckende Datei normalerweise zuerst in einem so genannten *Spool*-Verzeichnis für den Drucker vorformatiert abgelegt und anschließend erst gedruckt. Diese Datei wird normalerweise unmittelbar nach dem Druckvorgang wieder gelöscht, ist aber für einen gewissen Zeitraum eventuell unverschlüsselt verfügbar. Aus dem oben Genannten ergeben sich also zwei Lösungsmöglichkeiten: Die Deaktivierung des *Spoolings* oder die Verschlüsselung des *Spool*-Verzeichnis.

3.8 Allgemeine Dateioperationen

Im alltäglichen Gebrauch werden Dateien nicht nur editiert sondern auch verschoben, kopiert oder auch gelöscht. Obwohl durch die Möglichkeit, ganze Ordner als verschlüsselt zu kennzeichnen, ein anderer Eindruck erweckt wird, handelt es sich beim *EFS* um eine Verschlüsselung auf Dateiebene. Nicht das Verzeichnis wird verschlüsselt, sondern die darin enthaltenen einzelnen Dateien.

Selbst wenn ein Benutzer kein passendes Zertifikat zum Entschlüsseln, aber die notwendigen Schreib- und Leserechte besitzt, kann er sich den Inhalt – also zumindest Namen und Größe der enthaltenen Dateien – eines „verschlüsselten" Verzeichnisses ansehen und, die entsprechenden Rechte vorausgesetzt (zum Beispiel als lokaler Administrator), sogar verschlüsselte Dateien löschen. Verschieben oder Kopieren kann er hingegen nur, wenn er ein passendes Zertifikat besitzt. Man ist also weiterhin darauf angewiesen, die nötige Vorsicht bei der Rechtevergabe walten zu lassen, da Verschlüsselung und Schreib-Lese-Rechte unabhängig von einander vergeben werden können.

Ein weiteres Problem – vor allem unter Windows 2000 – ist der Unterschied zwischen dem Verschieben einer unverschlüsselten Datei in ein verschlüsseltes Verzeichnis und dem Kopieren der Datei in das Verzeichnis. Beim Kopieren wird die Datei effektiv im Zielverzeichnis neu angelegt und daher auch verschlüsselt. Beim Verschieben allerdings wird nur die Pfadangabe geändert, und daher wird die Datei im Zielverzeichnis auch nicht verschlüsselt. Wird eine Datei aus einem verschlüsselten Verzeichnis verschoben oder kopiert, so bleibt sie, und auch die Kopie, weiterhin verschlüsselt.

Erst mit Windows XP kam die Möglichkeit, verschlüsselte Dateien andersfarbig darzustellen und damit optisch erkennbar zu machen, ob eine Datei wirklich verschlüsselt ist oder nicht. Gerade unter Windows 2000 ist also generell Vorsicht beim Umgang mit verschlüsselten Dateien geboten.

4 Gefahren

Wenn man von Gefahren im Zusammenhang mit verschlüsselten Dateisystemen redet, kommen zwei Kategorien in Betracht: Gefahrenquellen, die eventuell zu Datenverlusten führen, und solche, die Unberechtigten Zugriff auf die zu schützenden Daten gewähren.

In die erste Kategorie fällt eigentlich nur der Verlust des eigenen privaten Schlüssels, der zur Entschlüsselung benötigt wird. In der zweiten Kategorie geht es zum einen um die eher theoretischen Angriffe auf die Verschlüsselung selbst, zum anderen um das Ausnutzen der Transparenz des *EFS*.

4.1 Verlust des privaten Schlüssels

Eine der größten Gefahren beim Einsatz von *EFS* ist der Verlust des eigenen privaten Schlüssels. Ohne das jeweilige Zertifikat ist eine Entschlüsselung der geschützten Dateien nicht mehr möglich. Es ist daher unbedingt empfehlenswert bei einem weiträumigen Einsatz von *EFS* mindestens einen *Recovery Agent* zu definieren, empfohlen werden von Microsoft mindestens zwei. Auch sollte der einzelne Benutzer nach Möglichkeit sein Zertifikat – inklusive privatem Schlüssel – mit einem ausreichend komplexen Passwort exportieren und sichern. Nur so können auf Dauer Datenverluste vermieden werden.

4.2 Zurücksetzen des Account-Passworts

Die einfachste Methode um die Verschlüsselung des *EFS* zu umgehen, besteht darin, sich die Transparenz des Systems zu Nutze zu machen. Gelingt es dem Angreifer, sich Zugang zum Account des jeweiligen Benutzers zu verschaffen, hat er auch vollständigen Zugriff auf alle von ihm verschlüsselten Dateien. Besonders gefährlich ist dies im Fall des *Recovery Agent*. Hat der Angreifer Zugriff auf dessen Konto, kann er **alle** entsprechend verschlüsselten Daten einsehen. Unter Windows 2000 ist es beispielsweise unter bestimmten Voraussetzungen möglich, von Startdiskette zu booten, das Zugriffspasswort zu ändern und anschließend auf die verschlüsselten Dateien zuzugreifen. Dieser Bug ging im Herbst 2002 als „EFS-Backdoor“ durch die Presse.

Um etwas Derartiges zu vermeiden, sollten vor allem Passwort-Richtlinien erlassen werden, die einfach zu erratende Passwörter verhindern. Außerdem sollte auf jeden Fall eine Domänenstruktur im Firmennetzwerk verwendet werden, da dann die Benutzerkennwörter nicht mehr in der *SAM*-Datenbank auf dem lokalen Rechner gespeichert werden, wodurch zum Beispiel Attacken mit Programmen wie *l0phtcrack* verhindert werden.

Des Weiteren sollte man darauf achten, regelmäßig etwaige Security-Patches auf allen Computern einzuspielen, um zu verhindern, dass ein Angreifer Lücken im Betriebssystem oder in der Anwendungs-Software ausnützen kann.

Zu guter Letzt sollte der private Schlüssel des *Recovery Agent* niemals auf einem Rechner gespeichert werden, sondern sicher auf einem anderen Medium – wie zum Beispiel einer CD-ROM – verwahrt werden, von dem er im Notfall importiert werden kann.

Fazit

Gerade in einer homogenen Windows 2000 bzw. Windows XP Umgebung bietet *EFS* eine kostengünstige Möglichkeit der Datei- und Verzeichnisverschlüsselung.

Wie bei den meisten *Out-of-the-Box*-Systemen ist eine nachträgliche Konfiguration allerdings unumgänglich, um eine ausreichende Sicherheit zu erreichen. Wichtig ist vor allem eine sinnvolle Passwort- und *Recovery*-Richtlinie. Wenn man als Betrieb stark auf Gruppenfunktionalitäten oder *Roaming Profiles* angewiesen ist und diese Bereiche auch verschlüsseln möchte, sollte man den Einschränkungen in diesen Gebieten besondere Aufmerksamkeit schenken.

Mit der Einführung von Windows XP wurden viele, wenn auch noch nicht alle, Schwachstellen im *EFS*, die in Windows 2000 noch vorhanden waren, beseitigt. Daher sollte zumindest für den Einsatz von *EFS* Windows XP bevorzugt werden.

Literatur

[1] Win 2000 Kompendium; Peter Monadjemi / Eric Tierling; Markt & Technik; 2000

[2] Microsoft TechNet

[3] Datei-Schlösser; Peter Dassow; c’t 15/2002 S. 202ff.

[4] Datenschutz bei Windows 2000; Ulrich Kühn, Dr. Sebastian Wirth; Hrg. Datenschutzbeauftragter Hamburg; 2002 sowie DuD 7/2002, S. 417-422.

[5] Guide to Securing Microsoft Windows 2000 Encrypting File System; Graham Bucholz, Harley Parkes; National Security Agency; Version 1.0 Januar 2001

[6] Dateiverschlüsselung von Windows 2000 unsicher; Andreas Marx, Andreas Beier, c’t 23/2002 S. 33