# Global Information Assurance Certification Paper

**Locking Lonely Laptop Data**

Robert F. Dienhart, MCSE
GSEC Practical Version 1.4b, Option 1
March 15, 2003

## Abstract

This paper addresses the issue of data protection in laptops for workgroup and individual road warrior environments. It assumes that a properly configured, and administered, Windows 2000 domain is not available. There is no grand goal to address all of the possible pitfalls and there is no discussion of defense against computer Viruses, Trojans, or network based intrusion attempts. The objective is to offer practical steps for wrapping several layers of protection around the data stored in a laptop to protect against unauthorized access.

The topic is divided into three major areas:

- Physical Security – Briefly touches on physically protecting a laptop, which is an essential part of defending one's data. It is not, however, the focus of this paper.
- Access Security – Presents practical steps for limiting access to a laptop's contents to those with proper credentials by focusing on the Authentication subsystem.
- Data Security – Addresses the permission structure related to file/data access, proper use of the NTFS Encrypting File System (EFS) to protect data in the event of unauthorized access, and a proposal for using PGP as an alternative for encryption that might be combined with EFS to capitalize on the strengths of both.

Only Windows 2000 Professional and Windows XP Professional are considered. Windows 9x systems are not discussed because, in their native modes, they offer no means of achieving any form of real Access or Data Security. Windows XP Home is not covered because EFS is not available on this platform.

## Background

> ". . . In a survey of 503 security professionals, 134 reported instances of laptop theft, with a dollar loss of $11,766,500--nearly $88,000 per incident, with the bulk of the damage from the loss of proprietary information." (DeMaria)

According to a 20 September 2000 article that appeared on a CNN.Com - Technology web page, the Computer Security Institute estimated 57 percent of member firms lost laptops to theft in 1999. [24] The insurance industry projected thefts numbering 319,000 laptops for the same year. 319,000. That is a lot of missing laptops.

At a consulting assignment, I was tasked with rebuilding the laptop for a city planner. The user toted this machine to meetings, made presentations with it, left it on his desk, probably left it in the car, and used it from home to access the office network. As I was installing Windows 2000 Professional and adding all the latest updates, security patches and needed applications, it occurred to me that, save for a weak password, the data on this laptop was not protected at all.

What might happen if this computer was lost and it contained highly proprietary data regarding corporate client financing? If this kind of information got into the wrong hands, the dollar implications to the city could be very high. The replacement value of a stolen laptop can be trivial when compared to the value of the data within.

**Physical Security**:

Computer security is best achieved through several *layers* of protection. Security experts refer to the concept as Defense in Depth.

A fundamental layer in protecting the data held in any computer, and a laptop in particular, is Physical Security. Numerous articles have been written about hardware and software to lock a laptop, alarm it, track it if stolen, and carry it securely during travel. There are papers in the SANS Reading Room [16, 19] about this as well as in print media and on the web. The DeMaria article [2] offers excellent information, including a number of related product reviews. Physical security would seem to be common sense, but thieves are opportunists and accidents happen.

Beyond the lock it, alarm it, and track it options, a step that can raise the machine from the lowest branches of vulnerability is disabling floppy and CD-ROM boot access. Do this in the BIOS and set a BIOS password, if the laptop supports that. This password, because of infrequent use, should be written down and locked up safely, and separately from the protected laptop. Alternatively, separate these boot devices from the laptop while traveling with it. Pack them in a separate bag – or even leave them at home or in the office.

Although trivial to overcome for a well-informed thief, blocking alternative boot access will at least prevent a casual attack on passwords and limit data access using bootable media. Administrators have been known to forget administrative passwords for workstations they support. Creative problem solvers, therefore, have developed "tools" – password cracking and modification utilities – to overcome such human weaknesses. While these are intended to be just that, administrative tools, in the wrong hands they become instruments of unauthorized access. [17,23,27]

Real security is a layered process. The above is only the beginning layer. It will frustrate and delay access to your data by a less skilled thief or one with time constraints, but it will not protect the data forever. Once the machine is in the possession of a thief, time is no longer an issue. They have all the time they need.

**Access Security**:

Following Physical Security, Access Security, or *Authentication*, is the next layer of defense. The components of Access Security are defined as:

1. Require Authentication (vs. the Auto Logon option)
2. Password and Authentication Policies
3. Usernames and Privileged Accounts
4. Passwords

**Require Authentication**:

Is Auto Logon enabled? Turn it off! It might be easier to start working after booting the machine at home or in a hotel room if logon credentials aren't required. However, is that convenience really worth it? Guess who else finds it easier to start working on the data in a stolen laptop. A thief only needs to turn the laptop on and they are in. Thank you. To turn off Auto Logon and require that users enter complete credentials at a logon screen, even when working at home, do the following:

- Windows 2000 Professional: My Computer > Properties > Network Identification tab > Network ID > Next > Accept "This computer is for home use . . ." > Next > Select "Users must enter a username and password . . ." > Click Finish

- Windows XP: Start > Run > In the Open box, enter the following "control userpasswords2" (without quotes) > Enter > Select "Users must enter a username and password . . . ". > OK

  Note: Disabling Auto Logon as described above assumes knowledge about configuring a user account and password as well as knowing the current administrator account password.

**Policies**

Windows 2000 and Windows XP both offer *Password Policies* and *Account Policies*. Password and Account Policies form *rules* that control a number of elements related to the authentication process. These elements include: require passwords, minimum acceptable password length, require password complexity, force periodic password changes, the number of failed logon attempts that will trigger an account lockout, and how long an account remains locked out before the lockout status is automatically cleared.

These policy settings can be found in both operating systems through the following path:

- Start > Settings > Control Panel > Administrative Tools > Local Security Policy > Account Policies > Password Policy/Account Policy

The Password and Account Policy names, default settings, and Center for Internet Security (CIS) recommended settings [26] are shown in the following table:

| **Policy** | **Default Setting** (Not Recommended) | **Recommended Setting** (CIS) |
|---|---|---|
| Enforce password History (Uniqueness) | 0 passwords remembered | 24 passwords remembered |
| Maximum password Age | 42 days | 90 days |
| Minimum password Age | 0 days | 1 day |
| Minimum password Length | 0 characters (password not required) | 8 |
| Passwords must meet complexity requirements | Disabled | Enabled |
| Store passwords using Reversible encryption | Disabled | Disabled (Note 2) |
| Account Lockout Duration | Not Defined | 15 Minutes |
| Account Lockout Threshold | 0 invalid logon attempts (Account will not lock out) | 3 invalid logon attempts |
| Reset account lockout counter after | Not defined | 15 minutes |

Note 1: Alternative references for these policy settings include “Password Policies” by Douglas Ludens [8] and “Windows 2000 Home User Self Defence Guide – Password Policies” (a UK site, hence the spelling of “Defence”) [25].

Note 2: Douglas Ludens's [8] recommendation for storing passwords with reversible encryption appears to be misinformed. According to the topic Windows XP Help and Support, this policy, which applies to both Windows 2000 and Windows XP, should not be used without good reason. An excerpt from the related help topic explains why:

> "This policy provides support for applications that use protocols that require knowledge of the user's password for authentication purposes. Storing passwords using reversible encryption is essentially the same as storing plaintext versions of the passwords. For this reason, this policy should never be enabled unless application requirements outweigh the need to protect password information." (Windows XP Built-in Help and Support: "Store Passwords Using Reversible Encryption for All Users in the Domain")

**Usernames and Privileged Accounts**

Choose usernames that are not obvious. There is a high probability that a business card will be with the laptop. Most corporate environments use predictable username configurations so guessing a functional username given access to a business card or other personal data will not be difficult. Again, good security involves multiple layers of defense. The *username* comprises one layer.

Apply the Local Security policy setting that disables display of the last logged on username. The less information a thief is gratuitously given, the harder they must work to achieve access and the more secure your data will be. Hiding a username will not do much to stop a determined thief with possession of the laptop from accessing the contents of the hard drive, but it will help thwart a casual or opportunistic thief. To set this policy:

- Windows 2000: Start > Settings > Control Panel > Administrative Tools > Local Security Policy > Local Policies > Security Options > "Do not display last user name in logon screen" > Enabled > OK

- Windows XP: Start > Settings > Control Panel > Local Security Policy > Local Policies > Security Options > "Interactive Logon: Do not display last user name" > Enabled > OK

Every Windows NT based system, and this includes Windows 2000 and Windows XP, includes a built-in *Administrator* account that is configured when the operating system is installed. This is a fully privileged account. Protection of the built-in Administrator account is essential because the local Account Lockout Policy settings will not protect it. To protect this account, apply the following steps:

1. Rename the Administrator account to something ordinary. Change the description of this account to "User account" or whatever standard description is used for other *un*privileged user accounts on the machine. Be consistent here so no single account stands out. To rename the administrator account correctly, use the local security policy:
    - Windows 2000: Start > Settings > Control Panel > Administrative Tools > Local Security Policy > Local Policies > Security Options > Rename Administrator Account > Enter the new name > OK
    - Windows XP: Start > Settings > Control Panel > Local Security Policy > Local Policies > Security Options > Accounts: Rename Administrator Account > Enter the new name > OK
2. Give this re-named administrator account a very long and complex password. A *passphrase* might even be appropriate (see "Good Passphrase Hygiene") [7]. Windows 2000 and Windows XP support passwords up to 127 characters long. Write this password or passphrase down and lock it up somewhere safe, away from the laptop and preferably at the office.
3. Create *unprivileged* user accounts for anyone that might use the machine. Give each user account a password that meets the guidelines identified above, and use a standard account description.
4. Create a second, unique, *privileged* account for each user that needs administrative access. Give these accounts ordinary names that cannot be easily differentiated as "special." Again, give each account a strong password that meets the Password Policy recommendations above. Users should never login with privileged accounts unless necessary. These privileged accounts, however, if under attack, *will* be protected by the Account Policy lockout settings. Never leave a privileged account logged in and unattended.
5. Always use the laptop in the login context of an unprivileged account. This should be the *normal* login context. If admin rights are occasionally needed to perform specific tasks, users should learn to use the power of the built-in RunAs command.
6. Create a new account named "Administrator" and give it the same description that the original built-in administrator account had. Assign this account, which will never really be used, a very strong password and put it in the Guests group where it will have minimal privileges.
7. Disable the built-in Guest account.
8. Set the screen saver to initiate automatically after 15 minutes and password lock it.

**Passwords**

Always use a password. Always use a strong password. Strong passwords look ugly and are hard to remember. Nevertheless, they are more secure because they are harder to guess and take longer to crack – "longer to crack" could mean months versus hours. Long, strong passwords are the best – but they are the hardest for users to remember and enter. Reality check – what is the data worth?

Try to make passwords at least eight characters long. [26] Moreover, never use dictionary words or things related to you like parts of your Social Security Number, your birth date, a pet's name, favorite sports team, the current month followed by the year, etc. [6]

A clever way to build a strong password that may be easier to remember is to use an *uncommon* quotation from a piece of literature and represent it with the first character of each word in that quotation. As an example, take this Sir Winston Churchill quotation and apply the suggested method:

"A fanatic is one who can't change his mind and won't change the subject."

This becomes:

afiowcchmawcts

as a password. Mixing in upper and lower case characters plus one or more non-alphabetic characters (e.g. *aFiowcchM&wct#!s*) will further strengthen this password – and meet the Password Complexity requirements identified in the recommended Password Policy settings. This password will look like gibberish to anyone who sees it. It will help deter over-the-shoulder password theft. In addition, it should give any password-cracking program a challenge. This means that it will be much harder for a thief to access data through a simple "guessed" authentication.

**Data Security**:

The next layer to consider is securing the data that is stored on the hard drive. This means making it harder for a thief to get at the data after stealing the laptop and managing to get authenticated to it. Again, the goal is security-in-depth.

**NTFS Permissions**

The focus here is effective use of the NTFS file system. NTFS offers several advantages over legacy FAT systems. The two most important of these are assignment of local access permissions and use of the Encrypting File System (EFS). Unlike share level permissions, NTFS permissions are able to limit access for both network-connected users and for those accessing the files and folders directly at the laptop. The NTFS permission structure is also much more granular and offers a greater level of control over access rights.

Both Windows 2000 and Windows XP Professional should default to the NTFS file system when the operating system is installed. However, FAT is available as an option and may have been chosen when the laptop was built. In addition, if the laptop was upgraded from an earlier windows operating system in which NTFS was not available, it may still be configured with a FAT file system. Take the time to verify what is being used. It's the data that is at stake.

To confirm what file system is in use, do the following in either system:

- My Computer > Properties > Disk Management > Check the column titled "File System". The number of drives/partitions configured on the computer and the related formats will be displayed. "NTFS" is the desired format.

Any drives or partitions currently configured as FAT or FAT32 can be easily converted to NTFS. Enter the following in a command screen:

convert *x*: /fs:ntfs

where "*x*:" is the drive that is to be converted. Make a backup before converting a volume because there is no built in way to reverse the process.

Consider converting each partition to NTFS if there is more than one, but certainly convert the partition that holds the operating system as well as the partition, or partitions, that hold the data. Partitions other than the active system partition (the "boot partition" in Microsoft parlance) will convert immediately. The active system partition will convert on the next boot cycle. Be forewarned that a dual boot machine with a Windows 9x boot option will no longer be able to access that OS following conversion to NTFS.

**"Everyone"**

Both Windows 2000 and Windows XP carry a built-in group called "Everyone" that is assigned NTFS access rights to virtually everything on the hard drive and to anyone who can gain access to the computer.

This is a potentially risky situation because the administrator of the computer has no control over the membership of the *Everyone* group. Membership is controlled by the Operating System and includes anyone who can get authenticated on the machine, including members of the "Guest" group. Default local access control is thus very loose in Windows 2000 because the *Everyone* group is assigned the Full Control permission. Windows XP (with SP1) has seen improvement, but the situation is still not great. In XP (with SP1), the default *Everyone* group is limited to Read and Execute permissions only. This is better, but it may be undesirable access just the same.

NTFS permissions follow an inheritance hierarchy. This means that a sub-folder automatically inherits the NTFS permissions assigned to a parent folder unless specific action is taken. The default in both Windows 2000 and Windows XP is for the *Everyone* group to be given access beginning at the root of a drive or partition. This means that every single sub-folder and file added to that drive or partition will inherit the permission attributes the *Everyone* group has. In Windows 2000, members of the *Everyone* group will have Full Control access to essentially the entire drive. In Windows XP (with SP1), members of the *Everyone* group will have Read and Execute access to essentially the entire drive. Either way, unless specific action is taken, "*Everyone*", which includes anyone that can gain a form of authentication on the laptop, will be able to access files on a drive.

So, be proactive. On any folder that contains sensitive data, or on which control over access is desired, disable the inheritance attribute. At the folder level where control is desired, do the following:

- Windows 2000: Select Folder > Properties > Security tab > Uncheck the box at the bottom titled "Allow inheritable permissions from parent . . ." > Select "Copy" at the next screen

- Windows XP: Select folder > Properties > Security tab > Advanced > Uncheck the box titled "Inherit from parent the permission entries . . ." > Select "Copy" at the next screen

  Note: Selecting "Copy" will retain all previously inherited permissions. The permission check boxes that were

originally grayed out, because the settings were inherited, will switch to white and allow changes.

Complete control of permission assignment is now enabled. At a minimum, replace the *Everyone* group with *Authenticated Users*. Even better, limit desired permissions to specific groups and individuals who should have access:

- Select Everyone > Click "Remove" > Click "Add" > Select specific user/users/group to have access > Assign desired access level > OK

  Note: In Windows XP, access to the Security tab, when looking at the Properties of a folder, may be hidden. This is typical on a machine that has been setup as a standalone or that is part of a workgroup rather than a domain. Such computers are configured with the default of "simple file sharing." Again, be proactive. Take control of the NTFS permissions being assigned to sensitive files and folders. To disable "simple file sharing" and enable access to the security tab where proactive control is available, do the following:

  - Start > Settings > control Panel > Folder Options > View tab > Uncheck "Use simple folder sharing (Recommended)"

What folders should be given this attention? Any folders created on the hard drive to store and organize data are good candidates. Limit access to these folders to specific accounts and groups for maximum control. The "My Documents" folder on a desktop is part of the user profile and is normally configured with tight security (access only by the specific user that belongs to the profile) by default. It would not hurt to verify that a user has not loosened security on this folder.

Permission changes made at a parent folder level will affect any new files/folders that are created within that folder. Permission changes will also be pushed down through the folder hierarchy and appear on existing subfolders and files on which inheritance is enabled. However, permissions that were explicitly assigned to sub-folders/files will not be changed unless this is forced. To force new permissions through out a folder hierarchy, including removal of any specifically configured settings, do the following:

- Windows 2000: Select Folder > Properties > Security tab > Advanced > "Reset permissions on all child objects . . ."

- Windows XP: Select Folder > Properties > Security tab > Advanced > "Replace permission entries on all child objects . . ."

**Data Encryption**

In any NT based operating system, a user account with administrator privileges can always access any file or folder and take ownership of the contents, regardless of the configured NTFS permissions. This capability will enable a thief to change (as in, loosen) any restrictive permissions by doing one of the following:

- Achieving authentication on the target using an illegally accessed administrator level account;
- Installing a second instance of a Windows 2000 or XP operating system on the target and using that system's administrator account;
- Physically removing the hard drive from the target and connecting it to a different computer with a suitable operating system on which the thief does have administrative rights.

This leads to the last layer of security, which is use of a data encryption mechanism. All of the prior steps that have been identified form an effective layered security strategy. However, they will merely slow a determined and skilled attacker's access to the data on stolen laptop. The only way to protect against a thief accessing data at this point is either to encrypt the data, or to simply not carry the data on the laptop. If the data isn't there, it can't be accessed.

Both the Windows 2000 Professional and Windows XP Professional operating systems offer a built in Encrypting File System (EFS). This provision by Microsoft is a boon for elevating the level of protection available at the local machine level. The benefit is provided as a self-signed, certificate-based data encryption process that performs block file encryption. It is quite fast, and it is completely transparent to the user. (The technical specifics of the EFS encryption feature are not a focus of this paper. See the Microsoft White Paper "Encrypting File System for Windows 2000" for information on how EFS operates. [13])

Encrypting a folder is accomplished the same way in both operating systems:

- Select the folder > Properties > Advanced >

  There are two options at this point: *Encrypt* and *Compress*. These are mutually exclusive which means that either Encryption *or* Compression can be selected but not both. *Neither* (both check boxes blank) is also a valid choice.

- Select *Encrypt contents to secure data* > Okay > Apply

A new window will appear with two more options: *Apply changes to this folder only* or *Apply changes to this folder, subfolders and files*. The second option is the better choice for the purpose of this discussion.

- Select *Apply changes to this folder, subfolders and files* > Okay

When using the built in EFS data encryption process, make it a point to apply the following *best practices* [4, 14, 28]:

- Encrypt folders and their contents rather than files
- Encrypt the "My Documents" folder
- Encrypt any other special data folders
- Encrypt the Print Spooler folder
- Encrypt the Temp Directory
- Configure the Local Security Policy to clear the Pagefile on Shutdown

There are, however, some caveats regarding use of the built-in Encrypting File System. Consider these carefully before deciding that EFS is the right solution for protecting sensitive data on a laptop:

1. Both operating systems will store a user's related Private Key, used to encrypt the actual File Encryption Key (FEK), in the user's profile (read "Registry") on the local machine. It *is not* possible to export a user's Private Key (copy it to removable media) and delete it from a standalone or workgroup member machine in Windows 2000. This means that an attacker, able to authenticate on the laptop with the user account by whatever means, will be able to decrypt and read all of the user's encrypted files. Windows XP Professional, on the other hand, *does* allow the export and removal of a user's Private Key on a standalone/workgroup member machine.
2. Windows 2000 automatically creates a second Private Key for the *Recovery Agent*. The Recovery Agent role is automatically assigned to the built-in Administrator account on any stand-alone or workgroup member computer and serves as an alternate route to unlock an encrypted file if the original Private Key is lost or becomes corrupt. It *is* possible to export the Recovery Agent's Private Key and delete it from the local machine in Windows 2000. This is good practice, particularly for mobile computers. [4, 14, 28.] The Recovery Agent is *good* in that it offers a second avenue for decrypting a file if the originator's private key is lost or becomes corrupt. However, it also provides a second Key that is susceptible to an attack on the account and that will yield

access to *all* encrypted data on the particular computer, regardless of the user who did the encrypting. The *best practice* is to export the Recovery Agent key to other media and remove it from the system.

3. Windows XP does not create a Recovery Agent by default. This is good in that only one Private Key per encrypted file is available, rather than two, in the event the computer is stolen and the data is under attack. It is a weakness if the originator's Private Key is lost or corrupted and not available for decrypting files. There is no practical avenue to recover the data in an encrypted file at this point.
4. One might consider removing the EFS encrypted data from the laptop by transferring it to removable media to protect it. Consider that EFS encrypted files cannot be transferred to offline media (e.g. floppy disk, or CD-R) in the encrypted state. Destination media must support the NTFS file system to support EFS. The only way to accomplish this is to back-up the encrypted data to a file (use Windows backup or similar), copy that backup file to the removable media, delete the original encrypted file from the laptop, and carry the media as separate from the laptop. There would then be *less* easily available data on a stolen laptop if successfully hacked, but the deleted data wouldn't be totally removed. (Files simply "deleted" are not "erased".) In order to access, and work with, the encrypted data, the backed up file must be "restored" to the hard drive using the same backup program. It could then be accessed normally. But, when it is time to secure the data again, it will have to be removed following the same process as before. This seems cumbersome for the typical laptop toting business traveler.
5. An alternative approach to protecting the EFS encrypted data is to export all of the Private Keys to removable media and delete them from the laptop. This process requires password protection of the exported keys on the media, that the media carrying the Keys not be with the laptop when being transported, and that this media be carefully protected. However, to access the data, the required Private Key must be imported to be restored for use, and then deleted again, before the laptop is on the move. Another cumbersome process. (Except for the built-in administrator account, exporting a user's Private Key is *not possible* in Windows 2000 on a stand-alone or workgroup member laptop.)
6. Further complications include:
    a. An EFS encrypted file sent across a network will be decrypted first. It then travels across the network in an unencrypted state. EFS does not support encryption-on-the-wire.
    b. EFS encrypted files copied to FAT based partitions or sent in an email are decrypted. If the destination in a network file

transfer is not an NTFS based file system, the copy at the destination will remain unencrypted.

The Encrypting File System does indeed provide a valuable benefit with its built-in, certificate-based, easy to use data encryption process. However, as discussed, it comes with potentially serious limitations when applied to a stand-alone or workgroup member laptop.

EFS works best in a Windows 2000 domain environment. [1] In a properly configured domain, Group Policy can be used to control the use of EFS. A carefully managed Certificate Authority (CA) will issue encryption and recovery certificates. Private Keys used for encryption and recovery based on CA issued certificates become part of a proper backup discipline which guards against loss and possible corruption. They can also be exported and removed from any domain member computer, whether 2000 or XP. This is certainly the better system architecture considering the importance of the encryption and recovery keys. It is not the kind of system architecture, however, that is likely to be found in the typical sole proprietorship or small company.

If the Encrypting File System that is part of Windows 2000 Professional and Windows XP Professional seems impractical for the individual or small business user, what option is there for this final layer of security?

Consider Pretty Good Privacy, better known as "PGP." PGP has been evolving since at least 1991. [29] It is a highly developed, well-documented, strongly supported product. Both commercial and free versions are available for the enterprise as well as for the individual. [5,15,21] The encryption algorithms used are based on established technical standards and proven encryption solutions whose source code is available for peer review.

PGP encryption offers end-to-end security that is not dependent on the underlying operating system or file structure. It works with Windows 9x products, Windows NT4, Windows 2000 and Windows XP – both Home and Professional. It is compatible with FAT16, FAT32 and NTFS formatted hard drives, with floppy disks, CD-ROMS, and USB based external storage systems, including the Flash memory based portable "hard drives."

It is not dependent on the binary identity of a user (e.g. Windows 2000 user SID) and a related private key that could be lost or damaged or compromised.

PGP uses a *passphrase* to generate the encryption key. The passphrase is not automatically stored in a user's profile or anywhere else on the computer, unless, of course, the user is undisciplined enough to put it there. The passphrase, like a password, is ideally stored in the user's memory. Give PGP the same passphrase used to encrypt the data, get the unencrypted data back. There are no *keys* to, administratively, deal with.

PGP encrypted data can be sent across a network, emailed to an associate, copied from an NTFS formatted partition to a FAT formatted partition and back again. It can be transferred to a floppy disk, an external hard drive, or a flash memory based thumb drive to *securely* separate sensitive data from a theft susceptible laptop. All can be done without concern about the state of encryption after the action is taken. If the data moved was encrypted at the source, it will be encrypted at the destination, unless, of course, specific action is taken to decrypt the data prior to transfer.

The quality of the *passphrase* used to encrypt data in PGP is central to the security of the protected data. Unlike the password used to authenticate to the system hosting EFS, there are no *Policies* to enforce "good" passphrase discipline. Fortunately, there are guidelines and tools to help with the creation of an effective passphrase, just as there are guidelines and tools to guide the creation of an effective password. [3, 7, 22]

The use of PGP based encryption, then, seems more practical for the isolated laptop user than is the use of EFS. It certainly seems simpler to use, from the standpoint of protecting the encryption/decryption status of sensitive data and the safety of related Keys.

Are there shortcomings to PGP? Of course there are:

1. The encryption/decryption process is not transparent to the user. Direct interaction by the user, following authentication, is necessary to access and work with PGP encrypted data. This interaction isn't necessarily difficult. The purchased product from PGP Corporation makes it easy to configure one or more folders or *virtual drives* for storage of encrypted data. These containers can be configured to request *mounting* at authentication, at which point the passphrase is needed, and to automatically *unmount* (close and lock) after a definable period of inactivity. Anything put in one of these containers when open will be encrypted and protected when the container is closed.
2. Unlike EFS, in which one can simply drag-and-drop a file into an EFS encrypted folder to encrypt the file, an object cannot be drag-and-dropped into an *unmounted* container.
3. The Encrypting File System (EFS) is native to the Windows 2000 and Windows XP Professional operating systems and requires no additional software. PGP requires the installation of third party software.
4. It may not be practical to apply PGP encryption to the Print Spooler, Temp Directory, and My Documents folders.

Where do these differences leave one if interested in protecting the data used in a standalone or workgroup member laptop?

Perhaps the best approach is avoiding exclusive reliance on one or the other. A combination of EFS and PGP may make an acceptable hybrid. Use EFS where its strengths make sense – protection of system files (Print Spooler, Temp Directory, etc) and less sensitive data, where loss of, or compromise of, a Private Key does not spell the end of the world. Use PGP where it shines – easier export of encrypted data to removable media in a portable, accessible, protected format that separates the data from the laptop. A copy of the PGP program could even be stored on the media with the encrypted data. The passphrase, stored in the user's memory, is what protects the data, not the absence of the PGP program used or the authentication credentials in a particular computer. In addition, if the laptop is stolen or lost, the data can still be accessed and used, albeit in a different machine.

## Summary

The value of a stolen laptop computer is very tangible when considering the price of its physical replacement. The cost may be uncomfortable to bear but is certainly finite. Products are available to protect against physical theft yet the undesirable may still happen. The value of data that is lost in a stolen laptop, however, is more difficult to assess. It may be trivial; in which case replacing the laptop is a financial inconvenience. On the other hand, the value of the data may border on priceless when one considers the time invested in creating the data or the potential fallout that can ensue if sensitive data gets into the wrong hands. In this case, the price of replacing the laptop is irrelevant – the data that was in it is the gem and is gone.

Layers of security settings, intended to bring a lone laptop up to a reasonable degree of protection, have been discussed. Most of these fall into the realm of "best practices" and focus on controlling legitimate access to a laptop and the data on it.

However, as one of The Ten Immutable Laws of Security states, "If a bad guy has unrestricted physical access to your computer, it's not your computer anymore." (Microsoft Technet) Discussion then focused on the use of a file encryption process to protect the data in the laptop.

Windows 2000 and Windows XP Professional both offer a native Encrypting File System. It is easy to use, fast, and effective on the surface. However, there are some potential "gotchas" when the hood is lifted and consideration is given to the all-important Private Keys. Also, exporting EFS encrypted data to external media is less than convenient.

PGP, an alternative encryption process, was then presented. While very straightforward to use, it is a third party application rather than a native part of an operating system. This has good points and bad. The benefits of PGP include reliance on a strong passphrase, that is not stored anywhere on the local machine, rather than potentially weaker authentication credentials that are stored in the local Registry and are susceptible to cracking. Additionally, PGP encrypted files are more readily compatible with portable media. PGP, however, lacks the transparency of use available in Windows based EFS.

The final argument is that laptops are stolen and the data on them is at risk. Following the theft of a laptop, all of the layers of security will impede, but never thwart, a thief determined to access the data. The best protection against a thief accessing critical data on a stolen laptop is the combination of a good security configuration, use of both EFS and PGP for encryption (each in its place), and to consider carrying the critical data, in encrypted form, apart from the laptop.

**Works Cited:**


1. Bragg, Roberta. "Applied Cryptography – Hardening EFS." Information Security. February 2001. URL: http://www.infosecuritymag.com/articles/february01/features_applied_crypto.shtml (15 March 2003)

2. DeMaria, Mike. "Gone in 6.0 Seconds." Network Computing Magazine. September 30, 2002. URL: http://www.networkcomputing.com/1320/1320f4.html (15 March 2003)

3. Diceware Passphrase Homepage. URL: http://world.std.com/~reinhold/diceware.html (15 March 2003)

4. Fasching, Chuck "Spence". "A Discussion of Best Practices for Microsoft's Encrypted File System." July 28, 2001. http://www.sans.org/rr/win2000/EFS.php (15 March 2003)

5. Freeware PGP Versions. URL: http://www.pgpi.org/products/pgp/versions/freeware/ (15 March 2003)

6. "Good and Bad Passwords How-To." GeodSoft's Website Consulting. URL: http://geodsoft.com/howto/password/ (15 March 2003)

7. Good Passphrase Hygiene. URL: http://security.tao.ca/pswdhygn.shtml (15 March 2003)

8. Ludens, Douglas. "Password Policies." About.com - Focus on Windows. URL: http://windows.about.com/library/weekly/aa000910a.htm (15 March 2003)

9. Microsoft. "Checklist: Create Strong Passwords." April 2002. URL: http://www.microsoft.com/security/articles/password.asp (15 March 2003)

10. Microsoft Technet. "5-Minute Security Advisor – The Road Warrior's Guide to Laptop Protection." URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/columns/security/5min/5min-205.asp (15 March 2003)

11. Microsoft Technet. "Best Practices for Encrypting File System." Knowledge Base Article 223316. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q223316&sd=tech (15 March 2003)

12. Microsoft Technet. "The Ten Immutable Laws of Computer Security." URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/columns/security/essays/10imlaws.asp (15 March 2003)

13. Microsoft White Paper. "Encrypting File System for Windows 2000." URL: http://www.microsoft.com/windows2000/docs/encrypt.doc (15 March 2003)

14. Microsoft Technet. "Increasing Security for Open Encrypted Files." URL: http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/winxppro/reskit/prnb_efs_qyxz.asp (15 March 2003)

15. MIT Distribution Site for PGP. URL: http://web.mit.edu/network/pgp.html (15 March 2003)

16. Mueller, Andrew. "Laptop Security: Past, Present." July 10, 2001. URL: http://www.sans.org/rr/travel/sec_revisited.php (15 March 2003)

17. Offline NT Password & Registry Editor. URL: http://home.eunet.no/~pnordahl/ntpasswd/ (15 March 2003)

18. "Operating Systems. Windows Security." Ernest Orlando Lawrence Berkley National Laboratory. Computer Protection Program. URL: http://www.lbl.gov/ICSD/Security/systems/windows.html (15 March 2003)

19. Palmer, Thomas. "Basic Travel Security Revisited." August 6, 2001. URL: http://www.sans.org/rr/travel/sec_revisited.php (15 March 2003)

20. Lemos, Rob. "Passwords: The Weakest Link? Hackers can Crack Most in Less Than a Minute." Cnet News.com. May 22, 2002. URL: http://news.com.com/2009-1001-916719.html (15 March 2003)

21. PGP Corporation. PGP Products – Key Features. URL: http://www.pgp.com/products/ (15 March 2003)

22. Random Passphrase Generator. URL: http://www.leemon.com/crypto/MakePass.html (15 March 2003)

23. Sysinternals. NTFSDOS Freeware. URL: http://www.sysinternals.com/ntw2k/freeware/NTFSDOS.shtml (15 March 2003)

24. Vincent, Christie; Vaughan, Jack. "Security Experts Seek to Combat Laptop Theft." CNN.com Technology. September 20, 2000. URL: http://www.cnn.com/2000/TECH/computing/09/20/laptop.security.idg/ (15 March 2003)

25. "Windows 2000 Home User Self Defence Guide – Password Policies." URL: http://www.uksecurityonline.com/husdg/windows2000/passwordpolicy.htm (15 March 2003)

26. “Windows 2000 Professional Benchmark – Consensus Baseline.” Center for Internet Security. (CIS) URL: http://www.cisecurity.org/ (15 March 2003)

27. Windows XP Administrator Password Lost !!! URL: http://www.experts-exchange.com/Operating_Systems/WinXP/Q_20348448.html (15 March 2003)

28. Valentine, Kayron C. "Encrypting File System Primer: Basics and Best Practices". July 6, 2001. URL: http://www.sans.org/rr/win2000/EFS_primer.php (15 March 2003)

29. Back, Adam. "PGP Timeline." URL: http://www.cypherspace.org/~adam/timeline/ (15 March 2003)

Last Updated: July 25th, 2017

# Upcoming Training


| | | | |
|---|---|---|---|
| SANS Prague 2017 | Prague, Czech Republic | Aug 07, 2017 - Aug 12, 2017 | Live Event |
| SANS Boston 2017 | Boston, MA | Aug 07, 2017 - Aug 12, 2017 | Live Event |
| SANS Salt Lake City 2017 | Salt Lake City, UT | Aug 14, 2017 - Aug 19, 2017 | Live Event |
| Community SANS Omaha SEC401* | Omaha, NE | Aug 14, 2017 - Aug 19, 2017 | Community SANS |
| SANS New York City 2017 | New York City, NY | Aug 14, 2017 - Aug 19, 2017 | Live Event |
| SANS Chicago 2017 | Chicago, IL | Aug 21, 2017 - Aug 26, 2017 | Live Event |
| SANS Virginia Beach 2017 | Virginia Beach, VA | Aug 21, 2017 - Sep 01, 2017 | Live Event |
| SANS Adelaide 2017 | Adelaide, Australia | Aug 21, 2017 - Aug 26, 2017 | Live Event |
| Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style | Virginia Beach, VA | Aug 21, 2017 - Aug 26, 2017 | vLive |
| Community SANS Pasadena SEC401 @ NASA | Pasadena, CA | Aug 23, 2017 - Aug 30, 2017 | Community SANS |
| Mentor Session - SEC401 | Minneapolis, MN | Aug 29, 2017 - Oct 10, 2017 | Mentor |
| SANS San Francisco Fall 2017 | San Francisco, CA | Sep 05, 2017 - Sep 10, 2017 | Live Event |
| SANS Tampa - Clearwater 2017 | Clearwater, FL | Sep 05, 2017 - Sep 10, 2017 | Live Event |
| Mentor Session - SEC401 | Edmonton, AB | Sep 06, 2017 - Oct 18, 2017 | Mentor |
| SANS Network Security 2017 | Las Vegas, NV | Sep 10, 2017 - Sep 17, 2017 | Live Event |
| Community SANS Albany SEC401 | Albany, NY | Sep 11, 2017 - Sep 16, 2017 | Community SANS |
| Mentor Session - SEC401 | Ventura, CA | Sep 11, 2017 - Oct 12, 2017 | Mentor |
| Community SANS Columbia SEC401 | Columbia, MD | Sep 18, 2017 - Sep 23, 2017 | Community SANS |
| Community SANS Dallas SEC401 | Dallas, TX | Sep 18, 2017 - Sep 23, 2017 | Community SANS |
| Community SANS Boise SEC401 | Boise, ID | Sep 25, 2017 - Sep 30, 2017 | Community SANS |
| Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style | Baltimore, MD | Sep 25, 2017 - Sep 30, 2017 | vLive |
| Community SANS New York SEC401 | New York, NY | Sep 25, 2017 - Sep 30, 2017 | Community SANS |
| Rocky Mountain Fall 2017 | Denver, CO | Sep 25, 2017 - Sep 30, 2017 | Live Event |
| SANS London September 2017 | London, United Kingdom | Sep 25, 2017 - Sep 30, 2017 | Live Event |
| SANS Baltimore Fall 2017 | Baltimore, MD | Sep 25, 2017 - Sep 30, 2017 | Live Event |
| SANS Copenhagen 2017 | Copenhagen, Denmark | Sep 25, 2017 - Sep 30, 2017 | Live Event |
| Community SANS Sacramento SEC401 | Sacramento, CA | Oct 02, 2017 - Oct 07, 2017 | Community SANS |
| SANS DFIR Prague 2017 | Prague, Czech Republic | Oct 02, 2017 - Oct 08, 2017 | Live Event |
| Community SANS Charleston SEC401 | Charleston, SC | Oct 02, 2017 - Oct 07, 2017 | Community SANS |
| Mentor Session - SEC401 | Arlington, VA | Oct 04, 2017 - Nov 15, 2017 | Mentor |
| SANS October Singapore 2017 | Singapore, Singapore | Oct 09, 2017 - Oct 28, 2017 | Live Event |