

2008

A Computer Forensic Response to Hard Drive Encryption

Joseph Daniel Painter
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Painter, Joseph Daniel, "A Computer Forensic Response to Hard Drive Encryption" (2008). *Graduate Theses, Dissertations, and Problem Reports*. 4412.
<https://researchrepository.wvu.edu/etd/4412>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

A Computer Forensic Response to Hard Drive Encryption

Joseph Daniel Painter

**Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of**

Master of Science in Computer Science

Dr. Roy Nutter, Ph.D., Chair

Dr. Bojan Cukic, Ph.D.

Dr. Katerina Goseva-Popstojanova, Ph.D

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia

2008

Keywords: TPM, Encryption, Hard Drive Encryption, Computer Forensics

Copyright 2008 Joseph Daniel Painter

Abstract:

A Computer Forensic Response to Hard Drive Encryption

Joseph Daniel Painter

Standard procedures used by law enforcement for evidence preservation require removing power from the computer and going further analysis in a digital forensics laboratory. With the advent of easy to use, highly accessible hard drive encryption, access to potential evidence becomes much more volatile problem than in the past. This paper will look at several encryption methods available on the commercial market, discuss their impact on current best practices for hard drive recovery, and propose new procedures to deal with the possibility of encrypted information on a target computer.

Table of Contents

Abstract: ii

 Figures Table vi

Problem Statement: 1

1 Introduction: 1

 1.1 TPM Background 2

 1.2 TPM Application..... 4

 1.3 Keys..... 6

 1.4 Hardware Encryption: Full Disk Encryption 8

 1.4.1 Trusted drives 9

 1.5 Hybrid Encryption: BitLocker 9

 1.5.1 Transparent Operation Mode10

 1.5.2 User Authentication Mode10

 1.5.3 USB Key Mode10

 1.6 Software Based Encryption: TrueCrypt.....11

2 Methods of Analysis and Recovery12

 2.1 Software Based Encryption Analysis12

 2.2 Hardware Based Full Disk Encryption Analysis14

 2.3 TPM and TPM Based Encrypted Volume Analysis16

 2.4 BitLocker Analysis.....17

3 Final Issue.....18

4 Conclusion.....19

5 Appendix A: RSA Encryption and signing	19
6 Appendix B: TPM in Vista	20
6.1 TPM Management Control Console.....	20
6.2 Turn on TPM	22
6.3 Activating the TPM (taking ownership)	25
6.4 Turn TPM off.....	32
6.5 Change owner password.....	34
6.6 Clear TPM	34
6.7 Command Management: tpm.msc.....	35
7 Appendix C: Third Party TPM Management	36
7.1 Outside programs	36
7.2 Enable and Activate the TPM	36
7.3 Establish Ownership of the TPM	37
7.4 Configure Document Manager (Vault).....	40
7.5 Back Up the TPM Encryption Keys	44
8 Appendix D: Embassy Trusted Drive Manager	46
8.1 Additional Windows Login Options	49
9 Appendix E: Windows BitLocker Disk Encryption.....	50
9.1 BitLocker Encryption Analysis without TPM	50
9.2 BitLocker Encryption Analysis with TPM	54
9.3 Using Recovery Keys	55
10 Appendix F: Expanded TrueCrypt Experiments	56
10.1 Standard TrueCrypt Volume Analysis:	56
10.2 Hidden TrueCrypt Volume Analysis:.....	58

10.3 Full Drive Encryption Analysis:	59
11 Appendix G: Computer Forensic Encryption Checklist	59
12 Appendix H: Experiments	61
12.1 TPM Verification	61
12.2 TrueCrypt Test 1	64
12.3 TrueCrypt Test 2	66
12.4 TrueCrypt Tests 3	68
12.5 Vault Tests 1 (TPM based encryption)	70
12.6 Vault Tests 2 (TPM based encryption)	74
12.7 Vault Tests 3 (TPM based encryption)	77
12.8 BitLocker Tests	79
12.9 Hardware Based Full Disk Encryption Tests	81
References	83

Figures Table

Figure 1: Sample Boot Process [3]	5
Figure 2: tpm.msc	21
Figure 3: run tpm.msc as administrator	22
Figure 4: TPM Unowned and turned off	24
Figure 5: TPM unowned and turned on	27
Figure 6: Initialization of TPM- password creation	27
Figure 7: Initialization of TPM: password saving and printing	28
Figure 8: TPM owner password backup file	29
Figure 9: TPM initialization complete	29
Figure 10: TPM initialization complete (cont.)	30
Figure 11: TPM owned and turned on	31
Figure 12: Turn off the TPM	32
Figure 13: TPM owned and turned off	33
Figure 14: Clear the TPM	35
Figure 15: TPM not enabled	38
Figure 16: TPM enabled and in the process of being owned	39
Figure 17: TPM enabled and owned	40
Figure 18: Opened Vault with Text File	41
Figure 19: Vault options when locked (user logged out)	42
Figure 20: Vault options when unlocked (user logged in)	43
Figure 21: Creating a key archive	45
Figure 22: Trusted Drive Main	47
Figure 23: Trusted Drive Advanced Options	48

Figure 24: Encrypted hard drive password backup file49

Figure 25: Operating System Name and Version.....50

Figure 26: Windows BitLocker Volume51

Figure 27: Run cmd.exe as Administrator52

Figure 28: Manage-bde.wsf53

Figure 29: BitLocker recovery key53

Figure 30: Mounting an Encrypted Volume One.....57

Figure 31: Mounting an Encrypted Volume Two58

Problem Statement:

Hard drive encryption has become a hurdle to digital forensics. Data protected by BitLocker, hardware full disk encryption, software based encryption, and trusted platform modules is now becoming a problem to law enforcement during computer forensic investigations. This paper analyzes the issues that now face the field of computer forensics, due to hard drive encryption, and provides methods to recover data protected by the different methods of encryption.

1Introduction:

With the popularization of disk encryption methods and of new operating systems, such as Windows Vista, current evidence collection and analysis methods, used by law enforcement, no longer should be considered as best practice. At a crime scene, it is common practice to acquire the computer by removing the power cable from the back of desktop machine and tagging the machine as evidence. In the case of a laptop, the typical procedure is to remove the battery pack from the laptop and then tag the machine as evidence. The purpose of removing power to the machine is to “preserve evidence”. Once the machine is powered down, the typical procedure is to make an image of the hard drive. The image is then analyzed in a laboratory.

With hard drive encryption becoming more common, evidence on a drive can easily be lost if normal procedures are followed. In most cases, the only option for collecting possible encrypted evidence is live analysis, before the machine is turned off. In many cases, if the machine remains powered up, encrypted files continue to be readable [4]. Once the machine is powered down, the keys to the encrypted data are discarded from memory. Thus, the encrypted files are lost to the investigator.

Although making a bit-wise image of a powered down hard drive, with encryption enabled, is still possible, the drive's image is effectively worthless as there are no keys to decrypt the encrypted data.

Currently, there are three major ways in which files can be encrypted: software encryption, hardware encryption, and a software / hardware hybrid encryption. The first such encryption technique is a software based encryption, such as the open source program TrueCrypt[7]. Software based encryption is somewhat slow, but is rather secure and requires a facilitating program. Since the encryption is based in software, the encrypted files are tied to a set of encryption keys, not necessarily tied to a specific machine. The second type of encryption is hardware based encryption. Recently, some hard drive manufactures, such as Seagate, have developed hardware based full disk encryption. Hardware, located on the disk drive itself, manages the encryption / decryption process, resulting in strong data security without sacrificing speed. In hardware encryption, the entire drive including the boot sector, operating system, temporary files, and swap files are all encrypted. The final type of encryption is a hybrid software/hardware encryption such as Windows BitLocker. The hybrid encryption uses software to encrypt but can be tied to a specific hardware configuration as well. Hardware and hybrid encryption techniques make use of a TPM (Trusted Platform Module) chip, which can tie a machine's hard drive, at a hardware level, to the TPM chip on the motherboard. Each type of encryption will be discussed and analyzed in further detail, including techniques and recommendations to law enforcement for dealing with each respective encryption method.

1.1 TPM Background

Trusted Platform Module (TPM) chips are the result of collaboration between industry representatives who have worked over several years developing the TPM standards. This collaborative group, referred to as the Trusted Computing Group, is comprised of representatives from both hardware manufacturers and software developers. These representatives include, among others,

AMD, Hewlett-Packard, IBM, Infineon, Intel, Lenovo Holdings Limited, Microsoft, and Sun Microsystems. The goal of the Trusted Computing Group is to create a set of standards that allow chip manufacturers, computer manufacturers, and disk manufacturers to lock data, using hardware that is built into the computer when the computer is manufactured. This provides a computer user with a high level of security, protecting users' information assets from compromise, while providing a high level of privacy [4].

All TPM chips that are manufactured and sold thus should conform to a set of standards defined by the TCG. The TCG standards not only define the hardware specifications of the chip, but also outline specific functionality that must be supported by the chip. In the original standards, referred to as 1.1b, the required functionality for each chip was defined by the TCG. However, the decision of how to implement the required functionality was left up to the individual TPM chip manufacturer. As a result, any programs developed to use a TPM chip had to be designed around a specific chip's architecture. This lack of a uniform chip implementation led to confusion among developers, code compatibility issues, and reduced the "real world" effectiveness of the chips. In order to promote development and make the chips more appealing for use, the group developed the Version 1.2 specifications [1].

The Version 1.2 specifications refined and expanded the Version 1.1b specifications in two major ways. First, the 1.2 specifications define how a TPM chip must implement some parts of functionality from a software point of view. In this regard, all 1.2 chips use the same programming interface for input and output. How the chip implements each function internally is still more or less up to the manufacturer. In this way, programs can be developed for all 1.2 TPM chips [1]. In the 1.2 specifications, the TCG also expanded the functionality of the chips to provide more uses and improve the protection of user privacy, even more than before [2].

Though TPM chips are now being manufactured under 1.2 specifications, the specifications are not yet complete and are continuing to evolve. Therefore, developers typically are only using the functions provided in 1.2 that are backwards compatible with 1.1b [4].

1.2 TPM Application

The functionality of TPM chips is rather open ended. The TCG wanted the chips to be able to be used under any number of circumstances and by any number of programs. With regards to this paper, three commonly used functions, important to law enforcement, will be analyzed. These functions are the identification of other trusted systems, sealing a system, and file encryption.

The ability to identify trusted systems encompasses both hardware and software. Hardware identification comes in several forms with TPM chips. At a local level, it is the goal of the TCG for TPM chips to be used in any major component of a computer system. The chips would be able to interact and verify one another for further communication. For example, if a TPM chip on a motherboard was programmed only to allow certain hard drives to be used, whenever a hard drive would be plugged into the system, the motherboard could verify or reject the drive based on the information stored on the TPM chip located on the hard drive. Likewise, the hard drive could verify or reject the motherboard. In this way the system could be sure that all communication between the hard drive and the motherboard would be secure and legitimate. Another hardware implementation of system identification would be remote computer usage. A system would only remotely communicate with another system if it could be identified as secure by the agreement of chips in both computers. From the software perspective, a file could be coded in such a way that only authorized programs (verified with the TPM) could access it [1].

The concept of “sealing” a system, using TPM, seems relatively simple but is harder to implement. A system is sealed by measuring, or hashing, different pieces of the system and storing

the resulting hashes on the TPM chip in areas called Platform Configuration Registers (PCRs). Then, when the system is restarted, a hand-off style boot is performed. The boot process is a hand-off due to the fact that before each step of the process, a hash is taken and compared against the hash value stored on the TPM for that step. Using Figure 1 as an example, the process can be more easily explained. The boot process starts with the “Core Root of Trust Measurement” (CRTM). This area is always considered to be trust worthy during the boot process. Before moving to the second stage, the BIOS is hashed (arrow 1) and compared to the stored value in the platform configuration registers (arrow 2)(recall that the PCR is inside the TPM chip). If the value matches, the boot process allows the BIOS to take control (arrow 3) and proceed with booting. If the value does not match, at any point, the system will not boot [3].

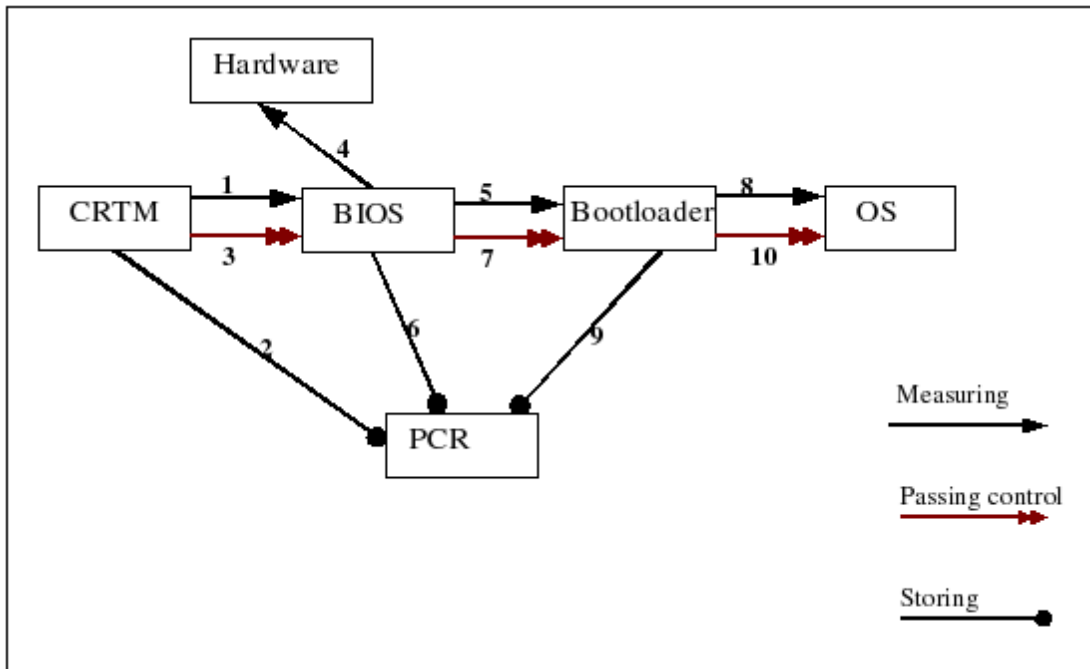


Figure 1: Sample Boot Process [3]

When at the BIOS, the systems hardware and boot loader are hashed (arrows 4 and 5) and again compared to values stored in the PCRs (arrow 6). This part of the process allows the system to

verify that no hardware has been added or removed from the system and potentially compromised. At the same time, the integrity of the boot loader is confirmed to be free of malicious code. This process of measuring and comparing continues until the system is able to finally load the operating system [3].

The final major use for TPM, as discussed by this document, is file encryption. However, with the introduction of TPM chips, files can be secured in a much stronger fashion. When a file is encrypted or decrypted using TPM, the encryption and decryption is done completely within the TPM chip itself. In this way, the encryption key used by the chip cannot be easily found or seen “in the clear”. The encryption and decryption process is independent of both the main processor and RAM.

1.3 Keys

In order to fulfill the requirements of trust, verification, and privacy, TPM chips use a series of different encryption keys. Each action taken by a TPM chip is associated with a set of keys. The style of key generated and used by the chip is based upon the RSA algorithm, discussed in appendix A.

Keys stored on a TPM chip are classified as either migratable or non-migratable. When a key is classified as migratable it means that it is possible to transfer the key from one TPM chip to another. Migratable keys are used for things such as back-ups, in case of hardware malfunction. It should be noted that while migrating a key, privacy is still protected at all times as the keys are never seen “in the clear”. When a key is classified as non-migratable it can never leave the chip. As such, if a file had to be decrypted using a non-migratable key and the chip failed, the file would be lost forever [2].

The two main categories of encryption are symmetric and asymmetric key systems. In a symmetric key system, the same key that is used to encrypt a file is also used to decrypt it. A symmetric key system is analogous to locking a door with a simple dead bolt. Unlocking or locking the door requires a single key. A problem with this system is that it is possible for multiple copies of a

key to exist for the same lock. So, if a key is compromised, it can be used to steal information without detection. To further complicate matters, a symmetric key system, by definition, requires keys to be distributed to each entity in the system, providing greater opportunity for keys to be stolen. Conversely, in an asymmetric system, key pairs are used to encrypt and decrypt information. These pairs work in a one way only fashion. For example, if key “A” is used to encrypt (lock) a file, only key “B” can be used to decrypt (unlock) the file. In the same way, if key “B” encrypts a file, only key “A” can then decrypt it. As such, no key distribution is necessary, and a greater inherent safety factor is provided by this method. The main disadvantage of an asymmetric keys system is the computational cost and subsequent loss of speed that comes with performing modulus and exponentiation operations on large numbers. However, the hardware based nature of the TPM reduces the cost of the asymmetric calculations, making it the better choice. Therefore, TPM chips use an RSA encryption scheme with asymmetric keys [2].

In order to provide better protection and privacy, TPM chips use several different keys to fulfill different function requirements. In this way, there is never any single “master key” that can compromise security and give a user full access. The following is a list of the different styles of keys provided by a TPM chip.

- **Endorsement Key:**

The Endorsement key of a TPM chip is unique in its functionality. The key is placed on the chip during the process of being manufactured. The primary function of the endorsement key is ensure other TPM chips that a piece of information could only have been encrypted using a TPM chip and is trusted. The private portion of this key may never leave the chip or be replaced. If the key is lost due to hardware damage, it is impossible to recover [2].

- **Storage Root Key (SRK):**

The Storage Root Key (SRK) is a unique key that is associated with the owner of a TPM chip. This key

is non-migratable and considered the parent of all other keys. In this way the SRK is guaranteed to always be present on the chip. As the parent, the SRK must be verified before using any child key in the TPM. It should be noted that due to the SRK being tied to the owner of the chip, if the owner changes, the SRK will also be changed. Any keys encrypted with the old SRK will be permanently lost. [2]

- **Storage Keys:**

Storage keys are used by a TPM chip to securely store any type of key (including other storage keys). These keys can be classified as migratable or non-migratable as needed by the chip. [2]

- **Binding Keys:**

Binding keys are used by a TPM chip to securely store symmetric keys used by a user. In this way, symmetric keys are given asymmetric protection and are never seen “in the clear” where they could be compromised. [2]

- **Identity Keys:**

Identity keys are a special class of non-migratable keys that are used to ensure that a stored hash value or other non-migratable key is used for a specific TPM chip. These keys are always created as children of the SRK to further ensure the key exists within the TPM chip. [2]

1.4 Hardware Encryption: Full Disk Encryption

Hardware based encryption provides the security and prevents data access by unauthorized persons. Trusted Drives (Drives encrypted by a manufacturer) have hardware encryption embedded in the drive hardware which ensures that all data stored on the drive is encrypted. Full disk encryption does not require the presence of a TPM chip. However, the TPM chip can be used to wrap the decryption key for the drive using a storage key. By wrapping the storage key, the TPM ties the encrypted drive to a particular system. Therefore, full disk encryption in combination with TPM ensures that the only authorized users can access encrypted data [6].

1.4.1 Trusted drives

Trusted drives have two modes of operation, un-locked mode and locked mode. When the trusted drive is in unlocked mode it behaves like a normal ata drive, drive data can be read by anyone who has platform access. Drive data can also be read if the drive is plugged into another PC as a secondary drive. "Unlocked mode" is the default mode.

When a trusted drive is in "locked mode", only users with the correct drive authentication credentials can access the drive data. The trusted drive will not boot unless the user enters the correct user name and password into the pre-boot screen. If the drive is plugged into another machine as a secondary drive, the trusted drive cannot be read. All of the data, even the file system, is fully encrypted [6].

1.5 Hybrid Encryption: BitLocker

Windows BitLocker is a hard drive encryption tool that is, at this time, only built into Windows Vista Enterprise, Windows Vista Ultimate and Windows Server 2008 editions. BitLocker helps prevent unauthorized access to data on lost or stolen computers by a combination of protection methods. Primary methods that are used: Encrypting the Windows operating system volume on the hard drive and verification of boot components and configuration data. BitLocker can be used with or without a TPM chip. However, a BitLocker enabled system without a TPM chip is limited to only encrypting the Windows operating system volume. The system cannot perform verification of any boot components, as a TPM chip is required to compute and safely store such measurements.

1.5.1 Transparent Operation Mode

BitLocker can be configured in three major modes: Transparent operation mode, User authentication mode, and USB key mode. Transparent operation mode requires a TPM chip to be present and activated on the motherboard. A TPM chip is turned on and activated through the BIOS menu, typically accessed at startup. Details of TPM activation can be found in appendices B and C. If TOM is set up, a user can log onto Vista normally. The key used for the BitLocker encryption is sealed by the TPM chip. The release of the encryption key is based on the integrity of the boot components. If the early boot files are modified, the drive encryption key will not be released, but if the boot files are unmodified, the drive will boot normally [5].

1.5.2 User Authentication Mode

User authentication mode requires that a user provide some kind of authentication to the pre-boot environment before the Windows volume can be decrypted. During the initial drive encryption a PIN of four to twenty characters is created by the user. Once the drive is encrypted, the correct PIN must be provided at boot time for the Windows volume to be decrypted. PIN authentication is only available on a computer that has a TPM chip [5].

1.5.3 USB Key Mode

In USB key mode, the user must insert a USB device that holds the startup key before boot (created during initial drive encryption). In order to use USB key mode the machine's BIOS must support reading USB devices in a pre-OS environment. Machines that do not have TPM chips must use USB key mode authentication [5].

It should be noted that both forms of authentication require that an external backup file be created in case of an emergency. The backup file contains the BitLocker recovery password that,

when entered, will allow the hard drive to be decrypted. It is also possible to use combinations of the forms of authentication such as a TPM+PIN, TPM+USB Key, or even TPM+PIN+USB Key (only available after Vista Service Pack 1) [5].

1.6 Software Based Encryption: TrueCrypt

TrueCrypt is a free, open source software based encryption program. TrueCrypt is not unique as a program, but it is a good representation of a software based encryption system that could be operated by a typical computer user. As stated previously, software based encryption relies completely on a separate software program to do encryption and decryption. The nature of software allows TrueCrypt to have a high level of flexibility in its functions. TrueCrypt and similar programs can create both fully encrypted drives and virtual encrypted partitions on standard hard drives, CDs, DVDs, and USB drives. With TrueCrypt, it is also possible to create layered encryption. For example, it is possible to create an encrypted partition within a file and use that partition to house another second encrypted partition. TrueCrypt leaves few traces when used to encrypt a device. TrueCrypt can even create a hidden operating system, contained within a volume. It should be noted that software encryption programs do not necessarily require installations. TrueCrypt and other software encryption programs can be run from standalone executables [7].

When installing TrueCrypt, a user is able to specify the type of encryption desired for the encrypted space. Once the encryption type is chosen, TrueCrypt prompts the user for a password. The user created password is used as a key for the encrypted space. When the user enters the correct password, the appropriate virtual partition is mounted and can be accessed. Without the password, there is no way to view the unencrypted data, as there is no data recovery tool built into TrueCrypt [7]. In the case of a second, hidden operating system, the password can be used to misdirect intruders. For example, a user could set up his computer such that entering an incorrect password at startup would

cause the computer to boot to a secondary operating system. The secondary operating system would be completely functional, fooling the intruder, while the actual operating system and sensitive data would remain encrypted on a different part of the hard drive.

2 Methods of Analysis and Recovery

As mentioned previously, a computer forensic responder will traditionally unplug the power from the back of a desktop computer, at the scene of an incident. With the increasing use of hard drive encryption, immediately removing power is no longer a best practice, as the data on the drive would then become recoverable. Live analysis should be done first to determine if encryption is present on a system and to recover as much of the system as possible before committing the drastic measure of pulling the power. Due to the various types of encryption mentioned above and in the appendices, it is not time to consider a change in the guidelines for computer recovery.

2.1 Software Based Encryption Analysis

In the case of pure software encryption, the best indicator of its presence is in the Windows registry and start menu. In many cases, software based encryption requires an installation. Checking the registry values under HKEY_CURRENT_USER\Software should give a good indication if any known encryption software is installed on the machine. Also, checking “All Programs” under the Windows start menu for known encryption software should not be overlooked. If the user uses an encrypted volume regularly, the program used to decrypt and re-encrypt the volume may be visible on “recently used programs” part of the start menu or in the “recent items” folder.

Software based encryption schemes are somewhat difficult to attack once they are discovered. Many times there is no backup file for recovery and no indication of the encrypted data’s location. The most successful attack against software encryption has been the Princeton cold boot attack described here <http://citp.princeton.edu/memory/>. Specifically, the weaknesses of software encryption

seem to be that the software is not tied closely with the operating system and the fact that the encryption program relies on RAM while doing the computation for encryption. A few times, during testing, some pieces of plaintext versions of encrypted files were found outside of the encrypted space. The unencrypted data is visible due to the fact that although the volume containing the file is encrypted, editing files and moving them around seems to still occasionally use operating system functions that use areas such as temp files. Similarly, remainders of files that are copied into encrypted space and deleted from regular space can still be found in slack space, as the operating system does nothing special to clean the unencrypted file from the drive when it is deleted from unencrypted space. Since the external program used to encrypt data is not married closely to the operating system, temp files and slack space should be recovered and analyzed if software encryption is found. More information on TrueCrypt and software encryption can be found in appendix F.

If software encryption is found on a system, a copy of ram needs to be dumped for later analysis. While working on Vista and XP, it was discovered that Windows keeps copies of its page files for quite a long time. With respect to the specific experiments (appendix H) logged in this paper, analysis showed that the main computer used for testing retained over fifty page files from the time of the purchase. Windows seems to keep page files forever, unless it needs the space for other things. With the capacity of today's drives, page file retention can be quite extensive. When examining some of the page files with FTK, it was discovered that the actual plain text password for the TrueCrypt volume was found by a string search. Along with the correct plain text password were other words that were recognized as previous attempts (incorrect passwords) that had been entered into TrueCrypt for earlier testing. Unfortunately, there was not a clear indicator of where in the page file that these entries could be consistently found. When a RAM dump was parsed using the algorithm provided in the Princeton cold boot paper, several AES and RSA keys were found. Although a "surefire" way to

break software based encryption was not discovered during testing, discoveries point toward RAM and page file analysis as significant possibilities for the future.

2.2 Hardware Based Full Disk Encryption Analysis

Hardware based full disk encryption is, to an extent, easier to identify than software based encryption, but there is still not a surefire method to determine its presence without invoking it. In order for full disk encryption to be utilized, there must be an external program to manage it. Currently, there are only a few vendors, such as Seagate and Hitachi that provide hardware based full disk encryption (FDE) drives. Each vendor has its own proprietary program that can be used to invoke the full disk encryption (FDE). In the same way, a few third party applications, such as Wave Systems Security can also manage the hardware based full disk encryption drives. Finding these programs of a running machine in the Windows start menu is an indicator that FDE may be present. The icon representing a Seagate FDE drive under “My Computer” also has lock at the bottom of it. The lock on the icon can look locked or unlocked, depending on whether FDE is enabled or disabled.

Once the full disk encryption is enabled and the password is set, the password must be entered if the drive ever spins down (turning off the computer, computer going to power-save mode, shutting the lid to a laptop) to get the drive to spin up again. While the drive is active and spinning, all data on the drive is in the clear and recoverable. It should be noted that restarting a computer will not cause the drive to spin down. Therefore, once the computer is turned on, and the password for the FDE is entered, it can be restarted without having to re-enter the password. However, it should also be noted that during the FDE activation process the “normal” sleep function of a computer is changed to a “hibernate” type sleep where the drive will spin down. Therefore, if the computer goes to sleep or screensaver at any time, the FDE password must be re-entered. In light of the previous revelations, if a computer is found to have hardware FDE enabled, do not let it go to sleep. In fact, during

experimentation, a computer with active full disk encryption went to sleep while in the process of being recovered by FTK imager. The computer froze and the drive was unrecoverable without re-entering the password. Therefore, if a computer is found to have active hardware FDE, the utmost care must be taken to keep the computer active, even during live recovery and analysis.

The main weakness of hardware based full disk encryption systems is the propensity of their managing programs to force the creation of recovery keys. During the FDE activation process, the user is forced to create a recovery file. The recovery file may be stored locally (which it never should be, as the local drive is typically what is inaccessible if recovery is ever needed), or on some sort of external media such as a USB, floppy disk, or SD card. In the case of a Seagate FDE drive, enabled using a Wave Systems Security interface (default FDE and TPM management program given by Dell), the backup file contained the usernames and passwords of all authenticated users on the system, in plain text. More details of the drive manager by Wave Systems can be found in appendix D. If the recovery files for the system can be found, the drive can be decrypted and analyzed in a standard fashion. Therefore, investigators should recover and search all forms of external media for usernames and passwords if hardware based full disk encryption is found on a computer.

Another point of note is that a hardware full disk encrypted drive may be used in conjunction with TPM. Theoretically, the Seagate FDE used for experimentation should be able to use the TPM chip to marry the hard drive to the motherboard. In practice, it was discovered that activating the FDE drive on a TPM enabled machine did not marry the hard drive to the specific motherboard. During testing, the hard drive was removed from a hardware full disk encryption; TPM enabled laptop and placed into a different TPM enabled laptop, with identical hardware. The FDE prompted the user for the drive encryption password on the new laptop. When the password was correctly entered, the drive booted normally. Although, in this case, the TPM marrying was not enforced, if investigators

find a computer with a hardware FDE drive and an enabled TPM, the entire computer should be recovered as evidence, as the drive may be married to the motherboard of the computer in question.

2.3 TPM and TPM Based Encrypted Volume Analysis

Computers with TPMs have the ability to create encrypted spaces, using the TPM for key management. Testing done using the TPM and its encryption methods suggest that TPM based encryption is very similar to software based encryption in its final result. However, since the TPM is used to do all encryption and decryption, it is considerably less “messy” than pure software based encryption. TPM methods leave few traces in slack space, temp files, and RAM. Additionally, space encrypted with a TPM is married to that specific TPM chip. Therefore, removing the hard drive from a computer with TPM encrypted space and placing it in a different computer with a TPM will result in the original data remaining encrypted, even if the user has the correct password to the encrypted space. Much more about TPM and TPM based encryption can be found in appendices B and C.

If a Windows Vista computer is found and suspected to have a TPM, the status / activation level of the TPM can be found by running tpm.msc (Appendix B). A Windows XP machine does not have a native interface for TPM, so the TPM must be managed and activated with a third party application. Looking for the third party application in the registry or start menu is recommended to confirm the presence of TPM. If a TPM is found on the system, its activation level should be determined. Again, XP machines must use the third party application to determine the status of the TPM. Vista machines can use either the third party application or tpm.msc. If the TPM on the machine is found to be activated, further analysis and recovery should be done.

The major weakness of TPM based encryption is the same as hardware based full disk encryption. Upon TPM activation and password creation, a password backup file is created, which can be used to restore the TPM keys or reset the TPM owner password if anything goes wrong.

Similarly, there is an option to print the recovery password if desired. Much like with hardware full disk encryption, it is important for an investigator to seize any external media storage in order to find the backup keys. If an activated TPM is found, the area around the computer should also be searched for a printed key as well. When TPM protected storage is created, backup keys are generated. It is also possible to create full backups of TPM protected storage if desired. Much more about TPM protected storage can be found in appendix C. Since, by nature, a TPM marries the information that it protects to itself, the entire machine needs to be recovered, if it is found to have an activated TPM.

2.4 BitLocker Analysis

BitLocker is unique as it is software based encryption that can also use a TPM for security and sealing. BitLocker is only compatible with Windows Vista Enterprise and Ultimate Editions. If the computer being examined is not using one of those versions, then it does not have BitLocker. Determining the version of Vista running is explained in appendix E. Also, when enabled, BitLocker creates a minimum 1.5 GB partition with the default label “S:”. BitLocker can be configured in several different modes, which are explained in appendix E.

Fortunately for investigators, BitLocker has the weaknesses of both hardware and software encryption. When enabling BitLocker, a recovery key must be created. Therefore, it is, again, vital to recover all external storage devices if BitLocker is found. Since BitLocker is software based, encryption and decryption takes place in RAM. On a live system with BitLocker enabled, the RAM can be analyzed to find BitLocker keys. However, the most glaring weakness of BitLocker is the recovery console. Using the technique explained in appendix E, the BitLocker recovery key can be obtained from a live machine.

If BitLocker is discovered, the entire machine should always be recovered as there is a possibility that TPM could be protecting the system in conjunction with the BitLocker encryption. In

many cases, the actual BitLocker security is transparent. That is, there is no required authentication unless something in the system changes (e.g. unplugging the hard drive and putting it in another computer).

3 Final Issue

One of the final and most key issues law enforcement and computer forensics should be aware of is not how encryption affects hard drive analysis, but how encryption can aid in the destruction of evidence. Previously, destroying evidence on a hard drive was a long process. To completely wipe an 80 GB drive, a computer typically takes around twenty to thirty minutes. Given the fact that today's drives are many times that size, quickly destroying evidence on a traditional unencrypted hard drive is basically impossible (unless there is a very powerful degausser handy). However, encryption keys can be destroyed or reset in a matter of seconds. During testing, 10 GB worth of TPM protected data was rendered unrecoverable in approximately twelve seconds. The twelve seconds included opening up tpm.msc selecting the option to reset the TPM keys, and typing in the TPM owner password. Similarly, the managing program for the Seagate FDE drive has a "nuke" button that, when clicked, throws away the current encryption keys, in effect deleting the entire drive. In testing, when the Seagate drive was told to reset, Windows immediately crashed and the screen went black in less than ten seconds. All data previously stored on the drive was unrecoverable. Due to fact that digital evidence can now be destroyed almost instantaneously, investigators must adjust their approach. Investigators should not leave suspects with their computer alone for any amount of time. Additionally, in a digital evidence case, law enforcement may now need to treat a significant delay in responding, after the officer has announced himself, as a reason to immediately enter the house (if a warrant has been issued).

4 Conclusion

Because hard drive encryption is becoming more common every day, a new approach that includes immediate, on scene, live recovery efforts needs to be taken to effectively obtain and preserve digital evidence. Although hard drive encryption seems to be a difficult hurdle, there are still ways to effectively recover digital evidence. This paper has given procedures to recover data from drives protected by BitLocker, hardware full disk encryption, software encryption, and TPM protected space. The procedures given should be incorporated into standard computer forensics procedures.

5 Appendix A: RSA Encryption and signing

RSA is an encryption scheme used in computer security. RSA uses a public and private key system to encrypt data. Public and private keys are generated by using the exponentiation and modulus, using the product of large prime numbers. Once the large numbers are run through the key generator, a public and private key pair is created for an entity. The public key is shared for the world to see, but the private key is kept secret. Using these keys it is possible for an entity to have both authenticity (that a message can only be generated by a trusted source) and integrity (that a message is secret and has not been changed) in transactions.

The main concept of RSA is that a message encrypted with an entity's (typically a computer) public key can only be decrypted with that same entity's private key. In the same way, a message encrypted with an entity's private key, can only be decrypted with that same entity's public key. Given that many entities have their own public-private key pairs, security can be achieved. For example, let's say that two entities, Bob and Alice, want to communicate securely. Bob and Alice have their own public-private key pairs. Let's say that Bob wants to send a secure message to Alice.

First, Bob would encrypt the message with his private key and then (second) he would encrypt the message with Alice's public key (Bob can see Alice's public key, because it is public). The message would have two layers of encryption when it got to Alice. The outer layer would be encrypted with Alice's public key, and the inner layer would be encrypted with Bob's private key. Once the message is received, Alice would decrypt the outer layer with her private key (remember that only Alice's private key can decrypt something encrypted with her public key: see main concept of RSA). Since the only person who could decrypt the outer layer of the original message is Alice, the message satisfies the integrity standard (the message could not have been changed in transit). Finally, Alice decrypts the inner part of the message (which was encrypted with Bob's private key) with Bob's public key. Since Bob's public key could only be used to decrypt something encrypted with Bob's private key, Alice knows that the message is actually from Bob. Therefore, the authenticity of the message is assured. The act of Bob using his private key to encrypt the message is called 'signing' because it proves (when the message is finally decrypted) that only Bob could have been the origin.

6 Appendix B: TPM in Vista

6.1 TPM Management Control Console

TPM 1.2 is supported by all versions of Windows Vista and Windows XP, even though only the Enterprise and Ultimate editions of Vista can use Windows BitLocker encryption. Vista supplies a native management control console (msc) interface to the TPM. The TPM management console can be invoked in several ways. The most common way to invoke the console is to type "tpm.msc" into the search bar at the bottom of the Start menu, as shown in the following figure.

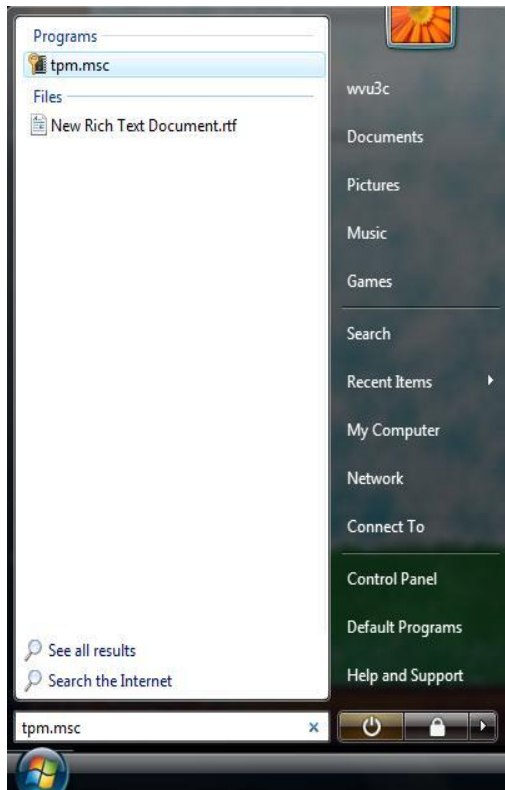


Figure 2: tpm.msc

Other methods to invoke the TPM management console include: typing "tpm.msc" at command prompt, going to Start->All Programs->Accessories->Run and type "tpm.msc" in the run box, and navigating to [HomeDrive]:\Windows\System32 and double clicking on tpm.msc (where HomeDrive is typically C:). It is important to note that a user without administrator privileges cannot typically run tpm.msc. In order for a non-admin to run the TPM command console, the user must navigate to the tpm.msc file located in the Windows\System32 folder, right click on tpm.msc and select "Run as Administrator".

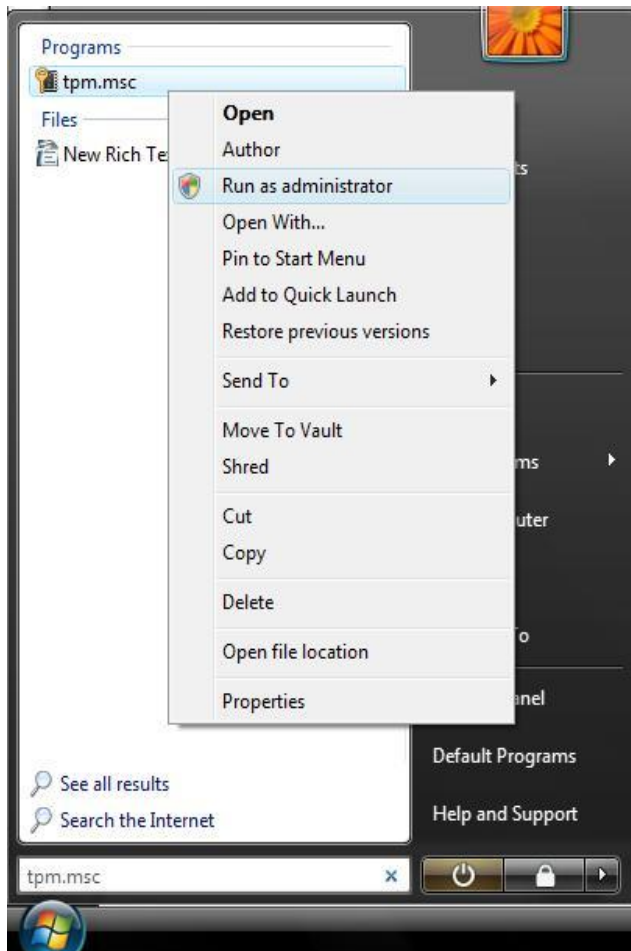


Figure 3: run tpm.msc as administrator

6.2 Turn on TPM

Before using tpm.msc, it is important to understand that a TPM can be in one of four states:

- Unowned and turned off
- Unowned and turned on
- Owned but turned off
- Owned and turned on

In order for TPM to be used to secure a computer, it must be turned on and owned. In its initial state, the TPM is unowned and turned off. Running tpm.msc with the TPM in the unowned/off state will result in the command console not recognizing that a chip TPM is present.

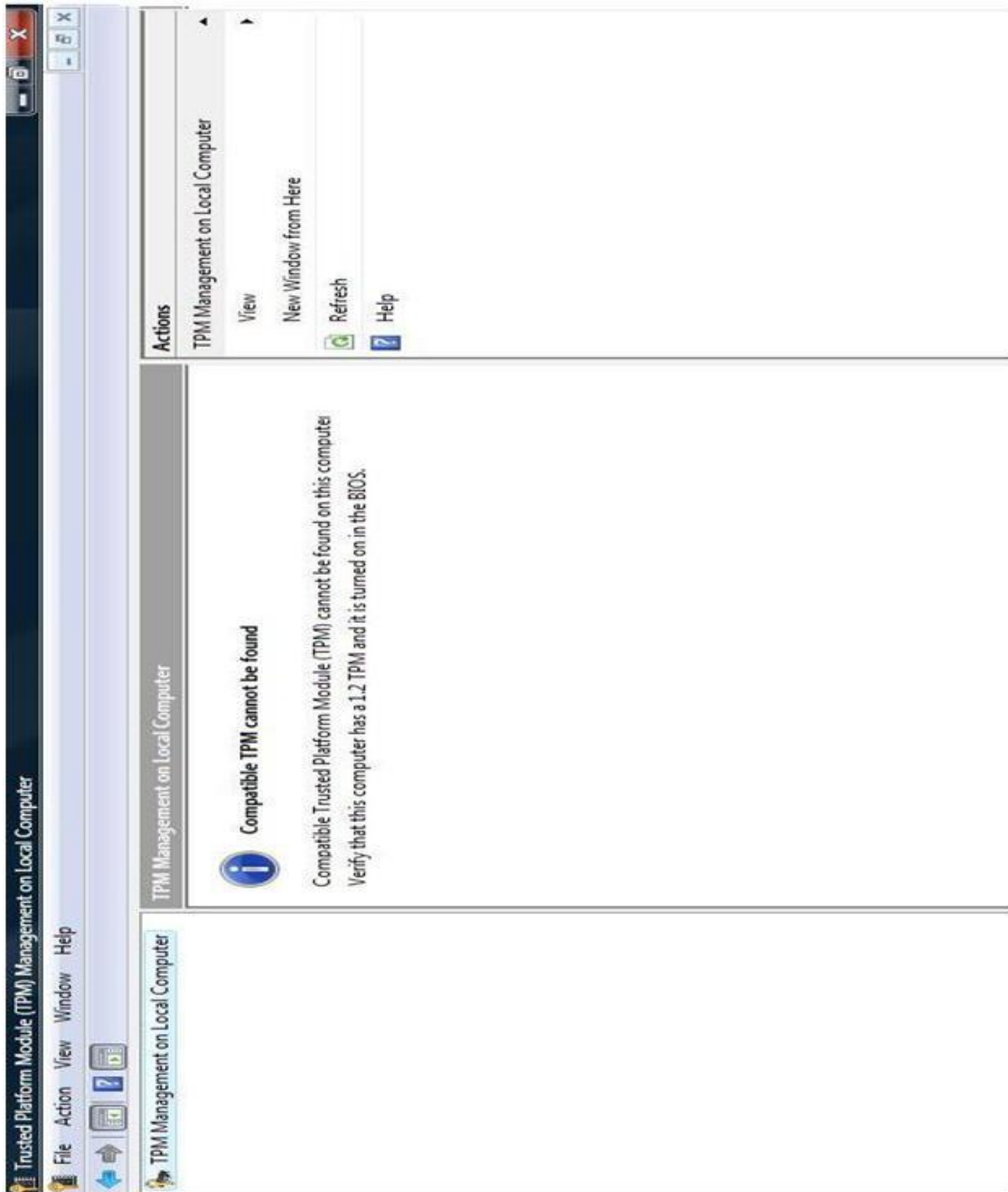


Figure 4: TPM Unowned and turned off

To turn on the TPM the computer has to be restarted and booted to the BIOS (the function key to get into the BIOS on restart vary by manufacturer, but F2 is the most common). Once in the BIOS, navigate to Security->TPM Security. Turn TPM Security 'on'. A new option called “TPM Activation” appears. There are three options in the activation menu: Activate, Deactivate, and Clear. Activate turns on the TPM chip, but does not generate keys for use. In the same manner, Deactivate turns the TPM chip off, but does not affect any keys that are already generated. Clear destroys all present root keys in the TPM, rendering anything encrypted / protected using them unrecoverable. To turn on the TPM select Activate and press enter. Exit BIOS and save the new settings (Press the escape key to exit BIOS).

6.3 Activating the TPM (taking ownership)

Even though the TPM is turned on, it is unable to be used to secure a computer without someone taking ownership of it. The combination of turning on and taking ownership of a TPM is called initialization. When the TPM is owned, it generates the set of root keys that used for encryption / security in the future. Again, to take ownership, a user must be a local Administrator or run tpm.msc with administrative privileges.

Now that the TPM is turned on, the option to activate the TPM becomes available using tpm.msc.

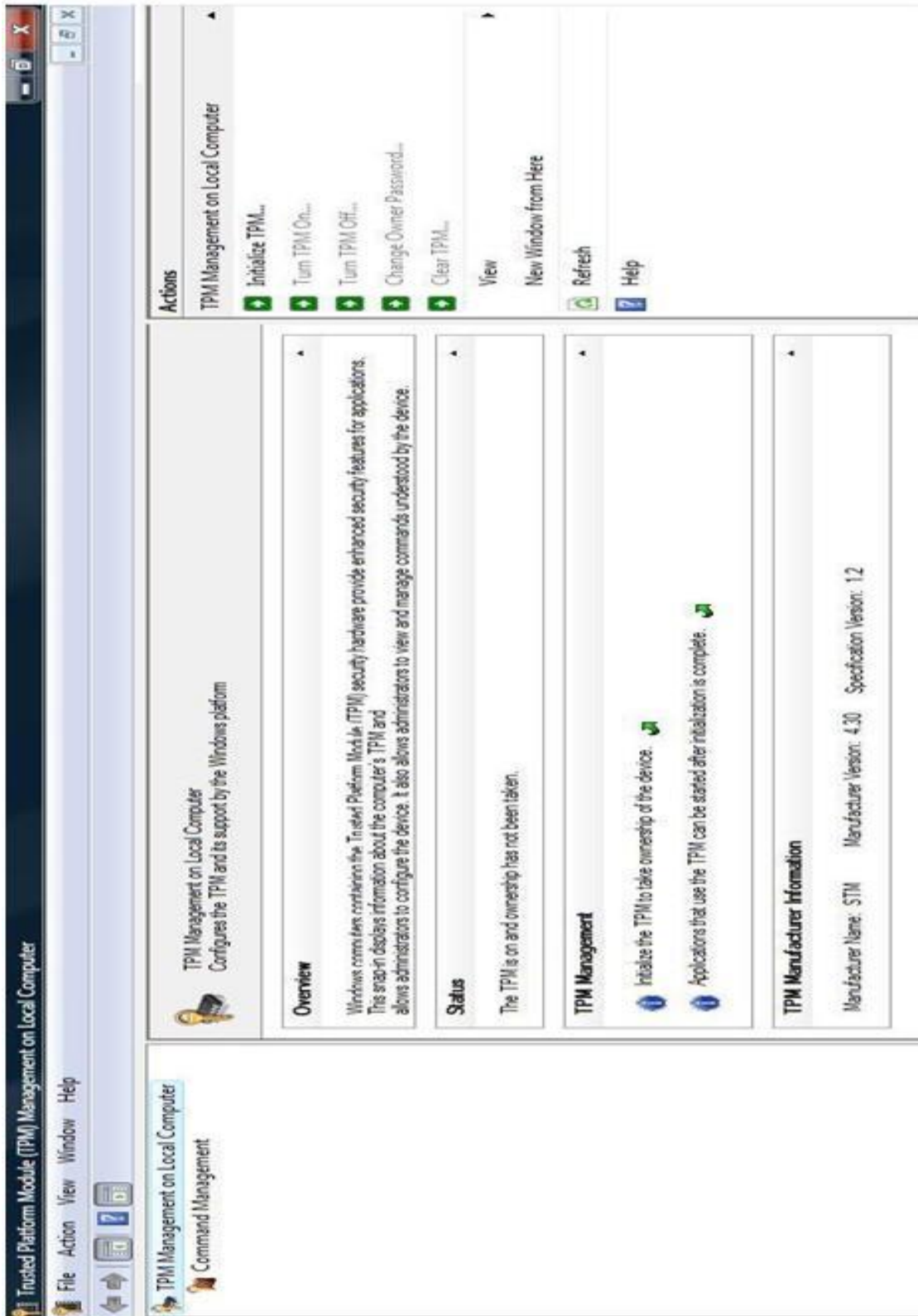


Figure 5: TPM unowned and turned on

Selecting “initialize TPM” begins the process of ownership. A window pops up prompting the user to create or let the computer generate a TPM owner password. An owner defined password must be at least eight characters, and a computer generated password is eight blocks of five digits (in the form xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx) The owner password is linked to the entire machine (the owner password is the same for all users) and is linked to the specific set of keys generated by the TPM during the “taking ownership” process. The owner password cannot be reset without destroying the current set of encryption keys. However, the owner password can be changed to something else if a user knows the current owner password.



Figure 6: Initialization of TPM- password creation

Once the password is created, Vista gives you the option to store the password in a backup file or print it out. The idea of a backup file is that migrating the owner password to a mobile device creates a sort of smart card scenario, where the user can use the device to authenticate to the TPM.

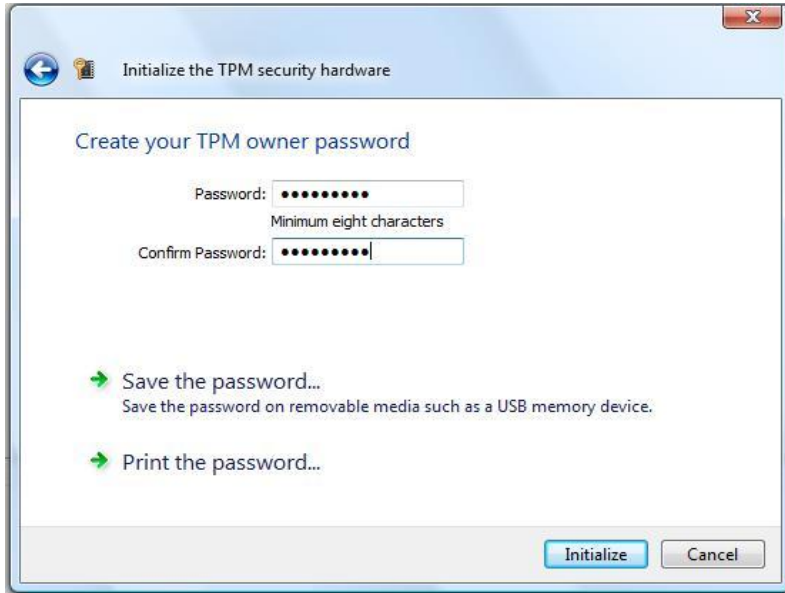


Figure 7: Initialization of TPM: password saving and printing

It should be noted that it is possible to store the backup password on any media connected to the computer, including the local drive. The TPM owner password backup file is an xml file with a .tpm file extension. The backup file contains the version of the backup file (currently 1.0), the version of Windows on which the file was created, the creation date of the backup file, the name of the machine where the password backup was created, the name of the user who created the backup, the unique manufacturer id of the TPM manufacturer, and a hash of authentication password. An example of a TPM owner password backup file can be seen below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
This page is a backup of Trusted Platform Module (TPM) owner
authorization information. Upon request, use the authorization information to
prove ownership of the computer's TPM.

IMPORTANT: Please keep this file in a secure location away from your computer's
local hard drive.
-->
<tpmOwnerData version="1.0" softwareAuthor="Microsoft windows [Version 6.0.6000]"
creationDate="2008-06-24T13:10:29-05:00" creationUser="wvu3c-PC\wvu3c"
machineName="wvu3c-PC">
  <tpmInfo manufacturerId="1398033696"/>
  <ownerAuth>Ya9wg/sxBV5D7MYd1hjmHq/OmM4=</ownerAuth>
</tpmOwnerData>
```

Figure 8: TPM owner password backup file

Clicking on initialize prompts the computer to generate the TPM root keys. Once the root keys are generated, any applications that use the TPM can run properly.

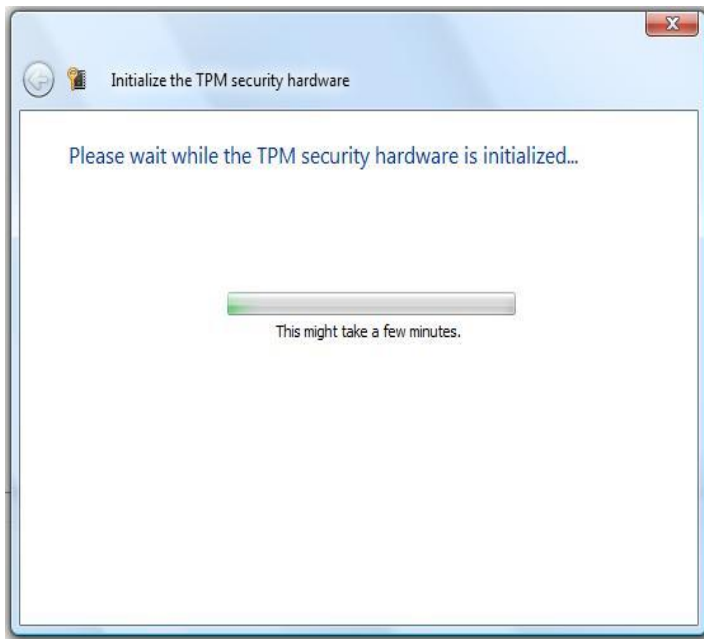


Figure 9: TPM initialization complete

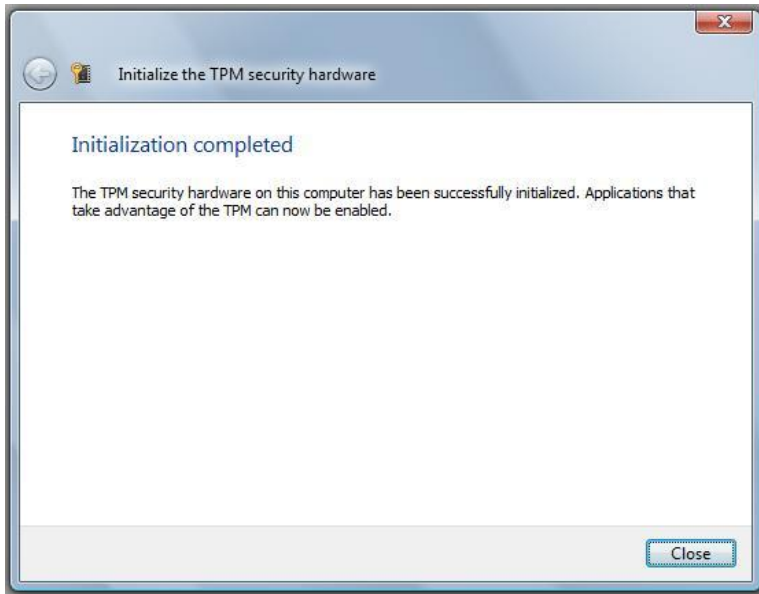


Figure 10: TPM initialization complete (cont.)

When the TPM chip is fully initialized, the management control console enables more options. Some of these options are Turn TPM off, Change Owner Password, Clear TPM and Command Management (Located on the left side of the control console, under “TPM Management on Local Computer”)

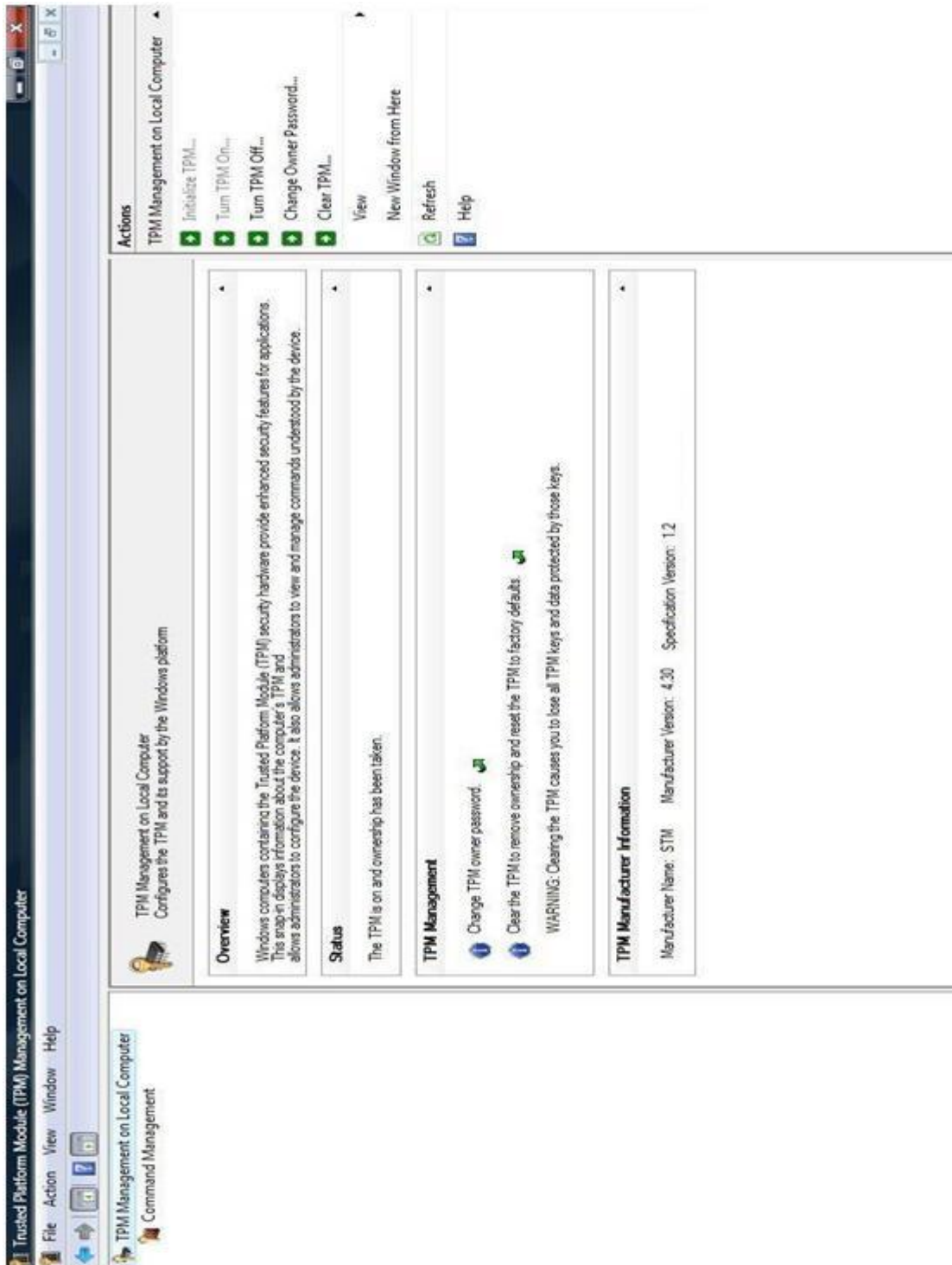


Figure 11: TPM owned and turned on

6.4 Turn TPM off

Through tpm.msc local administrators have the ability to temporarily shut off the TPM chip, regardless of whether they have the TPM owner password. When selected, the “Turn off TPM” gives options to enter the TPM owner password through typing or backup file. If the correct password is entered, the TPM turns off instantly. However, a user with local administrative privileges can still turn off the TPM, but the computer has to be restarted. Turning off the TPM does not destroy or reset any of the root keys associated with the TPM owner password, but it does render all data encrypted using the TPM unavailable until the TPM is turned back on.



Figure 12: Turn off the TPM

The following is a screenshot of tpm.msc with an owned but turned off TPM.

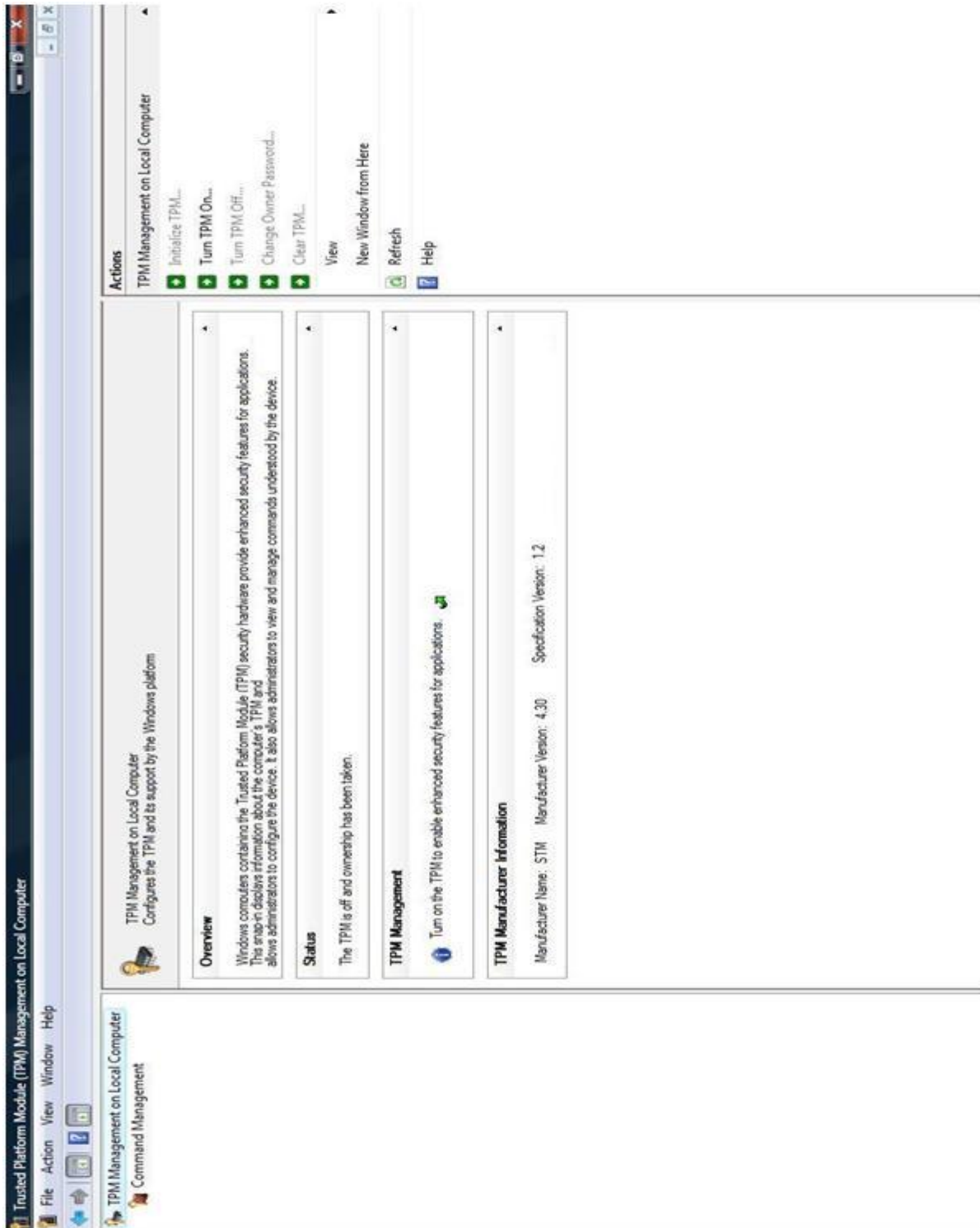


Figure 13: TPM owned and turned off

Once the TPM is turned off, the only available option is to turn it on again. Turning the TPM back on is similar to turning it off in the respect that tpm.msc asks for the TPM owner password, but allows local administrators to turn the TPM on without it (computer must be restarted in the case of no password).

6.5 Change owner password

Changing the owner password to something new requires that the user know the current owner password or have the correct .tpm file containing the owner password. Once current password is supplied, a new owner password can be created or generated by the computer. Again, the user is able to print the password or save a .tpm backup file onto disk (or any other connected media).

6.6 Clear TPM

Clearing the TPM resets the chip to factory defaults. The root keys created by initialization are destroyed and all data encrypted by the root keys is unrecoverable. Like turning off the TPM, clearing the TPM does not require the TPM owner password to execute. The combination of physical presence at the keyboard and local administrator privileges is all that is required to reset the TPM keys. Resetting the keys only takes a matter of seconds. Once the keys are reset, the TPM reverts back to the unowned and on state. To use the TPM again a new owner password must be created and new keys generated.



Figure 14: Clear the TPM

6.7 Command Management: tpm.msc

An owned and enabled TPM, by itself, does nothing to protect a computer or encrypt data. In order to do anything useful with the TPM, outside programs must be able to interact with the TPM chip. Although each TPM chip is slightly different, there is a standard set of commands that all TPMs use, as defined by the Trusted Computing Group. The Trusted Computing Group develops industry standards for the TPM and defines specifications for chip manufacturers. The command management option, located on the left side of the tpm.msc interface, allows a local administrator to block or allow specific commands inside the TPM.

7 Appendix C: Third Party TPM Management

7.1 Outside programs

As mentioned before, the TPM holds a list of commands that can be used to do security related tasks. Usually, any computer shipped with a TPM chip has a security suite developed to make use of the TPM chip. Windows XP, which lacks a native TPM interface, is forced to rely on such outside applications to manage TPM related security. EMBASSY Trust Suite is a program provided by Dell to customers who purchase Dell computers with TPM chips (including both Windows XP and Windows Vista machines). EMBASSY Trust Suite (ETS) is one of many outside applications that can communicate with TPMs and provides a more user friendly way to use the TPM. ETS uses the TPM to perform various security related enhancements. Three of the most important security features include: creating an encrypted partition on the local hard drive, utilizing full disk encryption, and backing up generated TPM keys.

It is important to note that although Windows XP requires use of an outside interface to TPM, the EMBASSY Trust Suite (or similar program) is included in Windows Vista installations as well. Therefore, although some small details (such as paths to the ETS programs) may differ, all of the following analysis applies not only to Windows XP machines but machines using Vista (unless otherwise noted).

7.2 Enable and Activate the TPM

Like Windows Vista, the TPM for a Windows XP machine is disabled from the factory by default. Before users are able to use any external security suite, the TPM must be enabled. To enable the TPM, press the 'F2' key (key may vary) during computer power-up to enter the BIOS configuration. Select security then 'TPM security' from the menu and set the TPM power state to 'On'. Press the 'Escape' key and choose 'Save/Exit'. When the computer boots again, enter the BIOS setup

using the F2 key as before. Select 'Security' then 'TPM Activation' from the menu then set the TPM activation state to Active. Note** A machine using Windows Vista does not require a restart between turning on and activating the TPM. Both actions can be done in one step.

7.3 Establish Ownership of the TPM

Now that the TPM is activated, an owner must be set using the EMBASSY software. From the Windows start menu navigate to All Programs -> Security by Wave Systems -> EMBASSY Security Center. At the bottom of the Security Center there should be a green check mark beside the word 'Enabled' indicating that the TPM has been turned on and enabled. From the EMBASSY Security Center, select Trusted Platform Module from the menu on the left, and then click on the 'Owner' tab. Clicking on the 'Establish' button under the 'Ownership' heading begins the process of taking ownership of the TPM. At the end of the 'establish ownership' process, the user is prompted to enter the owner password. Unlike Windows Vista, a random password cannot be generated and no backup of the password can be made. Once the new owner password is created, the TPM generates new root keys and can be used for other security functions. A green check mark beside the 'Owned' label at the bottom of the EMBASSY Security Center indicates that a TPM owner has been established.



Figure 15: TPM not enabled

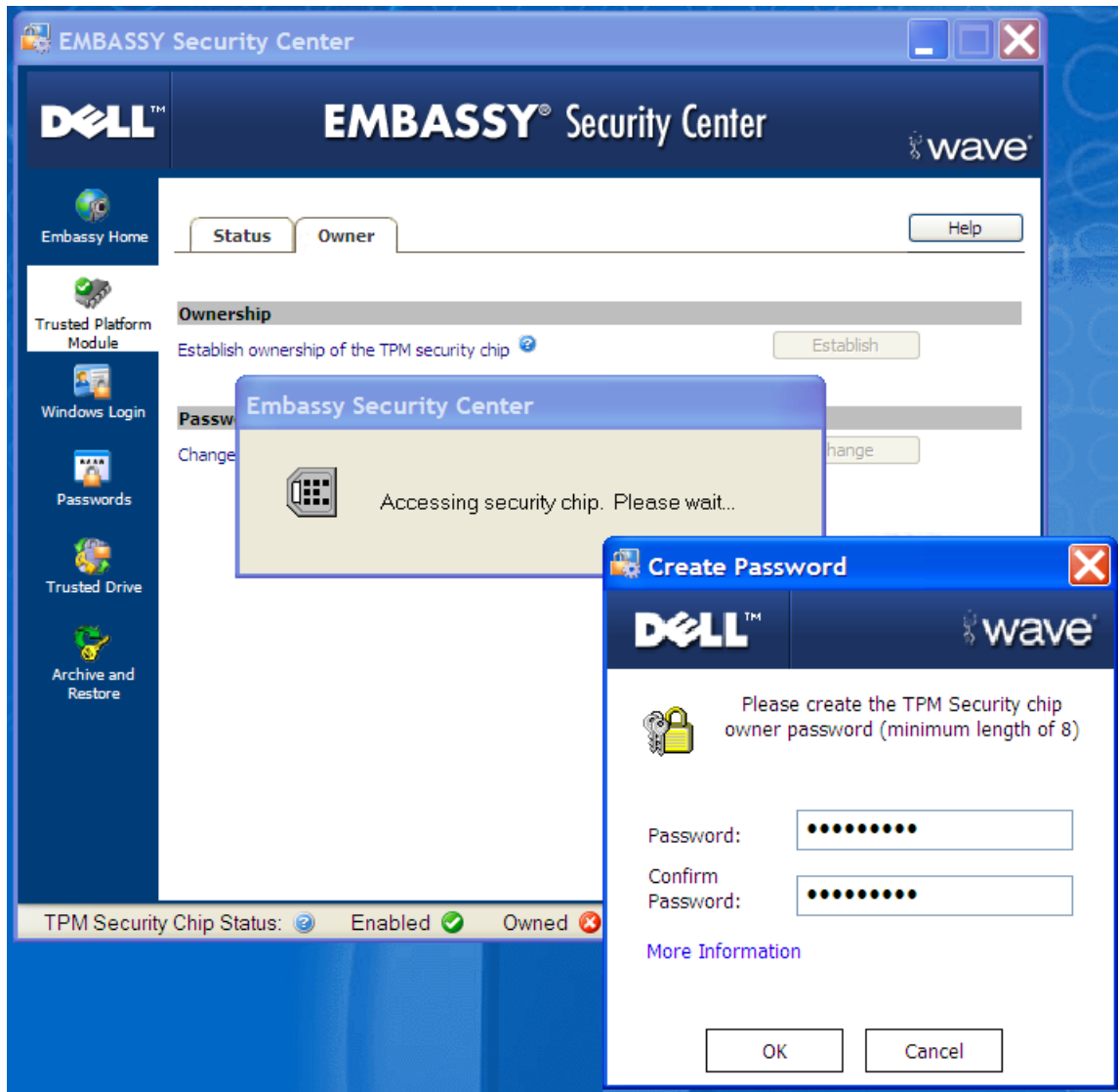


Figure 16: TPM enabled and in the process of being owned

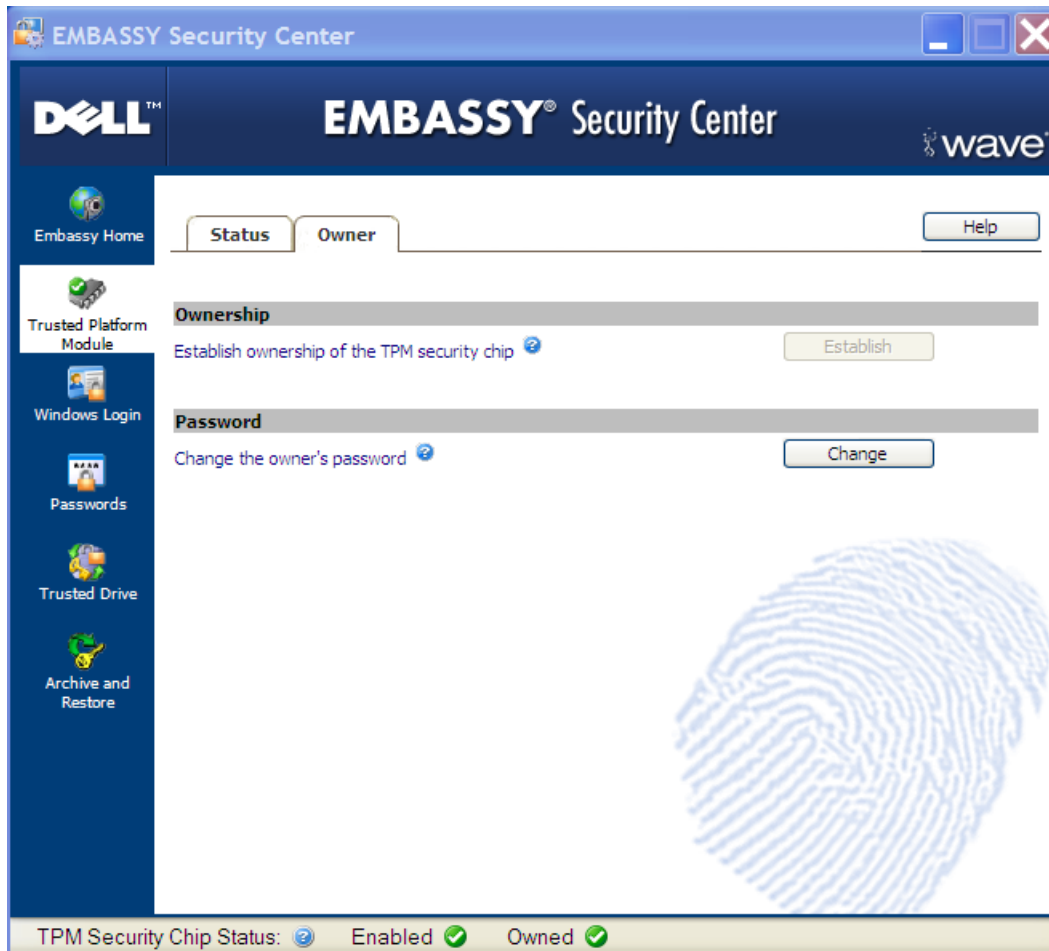


Figure 17: TPM enabled and owned

7.4 Configure Document Manager (Vault)

Document Manager creates an encrypted vault for general storage. Files that are saved into the Vault (or dragged-and-dropped there) are encrypted automatically. To create a Vault, go to the start menu and select All Programs -> Security by Wave Systems->Advanced->Document Manager. Click on the 'New' button. The Document Manager prompts the user for a new vault password. The EMBASSY Security Suite software accesses the TPM and generates a new 'Vault' or expandable encrypted file space locked using the keys generated by the TPM.

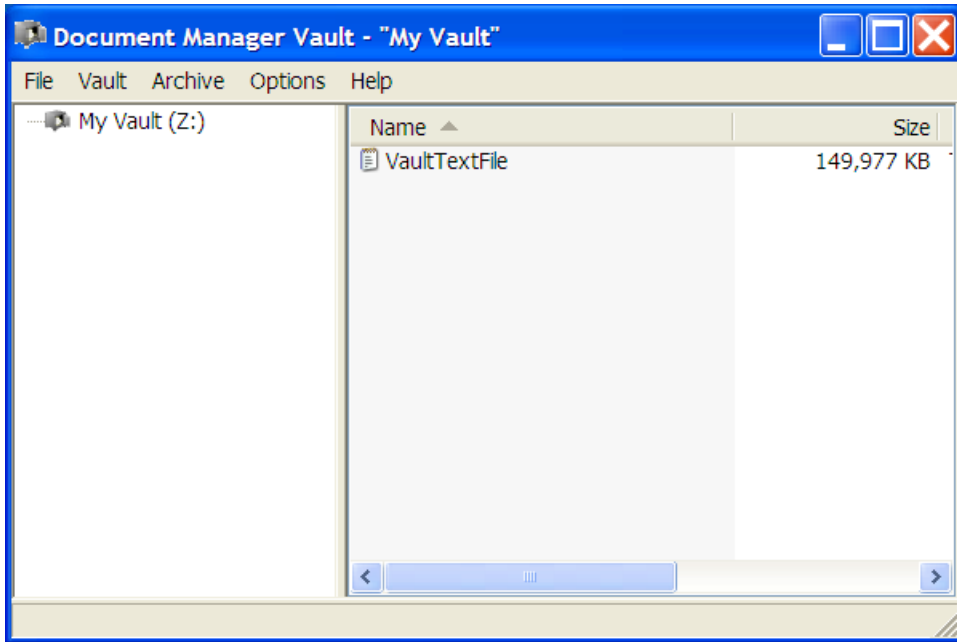


Figure 18: Opened Vault with Text File

Once the vault is created, files can be added to it as if it were a separate drive partition. The vault can most easily be accessed by clicking on My Computer and showing up as a local disk (typically Z:). A vault can either be in a locked or unlocked state (unlock and locked are also referred to as logged in or logged out at times by the vault software). If the vault is locked, double-clicking on its drive icon will result in a "vault not accessible, access denied" error. To log into the vault, a user may go through the document manager program or right-click on the vault drive icon and select 'Vault Login'. Once the correct password is entered, the vault can be accessed like any local drive. In the same manner, right-clicking on the vault drive icon and selecting 'Vault Log Out' will log the user out of the vault.

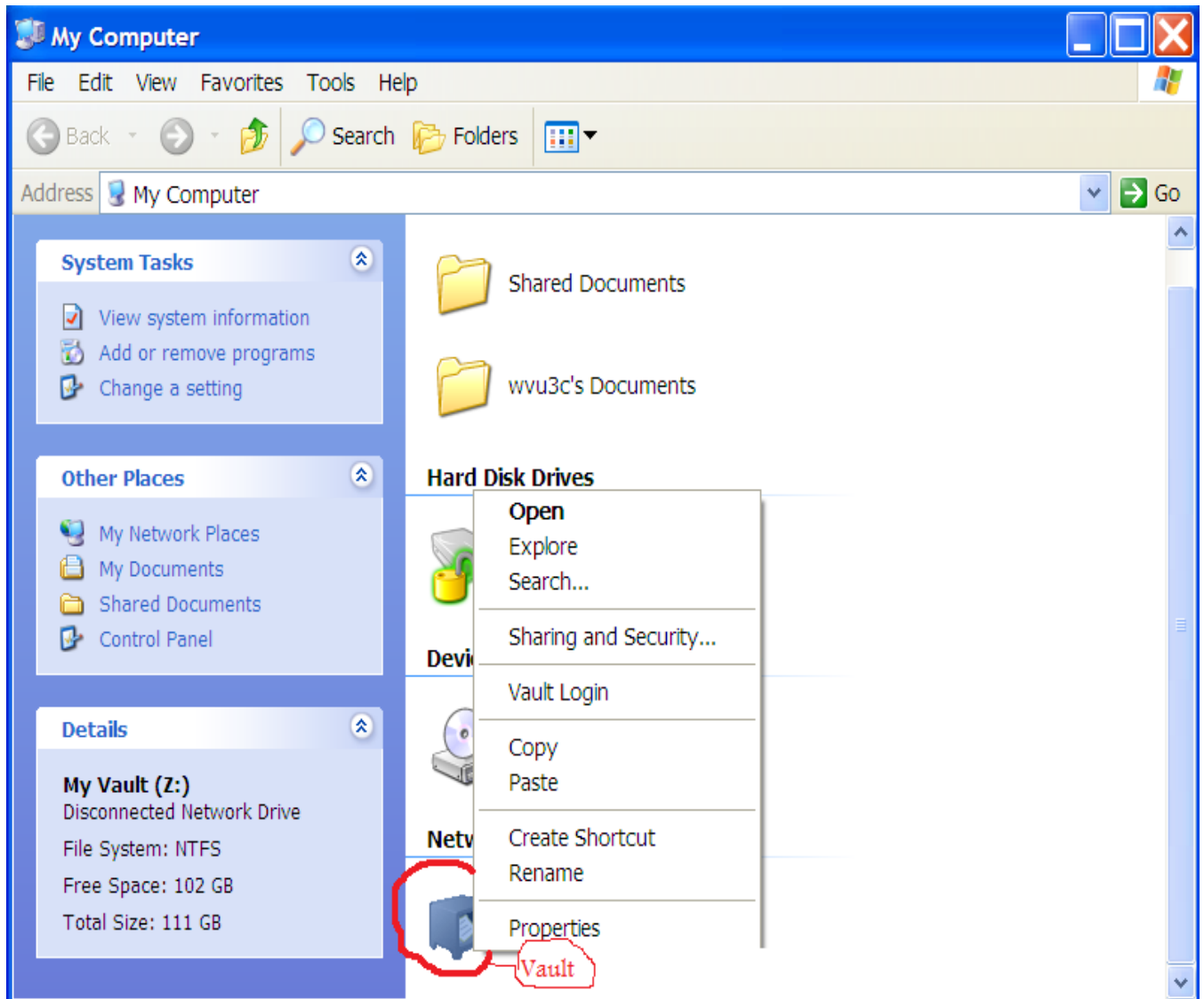


Figure 19: Vault options when locked (user logged out)

When a file is deleted from the vault, it is gone forever. Therefore, if a user is logged into the vault, the user can choose to back up the vault. Backing up the vault can be done by right clicking on the vault drive icon and selecting 'Vault Backup'. The backup function compresses the data in the vault and saves a backup as a .dmibku file. Once 'Vault Backup' is selected, the user must provide a new vault backup password to encrypt the vault backup. The user must then provide the TPM owner

password. The current user does not have re-enter the vault password as it is assumed that the user, already being logged into the vault (required to make a backup), knows the vault password. A compressed backup is then created at a user determined location.

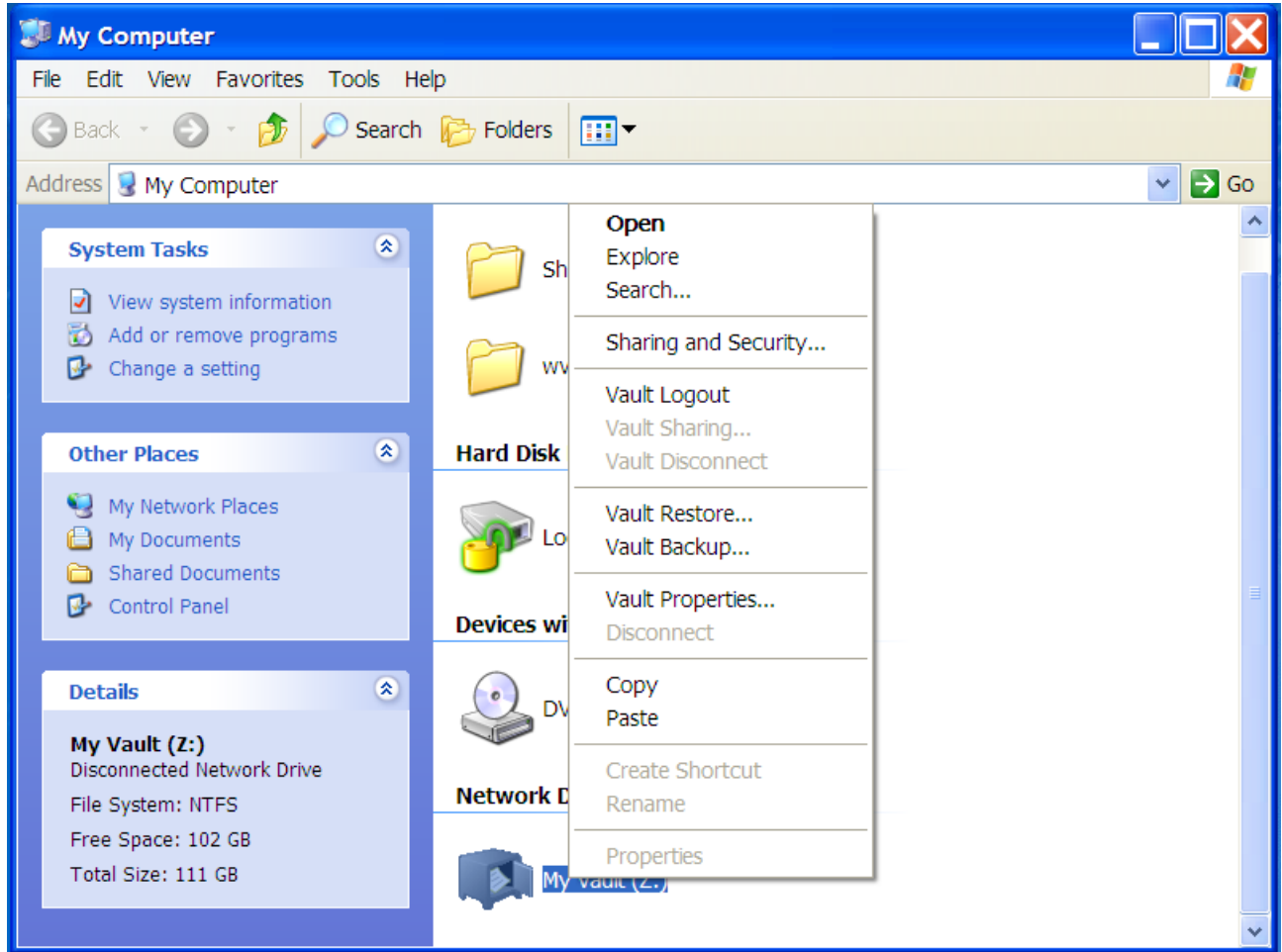


Figure 20: Vault options when unlocked (user logged in)

The vault backup varies in size, depending on the contents of the original vault. Most video, audio, and image files are compressed to begin with, so vault backups that contain only the aforementioned media will resemble the original vault in size. Text based files, on the other hand,

compress greatly when a backup is done. A vault of ~146Mb filled with various text files compresses to less than 1Mb as a backup.

To restore a backup, right click on the vault drive icon (user must be logged in) and select 'Vault Restore'. The user must specify the location of the vault backup file to restore the vault. As long as none of the TPM keys have changed (cleared and regenerated) the user does not have to supply the vault backup password to restore the vault. If TPM keys have changed, passwords should be supplied as prompted.

7.5 Back Up the TPM Encryption Keys

Using the EMBASSY Security Center, it is possible to create a backup archive of the TPM keys. Unlike using the tpm.msc interface in Windows Vista, TPM keys can easily be backed up in case of an emergency. If the TPM keys are lost, due to a malfunction in the system board, moving the hard drive to a different computer, or clearing the TPM, the data can still be recovered if the user has the encryption keys backed up. EMBASSY Security Center considers creating an archive of the TPM encryption keys as an essential step in protecting encrypted data. If the archive file is lost or damaged it may be impossible to recover any encrypted data.

In order to create a backup key file, navigate to start->All Programs -> Security by Wave Systems -> EMBASSY Security Center. Select the Archive and Restore tab. Select the Archive TPM keys for the current user Archive button.

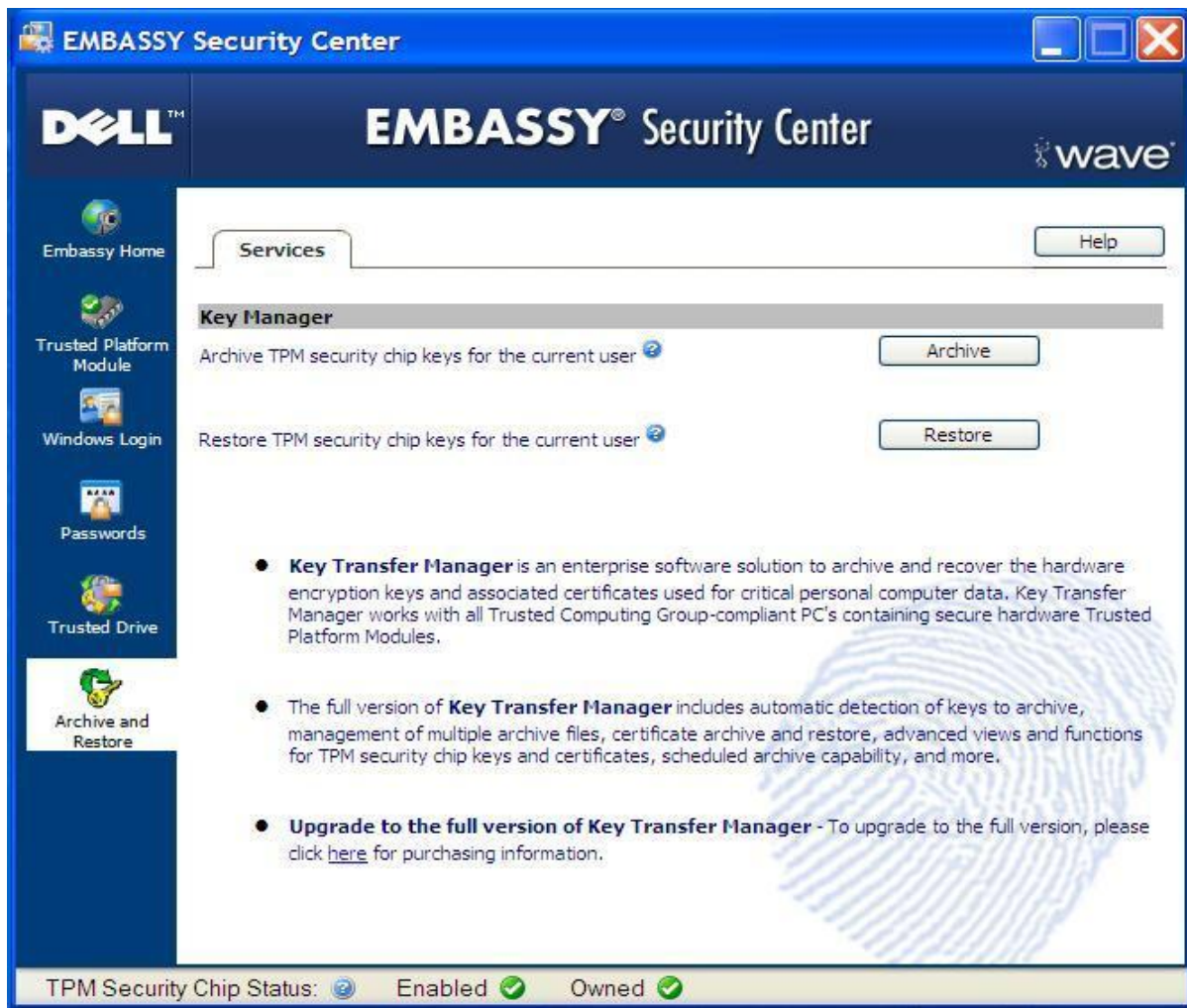


Figure 21: Creating a key archive

A browse window launches to allow the user to specify a location where the archive file will be saved. The backup file can be stored locally or on a different storage medium. As storing the file locally somewhat defeats the purpose of backing up the TPM keys, saving the archive file on an alternate storage media such as a network drive or removable USB memory device is recommended.

When prompted, the user must create a password for the key archive file. Without the key archive password it is impossible to restore the TPM keys. Enter the TPM owner password and Windows User password when prompted and select a location to save the backup key file.

8 Appendix D: Embassy Trusted Drive Manager

The EMBASSY Trust Suite also provides support for Full Disk Encryption (FDE) hard drives. FDE hard drives use hardware based full encryption technology. To enable hardware encryption, go to start->All Programs -> Security by Wave Systems -> EMBASSY Security Center and select the Trusted Drive Manager from the menu on the left. The embassy Trusted Drive Manager (ETDM) provides management functions for Trusted Drives. ETDM provides the security of hardware-based data protection and prevents data access by unauthorized persons. Trusted Drives have hardware encryption embedded in the drive hardware which ensures that all data stored on the drive is encrypted. FDE does not require the presence of a TPM chip. However, the TPM chip can be used to wrap the decryption key for the drive using a storage key. By wrapping the storage key, the TPM ties the encrypted drive to a particular system. Therefore, ETDM in combination with TPM ensures that the only authorized users can access encrypted data.

EDTM provides functions to initialize drive security, manage the drive security functions, and enroll / manage drive users. ETDM provides management functions for both the drive administrator and the drive user.

To enable hard drive encryption, select 'Initialize' from the Trusted Drive menu in the embassy Security Center. A user name of a currently valid windows user account must be entered, along with a password to continue. Although the user name must be of a user that exists, the password does not

have to be the password associated with that user name. Multiple users can be added to a single drive. After everyone is added, EDTM prompts the user to create a backup file containing the hard drive encryption password at a user specified location (the location, again, is typically not located on the encrypted local drive, as a user could not recover the password backup from an encrypted drive externally).

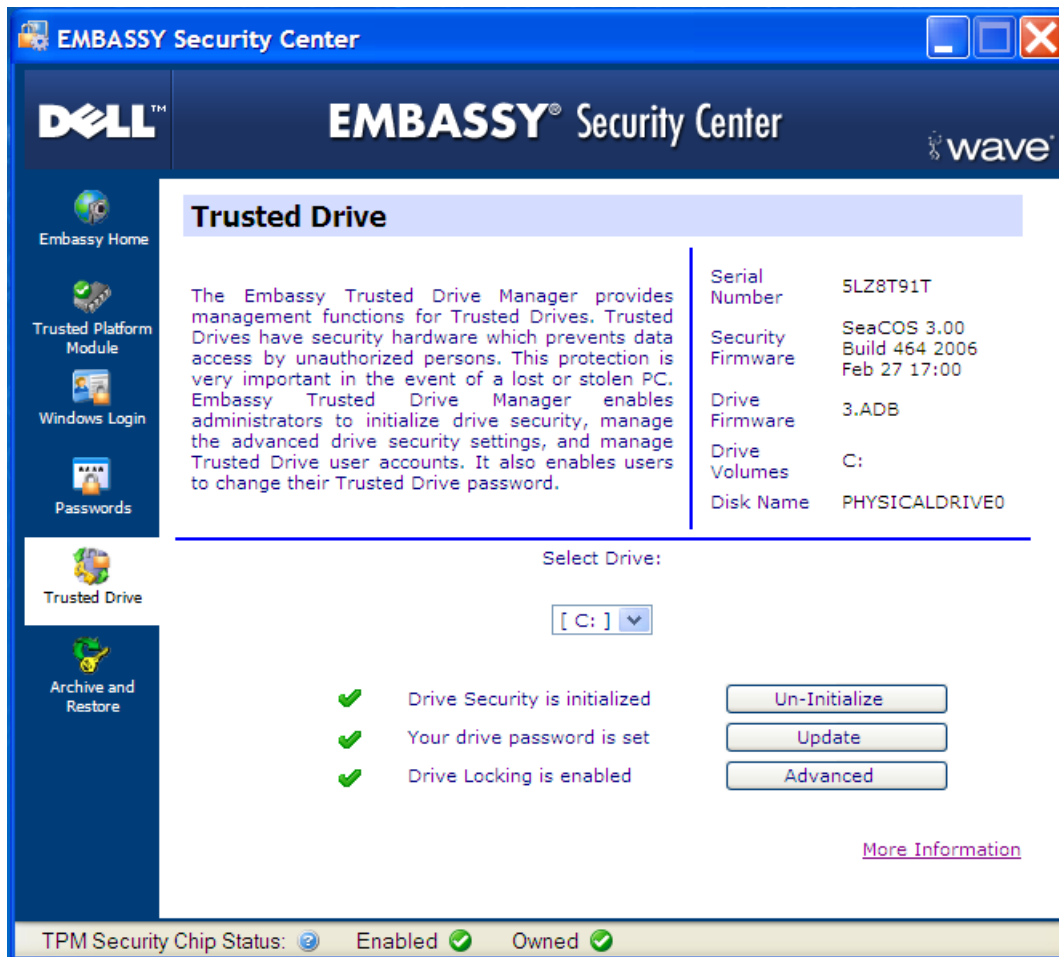


Figure 22: Trusted Drive Main

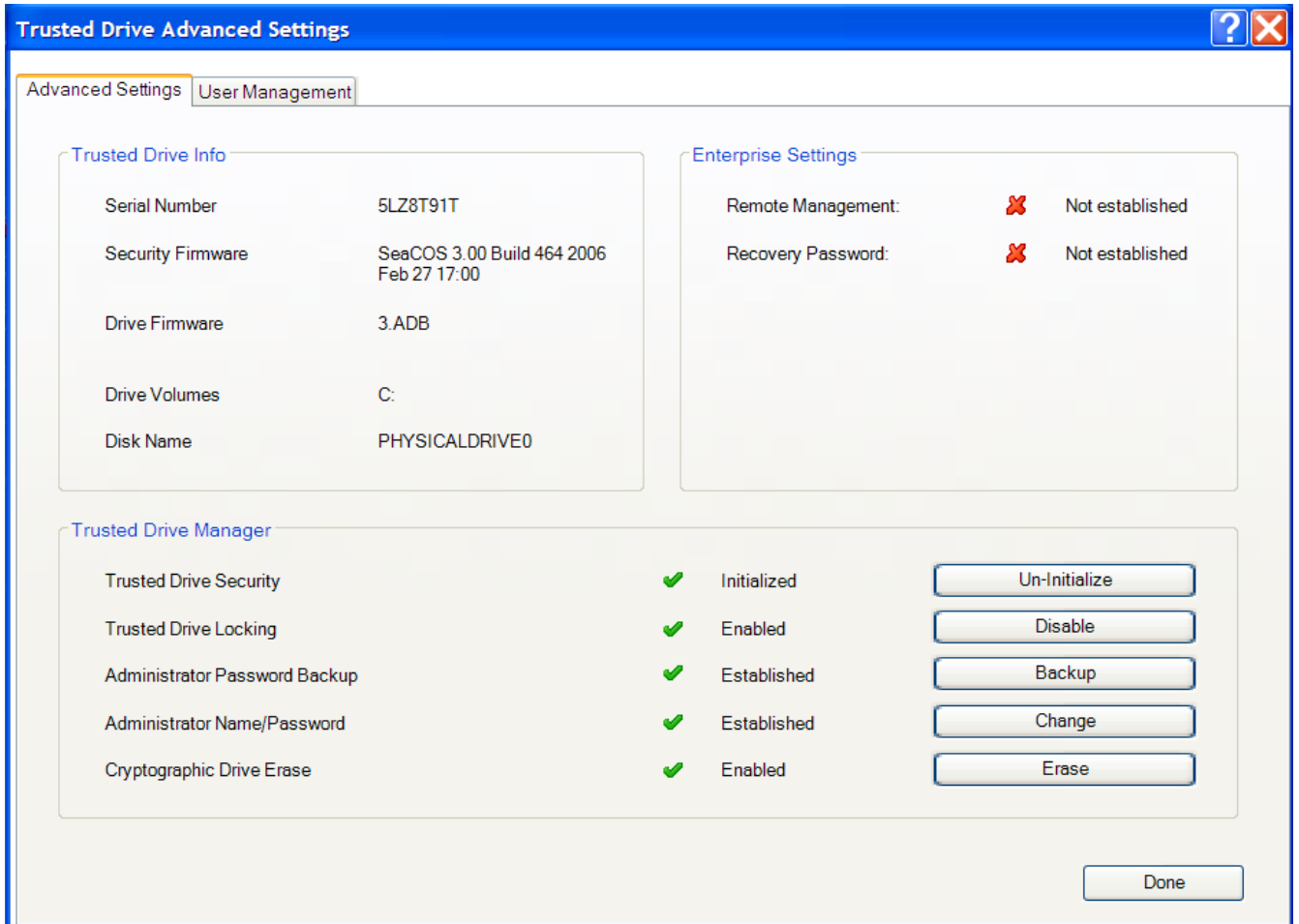


Figure 23: Trusted Drive Advanced Options

The hard drive password backup is an XML file labeled wave_tdm_backup_[year]_[month]_[day]_[hr]_[min] where variables are in brackets and represent the date/time when the backup was created. Inside the backup file, the user names and passwords for all authorized users of the encrypted drive are listed in plain text, along with the drive's serial, security ID, and Domain name. An example of a password backup can be seen below.

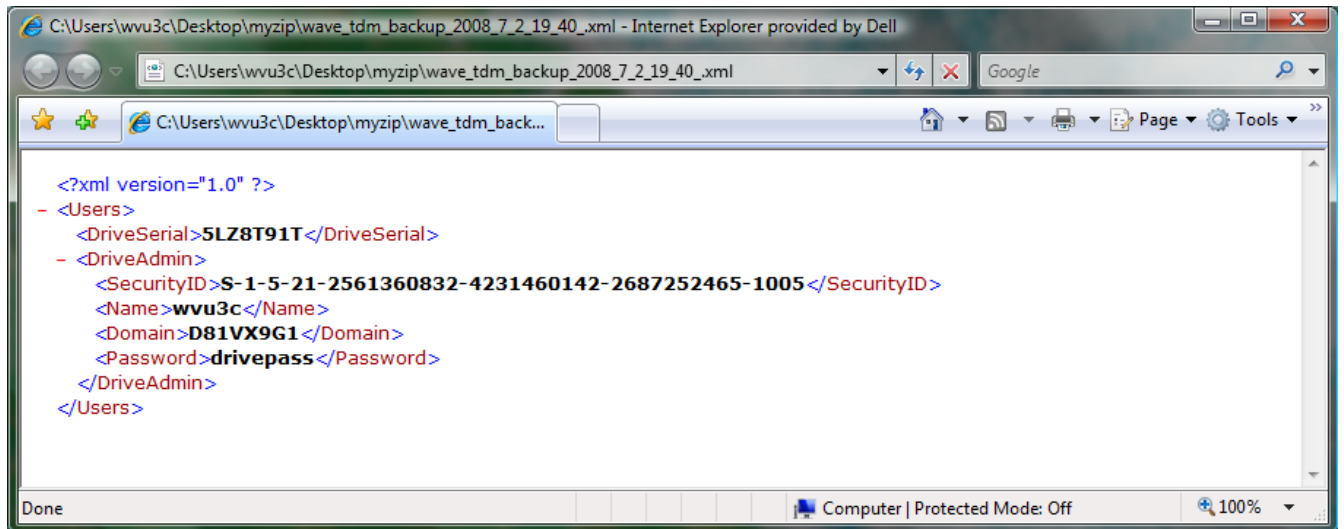


Figure 24: Encrypted hard drive password backup file

Once the drive hardware encryption is turned on, a screen prompts the user for a user name and password every time the computer is started. Failure to enter the correct user name and password, results in the drive remaining encrypted.

8.1 Additional Windows Login Options

The Windows Login tab provides two primary functions. It enables secure Windows login, which adds an additional layer of security on top of the standard Windows login and allows the user to enable other login methods, such as scanning a fingerprint with a biometric reader.

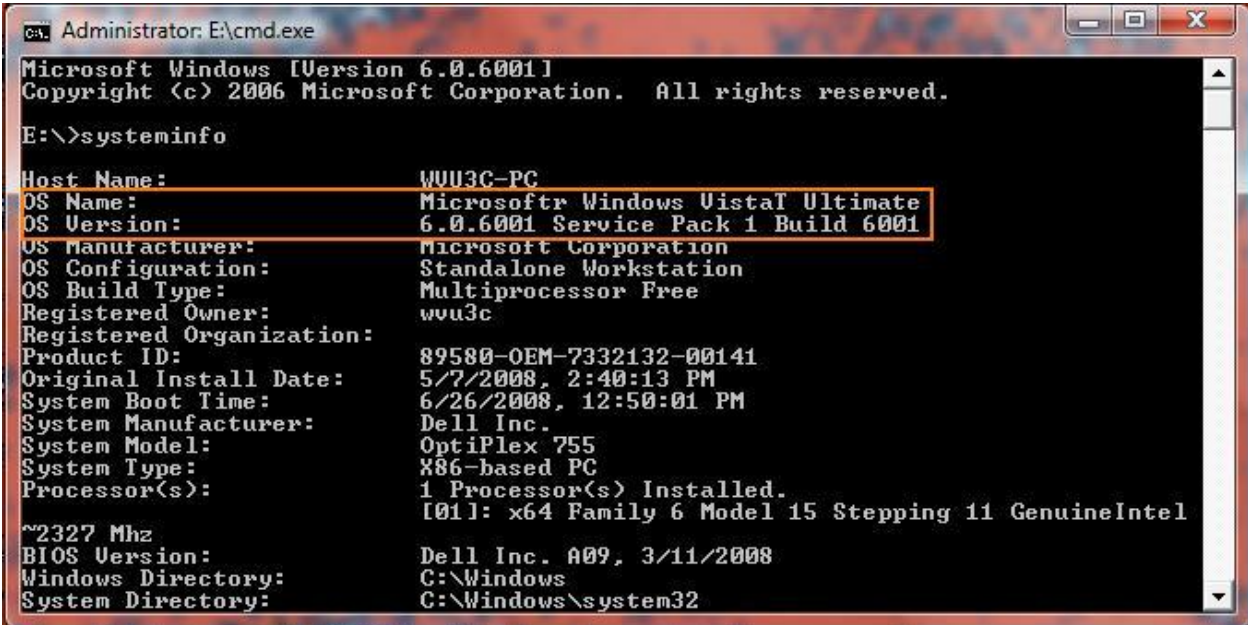
For systems equipped with a fingerprint reader, selecting this option allows the user fingerprint to be used instead of the Windows Login password.

To enable Secure Windows Login in the EMBASSY Security Center, select the Windows Login tab. Place a check in the Enable Secure Windows Login checkbox. Select the appropriate Authentication Type (the default is Password or Biometric). Click on the Apply button at the bottom of the application window. Reboot the computer.

9 Appendix E: Windows BitLocker Disk Encryption

9.1 BitLocker Encryption Analysis without TPM

In order to determine if BitLocker is present on a computer, determine the Windows version to see if BitLocker is supported. The operating system name and version can be found by double clicking on the “cmd.exe” file that has been copied from a trusted computer. After opening the cmd.exe program, type “systeminfo” and hit the “enter” key. Below, you can see the version of Windows and any service packs that are installed.



```
Administrator: E:\cmd.exe
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

E:\>systeminfo

Host Name:                WUU3C-PC
OS Name:                  Microsoft Windows Vista Ultimate
OS Version:               6.0.6001 Service Pack 1 Build 6001
OS Manufacturer:        Microsoft Corporation
OS Configuration:       Standalone Workstation
OS Build Type:            Multiprocessor Free
Registered Owner:        wwu3c
Registered Organization:
Product ID:               89580-OEM-7332132-00141
Original Install Date:    5/7/2008, 2:40:13 PM
System Boot Time:         6/26/2008, 12:50:01 PM
System Manufacturer:     Dell Inc.
System Model:              OptiPlex 755
System Type:              X86-based PC
Processor(s):             1 Processor(s) Installed.
                          [01]: x64 Family 6 Model 15 Stepping 11 GenuineIntel
                          ~2327 Mhz
BIOS Version:             Dell Inc. A09, 3/11/2008
Windows Directory:        C:\Windows
System Directory:         C:\Windows\system32
```

Figure 25: Operating System Name and Version

If the computer is running Windows Vista Ultimate or Windows Vista Enterprise, check to see if there are 2 or more drives shown in my computer. BitLocker needs a partition of ~1.5GB of space. Computers using BitLocker will have a drive similar to the one below that is labeled S: by default.

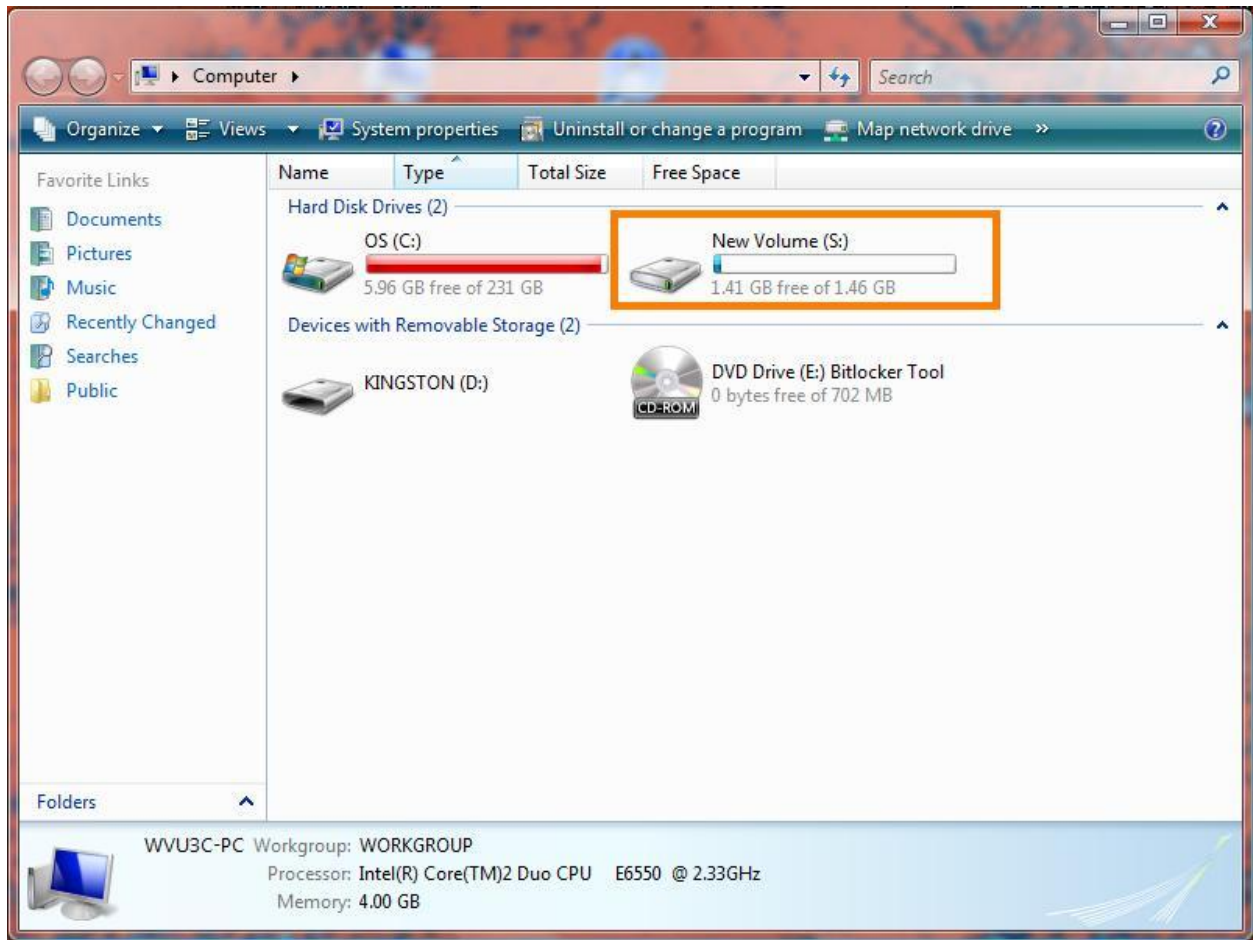


Figure 26: Windows BitLocker Volume

BitLocker user authentication without a TPM uses only a USB flash drive. A USB flash drive containing the startup key must be present when the computer is booted. . If this key is not present, the computer will not boot up normally. If the USB key is not present, but the machine is logged into Windows (which it should be if systeminfo was just executed), the BitLocker recovery key can be obtained. Once the recovery key is obtained, the drive can be decrypted and viewed offline. The recovery key can be obtained by doing the following:

- Right-click on the cmd.exe file that you copied from a trusted Windows Vista machine

- Click “run as Administrator”

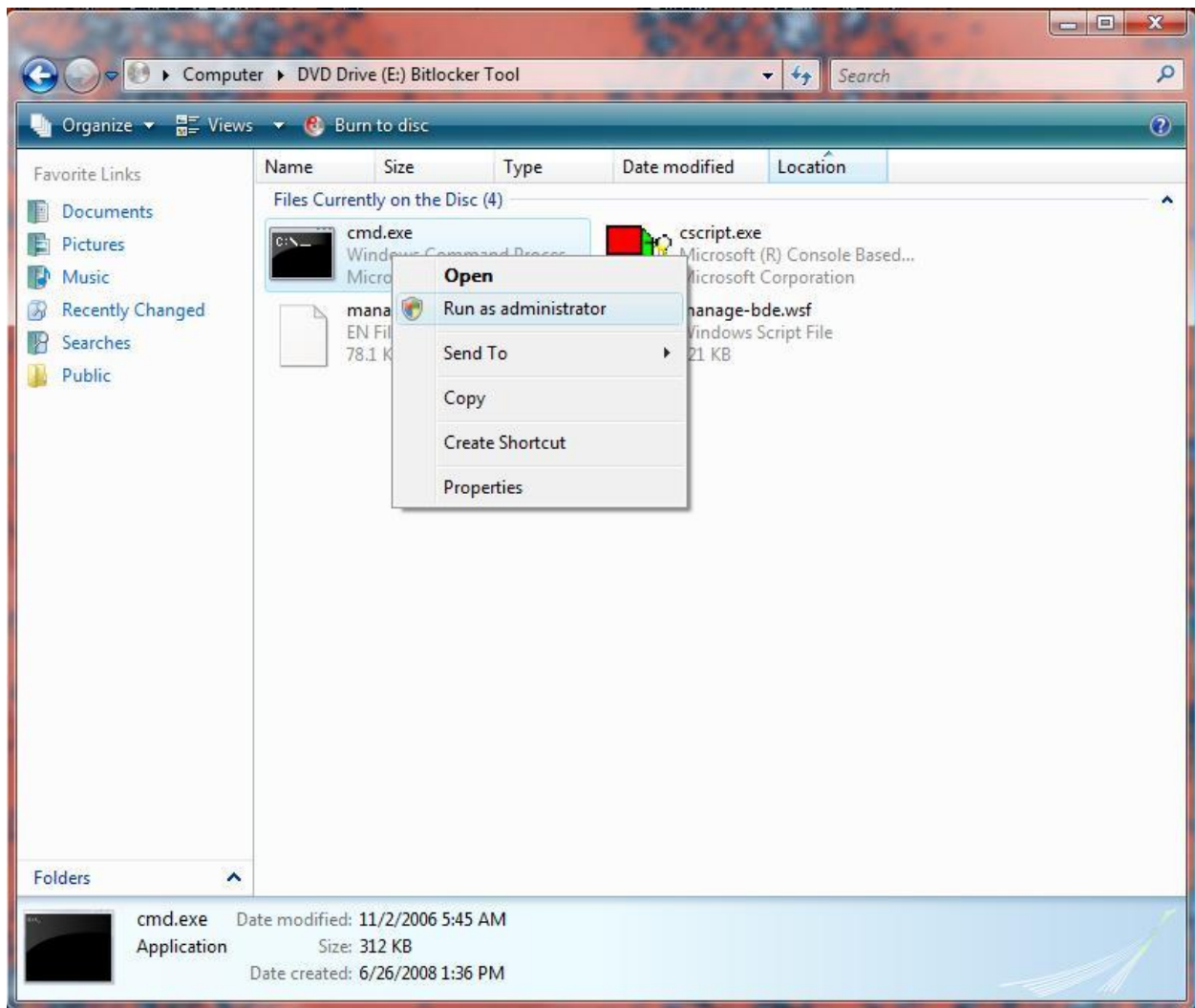


Figure 27: Run cmd.exe as Administrator

- type “cscript manage-bde.wsf –protectors –get c:”

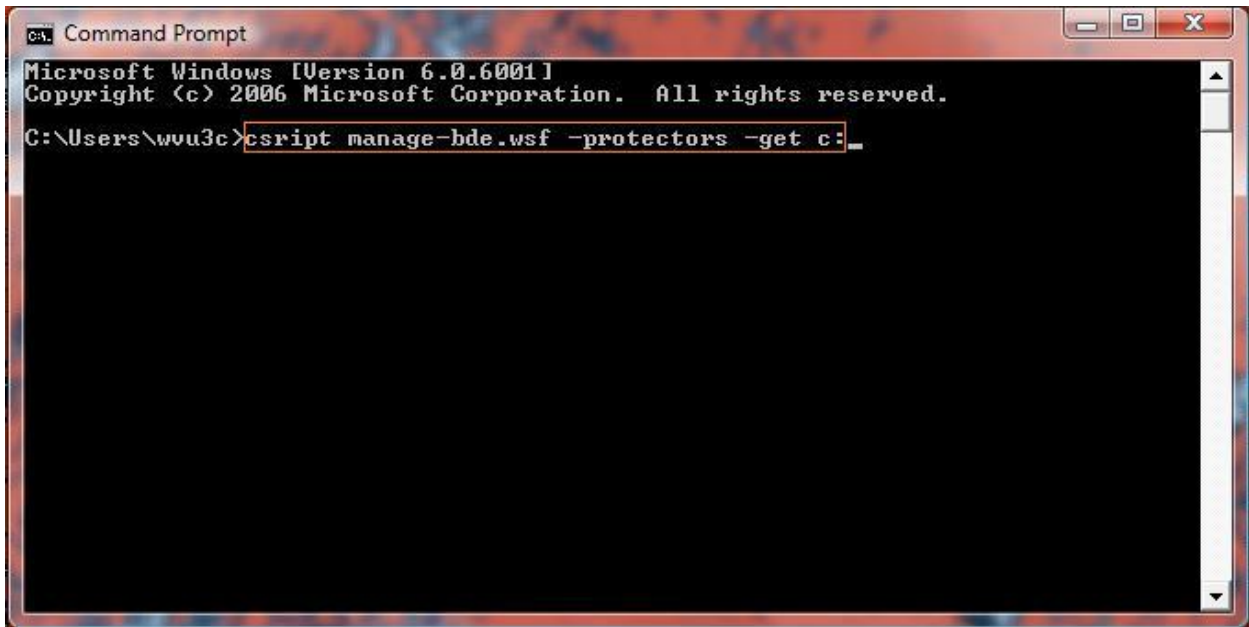


Figure 28: Manage-bde.wsf

- Write down the recovery key for later use

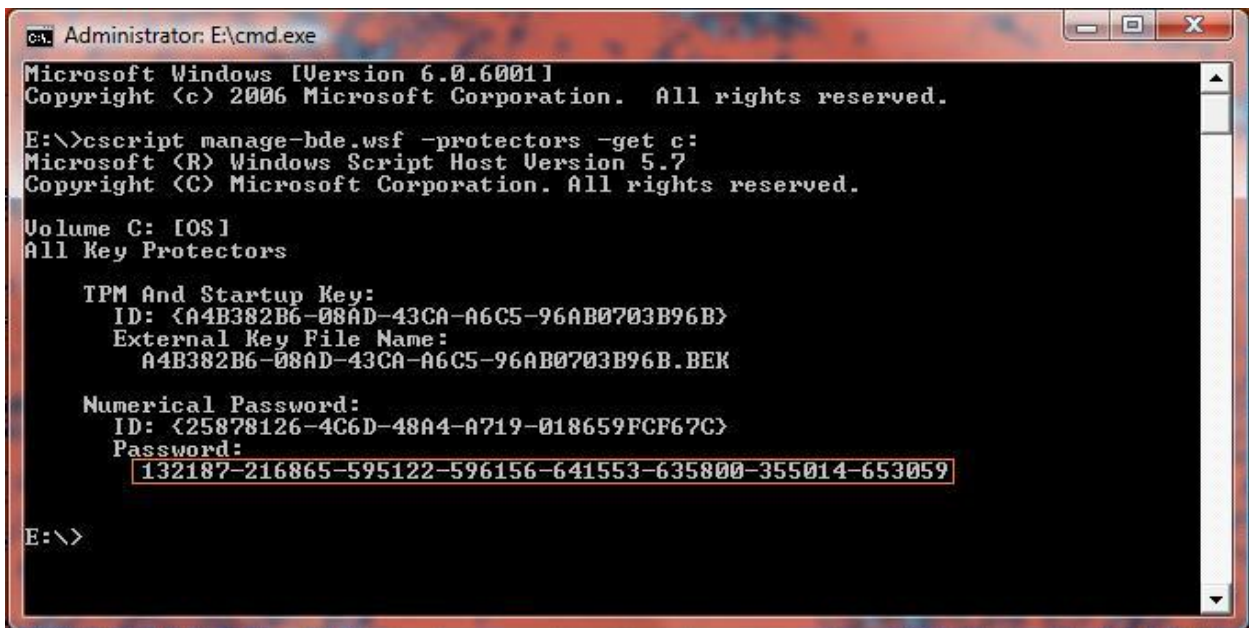


Figure 29: BitLocker recovery key

9.2 BitLocker Encryption Analysis with TPM

Using the same method mentioned previously, determine whether the current machine is able to use BitLocker. BitLocker with TPM can be in one of the four following scenarios:

1. TPM
2. TPM + PIN
3. TPM + USB flash drive
4. TPM + PIN + USB flash drive (must have Vista SP1 installed)

TPM without any additional authentication is the least difficult to bypass because it is implemented under default settings. As long as the hard drive has not been removed and the early boot sequence has not been modified, the computer will start up normally. If there is a password on the user account difficulties may arise as BitLocker does not allow booting from a CD, DVD, or USB device. Booting from any other device breaks the Root of Trust Measurement, which is checked by the TPM. If the machine is logged into Windows, the recovery keys can be obtained using the `manage-bde.wsf` method seen above.

Using a PIN, USB, or PIN+USB in conjunction with the TPM does not change much, as far as recovery practices go. Without the PIN / USB key the only option for recovery is to hope the machine is on and obtain the BitLocker recovery password using the `manage-bde.wsf` command. Further, the `manage-bde.wsf` command must be run with administrative access. If the account attempting to run the recovery command is not a local administrator account, a password may be required to run `cmd` in administrative mode.

9.3 Using Recovery Keys

Once obtained, the BitLocker recovery key can be used in conjunction with the BitLocker Repair Tool (available from Microsoft) to recover the encrypted files. The repair tool is a command line based program that must be run in a Windows Vista environment. The environment can be in the form of a separate Vista system or a Vista install DVD. From the command line, the repair tool provides two main techniques for attempting to recover the encrypted files.

The first method of recovery is a file decryption and transfer function. The process is activated by starting the program running the following line from cmd.exe with administrative privileges: "**repair-bde *InputVolume OutputVolume -RecoveryPassword NumericalPassword***". This method uses the encrypted drive as the "InputVolume" and a separate storage hard drive for the "OutputVolume". The tool locates any recoverable files on the encrypted drive and places a decrypted copy on the destination drive. It is important to note that all information on the "OutputVolume" will be erased during the copying procedure. Also, the "OutputVolume" should be at least the same size as the encrypted volume to ensure there is no data loss.

The second method of recovery attempts to repair specific errors on the encrypted drive. The recovery tool will attempt to repair the encrypted drive's boot sector in order to restore normal BitLocker functionality. As such, this method is primarily used only if the boot sector is corrupted and is unable to correctly boot. The process is activated by starting the program using the following line from cmd.exe with administrative privileges: "**repair-bde *InputVolume -NoOutputVolume -RecoveryPassword NumericalPassword***". It should be noted that using either method does not result in the drive being fully decrypted.

10 Appendix F: Expanded TrueCrypt Experiments

Once TrueCrypt is installed / executed on a computer, it can be used to fully encrypt a targeted hard drive or create an encrypted volume on the drive. In the following TrueCrypt experiments, two hard drives were used. One hard drive was fully encrypted using the utility and the other was partially encrypted (encrypted volume). Once the hard drives were encrypted, several recognizable files were copied to each respective disk. When the files were placed on the drives, computer forensics tools such as Access Data's Forensic Toolkit 1.71 (FTK) and Winhex 14.6 were used to check the hard drives both physically and logically for the test files. The computer used to conduct the TrueCrypt experiments was running Windows XP Service Pack 2 with updates.

10.1 Standard TrueCrypt Volume Analysis:

Access Data's Forensic Toolkit (FTK) and Winhex were used to test physical access. Logical analysis was not possible in the case of TrueCrypt. Remember that correct password authentication is used by TrueCrypt to mount an encrypted volume and assign it a drive letter. Without entering the correct password, the encrypted volume will not be seen as a logical drive. In effect, if an investigator does not know that a encrypted volume is present, it will probably remain hidden. Once the correct password is entered, the encrypted volume is mounted and assigned a letter.

The most basic way to use way to use TrueCrypt is to create an encrypted volume inside of a file or folder. The user can specify a file at a chosen location to house an encrypted volume. Once the volume is created and password is set, the user must mount the volume in TrueCrypt in order to access it. Once the user is authenticated, the contents of the volume can be manipulated. In forensic analysis of the hard drive containing the encrypted volume, the files were undetectable except for one notable exception. When the drive was analyzed using FTK, the actual file in which the encrypted volume resides showed up as pieces of a larger file. For example, if the user created an encrypted volume

inside a pdf file called “test.pdf”, the FTK results would show test.pdf_1, test.pdf_2, test.pdf_3 etc. Each variation of test.pdf is exactly the size of one block of free space on the drive. The number of test.pdf files depends on the size of the contents of the encrypted volume. There are as many test.pdf files as are needed to accommodate the size of the contents of the encrypted volume. Initially, it seems that FTK can be used to find hidden TrueCrypt volumes using the previously detailed information. In reality, the number of files on a computer limits the effectiveness of such a search. Unless investigators know that TrueCrypt has been used on the hard drive that they are testing, the results given by FTK could easily be overlooked.

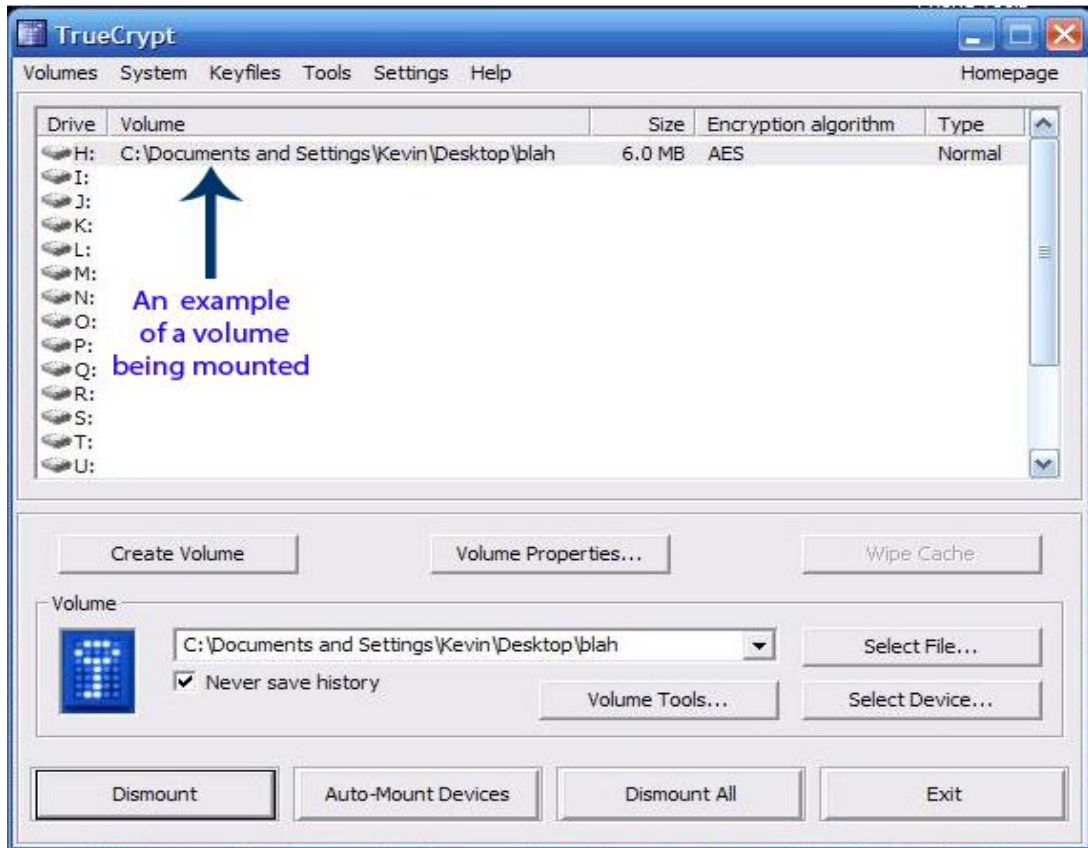


Figure 30: Mounting an Encrypted Volume One

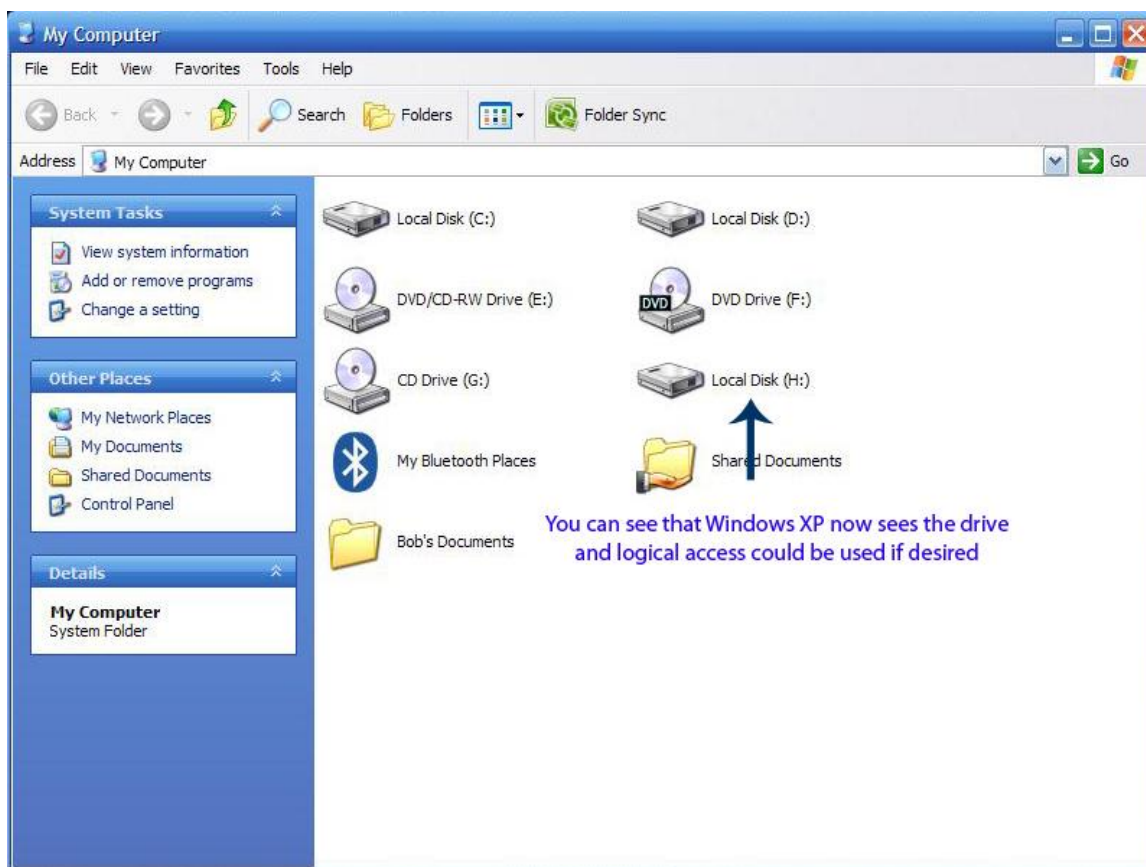


Figure 31: Mounting an Encrypted Volume Two

10.2 Hidden TrueCrypt Volume Analysis:

Using TrueCrypt, it is possible to create a hidden encrypted volume within a standard encrypted volume. When dealing with layered standard and hidden volumes, the forensics tools were unable to detect either volume if the standard volume is not mounted. When the standard volume was mounted by TrueCrypt and analyzed with Winhex and FTK, the existence of the hidden volume was not detected in any way.

10.3 Full Drive Encryption Analysis:

When TrueCrypt was used to encrypt the entire hard drive, everything was encrypted except for the first cylinder, which contained the boot-loader. When analyzing the drive using the forensic software, the only thing that was readable was boot -loader. The remainder of the drive was fully encrypted and unreadable.

11 Appendix G: Computer Forensic Encryption Checklist

Software Encryption

- Recover page files
- Recover a copy of RAM
- Recover any Windows backup files (they may hold unencrypted versions of currently encrypted files)
- Do a live image of the drive on scene

Hardware Based Full Disk Encryption

- If FDE is found to be enabled, do not let the computer go to sleep
- Recover and examine all external media from the scene to look for the backup file
- Do a live image of the drive on scene
- Recover the entire computer as evidence as the FDE drive can be married to the motherboard with TPM.

TPM based encryption

- Recover and examine all external media from the scene to look for backup files
- Do a live image of the drive on scene
- Recover the entire computer, as encrypted space can be married to the motherboard

BitLocker

- Attempt to use the BitLocker recovery console to disable BitLocker
- Recover and examine all media on the scene for backup files. Look for any papers that could have a BitLocker recovery key on them.
- Do a live image of the drive on scene
- Recover the entire computer, as BitLocker and TPM may be being used in conjunction

General Practice

- Do not give the suspect an opportunity to delete encrypted data
- Always recover all external storage devices and look at papers around the scene for passwords
- Record everything that is done on the computer
- If the computer has a TPM chip, take the whole thing as evidence

12 Appendix H: Experiments

12.1 TPM Verification

Test Plan

Test Name: TPM Verification

Test Title: Possible verification of the existence of an activated TPM on a Windows Vista machine

Purpose and Scope

This test is to determine if there is a way to verify that a TPM chip on a motherboard (of a computer that Windows Vista installed on its hard drive) has been activated, other than by using the tpm.msc interface.

Requirements

Two identical hard drives
Machine with TPM chip

Procedure

Microsoft Vista Ultimate will be installed on hard drive one. Once Vista is installed and bootable, a bit-copy (physical) of the drive one will be made onto drive two using BackTrack version 2. BackTrack is a live Linux/GNU distribution which focus on security and can be found at <http://www.remote-exploit.org>. The two drives will then be hashed to confirm that they are, indeed, identical.

Once the drives are confirmed as identical, a hash of every file on drive one will be taken and saved to a thumb drives using BackTrack. Drive two will then be booted. TPM will be turned on and enabled using the tpm.msc interface in Vista. It should be noted that part of the TPM activation must be done in the BIOS. When the TPM has been activated, a hash of every file on drive two will be taken and saved (again, using Backtrack). The hashes of all files from drive one (TPM not activated drive) and drive two (the TPM activated drive) will then be compared using BackTrack. The purpose of the drive comparison is to determine if activating TPM creates any discernable markers that could be used by computer forensic investigators, during live analysis, to confirm that a computer has an activated TPM. If a file

is found to have been changed from drive one to drive two, it will be checked for evidence of TPM related information.

Expected Results

- Activating the TPM should alter files on Vista machine in such a way that the changes will be visible to an investigator.

Test Setup

Test Drive 1:

Seagate Barracuda 7200.10
Capacity: 250 GB
S/N: 6RY7YKRF
Connection: SATA

Test Drive 2:

Seagate Barracuda 7200.10
Capacity: 250 GB
S/N: 6RY7YKVN
Connection: SATA

Test Computer:

Board: Dell Inc. 0GM819
Serial Number: CN1374081U0AQ8.
Bus Clock: 1333 megahertz
BIOS: Dell Inc. A09 03/11/2008
Enclosure Type: Desktop
Memory: 3326 Megabytes Installed Memory
Processor: 2.33 gigahertz Intel Core2 Duo
OS: Windows Vista Ultimate (build 6000)

Test Results

After comparison of the two drives' files, there were found to be over one hundred differing files between the two drives. Many of the files seemed irrelevant, in respect to tpm activation.

Also, many of the differences between files were found to be time stamps, most likely resulting from the boot required (drive two) to enable TPM. Of the differing files the following were deemed the most relevant and examined for traces of TPM activation:

Boot/bootstat.dat
Windows/inf/TPM.pnf
Sys32/config/components/SAM Components Security

Very little discernable trace of TPM activation could be found in the files. The only possible indicator found was in Windows/inf/TPM.pnf at hex offset 00000E80. At the aforementioned offset, a flag in the file was changed from D to E, possibly indicating a change from a disabled TPM to an enabled TPM.

Summary

There were no clear, easily testable indicators of TPM activation found outside of the tpm.msc interface on the Vista test machine.

12.2 TrueCrypt Test 1

Test Plan

Test Name: TrueCrypt Volume Test 1

Test Title: Find a TrueCrypt Volume

Purpose and Scope

This test is to determine if files inside an encrypted volume, created by TrueCrypt, are visible by using computer forensics tools.

Procedure

An encrypted volume will be created in a file called test.pdf, using TrueCrypt 4.3a. A file called encr_file.txt will be created inside of the encrypted volume. The file will be populated with the phrase “this file should be invisibled”. The phrase “invisibled” will be added to the file in order to make searching for traces of the encrypted file easier. The key phrase will be duplicated in the text file until the text file grows to a reasonably large size. In this case, the text file will be made to be around 5Mb. Once the file is created, an image of the drive will be taken using Forensics Tool Kit Imager Lite 2.54. The image will be saved onto an external drive. The image will then be analyzed using Forensics Tool Kit 1.8.

Expected Results

- 1) The file created within the TrueCrypt volume should remain invisible during analysis.

Test Setup

Test Drive 1:

Western Digital WD800
Model: WD800JB – 00CRA1
Capacity: 80 GB
S/N: WMA8E5918199
Connection: Parallel ATA

Test Computer:

Enclosure Type: Desktop

Board: ASUSTeK Computer INC. A7N8X REV 1.xx

Bus Clock: 133 megahertz

BIOS: Phoenix Technologies, LTD ASUS A7N8X Deluxe ACPI BIOS Rev 1002 Beta 006
01/28/2003

Memory: 512 Megabytes Installed Memory

OS: Windows XP Service Pack 2

Test Results

A string search for the phrase “invisible” yielded no results. The name of the file inside of the encrypted space, “encr_file.txt”, was found in “recent documents” but was not linked to the encrypted area.

Summary

There were no clear, easily testable indicators of TPM activation outside of the tpm.msc interface on the Vista test machine.

12.3 TrueCrypt Test 2

Test Plan

Test Name: TrueCrypt Test 2

Test Title: Full Encryption Analysis

Purpose and Scope

This test is to determine what data is recoverable from a drive using TrueCrypt 4.3a to encrypt an entire hard drive.

Procedure

TrueCrypt will be used to encrypt an entire hard drive. The drive will then be plugged in as a secondary hard drive and analyzed with FTK 1.8.

Expected Results

- 1) The volume encrypted with TrueCrypt should be unreadable

Test Setup

Test Drive :

Western Digital WD800
Model: WD800JB – 00CRA1
Capacity: 80 GB
S/N: WMA8E5918199
Connection: Parallel ATA

Test Computer:

Enclosure Type: Desktop
Board: ASUSTeK Computer INC. A7N8X REV 1.xx
Bus Clock: 133 megahertz
BIOS: Phoenix Technologies, LTD ASUS A7N8X Deluxe ACPI BIOS Rev 1002 Beta 006
01/28/2003

Memory: 512 Megabytes Installed Memory
OS: Windows XP Service Pack 2

Test Results

Upon analysis, everything was found to be encrypted except for the first cylinder of the hard drive, which contained the boot-loader. The remainder of the drive was fully encrypted and unreadable.

Summary

The hard drive encrypted using TrueCrypt was unreadable.

12.4 TrueCrypt Tests 3

Test Plan

Test Name: TrueCrypt Tests 3

Test Title: Using Indirect Analysis to Attack TrueCrypt

Purpose and Scope

Since direct analysis of the TrueCrypt volume yielded no result, an indirect approach will be tested. In previous tests (Vault Tests 2) Windows pagefiles have been shown to contain information that could be useful to computer forensic investigators. This test will analyze the Windows pagefiles of a machine with TrueCrypt to determine if there is any forensically useful information.

Procedure

An image of the test drive will be taken using Forensics Tool Kit Imager Lite 2.54. The image will be saved onto an external drive. The image will then be analyzed using Forensics Tool Kit 1.8. The encrypted volume contains several MS Word files.

Expected Results

- 1) Some evidence of the Word files and TrueCrypt should be found in the pagefiles.

Test Setup

Test Drive 1:

Western Digital WD800
Model: WD800JB – 00CRA1
Capacity: 80 GB
S/N: WMA8E5918199
Connection: Parallel ATA

Test Computer:

Enclosure Type: Desktop
Board: ASUSTeK Computer INC. A7N8X REV 1.xx

Bus Clock: 133 megahertz

BIOS: Phoenix Technologies, LTD ASUS A7N8X Deluxe ACPI BIOS Rev 1002 Beta 006
01/28/2003

Memory: 512 Megabytes Installed Memory

OS: Windows XP Service Pack 2

Test Results

Pieces of the contents of one of the Word files were found in the pagefile. Although the entire file was not found, there is evidence to indicate that files in a TrueCrypt volume can be found outside of the encrypted space. Like the TPM encryption experiments, the plain text password for the TrueCrypt volume was found throughout the pagefile. However, there was no clear indicator of where the password could be found. Therefore, until a clear indicator of password location can be found, the pagefile's use is somewhat limited.

Summary

Some contents of TrueCrypt volume were viewable outside of encrypted space. The pagefile contained the plain text password to the TrueCrypt volume, but there was no clear indicator to its location.

12.5 Vault Tests 1 (TPM based encryption)

Test Plan

Test Name: Vault Tests 1

Test Title: Tests on the “Vault” created by Wave Systems’ Embassy Security Suite.

Purpose and Scope

These tests are to discover tendencies about the Vault and TPM based encryption. A series of tests will be run on the Vault to determine what kind of computer forensic analysis is most effective for TPM based encryption. For the sake of removing confusion, “image”, for these tests, refers to a copy of a computer’s hard drive and “picture” refers to what would be traditionally associated with the word image (.jpg, .png, .gif etc).

Procedure

All tests will be run on a computer with a TPM based encrypted partition. For the first test, a bit-copy of the test computer’s hard drive will be made with BackTrack. The copy will then be analyzed using Forensics Tool Kit 1.8. This test represents the traditional method of analysis used in a computer forensic investigation.

- 1) A series of text files will be created inside of the Vault. The names of the files and the contents of the files will be searched for using FTK.

The next two tests deal with live analysis. After each test, an image of the hard drive will be taken with Forensics Tool Kit Imager Lite 2.54. The image will be saved onto an external drive. The image will then be analyzed using Forensics Tool Kit 1.8.

- 2) A series of text files will be created inside of the Vault. The names of the files and the contents of the files will be searched for using FTK. The vault will be closed (user not logged into the encrypted area) during live image recovery.
- 3) A series of text files will be created inside of the Vault. The names of the files and the contents of the files will be searched for using FTK. The vault will be open (logged into the encrypted area) during live image recovery.

The fourth and fifth tests, again, use live analysis. A live analysis of the machine will be done with Helix 1.7. In this case, no copy of the test drive will be created with Helix. Only live

preview of the files on the drive will be performed. Helix contains several tools that will be used during analysis including: a tool that shows all mounted drives on a system and a tool that gathers all pictures found on a drive.

- 4) A series of pictures will be placed into the Vault. The vault will be closed (user not logged into the encrypted area) during live preview of the system.
- 5) A series of pictures will be placed into the Vault. The vault will be open (logged into the encrypted area) during live preview of the system.

Requirements

- 1) Embassy Security Suites by Wave Systems
- 2) Machine with TPM chip

Expected Results

- 1) Since the machine is powered down, no evidence should be recoverable.
- 2) Since the vault is closed, no evidence should be recoverable
- 3) Since the vault is open during live imaging, the files in the vault should be recoverable
- 4) Since the vault is closed, no picture files should be recoverable
- 5) Since the vault is open the pictures in the vault should be viewable

Test Setup

Test Drive:

Seagate Barracuda 7200.10
Capacity: 250 GB
S/N: 6RY7YKRF
Connection: SATA

Test Computer:

Board: Dell Inc. 0GM819
Serial Number: CN1374081U0AQ8.
Bus Clock: 1333 megahertz
BIOS: Dell Inc. A09 03/11/2008
Enclosure Type: Desktop
Memory: 3326 Megabytes Installed Memory
Processor: 2.33 gigahertz Intel Core2 Duo
OS: Windows Vista Ultimate (build 6000)

Test Results

- 1) No evidence of an encrypted partition was found. It should be noted that FTK 1.8 contains a tool that automatically flags encrypted files that it finds. When the drive was processed, no files were flagged by FTK. String searches for the contents of the encrypted yielded no results. However, a string search for the names of the files, using FTK, yielded a positive match in the ntuser.dat file. The name of one of the files created in the Vault was found, in plain text, within the ntuser.dat file. As a computer forensic investigator would not typically know the exact name of the file that was to be searched for, especially if there was no evidence of encryption in the first place, (see first sentence in result) the name of a file in the Vault being found in ntuser.dat is not particularly useful in most cases.
- 2) It was immediately discovered, when trying to perform live analysis that FTK Imager Lite does not work on Windows Vista unless user access control is turned off. Once UAC was turned off, the imager ran as expected. The live image with the Vault closed, analyzed using FTK 1.8, yielded generally the same result as Test 1. Again, there was no direct of evidence encrypted data on the drive. Like the previous test, the searching for the contents of the encrypted files came up negative, but the names of the files came up in ntuser.dat. There were also a few instances of the filenames in the pagefile.sys. Again, unless the investigator knew the name of the file that was being looked for, the usefulness of finding instances of the encrypted files' names is somewhat limited.
- 3) Surprisingly, this test yielded the same results as the previous one. Having the Vault open did not aid in recovering the data when FTK Imager Lite.
- 4) As expected, the picture search did not find photos that were inside of the Vault. Helix also found no evidence of an encrypted virtual drive or partition. However, evidence of some picture files was found in the "recently used items". Pictures inside the Vault that were viewed still show up as being recently used. Looking at the "properties" of the picture shows the file path to the picture file. In this case, the Z: drive was the letter assigned to the Vault. The file path to the picture shows Z:[rest of the path to the picture file]. Clicking on the filename in the "recent items" area brings up a "bad shortcut" error message. Again, the usefulness of this information is limited, as a computer investigator may not know exactly what he/she is looking for, and the drive letter could just as easily be referring to a removable storage drive that is no longer plugged into the computer.
- 5) With the Vault open, the search found all picture files inside of encrypted space.

Additionally, the Z: drive (Vault) was visible as a mounted device and was able to be browsed. The only problem with this test was that the encrypted data was almost too transparent. An investigator that was not looking for encryption may mistake the results for normal, decrypted, information. In that case, the tendency may be toward shutting the computer down, which would render the data unrecoverable.

Summary

There was no concrete way found to identify TPM protected space. Imaging TPM protected space seems largely ineffective in recovering files. In fact, FTK Lite does not work on Windows Vista unless UAC has been turned off. Although some evidence of the encrypted files could be found using FTK, the usefulness of the evidence found was limited. Live analysis returned the best results, finding all of the encrypted files. However, an investigator must be vigilant, looking for encrypted space, as not recognizing evidence located in encrypted space may lead to the evidence being lost.

12.6 Vault Tests 2 (TPM based encryption)

Test Plan

Test Name: Vault Tests 2

Test Title: Getting around TPM protected space (the Vault)

Purpose and Scope

In previous experiments, it was shown that direct analysis of TPM protected space yielded limited results. The following experiments will more vigorously test for evidence of encrypted data outside of the actual protected space. TPM based encryption's reliance on third party applications to actually encrypt / decrypt data is a weakness that may be able to be exploited by examining operating system files. In these tests, operating system files will be examined to find any evidence of files hidden in the Vault (TPM protected space).

Procedure

All tests will be run on a computer with a TPM based encrypted partition. A series of picture files will be created inside of the Vault. The names of the files and the contents of the files will be searched for using FTK. After each test, an image of the hard drive will be taken with Forensics Tool Kit Imager Lite 2.54. The image will be saved onto an external drive. The image will then be analyzed using Forensics Tool Kit 1.8.

Windows Vista no longer uses individual thumbs.db files to store thumbnail previews. Instead, Vista employs several larger caches of thumbnail images which are used throughout the computer. The thumbcaches can be found at AppData->Local->Microsoft->Credentials->Windows->1033->Burn->Explorer.

- 1) The Vault will be set to "thumbnail view" with the pictures inside. Once the thumbnails have loaded, an image of the hard drive will be done with FTK Imager Lite. FTK 1.8 will then be used to examine the thumbcache files created by Vista.

In previous experiments, it was shown that the Windows pagefile (pagefile.sys) contained some evidence about files in the Vault.

- 2) In this experiment the computer will be checked for the password associated with the Vault (masterpass). The Vault password was created using Embassy Security Suites

by Wave Systems. As Embassy Security Suites is a third party application, there may be evidence of the password or password hash somewhere in the computer.

Requirements

- 1) Embassy Security Suites by Wave Systems
 - Machine with TPM chip

Expected Results

- 1) Since Vista caches thumbnails for quick loading, there should be evidence of the files in the Vault in Vista's thumbcaches.
- 2) Some evidence of the Vault password will be found.

Test Setup

Test Drive:

Seagate Barracuda 7200.10
Capacity: 250 GB
S/N: 6RY7YKRF
Connection: SATA

Test Computer:

Board: Dell Inc. 0GM819
Serial Number: CN1374081U0AQ8.
Bus Clock: 1333 megahertz
BIOS: Dell Inc. A09 03/11/2008
Enclosure Type: Desktop
Memory: 3326 Megabytes Installed Memory
Processor: 2.33 gigahertz Intel Core2 Duo
OS: Windows Vista Ultimate (build 6000)

Test Results

- 1) Upon analysis, it was found that Vista did cache the thumbnails of the files in encrypted space. The thumbcache.db files contained thumbnails of all the pictures in TPM encrypted space.

- 2) The string search for the Vault password (masterpass) yielded 152 matches. All of the matches were found in pagefile.sys files (Windows had 51 saved). Upon analysis of the pagefiles, most instances of “masterpass” were unable to be linked to anything specific within the file. However, in one of the earliest page files, the entire TPM setup and Vault creation process was logged in plain text. A small excerpt from the file can be found below:

...

Embassy Security Suites

...

Next

Password: masterpass

Next

...

To put these findings in perspective, the pagefile analysis occurred three months after the Vault was originally set up. Analysis of the pagefiles show that the third party programs used to manage TPM protected space may be a potential weakness. However, it should be noted that finding the password in a pagefile could still prove to be difficult. String searches on “Embassy” yielded over 700 results and “password” yielded over 1100 results respectively. No other clear indicator of a plain text password’s location was found.

Summary

Windows Vista stores caches of users’ thumbnails for quicker loading. Vista’s thumbaches.db contained thumbnails of pictures stored in TPM protected space. Plain text instances of the Vault’s password were found in Windows’ pagefiles. TPM’s reliance on third party software to do encryption is a weakness that may be able to be exploited.

12.7 Vault Tests 3 (TPM based encryption)

Test Plan

Test Name: Vault Tests 3

Test Title: Moving Files into the Vault

Purpose and Scope

In previous experiments, it was shown that files created inside the vault had no visible traces during forensic analysis. The purpose of these experiments is to test whether an old copy of a file moved into the Vault can be found in drive slack space.

Procedure

All tests will be run on a computer with a TPM based encrypted partition. A text file created outside of the vault will be moved (dragged and dropped) into the vault. After the test, an image of the hard drive will be taken with Forensics Tool Kit Imager Lite 2.54. The image will be saved onto an external drive. The image will then be analyzed using Forensics Tool Kit 1.8.

Requirements

- 1) Embassy Security Suites by Wave Systems
- 2) Machine with TPM chip

Expected Results

- 1) Since the file was present outside of the Vault, there should be some residue of the file left in slack space.

Test Setup

Test Drive:

Seagate Barracuda 7200.10
Capacity: 250 GB
S/N: 6RY7YKRF

Connection: SATA

Test Computer:

Board: Dell Inc. 0GM819

Serial Number: CN1374081U0AQ8.

Bus Clock: 1333 megahertz

BIOS: Dell Inc. A09 03/11/2008

Enclosure Type: Desktop

Memory: 3326 Megabytes Installed Memory

Processor: 2.33 gigahertz Intel Core2 Duo

OS: Windows Vista Ultimate (build 6000)

Test Results

- 1) There was no residue of the file found in slack space or temp files. After looking more closely at the Embassy Security Suites manual, it was discovered that dragging and dropping a file into the Vault shreds the original copy outside of the vault, making the file unrecoverable. In the same manner, deleting a file located inside the vault shreds the file, making it unrecoverable.

Summary

Files moved from outside of the vault to inside the vault were unable to be recovered using FTK 1.8.

12.8 BitLocker Tests

Test Plan

Test Name: BitLocker Tests

Test Title: BitLocker recovery through traditional and live analysis.

Purpose and Scope

BitLocker is an encryption method developed by Microsoft for Windows Vista Enterprise and Ultimate Edition. These tests will determine the effectiveness of traditional and live computer forensic analysis on BitLocker encryption.

Procedure

All tests will be run on a computer with BitLocker encryption. Once Vista is installed and bootable, a bit-copy (physical) of the BitLocker drive will be made onto a second drive using BackTrack version 2. BackTrack is a live Linux/GNU distribution which focus on security and can be found at <http://www.remote-exploit.org>. The two drives will then be hashed to confirm that they are, indeed, identical. After the hash check, the copied drive will be analyzed using Forensics Tool Kit 1.8. The first drive will be booted into Windows. At that point, an image of the hard drive will be taken with Forensics Tool Kit Imager Lite 2.54. The image will be saved onto an external drive. The image will then be analyzed using Forensics Tool Kit 1.8.

Requirements

- 1) Windows Ultimate or Enterprise Edition
- 2) Two identical hard drives

Expected Results

- 1) The drive protected by BitLocker will be unreadable.
- 2) Since the data on the drive is decrypted when the machine is logged in, all data on the drive should be recoverable.

Test Setup

Test Drive 1:

Seagate Barracuda 7200.10
Capacity: 250 GB
S/N: 6RY7YKRF
Connection: SATA

Test Drive 2:

Seagate Barracuda 7200.10
Capacity: 250 GB
S/N: 6RY7YKVN
Connection: SATA

Test Computer:

Board: Dell Inc. 0GM819
Serial Number: CN1374081U0AQ8.
Bus Clock: 1333 megahertz
BIOS: Dell Inc. A09 03/11/2008
Enclosure Type: Desktop
Memory: 3326 Megabytes Installed Memory
Processor: 2.33 gigahertz Intel Core2 Duo
OS: Windows Vista Ultimate (build 6000)

Test Results

- 1) The drive protected by BitLocker was unreadable.
- 2) FTK Imager Lite recovered all data on the drive. Everything in the image taken was decrypted and readable.

Summary

Traditional forensic analysis of the drive yielded no results. Live imaging of the BitLocker protected drive managed to recover all data on the drive in an unencrypted form.

12.9 Hardware Based Full Disk Encryption Tests

Test Plan

Test Name: Hardware Based Full Disk Encryption Tests

Test Title: Hardware Based Full Disk Encryption Recovery Through Traditional and Live Analysis.

Purpose and Scope

Hardware based full disk encryption is an encryption method gaining popularity. Hardware FDE uses a chip built into the drive to encrypt and decrypt files. These tests will determine the effectiveness of traditional and live computer forensic analysis on hardware FDE encryption.

Procedure

All tests will be run on a computer with hardware FDE enabled. A bit-copy (physical) of the FDE drive will be made onto a second drive using BackTrack version 2. BackTrack is a live Linux/GNU distribution which focus on security and can be found at <http://www.remote-exploit.org>. The two drives will then be hashed to confirm that they are, indeed, identical. After the hash check, the copied drive will be analyzed using Forensics Tool Kit 1.8. The first drive will be booted into Windows. At that point, an image of the hard drive will be taken with Forensics Tool Kit Imager Lite 2.54. The image will be saved onto an external drive. The image will then be analyzed using Forensics Tool Kit 1.8.

Requirements

- 1) FDE enabled hard drive

Expected Results

- 1) The drive protected by the hardware FDE will be unreadable.
- 2) Since the data on the drive is decrypted, once the machine is booted, all data on the drive should be recoverable.

Test Setup

Test Drive 1:

Seagate Momentus 5400 FDE .2
Capacity: 120 GB
S/N: 5LZ5SKZP
Connection: SATA

Test Drive 2:

Seagate Momentus 5400 FDE .2
Capacity: 120 GB
S/N: 5LZ8T91T
Connection: SATA

Test Computer:

Board: DELL - 27d80304
Serial Number: 76487-OEM-0011903-00102
Enclosure Type: Laptop
Memory: 2,038 MB Installed Memory
Processor: x86 Family 6 Model 15 Stepping 11 GenuineIntel ~2194 Mhz
OS: Microsoft Windows XP Professional 5.1.2600 Service Pack 3 Build 2600

Test Results

- 1) The drive protected by the hardware FDE was unreadable
- 2) FTK Imager Lite recovered all data on the drive. Everything in the image taken was decrypted and readable. It should be noted that during the original test, the computer with FDE went into hibernation while the image was being taken. Hibernation caused the drive to spin down and, consequently, re-encrypt itself. Once the drive was spun down, the only option was to re-enter the disk password to make the drive readable again. During the second test, the computer was kept from hibernating during imaging. Therefore, the decrypted data was recovered.

Summary

Traditional forensic analysis of the drive yielded no results. Live imaging of the hardware FDE drive managed to recover all data on the drive in an unencrypted form. Do not let the computer enter hibernation during the image recovery.

References

- [1] Challener, David, Yoder, Kent, Catherman, Ryan, Safford, David, and Leendert Van Doorn. A Practical Guide to Trusted Computing. Indianapolis: IBM Press 2008.
- [2] Mitchell, Chris. Trusted Computing. London: The Institution of Electrical Engineers 2005.
- [3] Ryan, Mark and Tien Tuan Anh Dinh. Trusted Computing: TCGProposals. Computer Security Notes. 2006. http://www.cs.bham.ac.uk/~mdr/teaching/modules/security/lectures/TrustedComputing_TCG.html
- [4] Trusted Computing Group. November 2008. <https://www.trustedcomputinggroup.org/>
- [5] “BitLocker Drive Encryption”. Microsoft. November 2008. <http://technet.microsoft.com/en-us/windows/aa905065.aspx>
- [6] Seagate. “DriveTrust Technology, a Technical Overview.” Seagate Technology LLC 2006.
- [7] TrueCrypt: Free Open-Source On-The-Fly Encryption. TrueCrypt. November 2008.
<http://www.truecrypt.org/>