



SANS Institute

Information Security Reading Room

A Guide to Encrypted Storage Incident Handling

Wylie Shanks

Copyright SANS Institute 2021. Author Retains Full Rights.

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

A Guide to Encrypted Storage Incident Handling

GIAC (GCIH) Gold Certification

Author: Wylie Shanks, wylie.shanks@gmail.com

Advisor: Carlos Cid

Accepted: April 6, 2009

Abstract

Encrypted storage solutions provide confidentiality of data; however, they have also created a need for effective incident response. Privacy legislation and regulatory compliance mandates are increasingly driving encrypted storage solution deployments. This document provides incident handlers with a few tools and processes in order to respond effectively to incidents involving encrypted storage solutions.

Table of Contents

Introduction..... 1

1 Encryption 1

 1.1 What is Encryption?..... 2

 1.2 Types or Forms of Encryption..... 2

 1.2.1 Symmetric Key Cryptography, with Block Cipher..... 3

 1.2.2 Stream Cipher..... 3

 1.2.3 Key Management 4

 1.2.4 Asymmetric Cryptography..... 4

2 Data Storage – Hard Disks 7

 2.1 Formatting and Partitioning a Hard Disk 7

 2.2 File Systems 8

 2.2.1 FAT 9

 2.2.2 NTFS 10

 2.2.3 EXT2/EXT3 10

 2.3 File Storage – Residual Data 12

3 Encrypted Storage Overview..... 12

 3.1 Encrypted Storage Technology 14

 3.1.1 File/Folder Encryption 14

 3.1.2 Full Disk / Whole Disk Encryption..... 14

 3.1.3 Virtual Disk Encryption 15

 3.1.4 Volume Encryption 15

3.2	Comparison of Encrypted Storage Solutions	16
4	Incident Handling Overview.....	16
4.1	Preparation.....	17
4.2	Identification	17
4.3	Containment	18
4.4	Eradication.....	18
4.5	Recovery.....	19
4.6	Lessons Learned.....	19
4.7	Additional Resources	19
5	Incident Response and Encrypted Storage	20
5.1	Incident Response Toolkit.....	20
5.2	Jump Bag.....	21
5.3	Accessing Encrypted Storage Solutions.....	22
5.4	Volatile and Nonvolatile Data.....	23
5.5	Collecting Volatile and Nonvolatile Data	24
5.5.1	Setting up Volatile Data Collection.....	24
5.5.2	How to Collect Volatile Data	25
5.5.3	Setting up Nonvolatile Data Collection.....	26
5.5.4	How to Collect Nonvolatile Data	26
5.6	Incident Analysis using a Disk Image.....	26
5.6.1	VMWare.....	27
6	Conclusion.....	28

References.....	29
Appendix A: Using the BartPE Utility.....	31
Select / Enable Plugins.....	33
Building BartPE Media.....	34
RAMDisk Plugin.....	35
Appendix B: PGP Plugin for BartPE.....	37
Appendix B.1: Accessing a PGP Encrypted System.....	37
Appendix C: SafeGuard Easy Plugin for BartPE.....	39
Appendix C.1: Accessing a SafeGuard Easy Encrypted System.....	40
Appendix D: SecureDoc Plugin for BartPE.....	41
Appendix D.1: Accessing a SecureDoc Encrypted System.....	42
Appendix E: TrueCrypt Plugin for BartPE.....	45
Appendix E.1: Installing TrueCrypt within a BartPE Environment.....	46
Appendix E.2: Accessing a TrueCrypt Encrypted System.....	47
Appendix F: Identifying Encrypted Storage Solutions.....	49
Appendix G: FAT16/FAT32/NTFS Cluster Size by Volume.....	51
Appendix H: FAT Volume Components.....	52
Appendix I: Metadata Files Stored in the MFT.....	53
Appendix J: NTFS Volume Components.....	54
Appendix K: Live View.....	55
Appendix L: Data Collection using Helix 3.....	58
Helix 3 - Non-Volatile Data Collection.....	58

A Guide to Encrypted Storage Incident Handling

Helix 3 - Volatile Data Collection	59
Helix 3 – Helix Forensic Command Shell	60
Helix 3 – Zero View.....	61

© SANS Institute 2009, Author retains full rights.

Introduction

Incident handling and response has become more complicated with the increased use of encrypted storage technology due, in part, to privacy legislation and regulatory compliance mandates. There are many works that have been created previously that discuss Incident Handling but very few relate to the current need to handle encrypted storage. Fortunately, there are tools and processes that aid the Incident Handler in performing their duties. In this paper, I give an overview of the Incident Handling process as it relates specifically to Encrypted Storage.

This paper is comprised of various sections relevant to Incident Handling and Encrypted Storage. Overviews of the following topics are provided in the ensuing sections of this paper:

- Encryption - Symmetric and Asymmetric encryption.
- Data Storage and File Systems - FAT, NTFS and EXT2/3 file systems.
- Encrypted Storage Technology.
- Incident Handling.
- Incident handling when confronted with Encrypted Storage.

1 Encryption

Most organizations are subject to one or more Acts that require, strongly recommend, or allude to data remaining confidential through the use of encryption. Some of the most popular Acts and regulations are show below.

Act / Regulation	Section / Requirement	Penalty for disclosing data
Payment Card Industry Data Security Standard (PCI-DSS)	Requirement 3 - Protect Cardholder Data	Potential fines imposed by PCI Standards Council, through Acquirer (Merchant's bank).
Health Insurance Portability and Accountability Act of 1996 (HIPAA)	Section 164.312(a)(2)(iv)- Encryption and Decryption	Non-compliance fines from \$100 per incident to a maximum of \$25,000 for identical requirements.
California Privacy Act	SB 1386	Can be sued, including Class

		Action, for damages related to unauthorized disclosure (data breach) of unencrypted personal information.
PIPEDA	Section 4.7.3 - Safeguards	Monetary damages can result if well founded concern taken to Federal Court. No ceiling on damage awards. Mediation and negotiation are preferred resolution methods.

1.1 What is Encryption?

A common means of securing data (ensuring data remains confidential) is through the use of encryption. The original data (plaintext) is transformed using a mathematical algorithm into ciphertext using a secret (cryptographic key) in such a way as to only allow those with the key to be able to read the original data. The encrypted data remains protected while the key is kept secret.

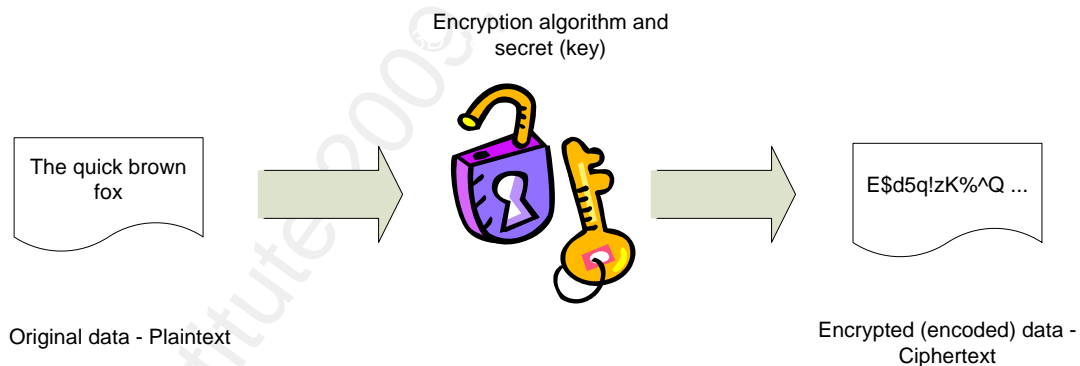


Figure 1 – Encrypting Data with a Secret (key).

Protecting the key is vital to ensuring the data remains confidential. This is what provides the security whether or not the encryption algorithm is known.

1.2 Types or Forms of Encryption

Data is encrypted either via a fixed size (block cipher) or bit-by-bit (stream cipher). How each algorithm performs the encryption function is beyond the scope of this document.

1.2.1 Symmetric Key Cryptography, with Block Cipher

Data at rest is better suited to being encrypted using a block cipher such as the Data Encryption Standard (DES) or the Advanced Encryption Standard (AES) algorithm as these algorithms are rather computationally intensive which provides comparatively better security and data is encrypted a chunk at a time rather than bit-by-bit.

1.2.1.1 Examples - DES & AES

As stated by Tipton & Henry (2007), "Symmetric algorithms operate with a single cryptographic key that is used for both encryption and decryption of the message" (p. 236). It is extremely important that the key is used, transmitted, and stored in a secure manner as the key is the only means by which the encrypted data can be unencrypted. The key is what primarily provides the security functionality. For example, when new homes are built in a community the builder uses door locks from the same manufacturer for all doors on each house. What stops a neighbor from entering another house is the key.

Advantages

Disadvantages

The encryption and decryption process is fast.	Key needs to be distributed out-of-band. (The key should be transmitted using a different method or medium than was used for the data.)
Secure method of providing confidentiality.	Key management – secure aspect of use, creation, deletion, recovery and storage of keys.
Provides some integrity and authentication for messages being stored or transmitted.	Cannot provide nonrepudiation of origin. It is impossible to know who altered a file protected with a symmetric key if two or more people share the key.
	Keys need to be randomly selected from the entire list of possible key values (keyspace). Poorly selected keys adversely affect the security of the data.

(Tipton & Henry, 2007, p. 252)

1.2.2 Stream Cipher

Stream ciphers are used to encrypt real-time data such as voice, video or data transmissions. RC4, developed by RSA Data Security, is the most widely used stream cipher deployed, for example,

in Secure Sockets Layer (SSL) and Wired Equivalent Privacy (WEP). (Tipton & Henry, 2007, p. 252) SSL is used, for example, to secure data in transit between a client and web server (i.e. Internet banking). This is implemented through the use of HTTPS (SSL). Wired Equivalent Privacy (WEP) has been used in wireless networks to provide confidentiality of data transmission. However, due to problems related to the implementation of the RC4 algorithm it does not provide the expected level of security. As a result, WEP keys can now be recovered very quickly.

1.2.3 Key Management

- Keys should be stored in a physically secure location rather than in another file. This may include using a tamper resident hardware device to store the keys.
- Hierarchical key management – Data keys can be encrypted with a master key that is derived from a passphrase (a long password) that is not stored on the system. A master key is a key-encrypting-key that would be used to encrypt the data key so that it could be stored in a file. The master key would need to be stored in a physically secure location.

(Sherwood, Clark, and Lynas, 2005, p. 333)

1.2.4 Asymmetric Cryptography

Asymmetric encryption, or Public Key Cryptography, is "...the concept of using two different keys (a key pair) to perform cryptographic operations." (Tipton & Henry, 2007, p. 253) The key pairs, a public and private key, are mathematically related. The public key is used for encryption and the private key is used for decryption.

Asymmetric key cryptography is based on one-way trapdoor functions. "... a one-way function is a mathematical function that is significantly easier to compute in one direction (the forward direction) than in the opposite direction (the inverse direction)." (Hanshe, Berti, and Hare, 2003, p. 415) Typically, it is easier to compute the forward function very quickly (for example, performing multiplication) whereas performing the inverse function (for example, factoring the product of the forward function into two large prime numbers) could take years. "A trapdoor one-way function is a one-way function for which the inverse direction is easy, given a certain piece of information (the

trapdoor), but difficult otherwise." (Hanshe, Berti, and Hare, 2003, pp. 415-416) The trapdoor, in this case, is the private key. "The forward direction is used for encryption and signature verification; the inverse direction is used for decryption and signature generation." (Hanshe, Berti, and Hare, 2003, p. 416) Thus, it is easy to encrypt data using the public key (forward function) but difficult to decrypt the data (inverse function) unless one has the private key.

The use of key pairs in asymmetric encryption allows for the secure transmission of data without agreeing on the use a pre-arranged secret key that needs to be transported out-of-band. Having to transport the secret key out-of-band was one of the deficiencies of using symmetric key encryption.

Asymmetric encryption can be used to overcome the difficulty of transmitting a symmetric key out-of-band. For example, if Gail wanted to send a symmetric key to Tim securely, Gail would use Tim's public key to encrypt the symmetric key. The encrypted symmetric key would then be sent to Tim and could only be decrypted using Tim's private key. Remember, the symmetric key in this example is simply data that is to be encrypted. Figure 2 demonstrates encryption and decryption using asymmetric keys.

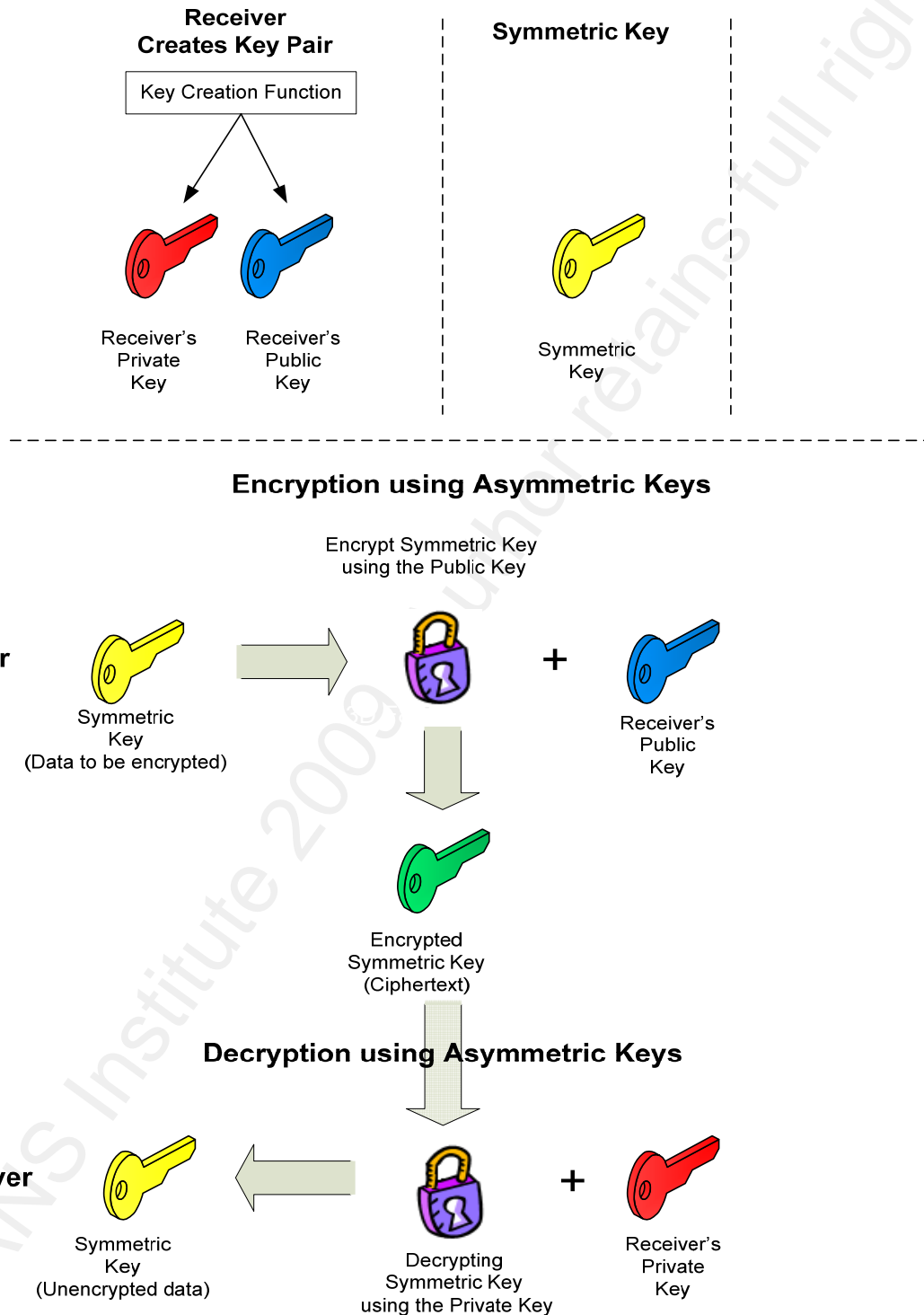


Figure 2 – Encryption and Decryption using Asymmetric Keys.

2 Data Storage – Hard Disks

A hard drive is comprised of one or more platters (disks) that spin at the same time. An arm on top of and below of each platter contains read/write heads that can move back and forth in order to access and store data. The hard disk is comprised of many tracks and sectors. A track is a concentric circle around the disk. Each track is broken into sectors which are the smallest unit of storage on a hard disk. (Carrier, 2005, p. 30)

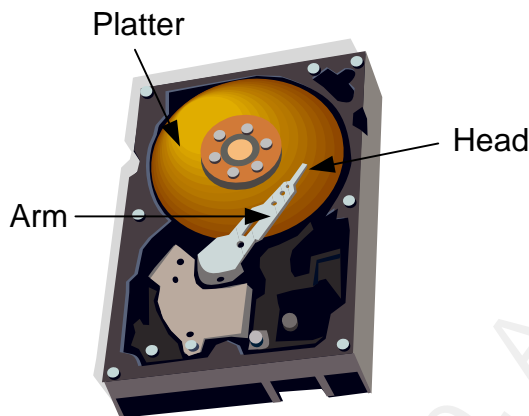


Figure 3 – An Example of a Hard Drive with Platter and Arm.

2.1 Formatting and Partitioning a Hard Disk

There are two types of hard formatting:

Low-level formatting – low-level formatting is conducted by the manufacturer.

High-level formatting – created by the file system.

The platters in the hard drive are completely blank until the manufacturer performs a low-level format of the hard drive at the factory which creates the tracks and sectors on each disk. (Carrier, 2005, p. 30) The sector size is typically 512 bytes.

During formatting, an empty file system is written to the disk which then allows for data and other files to be stored on it.

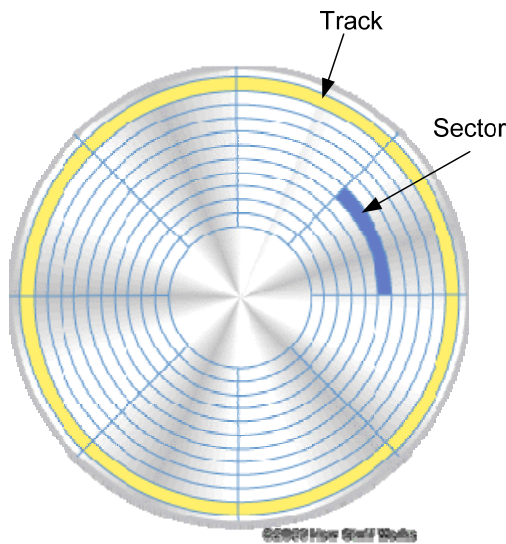


Figure 4 – Tracks, Sectors, and Clusters of a Hard Disk. (How Stuff Works, 2008).

As stated by Scarefone et al (2007):

Before media can be used to store files, the media must usually be partitioned and formatted into logical volumes. Partitioning is the act of logically dividing a media into portions that function as separate units. A logical volume is a partition or collection of partitions acting as a single entity that has been formatted with a file system. Some media types can contain only one partition (and consequently, one logical volume), while others can contain multiple partitions. (pp. 2-1-2-2)

2.2 File Systems

The file system determines how data will be accessed, stored, organized and named. Data can be stored in logical units or files which can be named (file name). Files can be further organized or grouped into directories or folders. (Scarfone et al, 2007, p. 2-1) File systems also map the hard drive's physical space to logical addresses. (Nolan et al, 2005, p. 38)

It is the file system that makes data storage and retrieval easy, relatable, and meaningful to users. Most users do not think about how their data is being stored in tracks and sectors on the disk. For Incident Handlers, however, this knowledge is important as relevant data can be found in these areas.

For instance:

Wylie Shanks

- It is important to know how and where files are stored as they could be hidden to the operating system.
- Files that are deleted are typically not removed by overwriting the sector(s) on the disk. An entry is made in the file system denoting that space is available to be used.
- Virtual memory that is saved to disk as swap space is available to be examined.

(Nolan et al, 2005, p. 38)

Three commonly used file systems are File Allocation Table (FAT), NT File System (NTFS) and EXT2/EXT3 which is used in Unix/Linux environments.

2.2.1 FAT

Microsoft's file allocation table (FAT) file system is the most widely used file system and is available, primarily, in two forms – FAT16 and FAT32. The file allocation table is a database that contains file and directory names, file MAC times (modified, accessed, and created), cluster number and other attributes (hidden, read-only etc.) (Nolan et al, 2005, pp. 40-41) This information can be useful to the Incident Handler in order to understand the order of events that occurred on the system including the times when files were accessed, created, or deleted.

2.2.1.1 FAT File Structure

The hard disk can be divided into separate logical sections called partitions. Partitions are typically labeled by the file system as C, D, E and so forth. (Nolan et al, 2005, p. 39)

Hard drive sectors are grouped together to form clusters. This is the smallest storage unit used to store files. The file system reduces overhead related to disk read/write operations as there are fewer unique storage areas available to maintain. (Nolan et al, 2005, p. 39) When the file system writes data to disk it is more efficient to write to clusters rather than sectors. These storage units, clusters and thus sectors, are progressively filled as data is stored on the disk. Cluster sizes vary depending on the file system and size of volume created. See Appendix G for further details.

On FAT volumes, the boot sector is located on the first logical sector of each partition and is

created when the partition is formatted. It contains executable code used to load the operating system into memory. (Microsoft, 2003a)

Boot Sector	Reserved Sectors	FAT 1	FAT 2 (duplicate)	Root folder	Other folders and all files
-------------	------------------	-------	-------------------	-------------	-----------------------------

Figure 5 - File Allocation Table File System Layout. (Microsoft, 2003a).

For a description of each FAT Volume Component see Appendix H.

2.2.2 NTFS

Microsoft's NT File System (NTFS) improves upon their earlier FAT design by including more comprehensive information about all files on the disk and improving the way files and directories are stored.

The Master File Table (MFT) replaces the FAT in the NTFS model. Again, the MFT and other structures are created when the partition is formatted, the boot sector is the first sector of the partition, and data is stored in clusters as is the case with FAT. See Appendix G for NTFS cluster size information.

NTFS Boot Sector	Master File Table	File System Data	Master File Table Copy
------------------	-------------------	------------------	------------------------

Figure 6 – NTFS File System Layout. (Microsoft, 2003b).

For a description of each NTFS Volume Component see Appendix J and see Appendix I for metadata information stored in the MFT.

2.2.3 EXT2/EXT3

The EXT2 and EXT3 file systems are primarily used in Linux and Unix environments. This file system is fast and reliable. There are copies of important data structures (superblocks), stored throughout the file system. The remaining sections of the file system are divided into block groups. These block groups store file names, metadata, and file content. (Carrier, 2005, p. 397) Like other file

systems, consecutive sectors are grouped together in this case to form blocks.

The superblock, which contains basic layout information including block size, number of blocks, blocks per group, the number of inodes and the number of free blocks and inodes, is a data structure stored at the beginning of the file system. (Carrier, 2005, pp. 398,400)

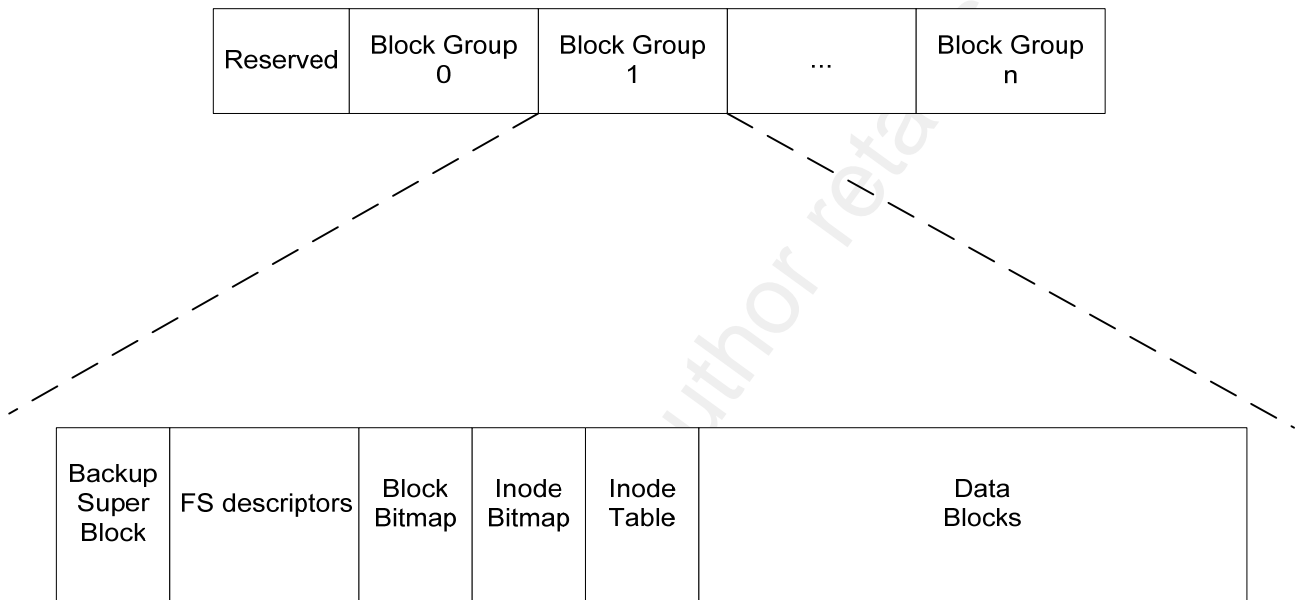


Figure 7 – EXT2/3 Structure. (Tso, 2006, p. 8).

Here is a description of the remaining block group members:

FS descriptors (Group descriptor table) - contains references to the location of the block bitmap, inode bitmap and inode table for each block group.

Block bitmap - maintains the allocation status (free or allocated) of each block in the group.

Inode bitmap – maintains the allocation status of each inode.

Inode table – a reference list of inodes.

Inodes are used to refer to files. "Each inode contains the description of the file: file type, access rights, owners, timestamps, size, pointers to data blocks. The addresses of data blocks allocated to a file are stored in its inode." (Nolan, Sullivan et al, 2005, p. 49)

Data blocks – used to directory listing, extended attributes and symbolic links.

2.3 File Storage – Residual Data

As mentioned previously, file systems store files and metadata. Data from previously deleted or temporary files (residual data) may still reside on the storage media. It may be possible to recover this data. Common forms of residual data include (Scarfone et al, 2007, p. 2-1):

- Unused File Allocation Units – file allocation units within a partition that are not being used by the file system.
- Slack space – The area remaining between the end of a file and the end of the file allocation unit. For example, if data is stored in 2KB blocks a 5KB file would use 3 blocks resulting in a 1KB area of slack space.
- Free space – The storage media area not currently allocated to a partition.

3 Encrypted Storage Overview

There are many mediums used to store digital information including USB enabled devices (thumb drives, Personal Digital Assistants (PDAs), cellular phones), memory cards, CD/DVD media, and hard drives. Many of these storage options allow one to easily transport data between devices and computer systems. While the ease with which information can be transported is, in many cases, a benefit to the end user there can be unintended consequences through the loss of information due to loss or theft.

The unauthorized access to sensitive data including personally identifiable information can result in financial loss through a tarnished name or brand which could result in a reduced stock price for a publicly traded company, lower product sales, result in regulatory fines, and potentially reduce future earnings. There can also be risk to competitive advantage if trade secrets or similarly sensitive information were lost. Thus, information must be secured not only to comply with regulatory mandates but to also reduce the financial risk associated with loss or disclosure of this information.

According to the National Institute of Standards and Technology (NIST), there are a few considerations when looking to implement encrypted storage solutions (Scarfone, Souppaya, and Sexton, 2007, p. ES-2):

1. "Organizations should use centralized management for all deployments of storage encryption except for standalone deployments and very small-scale deployments."
 - This allows for effective and efficient management of the encryption storage solution, key management and recovery, and policy verification.
2. "Organizations should ensure that all cryptographic keys used in a storage encryption solutions are secured and managed properly to support the security of the solution."
 - Key management includes key creation, deletion, recovery, use, and storage. The longer and more often a key is used the greater likelihood there is that the key can be intercepted and used by unauthorized individuals. Thus, keys need to be changed at regular intervals. This introduces a potential problem – which key was used to encrypt data? Generally, encrypted data is not decrypted and re-encrypted with a new key. Therefore, there must be a way to select which stored key is required for the decryption operation. In addition, when a key is no longer required for encryption or decryption it should be destroyed. Key management helps to simplify these and other issues.
3. "Organizations should select appropriate user authenticators for storage encryption solutions."
 - Only authorized individuals should be able to perform key management functions. Methods of authentication include password, PIN, tokens, biometrics, and smart cards. (Scarfone, Souppaya, and Sexton, 2007, p. 3-1) The number of distinct authentication methods used determines the number of factors. An example of two-factor authentication would be something you have such as a token and something you know such as a password. Three-factor authentication includes something you are (biometrics), something you have (a smart card), and something you know (a PIN). It becomes increasingly more difficult to authenticate successfully without all of the required factors. The level of difficulty when authenticating to the encrypted storage solution should be commensurate with the data being protected.

3.1 Encrypted Storage Technology

File/folder encryption, full disk encryption (FDE) / whole disk encryption (WDE), virtual disk encryption, and volume encryption are the primary options for encrypting data at rest. This section briefly describes each of these options at a high level.

3.1.1 File/Folder Encryption

Individual files or whole directories (folders) of files can be encrypted when using file/folder encryption. Files remain encrypted on the storage media but directory listings and file metadata is available. Once the user attempts to open an encrypted file the user must successfully authenticate before the file will be decrypted.

This solution requires the user to know which files and directories need to be encrypted. Thus, residual data is not likely to be encrypted.

3.1.2 Full Disk / Whole Disk Encryption

Full disk encryption (FDE) / whole disk encryption (WDE) solutions encrypt the full storage media including page or swap files, temporary files, and all other stored data. "FDE software works by redirecting a computer's master boot record (MBR), which is a reserved sector on bootable media that determines which software (e.g., OS utility) will be executed when the computer boots from the media." (Scarfone, K., Souppaya, M., & Sexton, M., 2007, p. 3-1) Prior to installing the FDE/WDE software, the MBR points to the primary operating system. Once installed, the MBR is redirected to the pre-boot environment (PBE). The user is then prompted to authenticate before booting the operating system. This process is called pre-boot authentication (PBA). (Scarfone, K., Souppaya, M., & Sexton, M., 2007, p. 3-1)

Once authenticated, the FDE/WDE software decrypts the operating system's boot sector and the operating system starts to load. The FDE/WDE software decrypts the required sectors of the hard drive as operating system files are loaded. Once the operating system has booted, the user authenticates to the operating system and operates the computer as they would normally. (Scarfone, K., Souppaya, M., & Sexton, M., 2007, p 3-1)

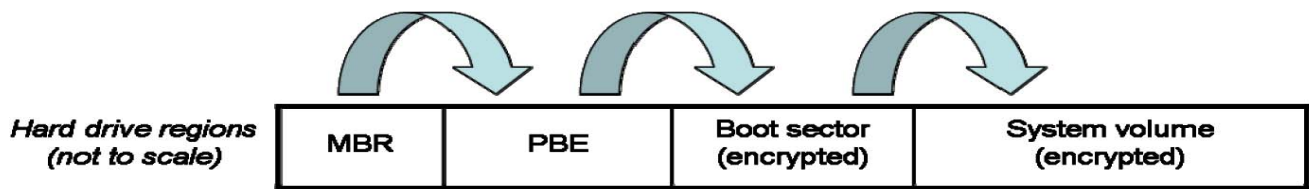


Figure 8 - Boot Sequence for FDE Software. (Scarfone, K., Souppaya, M., & Sexton, M., 2007, p 3-2)

Any residual data would be encrypted on the disk. This method provides the greatest protection for the confidentiality of data as all sectors are encrypted with the exception of the pre-boot authentication area. This area is typically not encrypted as it is required to launch the authentication software.

3.1.3 Virtual Disk Encryption

In virtual disk encryption, an encrypted file called a container is created and used to store files and directories. Access to the virtual disk is only authorized once the user has successfully authenticated. The virtual disk encryption software encrypts and decrypts sectors as required when writing to and reading from the container. Typically the container is mounted as a virtual disk. (Scarfone, K., Souppaya, M., & Sexton, M., 2007, p 3-3)

One of the benefits of virtual disk encryption is that containers are portable and can be easily backed up since containers are files. Thus they can be copied or burned to other media without affecting the ability to encrypt/decrypt data within the container.

Virtual disk encryption relies on the user to place files in the container in order to be encrypted. Temporary, swap files, deleted files and other residual data may remain within and outside of the virtual disk.

3.1.4 Volume Encryption

In volume encryption, the entire logical volume is encrypted and can only be accessed once user authentication is successful. This process is very similar to FDE/WDE, however, in this case only a volume is encrypted not the full storage media.

Residual data may still be accessible within or outside of the encrypted volume.

3.2 Comparison of Encrypted Storage Solutions

The chart below describes the various encrypted storage technology features available in several of the most common device encryption solution products:

Solution	Operating System	Full Disk Encryption	Volume/Partition Encryption	Virtual Disk Encryption	File/Folder Encryption
PGP Desktop Professional	Windows, Mac	Yes	Yes	Yes	Yes
TrueCrypt	Windows, Unix, Linux, Mac	Yes	Yes	Yes	No
SecureDoc	Windows, Linux, Mac	Yes	Yes	No	Yes
SafeGuard Easy	Windows	Yes	Yes	No	No

Figure 9 – Encrypted Storage Solutions.

4 Incident Handling Overview

An incident is an adverse event that affects an information system, for example, a denial of service. In addition, an adverse event could include system crashes, intrusions, unauthorized use of system privileges, and the malicious destruction of data. Knowing how to respond to incidents in a systematic and efficient manner helps to reduce the negative financial implications of system interruption and downtime and the loss or theft of data. (Scarfone, K., Grance, T, & Masone, K., 2008, pp. 2-1-2-2) Incident handling involves the creation of an incident response plan to deal with these

adverse events.

Incident Handling involves multiple phases including preparation, identification, containment, eradication, recovery, and lessons learned which are described below.

4.1 Preparation

Prior to an incident the tools, processes, company policies, and people should be in place to respond to any incident that may arise. Tools that may be required include forensic tools which will be discussed later in section 5 - Incident Response and Encrypted Storage. Escalation processes, handling procedures, and documentation should be created and the Incident Response Team should be trained in their use. Incident Response Teams should be comprised of members from various groups within the organization such as Human Resources, Legal, IT including network, server, and application teams, Information and physical security teams, and management. Not every incident would require the involvement of the full Incident Response Team.

Understanding the information and security systems of the business is very important. Knowing how these environments operate and what is considered normal activity helps the Incident Handler to properly assess a suspected incident when it arises and helps to ensure the proper response process is used.

A company-wide security policy aids in the protection of information assets of the organization and sets administrative or other penalties for policy violation. The policy would outline the authority and responsibility of various teams and thus gives direction and guidelines around appropriate actions teams can take during the incident. (Peikari, C, Chuvakin, A., 2004, p. 463)

4.2 Identification

Teaching other staff how to identify incidents (security awareness training) is helpful in detecting when an incident occurs and who to contact when an incident is suspected. Determining if the activity is an incident is crucial. Everything related to the incident must be carefully documented as it will be used in later stages of the incident response process. Maintaining control of this evidence, called Chain of Custody of Evidence, is important not just for cases that are tried in court but also

demonstrates that a company has a documented incident handling process in place and that it is being followed. Certain regulatory bodies require incident handling processes to be in place and organizations are audited against their own policies in this regard. Having a documented incident handling process that is being followed demonstrates due diligence and due care on the part of the organization. (Peikari, C, Chuvakin, A., 2004, p. 463)

Logs can play an important role in the identification phase. Application, firewall, system, and intrusion detection logs could contain valuable evidence including the source or the activity.

4.3 Containment

Once the incident has been confirmed, the next step is to stop the incident from spreading or causing additional harm. This can include disabling login accounts, changing passwords, stopping computer or network services, unplugging network connections, and applying filters to routers or rules to the firewall. However, it is important to limit changing the environment as much as possible to ensure that all potential evidence is collected intact. It is very important to know how commands or actions affect the system in order to know what steps can be taken. (Peikari, C, Chuvakin, A., 2004, p. 464) Communication with others on the Incident Response Team is very important. This includes advising the business unit or application owners and management what is occurring and what next steps are proposed before containment activities are initiated.

The Incident Response tool should be used at this point to backup the affected system(s). This will allow for future analysis of the system including any forensic analysis that may be required.

4.4 Eradication

In the eradication phase, the factors which caused the incident are mitigated or eliminated. This may include patching vulnerable systems, changing system configurations, restoring the system from a known good backup, or rebuilding the operating system and applications from scratch using original media then patching and configuring the system. In this phase, the incident handler must balance the business need of ensuring the system is accessible to users while being able to collect evidence, and stop the spread of damage, while working within their company's policies. (Peikari, C, Chuvakin, A.,

2004, p. 464) Additional steps may also be required to improve system defenses such as additional firewall and/or router rules, applying additional patches, further system hardening, and performing system and network vulnerability analysis, in part, through the use of a vulnerability scanner such as Nessus.

4.5 Recovery

Company operations are returned to normal in this phase having previously contained and eradicated the issue. It is recommended that increased monitoring is implemented to ensure the system is working correctly and that the issue has been properly resolved. Additional monitoring provides an increased level of protection for the affected system. (Peikari, C, Chuvakin, A., 2004, p. 465) The business or application owners should decide when to restore operations.

4.6 Lessons Learned

Learning from incidents helps to ensure they will not be repeated in the future, and thus, helps to improve the overall security posture. Notes taken during an incident should be reviewed and compared against the Incident Response Plan. An executive summary should be written for each incident and submitted to management along with proposed changes, a budget, and the impact of the recommendations. The incident response plan should be updated with the accepted changes to ensure the incident handling procedures are current. Other teams may receive a copy of this information in order to improve their processes and procedures as well as the incident handling process. (Peikari, C, Chuvakin, A., 2004, p. 465)

4.7 Additional Resources

Here are several helpful incident response resources:

- Incident response forms - <http://www.sans.org/score/incidentforms>.
- Linux Incident Response Cheat Sheet – <http://sans.org/resources/linsacheatsheet.pdf>
- Windows Incident Response Cheat Sheet – <http://sans.org/resources/winsacheatsheet.pdf>
- Security Incident Survey Cheat Sheet - <http://www.zeltser.com/network-os->

[security/security-incident-survey-cheat-sheet.pdf](http://www.zeltser.com/network-os-security/security-incident-survey-cheat-sheet.pdf)

- Security Incident Questionnaire Cheat Sheet - <http://www.zeltser.com/network-os-security/security-incident-questionnaire-cheat-sheet.pdf>

5 Incident Response and Encrypted Storage

First responders are the system and network administrators and other Incident Response team members who are assigned to handle computer security incidents. They should (Nolan, R., O'Sullivan, C, Branson, J, & Waits, C., 2005, p. 88):

- Determine the severity of the incident.
- Collect as much information about the incident as possible.
- Document all findings.
- Share this collected information to determine the root cause.

The first responder needs to be trained and knowledgeable in how to perform their duties. Information or evidence can be easily destroyed if mistakes are made. An incident response toolkit and predetermined incident response plan should be created in order to collect data. (Nolan, R., O'Sullivan, C, Branson, J, & Waits, C., 2005, p. 88)

5.1 Incident Response Toolkit

A popular, free, incident response toolkit is Helix 3 from e-fense. Recently, e-fense commercialized their product. The last freely available version of Helix 3 is Helix2008R1. A Helix 3 manual is available to members (<http://www.e-fense.com/register-overview.php>) who pay \$14.95 per month with a minimum one-year commitment.

Helix can be used to collect volatile and nonvolatile data for later analysis. The *Helix v1.7 for Beginners* manual can be downloaded from the link mentioned below. This document describes how to use the various tools included on the Helix v1.x CD. This documentation relates to Helix version 1.9 which can be also be downloaded via the link shown below.

Download Helix2008R1 (2.0) - <http://www.charlestendell.com/content/downloads>

Helix v1.7 for Beginners - <http://www.scm.uws.edu.au/units/2008.2/cfw/Materials/Helix0307.pdf>

Helix v1.9 - http://downloads.cs.txstate.edu/instructor/davis/forensics/Helix_V1.9-07-13a-2007.iso

BartPE is a program that allows one to create a bootable version of Windows XP or Windows 2003 on a CD/DVD, USB device, or ISO format. While the system created using BartPE is not forensically sound (updates to the file system can be made to the system being reviewed) it does offer several useful incident response and incident analysis features:

- Encrypted storage systems can be accessed, with proper authentication, from BartPE. Files can be viewed, copied, or modified.
- If dcfldd is stored on the BartPE bootable media an unencrypted backup or image can be made from the original encrypted image. The unencrypted image can be scanned for keywords, sectors can be viewed for residual data, and other forensic analysis can be conducted. Dcfldd can be downloaded from this location:
<http://dcfldd.sourceforge.net/#download>.

Appendix A through Appendix F reviews how to create BartPE bootable media, create and install plugins for several popular encrypted storage solutions including PGP, TrueCrypt, SecureDoc, and SafeGuard Easy, and how to identify the encrypted storage solution by viewing the initial sector of the storage media. Data acquisition using Helix 3 is described in Appendix L.

5.2 Jump Bag

A jump bag is a collection of tools and resources that may be useful when responding to an incident. A few important items to have in a jump bag are:

- Blank storage media including CD/DVD and blank USB storage devices.
- Incident response toolkit including statically linked binaries for dcfldd and forensic software that runs from a write-once media (CD/DVD).
- A laptop with at least 2GB of RAM, large hard drive (at least 200 GB), capable of

running multiple operating systems either through multi-boot, via CD/DVD, or virtualization software (i.e. VMWare).

- A variety of cables – patch cables of various lengths, cross-over cables, null modem cable, USB to serial cable, serial cable, and monitor cables (DVI, VGA etc.).
- USB to serial adapter.
- Hub, switch, and/or network tap.
- Screwdrivers (tools), flashlight, desiccants, and anti-static bags.
- Pens and blank notebooks to take notes.
- Blank incident response forms, cheat sheets, and incident response plan.
- Incident response team and other contact information.

5.3 Accessing Encrypted Storage Solutions

Whole disk encryption solutions require a means of successful authentication prior to allowing access to the system. A few authentication examples are:

- Password / passphrase
- Smart card (certificates)
- Token
- Biometric

If a system is powered on and successful authentication to the system can be achieved then standard incident response procedures can be followed in order to collect both volatile and nonvolatile data. This process will be described in several upcoming sections.

It will be very difficult to access an encrypted system without successfully authenticating. However, there are a few options that may be available to the incident handler in order to successfully access the encrypted storage solution:

- Bootable recovery media (CD/DVD/USB). Some solutions do not require authentication to use the recovery tool which may contain the decryption key.
- Enterprise Key Management System / Centralized Encryption Solution. These systems may have stored the decryption key which can be retrieved by the incident handler. Several vendors, including PGP, WinMagic and Utimaco, offer centralized a key storage solution for their product.
- Use of a centralized directory service such as Active Directory, or LDAP. If the encrypted storage solution utilizes a centralized directory service then an administrator could provide the necessary privileges to the incident handler in order to access the encrypted system.
- Other files such as the SAM, NTUSER.dat, registry, pagefile.sys, hibernation file, and crash dump may contain passwords. If a system is powered on, it may be possible to retrieve these files which may be helpful in obtaining the encrypted storage solution password.

It is not always possible to determine whether or not a system uses an encrypted storage solution by looking at the names of running processes or icons in the Windows status bar. The ZeroView tool from Technology Pathways allows one to see the contents of the first sector of the storage media. Typically, this is the location of the master boot record (MBR). Most encrypted storage solutions have an identifier located in this sector. See Appendix F - Identifying Encrypted Storage Solutions for further detail.

5.4 Volatile and Nonvolatile Data

Data stored in system memory that is lost when a machine loses its power, is rebooted, or shutdown is considered volatile data. This includes data stored in RAM, system registers, or is cached. (Nolan, R., O'Sullivan, C, Branson, J, & Waits, C., 2005, p. 89)

The incident handler should decide whether or not collecting volatile data is required for the particular type of incident. This decision should be made prior to the incident and documented in the

incident response plan in order to aid the incident handler in making the best decision at the time. This is particularly important when a system uses encryption to secure data as the computer's RAM may contain the passwords or password hash that can be used later if required. (Kent, K., Chevalier, S., Grance, T., & Dang, H., 2006, p. 5-5)

Nonvolatile or persistent data resides on storage media. This data still resides on the media when power is lost, or the machine is rebooted or shutdown.

5.5 Collecting Volatile and Nonvolatile Data

Data should be collected, when required, in the order of volatility (Nolan, R., O'Sullivan, C, Branson, J, & Waits, C., 2005, p. 91):

- Registers and cache.
- Routing table, arp cache, process table, kernel statistics, connections.
- Temporary file systems.
- Hard disk or other nonvolatile storage devices.
- Remote or off-site logging and monitoring data.
- Physical configuration and network topology.
- Archival media such as backup tapes or disk.

It is important to use known, trusted, static binary files for volatile/nonvolatile data collection as the system being investigated could have been compromised either through the replacement of, for example, important binary files or the kernel. Using known, trusted, static binaries eliminates this concern. This was discussed in section 5.1 - Incident Response Toolkit.

Remember to record all steps taken and output from any tools or commands that are executed.

5.5.1 Setting up Volatile Data Collection

One or more of the following steps may need to be followed in order to collect volatile data (Nolan, R., O'Sullivan, C, Branson, J, & Waits, C., 2005, pp. 100-101):

1. Establish a command shell using the trusted command shell from the incident response toolkit.
2. Establish a means of transmitting and storing collected information
 - a. Data can be collected and transmitted across the network, for example, using a trusted version of netcat.
 - b. Additional local storage (i.e. USB drive) can be used to store collected data.
 - c. One should not write collected data back to the system being investigated as this could overwrite important evidence.
3. Hash the collected data to ensure its integrity
 - a. Collected data should be hashed to ensure that its integrity remains intact when it is copied. Hash values can be compared to ensure the data has not been modified.

5.5.2 How to Collect Volatile Data

Here is a list of several types of volatile data and how it can be collected (Kent, K., Chevalier, S., Grance, T., & Dang, H., 2006, pp. 5-6-5-7):

- Contents of memory – dcfldd, an improved version of dd, is a third-party tools available for Linux/Unix and Windows
- Network configuration – ifconfig on Unix/Linux and ipconfig on Windows.
- Network connections – netstat command in Unix/Linux and Windows.
- Running processes – ps command in Linux/Unix and Task Manager in Windows.
- Open files – lsof in Linux/Unix. Third-party tools in Windows.
- Login sessions – w command in Linux/Unix. Third-party tools in Windows.
- Operating system time – date command in Linux/Unix and date and time commands in

Windows.

It can be important to collect volatile data in order to gain insight into the state of the system, the possible root cause of the problem, determine the timeline of the incident, and what next steps may be required including collection of persistent data. Volatile data can only be collected before the system is shut down or rebooted. (Nolan, R., O'Sullivan, C, Branson, J, & Waits, C., 2005, p. 92) These tools should be contained within the incident response toolkit.

5.5.3 Setting up Nonvolatile Data Collection

Whether or not the system should be shut down depends on the incident. A graceful shut down of the operating system can overwrite information contained in system and other logs, can change operating system data, and possibly delete temporary files. Powering off the system may corrupt files that were open and, as a result, may result in lost data. In addition, powering off the system may impact the ability to image disk volumes and partitions.

If the storage media is encrypted using full disk encryption, it may not be possible to access the data in the future unless it is possible to authenticate to the pre-boot authentication system. Thus, it may be necessary to image the system while it is powered on in order to create an unencrypted image. Otherwise, only an encrypted image can be created.

5.5.4 How to Collect Nonvolatile Data

Dcfldd is a utility that can be used to create a bit-for-bit image of storage media also known as an image. Dcfldd works in both Linux/Unix and Windows environments. The data collected using dcfldd can be hashed as it is copied and compared to the resulting image file to ensure the image was captured successfully. Adepto and dcfldd are contained within the Helix CD and can be used to create disk images.

5.6 Incident Analysis using a Disk Image

Once data has been collected it needs to be analyzed. If a disk image was created then:

- The file system can be used to locate files.
- The whole disk can be scanned for data or patterns.
- It is possible to search for root kits or malware that may be resident on the storage media.

Using virtualization software, such as VMWare Workstation, can be helpful when reviewing disk images. In addition, the Live View tool developed by the Software Engineering Institute (CERT) of Carnegie Mellon University can be used to create the necessary VMWare Workstation configuration files required to load the disk image. The disk image can be set to read-only or the virtual machine disk can be set not to write data to the image so that it remains intact.

5.6.1 VMWare

VMWare Workstation can be used to run operating systems within an operating system.

1. Download and install VMWare Workstation (<http://www.vmware.com/products/ws/>).
2. Use LiveView to create the files necessary to launch the disk image in VMWare Workstation. See Appendix K for further details.
3. Add a second hard drive to the virtual machine configuration. This hard drive should be larger than the disk image and can be used to store files recovered from the image.
4. Specify that the hard disk in the virtual machine settings is read only. This can be defined through the virtual machine settings in VMWare or when creating the virtual machine using Live View.
5. Set the boot option to CD/DVD. Specify the BartPE ISO file as the boot file.
6. Edit the virtual machine's VMX file and add the following line:

```
bios.bootDelay="5000"
```

This line delays the booting of the VMWare guest by five (5) seconds in order to be able to enter the BIOS or select the boot device.

6 Conclusion

The use of Encrypted Storage Technology has become more prevalent due to privacy legislation and regulatory compliance mandates. Responding to incidents requires Incident Handlers to be able to accurately identify that an encrypted storage solution being utilized on a system in question. In addition, they must be knowledgeable of incident handling and response processes, understand data storage technology including encrypted storage, and be familiar with the means by which to access systems using encrypted storage solutions.

The primary purpose of this paper was to bring awareness to the reader that there are tools available to the Incident Handler that, once successfully authenticated to the encrypted storage solution, allow for further incident response and analysis capabilities. For example, if a disgruntled employee deletes several important documents from an encrypted company laptop the incident handler should create an image of the laptop's hard drive, which can be later analyzed. One can appreciate the benefits of knowing how to access the encrypted system, what tools to use to acquire the image, and analyzing the image in a virtualized environment in order to recover the files.

A further benefit to increased awareness of Encrypted Storage Incident Handling is risk reduction. Improving one's incident handling policies and procedures can result in decreased risk to the organization through the improvement of incident response time and effectiveness. This can also reduce the overall impact of an incident. Some important factors to consider are the identification of encrypted storage solutions, as well as the prior acquisition of Incident Handling and analysis tools that are compatible with encrypted storage solutions. In addition, training Incident Handlers on the use of these tools contribute to risk reduction and improved incident response time. Businesses increasingly rely on the use of encrypted storage solutions to protect their data. Protecting the confidentiality of sensitive company information, while meeting privacy and regulatory compliance requirements helps to ensure that significant risks to the organization such as fines or other financial loss, intellectual property theft, as well as damage to the company's reputation can be mitigated.

References

- Carrier, B. (2005). *File System Forensic Analysis*. Crawfordsville: Addison-Wesley.
- Hansche, S., Berti, J., & Hare, C. (2004). *Official (ISC)2 Guide to the CISSP Exam*. Boca Raton: Auerbach Publications.
- How Stuff Works. (2008). *How Hard Disks Work*. Retrieved January 24, 2009 from: <http://computer.howstuffworks.com/hard-disk7.htm>
- Kent, K., Chevalier, S., Grance, T., & Dang, H. (2006). *Guide to Integrating Forensic Techniques into Incident Response*. (NIST Special Publication 800-86). Retrieved from National Institute of Standards and Technology: <http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf>
- Microsoft Corporation. (2003a, March). *How Fat Works*. Retrieved January 24, 2009 from: <http://technet.microsoft.com/en-us/library/cc776720.aspx>
- Microsoft Corporation. (2003b, March). *How NTFS Works*. Retrieved January 24, 2009 from: <http://technet.microsoft.com/en-us/library/cc781134.aspx>
- Nolan, R., Baker, M., Branson, J., Hammerstein, J., Rush, K., Waits, C., & Schweinsberg, E. (2005). *First Responders Guide to Computer Forensics: Advanced Topics*. Retrieved January 24, 2009 from: <http://www.sei.cmu.edu/pub/documents/05.reports/pdf/05hb003.pdf>
- Nolan, R., O'Sullivan, C, Branson, J, & Waits, C. (2005). *First Responders Guide to Computer Forensics*. Retrieved on January 24, 2009 from: http://www.cert.org/archive/pdf/FRGCF_v1.3.pdf
- Peikari, C, Chuvakin, A. (2004). *Security Warrior*. Sebastopol: O'Reilly Media.
- PGP Corporation. (2008, July). *Customizing the Windows Preinstallation Environment for PGP Whole Disk Encryption*. Retrieved November 2, 2008 from http://supportimg.pgp.com/pgp_pe/Tech_Note_Customizing_%20the_PE_for_PGP_Whole_Disk_Encryption_july_08.pdf
- Scarfone, K., Grance, T, & Masone, K. (2008). *Computer Security Incident Handling Guide*. (NIST Special Publication 800-61, Revision 1). Retrieved from National Institute of Standards and Technology: <http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf>
- Scarfone, K., Souppaya, M., & Sexton, M. (2007). *Guide to Storage Encryption Technologies for End*

User Devices. (NIST Special Publication 800-111). Retrieved from National Institute of Standards and Technology: <http://csrc.nist.gov/publications/nistpubs/800-111/SP800-111.pdf>

Sherwood, J., Clark, A., & Lynas, D. (2005). *Enterprise Security Architecture: A Business-Driven Approach*. San Francisco: CMP Books

Tipton, H.F. (Ed.), Henry, K. (Ed.). (2007). *Official (ISC)2 Guide to the CISSP CBK*. Boca Raton: Auerbach Publications.

Tso, T. (2006). *The Linux ext2/3/4 Filesystem: Past, Present, and Future*. Retrieved January 31, 2009 from http://jp.linuxfoundation.org/jp_uploads/seminar20060911/Ted_Tso.pdf

WinMagic Inc. (2007). *Recovering with BartPE*. [Software] Retrieved from http://www.disk-encryption.com/tech_support/downloads/BartPE_SP3.zip

Appendix A: Using the BartPE Utility.

A bootable live Windows CD/DVD can be created by using the BartPE utility. According to <http://www.nu2.nu/pebuilder/> the following are the licensing and build requirements of BartPE:

1. A properly licensed copy of the operating system.
2. The Windows Installation CD-ROM (Windows XP Home or Windows XP Professional SP1 or higher, Windows Server 2003 Web, Standard, or Enterprise Edition).
3. BartPE (PE Builder) run from a Windows 2000/XP/2003/BartPE system.
4. A CD/DVD burner if booting from CD/DVD media if required.

Creating the recovery media:

1. Download PE Builder from <http://www.nu2.nu/pebuilder/>.
2. Install PE Builder to a directory you choose (i.e. C:\pebuilder3110a).
3. Ensure all required plugins are installed and enabled. See below for specific encryption solution plugins.
4. Run PE Builder and respond to whether or not you accept the license agreement.
5. Specify the source path of the Windows installation files.
6. Specify the output directory for BartPE.
7. Specify media output:
 - a. None – This option populates the output directory (i.e. C:\pebuilder3110a\BartPe).
 - b. Create ISO image. This option populates the output directory and creates an ISO image file but does not write the contents to CD/DVD.
 - c. Burn to CD/DVD – This option populates the output directory and writes the data to CD/DVD. One can specify the CD/DVD write application, output device, auto erase

R/W media, and eject after burn settings.

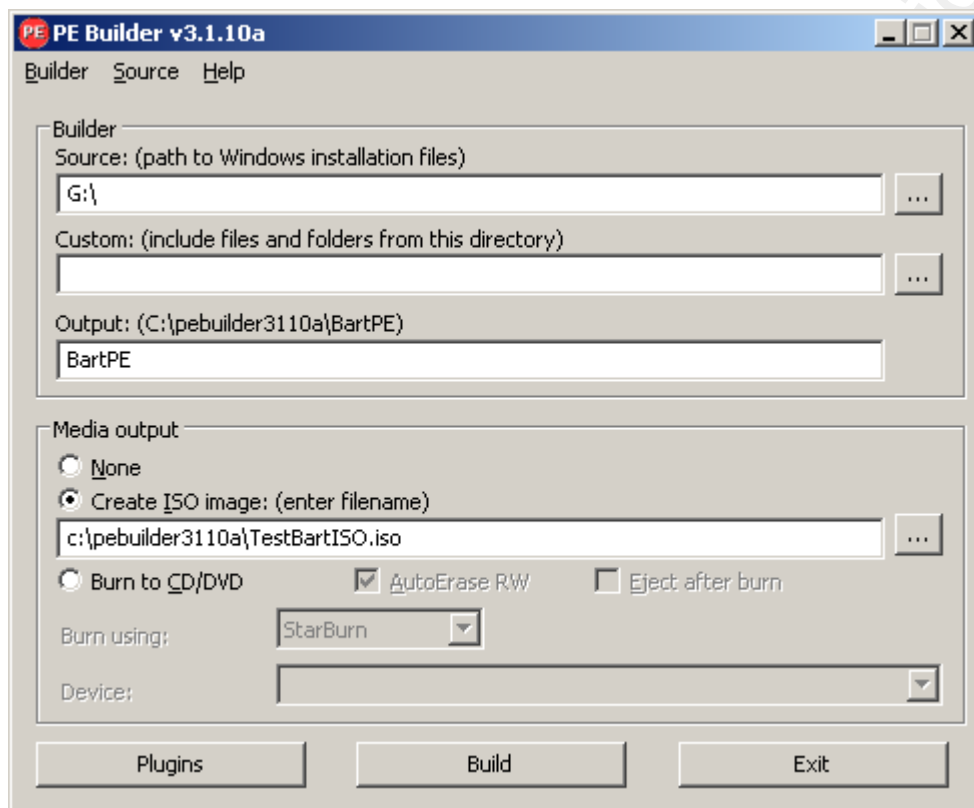


Figure 10 – PE Builder.

Clicking the "Plugins" button allows one to enable/disable plugins.

Select / Enable Plugins

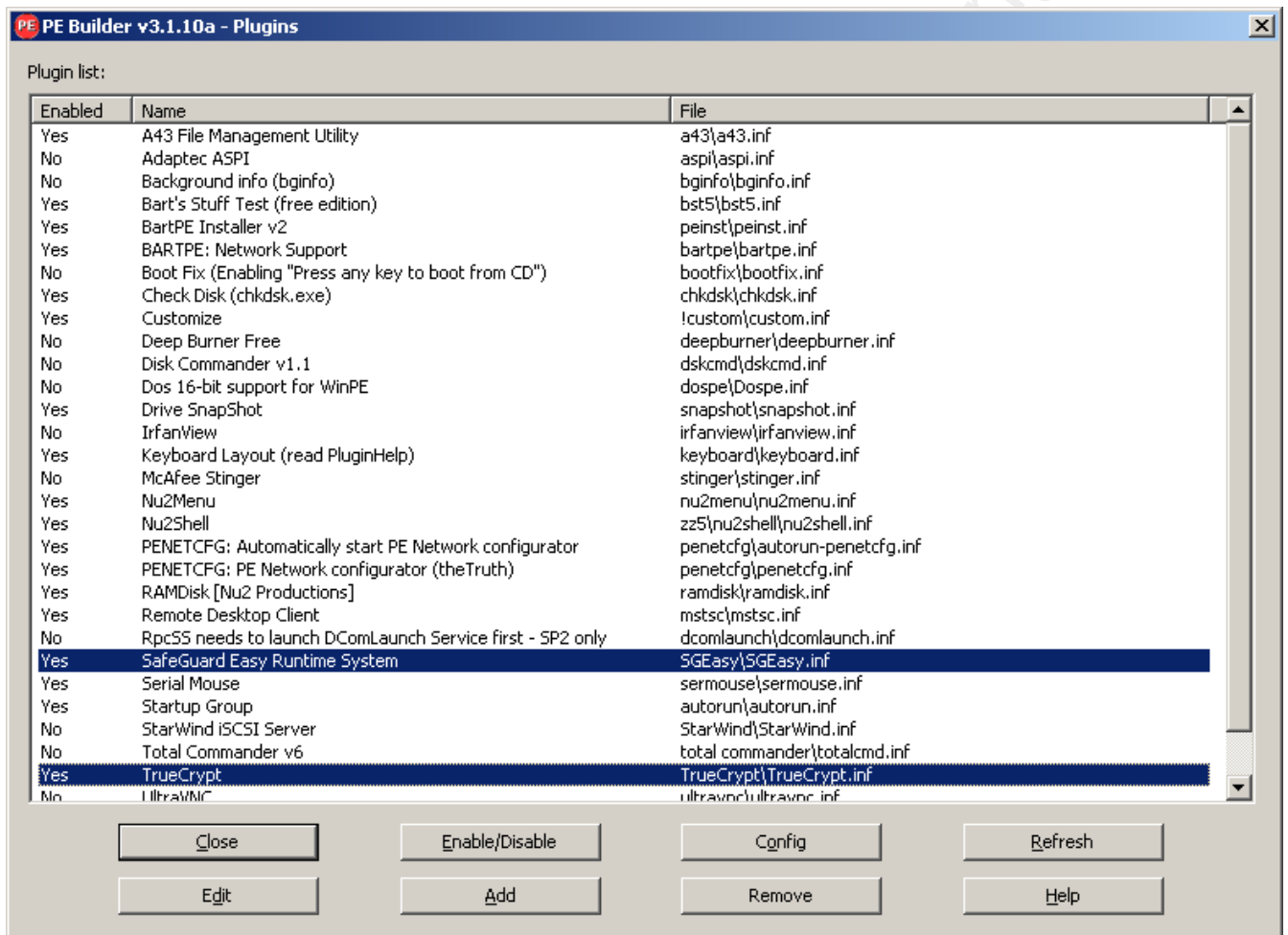


Figure 11 – PE Builder Plugins Screen.

The following are some of the options available from the plugins screen:

- An individual plugin can be selected by clicking on it. To select multiple plugins, hold down the CTRL button and click on each plugin.
- Plugins can be modified, once selected, by clicking the edit button. The file is then opened in notepad.
- Clicking "Close" closes this window and returns to the main PE Builder screen.

Building BartPE Media

Once the various options are specified the BartPE media can be built by clicking on the "Build" button. If the BartPE directory does not exist the user will be prompted to create it.

Clicking "Yes" begins the build process. If any errors are discovered during the process the user will be notified of them in order to correct them. Using the "<<" and ">>" buttons one can scroll through the error(s) and warning(s) discovered.

RAMDisk Plugin

It is advisable to increase the amount of memory available to the RAM disk as a few of the plugins require software to be installed. From the plugins window, select "RAMDisk" then click "Edit".

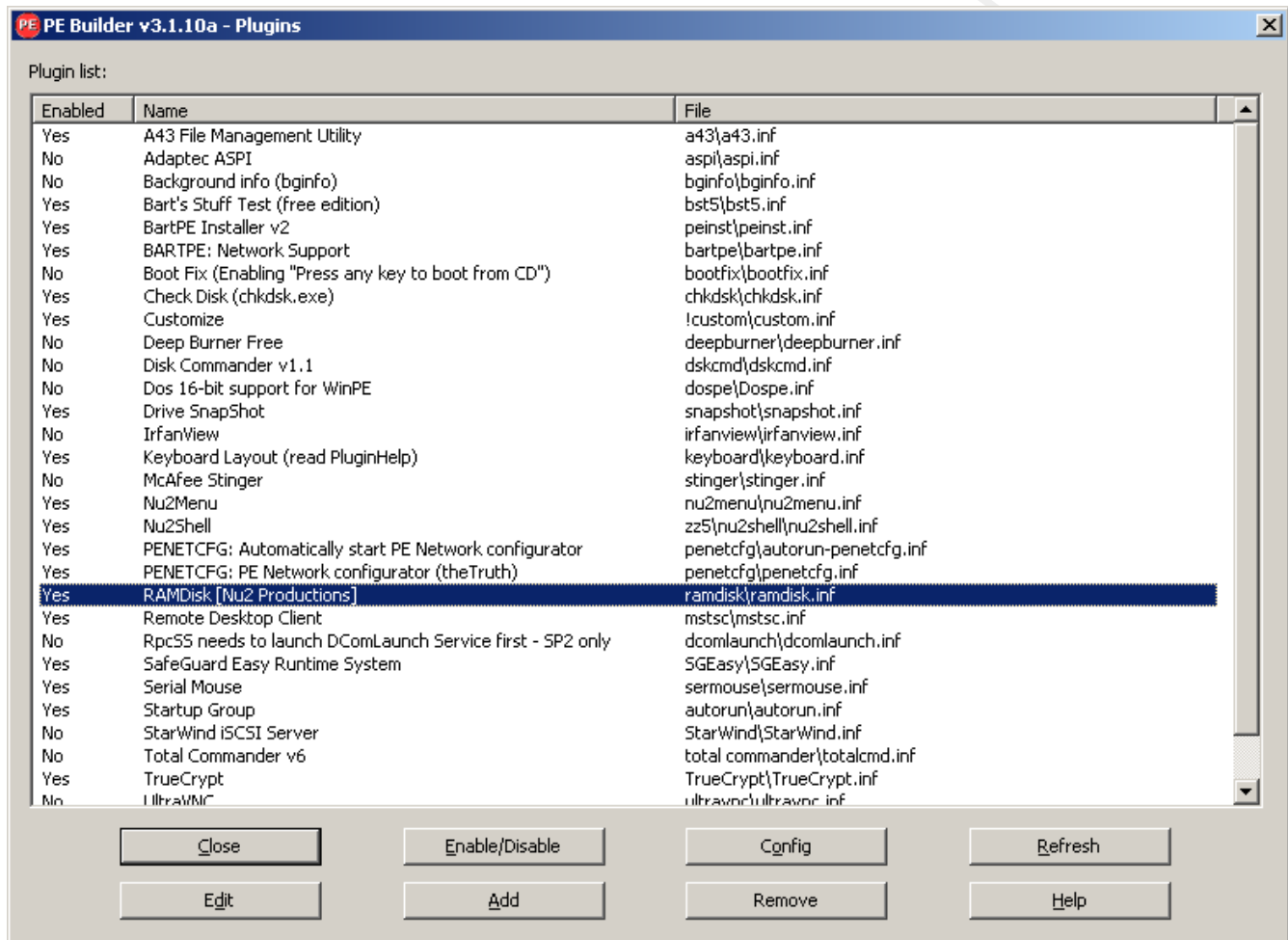
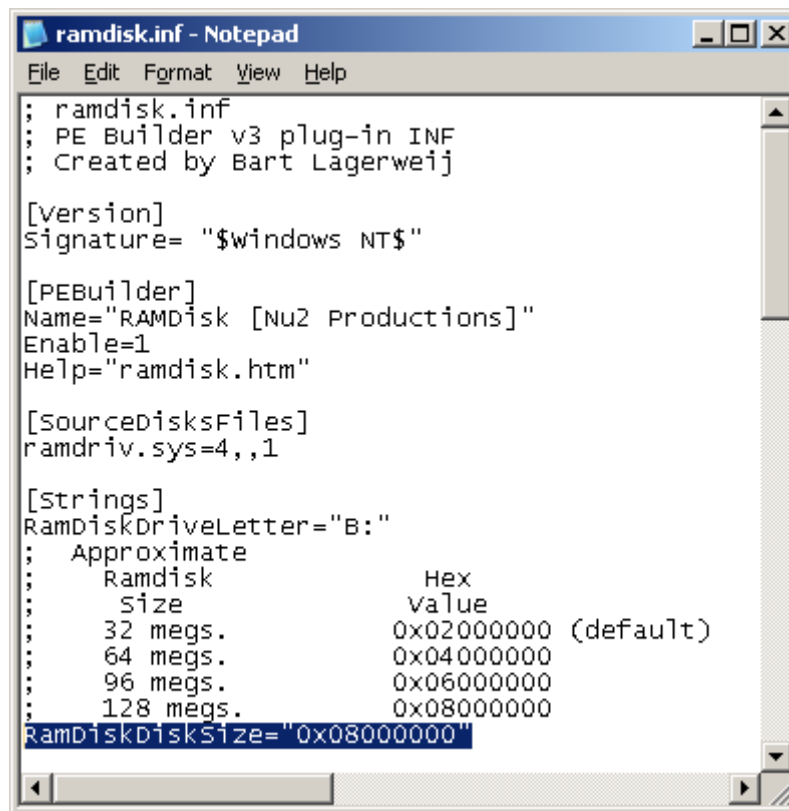


Figure 12 – RAMDisk Plugin.

Increasing the amount of memory available for the RAM disk is contingent on the system that will be booting the BartPE media. Thus, the system needs to have sufficient memory available in order to create the specified size of RAM disk.



```
ramdisk.inf - Notepad
File Edit Format View Help
; ramdisk.inf
; PE Builder v3 plug-in INF
; Created by Bart Lagerweij

[Version]
Signature= "$windows NT$"

[PEBuilder]
Name="RAMDisk [Nu2 Productions]"
Enable=1
Help="ramdisk.htm"

[SourceDisksFiles]
ramdriv.sys=4,,1

[Strings]
RamDiskDriveLetter="B:"
; Approximate
;   Ramdisk           Hex
;   Size             Value
;   32 megs.         0x02000000 (default)
;   64 megs.         0x04000000
;   96 megs.         0x06000000
;   128 megs.        0x08000000
RamDiskDiskSize="0x08000000"
```

Figure 13 – RAMDisk Configuration File.

The highlighted line in the image above shows the line and value to modify in order to use 128 MB of memory for the RAM disk. When the change has been made, click "File" and then "Save" to save the changes. The file can be closed by clicking "File" then "Exit".

Appendix B: PGP Plugin for BartPE

The PGP Plugin for BartPE can be downloaded from this location:

https://pgp.custhelp.com/cgi-bin/pgp.cfg/php/enduser/std_adp.php?p_faqid=807&p_topview=1. The plugin contains two files which should be stored in a temporary directory (i.e. D:\PGPtemp) (PGP Corporation, 2008):

- pgpstart.exe
 - pgppe.exe
1. The BartPE folder needs to have been created in order for this plugin to work correctly (i.e. C:\pebuilder3110a\BartPE).
 2. The following files need to be copied from a system that has PGP Desktop Professional installed to a temporary directory (i.e. D:\PGPtemp) (PGP Corporation, 2008):
 - C:\Program Files\PGP Corporation\PGP Desktop\pgpbootb.bin
 - C:\Program Files\PGP Corporation\PGP Desktop\pgpbootg.bin
 - %SYSTEMROOT%\system32\PGPsdk.dll
 - %SYSTEMROOT%\system32\pgpsdknl.dll
 - %SYSTEMROOT%\system32\PGPwd.dll
 - C:\Program Files\PGP Corporation\PGP Desktop\PGPwde.exe
 - %SYSTEMROOT%\system32\drivers\PGPwded.sys
 - C:\Program Files\PGP Corporation\PGP Desktop\Stage1
1. Run the following command from a command line:


```
D:\PGPtemp\pgppe.exe /winpe C:\pebuilder3110a\BartPE D:\PGPtemp
```
 2. A bootable ISO needs to be created of the C:\pebuilder3110a\BartPE directory which can be burned to CD/DVD later if required. The following command, run from a command line, can be used to create the bootable disk:


```
C:\pebuilder3110a\mkisofs.exe -iso-level 4 -volid "BartPE" -b bootsect.bin -no-emul-boot -boot-load-size 4 -hide bootsect.bin -hide boot.catalog -o "C:\pebuilder3110a\BartPE_PGP.ISO" "C:\pebuilder3110a\BartPE"
```

Appendix B.1: Accessing a PGP Encrypted System

There are three primary methods of accessing a system encrypted using PGP encryption.

Wylie Shanks

37

1. Boot normally.
 - a. Authenticate to PGP Whole Disk Encryption.
2. Boot BartPE.
 - a. Open a CMD prompt
 - b. Type: `pgpwde --disk <#> --auth --passphrase <passphrase>` where `<#>` is the disk number starting at zero (0) and `<passphrase>` is the password that is used to allow access to the encrypted disk.
 - c. If step b was successful, an "Authenticate disk completed" message should appear.
3. Boot PGP Recovery Disk.
 - a. Authenticate to PGP WDE.
 - b. Press "D" to decrypt the disk or any other key to boot normally.

PGP Whole Disk Encryption Recovery media can be downloaded from this location:

https://pgp.custhelp.com/cgi-bin/pgp.cfg/php/enduser/std_adp.php?p_faqid=471

Appendix C: SafeGuard Easy Plugin for BartPE

The SafeGuard Easy plugin is located in the Tools\SGE-BART-PE\SGEasy subdirectory of the SafeGuard Easy install point.

1. Copy the SGEasy directory and the related files and subdirectories to C:\pebuilder3110a\plugin directory.
2. Build a BartPE CD/DVD or ISO file.

Appendix C.1: Accessing a SafeGuard Easy Encrypted System

There are three methods of accessing a SafeGuard Easy Encrypted System:

1. Ensure the boot order has CD/DVD as the primary boot method.
 - a. Enter the BIOS when first booting to set the temporary boot order to hard disk.
 - b. When the Pre-Boot Authentication (PBA) screen appears enter the user ID, press tab, and enter the password.
 - c. Press F7 to boot the CD.
 - d. The encrypted hard disk contents should be accessible.

2. Ensure the hard disk is selected as the primary boot method.
 - a. Place the BartPE disk in the CD/DVD drive when the Pre-Boot Authentication screen appears.
 - b. Enter the user ID, press tab, and enter the password.
 - c. Press F7 to boot the CD.
 - d. The encrypted hard disk contents should be accessible.

3. Boot normally.

Be aware that any commands run against this device may alter the system including metadata and thus modify evidence.

Appendix D: SecureDoc Plugin for BartPE

Follow these steps to create the SecureDoc plugin for BartPE (WinMagic Inc., 2007):

1. The SecureDoc BartPE plugin can be downloaded from http://www.disk-encryption.com/tech_support/downloads/BartPE_SP3.zip.
2. Create the SecureDoc plugin directory (i.e. C:\pebuilder3110a\plugin\SecureDoc).
3. Extract the contents of the BartPE_SP3.zip file to the SecureDoc plugin directory created in step 2.
4. Build a BartPE CD/DVD or ISO file.

Appendix D.1: Accessing a SecureDoc Encrypted System

There are three primary methods of accessing a system encrypted using SecureDoc.

1. Boot the system normally.
2. Boot the system normally but boot from BartPE media.
 - a. Enter the key file (or press Enter for the default key file).
 - b. Enter the correct password.

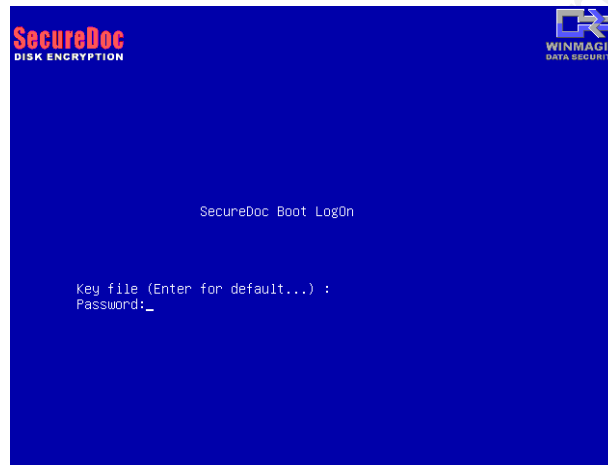


Figure 14 – SecureDoc Boot Logon screen.

- c. Press F8.
- d. Select boot from and change it to the boot media (i.e. CD/DVD).

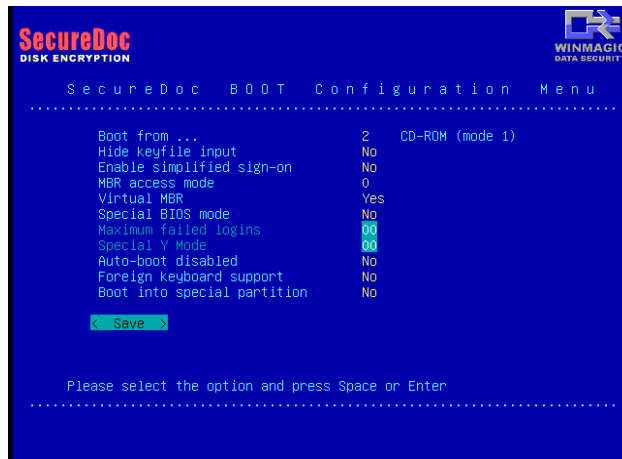


Figure 15 – SecureDoc Boot Configuration Menu.

- e. Select "Save" and press Enter.
- f. Ensure BartPE boot media is available and press any key to boot.



Figure 16 – SecureDoc Insert Bootable CD-Rom Screen.

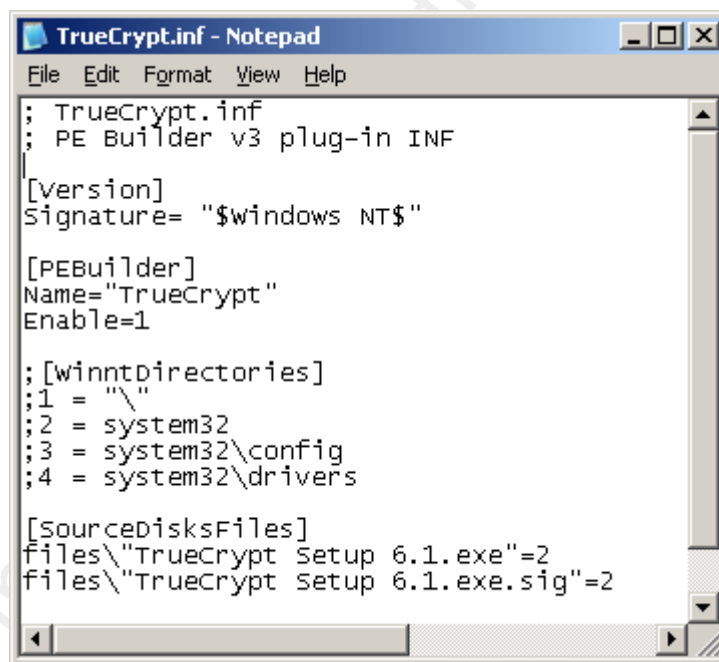
- g. The BartPE media will boot. The encrypted drive is now accessible.
3. Boot the BartPE CD (WinMagic, 2007).
 - a. Once the BartPE CD/DVD has booted, select "Programs".
 - b. Select "SecureDoc Control Center".
 - c. Select the "Boot logon key file drive".

- d. Enter the "File ID" and "UserID".
- e. Enter the "Logon password".
- f. Click the "Logon" button.
- g. The encrypted drive is now accessible.

Appendix E: TrueCrypt Plugin for BartPE

The TrueCrypt application, not a plugin, is required in order to access a system or file encrypted by TrueCrypt. It is best to include a copy of the TrueCrypt application on the BartPE CD/DVD from a known good source. TrueCrypt can be downloaded from: <http://www.truecrypt.org/downloads.php>. The file should be saved to a TrueCrypt\files subdirectory in the BartPE plugins directory (i.e. C:\pebuilder3110a\plugins\TrueCrypt\files).

The following plugin should be stored as C:\pebuilder3110a\plugins\TrueCrypt\TrueCrypt.inf. It should be created in order to copy the TrueCrypt binary into the BartPE directory before an ISO file is created or BartPE is written to CD/DVD.



```
TrueCrypt.inf - Notepad
File Edit Format View Help
; TrueCrypt.inf
; PE Builder v3 plug-in INF

[Version]
Signature= "$windows NT$"

[PEBuilder]
Name="TrueCrypt"
Enable=1

;[winntDirectories]
;1 = ""
;2 = system32
;3 = system32\config
;4 = system32\drivers

[sourceDisksFiles]
files\TrueCrypt Setup 6.1.exe=2
files\TrueCrypt Setup 6.1.exe.sig=2
```

Figure 17 – TrueCrypt Plugin Configuration File.

Appendix E.1: Installing TrueCrypt within a BartPE Environment

To launch the TrueCrypt application:

1. Boot BartPE.
2. Open a command (CMD) prompt.
3. Type, "TrueCrypt Setup 6.1.exe" and press Enter.
4. Select, "I accept and agree to be bound by the terms of the license terms".
5. Click "Accept".
6. Select "Extract" and click "Next".
7. Extract the files to "B:\TrueCrypt\" and click "Extract".
8. All files should be successfully extracted.
9. Click "OK" and "Finish" to close the program.
10. Run TrueCrypt by typing "B:\TrueCrypt\TrueCrypt.exe".
11. The TrueCrypt application should appear:

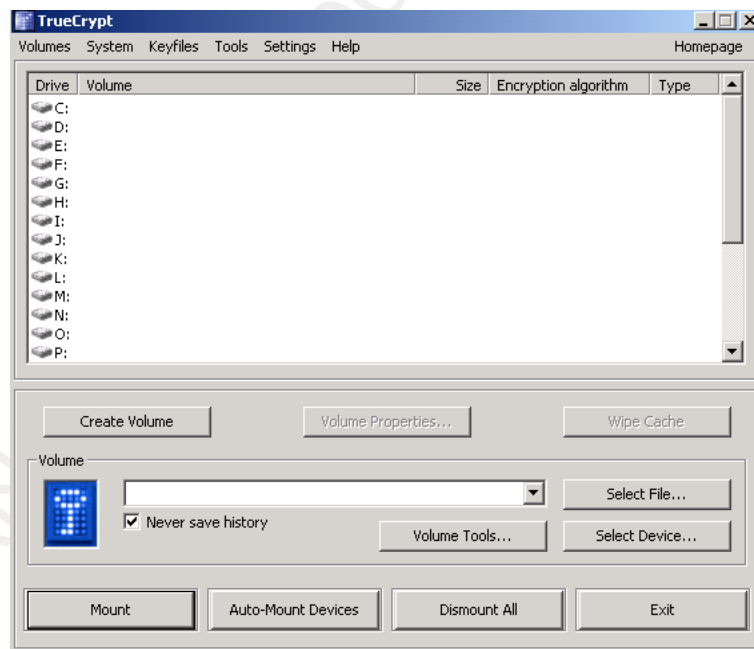


Figure 18 – Main TrueCrypt Screen.

Appendix E.2: Accessing a TrueCrypt Encrypted System

To access a whole-disk encrypted device via TrueCrypt:

1. Click on an available drive letter (i.e. C through Z).
2. Click "Select Device..." and choose the appropriate partition. For example:

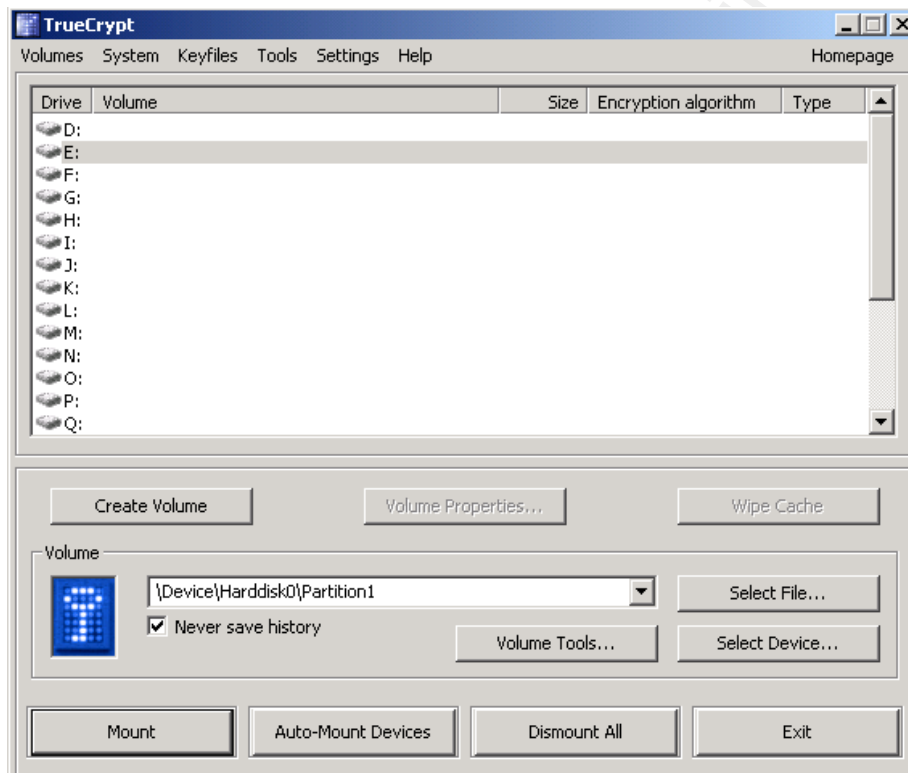


Figure 19 – Main TrueCrypt Screen with Volume Information.

3. Click "Mount" to mount the selected device.
4. Enter the password for the selected device.
5. Mount options can be specified if required as seen below, otherwise click "OK" to continue.

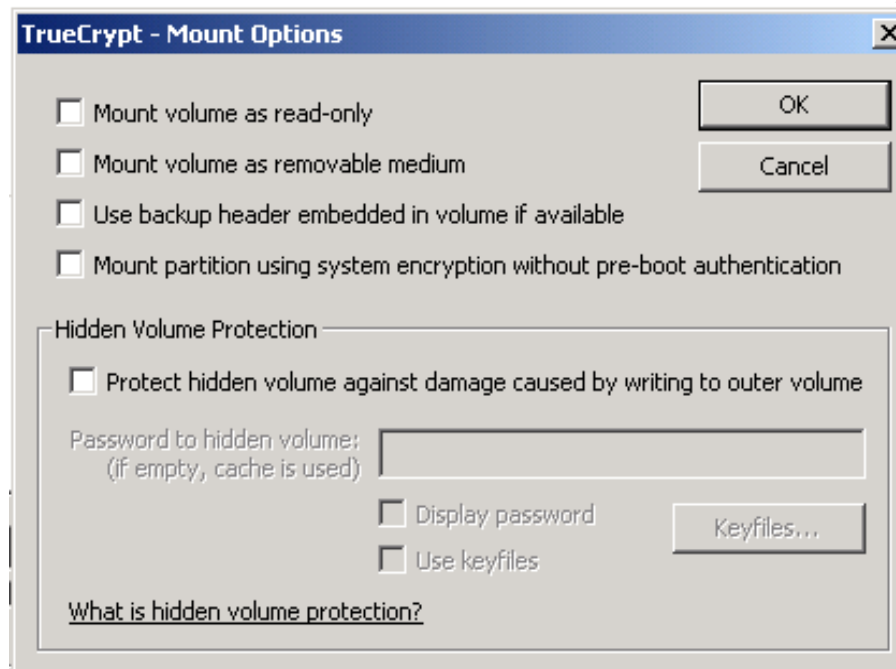


Figure 20 – TrueCrypt Mount Options Screen.

6. It may be necessary to select "Mount partition using system encryption without pre-boot authentication" if the pre-boot authentication screen was bypassed by directly booting the BartPE media.
7. The device should now be available. Be aware that any commands run against this device may alter the system and thus modify evidence.

Appendix F: Identifying Encrypted Storage Solutions

The use of ZeroView, a free tool from TechPathways, can aid the incident handler in determining if whole disk encryption is being used on a particular system. It is available from this location: <http://toorcon.techpathways.com/uploads/zeroview.zip>

Product Name	Sector 0 Offset	Product Identifier
PGP Whole Disk Encryption	0x03	PGPGUARD
WinMagic SecureDoc	0xF6	WMSD
Utimaco SafeGuard Easy	0x90	SGE400
TrueCrypt	0x06	TrueCrypt Boot Loader

ZeroView screenshots are shown below for each encryption solution:

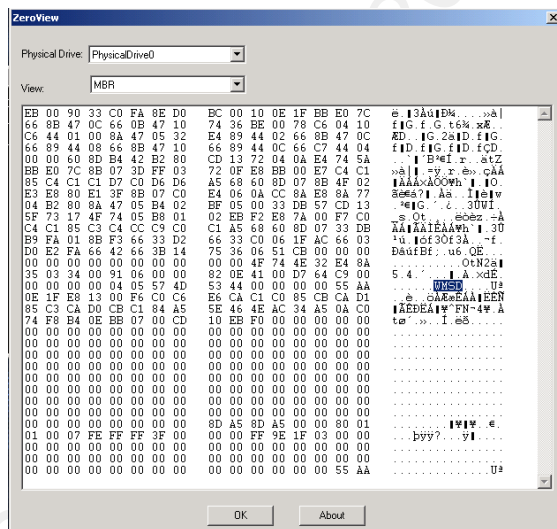


Figure 21 – SecureDoc MBR before PBA.

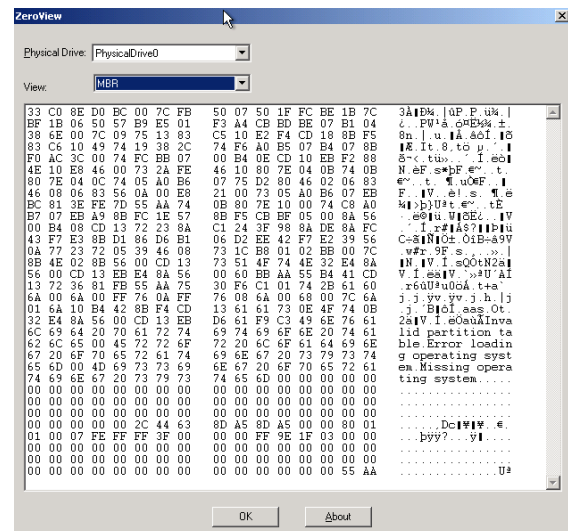


Figure 22 – SecureDoc MBR after PBA.

A Guide to Encrypted Storage Incident Handling

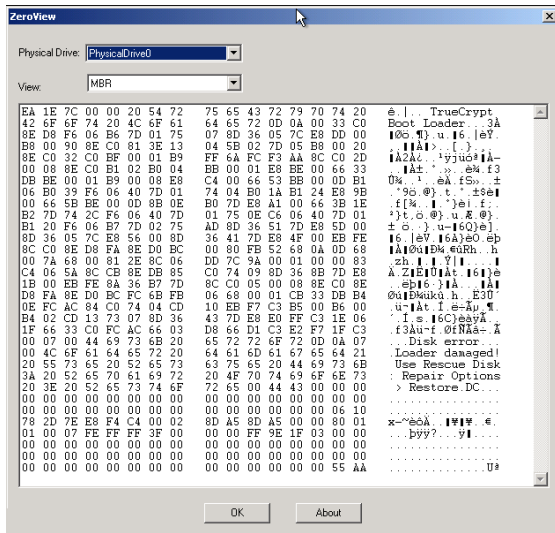


Figure 23 – TrueCrypt MBR.

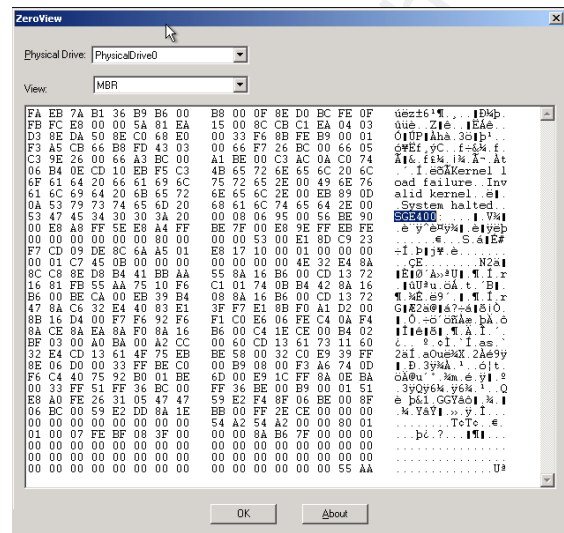


Figure 24 – SafeGuard Easy MBR.

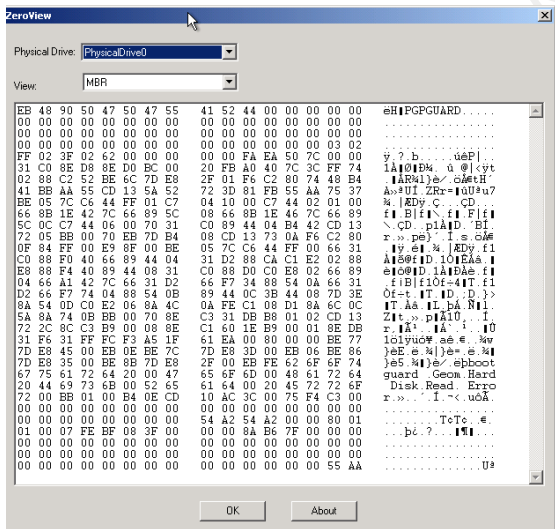


Figure 25 - PGP MBR.

Appendix G: FAT16/FAT32/NTFS Cluster Size by Volume

Volume Size	FAT16 Cluster Size	FAT32 Cluster Size	NTFS Cluster Size
7 MB – 16 MB	2 KB	Not supported	512 bytes
17 MB – 32 MB	512 bytes	Not supported	512 bytes
33 MB – 64 MB	1 KB	512 bytes	512 bytes
65 MB – 128 MB	2 KB	1 KB	512 bytes
129 MB – 256 MB	4 KB	2 KB	512 bytes
257 MB – 512 MB	8 KB	4 KB	512 bytes
513 MB – 1,024 MB	16 KB	4 KB	1 KB
1,025 MB – 2 GB	32 KB	4 KB	2 KB
2 GB – 4 GB	64 KB	4 KB	4 KB
4 GB – 8 GB	Not supported	4 KB	4 KB
8 GB – 16 GB	Not supported	8 KB	4 KB
16 GB – 32 GB	Not supported	16 KB	4 KB
32 GB – 2 TB	Not supported	Not supported	4 KB

Default Cluster Sizes for Volumes with Windows XP Professional File Systems (Nolan et al, 2005, p. 40).

Appendix H: FAT Volume Components

Component	Description
Boot Sector	...stores information about the layout of the volume and the file system structures, as well as the boot code that loads [the operating system].
Reserved Sectors	The number of sectors that precede the start of the first FAT, including the boot sector.
FAT 1	Original FAT.
FAT 2 (duplicate)	Backup copy of the FAT
Root folder	Describes the files and folders in the root of the partition.
Other folders and all files	Contains the data for the files and folders within the file system.

(Microsoft, 2003a).

Appendix I: Metadata Files Stored in the MFT

Master file table	\$Mft	0	Contains one base file record for each file and folder on an NTFS volume. If the allocation information for a file or folder is too large to fit within a single record, other file records are allocated as well.
Master file table mirror	\$MftMirr	1	Guarantees access to the MFT in case of a single-sector failure. It is a duplicate image of the first four records of the MFT.
Log file	\$LogFile	2	Contains information used by NTFS for faster recoverability. The log file is used by Windows Server 2003 to restore metadata consistency to NTFS after a system failure. The size of the log file depends on the size of the volume, but you can increase the size of the log file by using the Chkdsk command.
Volume	\$Volume	3	Contains information about the volume, such as the volume label and the volume version.
Attribute definitions	\$AttrDef	4	Lists attribute names, numbers, and descriptions.
Root file name index	.	5	The root folder.
Cluster bitmap	\$Bitmap	6	Represents the volume by showing free and unused clusters.
Boot sector	\$Boot	7	Includes the BPB used to mount the volume and additional bootstrap loader code used if the volume is bootable.
Bad cluster file	\$BadClus	8	Contains bad clusters for a volume.
Security file	\$Secure	9	Contains unique security descriptors for all files within a volume.
Uppcase table	\$Uppcase	10	Converts lowercase characters to matching Unicode uppercase characters.
NTFS extension file	\$Extend	11	Used for various optional extensions such as quotas, reparse point data, and object identifiers.
		12–15	Reserved for future use.

(Microsoft, 2003b).

Appendix J: NTFS Volume Components

Component	Description
NTFS Boot Sector	...stores information about the layout of the volume and the file system structures, as well as the boot code that loads [the operating system].
Master File Table	Contains the necessary information to retrieve files from the NTFS partition, such as the attributes of a file.
File System Data	Stores data that is not contained within the Master File Table.
Master File Table Copy	Includes copies of the records essential for the recovery of the file system if there is a problem with the original copy.

(Microsoft, 2003b).

Appendix K: Live View

Live View was created by the Software Engineering Institute (CERT) of Carnegie Mellon University. It is a java based tool that creates VMWare virtual machines from raw disk images (dcfldd) or physical disks. See Appendix L for further information regarding image acquisition.

Live View can be downloaded from <http://liveview.sourceforge.net/> and requires the VMWare Virtual Disk software in order to run. Virtual Disk can be downloaded from this location: <http://www.vmware.com/download/sdk/virtualdisk.html>.

This screen appears when Live View is launched:

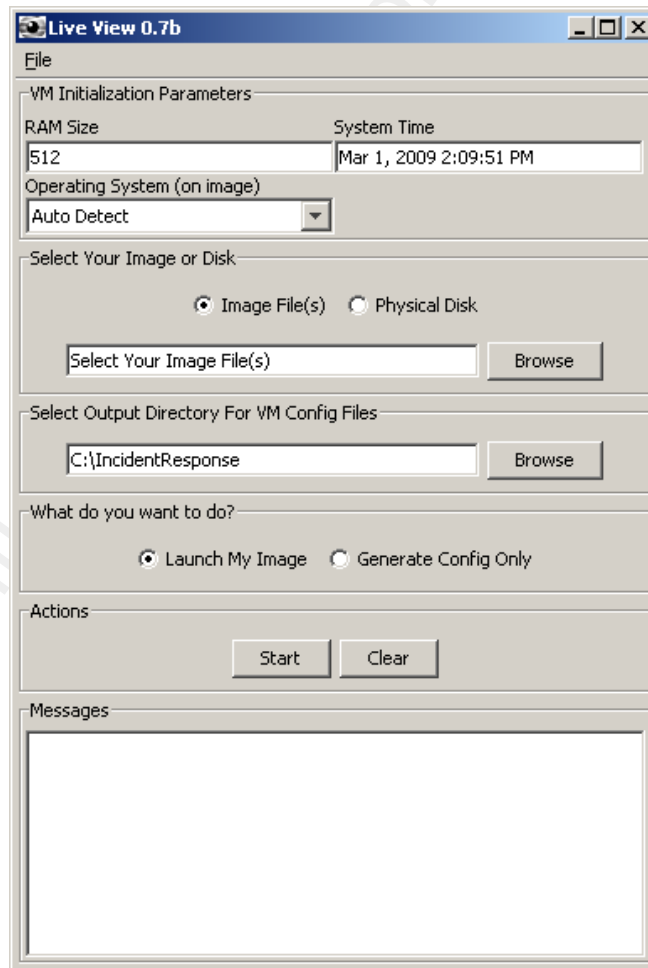


Figure 26 – Main Live View 0.7b Screen.

Configure the virtual machine by selecting:

- The amount of RAM.
- The operating system (on the image).
- Whether the image is an image file or physical disk. Image files can be of the .img, .dd, .raw or split format (i.e. .001, .002 etc.).
- The output directory for the VMWare guest configuration files.
- Whether you want to launch the image or generate the configuration files only.

When the Start button is clicked you may be prompted to make the image file read-only. The following message may appear:

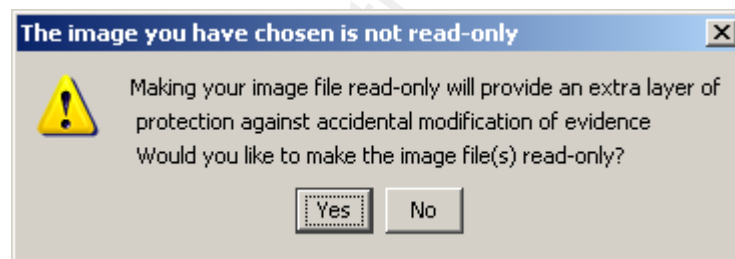


Figure 27 – Setting Image to Read Only.

Once a selection has been made and button clicked Live View begins the virtual machine creation process. This screen is shown on the next page. Once complete, the virtual machine is ready for use.

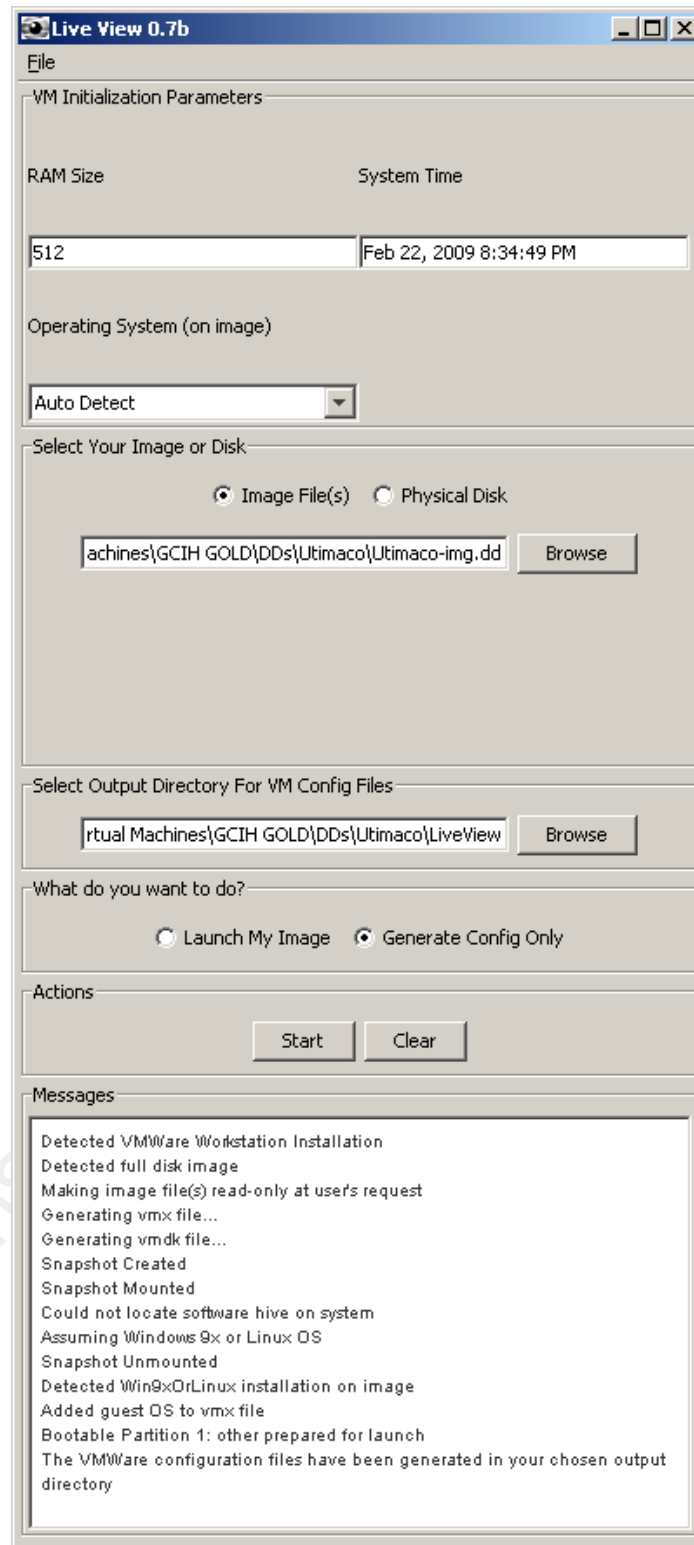


Figure 28 – Virtual Machine Created by Live View.

Appendix L: Data Collection using Helix 3

As previously mentioned, Helix can be used to collect both volatile and non-volatile data. The steps to collect this data are outlined below.

Helix 3 - Non-Volatile Data Collection

1. Boot Helix 3.
2. Mount the hard disk that will be used to store the image as read/write.
 - a. From a terminal prompt, type, "sudo su -". This will switch user to root. The terminal prompt can be accessed via the single monitor icon on the menu bar.
 - b. Type, "fdisk -l" to see a list of devices recognized by the operating system.
 - c. Mount the destination device as read/write. If you are unsure how to mount the file system then type, "man mount" in a terminal prompt window. The syntax is similar to, "mount -rw /dev/sda1 /media/sdb1".
3. Select "Applications" then select "Forensics & IR".
4. Select "Adepto". Adepto uses dcfldd as the image acquisition tool.
5. Enter a user name, and optionally, a case number then select "Go...".
6. Select the device to image using the drop down arrow.
7. Select the "Acquire" tab.
8. Optionally, you can specify an image name or use the default.
9. Optionally, you can specify image notes.
10. A destination needs to be specified. In this case a mount point needs to be specified and will be used as the destination location for the image to be acquired.
11. Optionally, the hashing algorithm can be specified or the default, md5, can be used.
12. Click "Start" to begin acquiring the image.
13. When completed, a "Verify Successful" message should appear at the bottom of the

Adepto window.

14. Click "Quit" to exit.

Helix 3 - Volatile Data Collection

1. Helix 3 can perform live incident response tasks within Windows. Thus, access to the Windows desktop is required.
2. Load the Helix 3 CD. If it does not launch automatically then launch helix.exe manually.
3. If you agree to the information contained in the warning screen then click "Accept" to continue.
4. Click the camera icon to load the live acquisition tool.



Figure 29 – Helix Incident Response Screen.

5. Select the source. This can be physical memory, or a logical or physical device.
6. Select the destination. Use separately attached storage to store this data such as a USB device.
7. Optionally, you can specify the name of the image.
8. Click "Acquire" to begin the live acquisition process.

9. You may be prompted if it is OK to proceed. If so, click "Yes".



Figure 30 – Helix Live Acquisition Prompt.

10. Image acquisition begins.



Figure 31 – Helix Live Acquisition Progress Screen.

11. When you close Helix you have the option to save a Helix audit log in PDF format which contains a list of action taken within Helix.

Helix 3 – Helix Forensic Command Shell

Another command that can be useful for data collection is the Helix Forensic Command Shell. It is accessible from the Quick Launch menu bar. This program can be useful when responding to incidents where malware or rootkits are suspected as the Forensic Command Shell uses trusted binaries from the Helix 3 CD.

Helix 3 – Zero View

Zero View is contained on the Helix 3 CD. Please see Appendix F for more information.