Forensic Implications of Windows Vista

Barrie Stewart

September 2007
Abstract

Windows XP was launched in 2001 and has since been involved in many digital investigations. Over the last few years, forensic practitioners have developed a thorough understanding of this operating system and are fully aware of any challenges it may create during an investigation. Its successor, Windows Vista, was launched in January 2007 and is fast on its way to becoming the platform of choice for new PCs. Vista introduces many new technologies and refines a number of features carried over from XP. This report focuses on some of these technologies and, in particular, what effect they have on digital investigations.

Acknowledgements

Thanks go to Ian Ferguson for arranging the placement at QinetiQ where the majority of this research was conducted. Also, thanks to Phil Turner and the staff at QinetiQ for providing valuable support and insight during the course of the research.
# Table of Contents

Abstract................................................................................................................. i

Acknowledgements............................................................................................. i

Table of Contents ............................................................................................... ii

Chapter 1 - Introduction..................................................................................... 1
  1.1 - Overview ..................................................................................................... 1
  1.2 - Methods and Tools .................................................................................. 1
  1.3 - Existing Work .......................................................................................... 2

Chapter 2 - User Account Control................................................................. 4
  2.1 - Introduction to User Account Control .................................................. 4
  2.2 - Standard User Accounts ....................................................................... 4
  2.3 - Administrator Accounts ....................................................................... 5
  2.4 - Effects of User Account Control ............................................................. 6
Chapter 9 – Internet Explorer 7 for Vista ......................................................... 54

9.1 – Introductory Note ..................................................................................... 54

9.2 – Protected Mode for IE7 ........................................................................... 54

9.3 – Internet Explorer 7 Cache and index.dat Files ...................................... 56

9.4 – Passwords .................................................................................................. 58

9.4.1 – Internet Credentials ............................................................................ 58

9.4.2 – AutoComplete Information ................................................................. 59

9.5 – Deleting Browsing History ..................................................................... 61

Chapter 10 – Additional Points of Interest ..................................................... 62

10.1 – The Recycle Bin ..................................................................................... 62

10.2 – Acquisition of Physical Memory ............................................................ 63

10.3 – ReadyBoost ............................................................................................. 63

10.4 – Transactional NTFS and Transactional Registry .................................. 63
10.5 – Disk Defragmenter ........................................................................................................64

Chapter 11 – Conclusion........................................................................................................65

Chapter 12 – Suggestions for Further Work.................................................................67

Appendices

Appendix A – References .......................................................................................................69
Appendix B – Features Comparison....................................................................................80
Appendix C – Registry Keys ................................................................................................85
Appendix D – The BitLocker Command-Line Interface ..................................................91
Chapter 1

Introduction

The report attempts to identify the technologies in Windows Vista which may have an impact on current digital forensic practices and provide a detailed overview of each one. In addition to providing an overview, the report also provides any possible solutions to new problems and highlights any differences when compared to the investigation of legacy Windows platforms.

1.1 – Overview

The report begins by discussing some of the methods and tools used to conduct the research. It then outlines some existing research material that looks at Windows Vista from a forensic perspective. The next chapter explains the User Account Control feature and, in particular, how it affects investigations in a live response scenario. Chapter 3 looks at one of the more fundamental changes in Vista – the changes to the basic directory structure. Chapter 4 details the BitLocker encryption system used in Vista while the next chapter discusses the changes to the thumbnail cache. Chapter 6 identifies the problems in trying to determine user file activity and Chapter 7 discusses some of Vista’s new backup features. The next two chapters discuss Windows Mail and Internet Explorer 7 respectively. Chapter 10 list a number of technologies which, while still important, did not merit having an entire chapter dedicated to them. The report then ends with the conclusion and outlines some suggestions for future work. Appendix B contains a comprehensive list of which features are available in the different editions of Vista. Appendix C lists a number of important registry keys that can be used during an investigation. The final appendix contains a list of useful commands for investigators dealing with BitLocker.

1.2 – Methods and Tools

The majority of the research for this report was conducted on four different machines:
- An Intel Pentium IV (2.8 GHz) PC with 1024 MB DDR running Vista Business.
- An Intel Pentium IV (2.8 GHz) PC with 512 MB DDR running Vista Ultimate.
- An AMD X2 4400+ (2 x 2.3 GHz) PC with 2048 MB DDR-II running Windows XP Pro.
- An Intel Core 2 Duo (1.66 GHz) laptop with 1024 MB DDR-II running Vista Home Premium and Ultimate.

The first machine was used to various features documented throughout the report. The second machine was used for conducting experiments with BitLocker. The third machine was used to run virtual machines and the laptop was used as a source of some of the test data used in the report. Unfortunately, a machine with a TPM (discussed later) was not available for testing the secure startup features of BitLocker.

VMware Workstation v6 [1] was used for creating virtual machines (VMs) which were used throughout the research. This included VMs running Windows XP, Vista Home Basic, Vista Home Premium, Vista Business, and Vista Ultimate.

WinHex [2] and Notepad++ [3] were used the majority of the time when examining logs and undocumented file types. Process Monitor [4] and Regshot [5] were used to monitor registry and file activity for processes and FTK Imager [6] was used for imaging drives.

The majority of testing was conducted on physical machines and then on virtual machines. While this validated results, it was also useful as the VMs could then be restored to snapshots taken immediately after Vista was installed.

1.3 – Existing Work

As Vista is currently less than a year old, there is not a great deal of detailed information currently available. “Notes on Vista Forensics” is a two-part article [7] [8] by Jamie Morris, founder of Forensic Focus [9]. The article provides a general look at some of the new features of Vista and their possible effects on forensic investigations. Lance Mueller of Guidance Software [10] made a presentation [11] at CEIC2007 that provided a more detailed look at some of the points available in [7] and [8]. More recently, there have been a number of Microsoft presentations [12] [13] which have contained material on a broad
range of forensic issues in Vista. Christopher Hargreaves and Howard Chivers of Cranfield University have written a paper [14] which covers many of the practical aspects of dealing with Vista systems.
Chapter 2

User Account Control

This chapter provides details of the User Account Control feature of Windows Vista and how it can affect investigations.

2.1 – Introduction to User Account Control

One of the major security features introduced with Vista is User Account Control (UAC). Microsoft’s main goal with UAC is to “reduce the exposure and attack surface of the operating system by requiring that all users run in standard user mode” [15]. It has been introduced as a mechanism to reduce the possibility of users inadvertently making changes to their system which may have adverse effects and also to reduce opportunities for malware to be introduced to a system.

2.2 – Standard User Accounts

Vista uses two types of user account: a standard user account and an administrator account. As part of the Vista installation process, an administrator account must be created. Any accounts that are subsequently created are standard user accounts by default, but there is also the option to create them as administrator accounts. All accounts in Vista work on the principle of least privilege whereby they are only allowed access to the resources that are necessary for their purpose. In comparison to previous versions of Windows, standard user accounts have been given additional privileges in order to perform common tasks (e.g. the ability to change the time zone) but for the most part function in a similar fashion to Windows XP limited user accounts. If a standard user attempts to perform an administrative task, they are greeted with the Credential UI shown in Figure 2.1. This allows the standard user to be “elevated” to administrator level (only if the correct password is provided) and the task can be completed.
2.3 – Administrator Accounts

In earlier versions of Windows, any process that an administrator ran would automatically run with administrative privileges. The problem with this scenario is that any malware that was run by an administrator had the potential to make changes to the system without having to explicitly ask for confirmation. UAC in Vista protects against this by running administrator accounts in Administrator Approval Mode. In Administrator Approval Mode, Vista runs most applications with standard user privileges even though the user is an administrator. When the user attempts to perform an action that requires administrative privileges, they are prompted by the Consent UI before the action can be completed (see Figure 2.2).
Vista manages to achieve this standard/administrator duality for administrator accounts by creating two access tokens for each member of the Administrator group: one token which has full administrative privileges and a second “filtered” token with User Account Protection [16]. It is this filtered token that is used by default in an administrator account with the privileged token only ever being used after the user has consented via the Consent UI. A user can explicitly tell UAC to use the privileged token by right-clicking on a program and using the “Run as administrator” option. This is often necessary even when running programs from an administrator account. For example, WinHex v14.3 refuses to open physical disk unless launched using the “Run as administrator” option.

An investigator can quickly determine what category a user falls into on a live machine by opening a command prompt and using the whoami /all command. Users belonging to the Administrators group will have a “BUILTIN\Administrators” entry under the Group Information section (note: the SID for the Administrators group is S-1-5-32-544).

2.4 – Effects of User Account Control

UAC is enabled by default and one of the major implications of it running is that many legacy applications were not compatible with standard user accounts. Vista gets around this problem by introducing file system and registry virtualization where writes to protected file system and registry locations are intercepted and written to safe, low integrity locations. It should be noted that some tools which are essential in a live incident response may refuse to run under a standard user account even when using virtualization. It is also fundamental to the new security model introduced in Internet Explorer 7 (see Chapter 9) and is also of significant when an investigator approaches a machine they may suspect is running with BitLocker encryption (see Chapter 4 for details).
Chapter 3

Changes to Windows Directory Structure

This chapter describes the methods used to create backwards-compatibility with the XP folder structure while allowing Vista to use a directory structure of its own. The chapter finishes by providing a list of legacy folders and their Vista counterpart.

3.1 – Symbolic Links and Junctions

Many of the folders familiar to XP users have been replaced in Vista. In order to maintain backwards-compatibility with legacy applications, Vista uses a combination of the new NTFS symbolic links and the older NTFS junction points. Symbolic links (symlinks) work for both files and directories in Vista. Instead of pointing directly to data, symlinks contain a symbolic path that the OS uses to identify the target file filesystem object. Symlinks appear transparent because when a user or application attempt to open, read, or write to a symbolic link, the operating system redirects the attempted action to the target object of the symlinks. By default, symlinks in Vista allow users to create symlinks to file system objects located across local volumes and are created as soft links.

Vista also provides the option to create hard symlinks. They can only be created for files and, unlike default symlinks, the target object must reside on the same NTFS volume as the symlink. They can be thought of as an N:1 association between a filename and a file system data object [17] as they use the same MFT entry as the original file. By adding a new hard link, a new name attribute is created in the MFT. In addition, the data object contains a reference count of how many filenames refer to it. The object can only be deleted if (and only if) there are no longer any filenames referencing it.

Junction points in NTFS are similar to symlinks except they are used exclusively for directories and volumes.
3.2 – Directory Structure Changes

The following tables (based on tables available from [18]) show lists of the directory structure changes in Vista and each folder's XP counterpart. Any time an application attempts to write to one of the legacy locations, Vista automatically redirects them to the new location specified in the junction/symlink. For the complete list of directory junctions in Vista see [18].

3.2.1 – Parent Folder Junction

<table>
<thead>
<tr>
<th>XP Folder</th>
<th>Vista Equivalent</th>
<th>Junction/Symlink</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Documents and Settings\</td>
<td>C:\Users\</td>
<td>Junction</td>
</tr>
</tbody>
</table>

3.2.2 – User Data Legacy Folder Junctions

<table>
<thead>
<tr>
<th>XP Folder</th>
<th>Vista Equivalent</th>
<th>Junction/Symlink</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Documents and Settings\ &lt;username&gt;\My Documents\</td>
<td>C:\Users&lt;username&gt;\Documents</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\ &lt;username&gt;\My Documents\My Music</td>
<td>C:\Users&lt;username&gt;\Music</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\ &lt;username&gt;\My Documents\My Pictures</td>
<td>C:\Users&lt;username&gt;\Pictures</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\ &lt;username&gt;\My Documents\My Videos</td>
<td>C:\Users&lt;username&gt;\Videos</td>
<td>Junction</td>
</tr>
</tbody>
</table>

3.2.3 – Per-User Application Data Legacy Folder Junctions

<table>
<thead>
<tr>
<th>XP Folder</th>
<th>Vista Equivalent</th>
<th>Junction/Symlink</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Documents and Settings\ &lt;username&gt;\Local Settings</td>
<td>C:\Users&lt;username&gt;\AppData\Local</td>
<td>Junction</td>
</tr>
</tbody>
</table>

- 8 -
<table>
<thead>
<tr>
<th>XP Folder</th>
<th>Vista Equivalent</th>
<th>Junction/Symlink</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Local Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Data</td>
<td>C:\Users&lt;username&gt;\AppData\Local</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Local Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporary Internet Files</td>
<td>C:\Users&lt;username&gt;\AppData\Local\Microsoft\Windows\Temporary Internet Files</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Local Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>C:\Users&lt;username&gt;\AppData\Local\Microsoft\Windows\History</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Application Data\</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C:\Users&lt;username&gt;\AppData\Roaming</td>
<td>Junction</td>
</tr>
</tbody>
</table>

### 3.2.4 – Per-User OS Settings Legacy Folder Junctions

<table>
<thead>
<tr>
<th>XP Folder</th>
<th>Vista Equivalent</th>
<th>Junction/Symlink</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Cookies\</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C:\Roaming\Microsoft\Windows\Cookies</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Recent\</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C:\Roaming\Microsoft\Windows\Recent</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Nethood\</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C:\Roaming\Microsoft\Windows\Network Shortcuts</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Printhood\</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C:\Roaming\Microsoft\Windows\Printer Shortcuts</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\SendTo\</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C:\Roaming\Microsoft\Windows\Send To</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\StartMenu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C:\Roaming\Microsoft\Windows\StartMenu</td>
<td>Junction</td>
</tr>
<tr>
<td>C:\Documents and Settings\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;username&gt;\Templates\</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C:\Roaming\Microsoft\Windows\Templates</td>
<td>Junction</td>
</tr>
</tbody>
</table>
### 3.2.5 – All Users Legacy Folder Symlink

<table>
<thead>
<tr>
<th>XP Folder</th>
<th>Vista Equivalent</th>
<th>Junction/Symlink</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Users\All Users</td>
<td>C:\ProgramData</td>
<td>Symlink</td>
</tr>
</tbody>
</table>

### 3.2.6 – Default User Legacy Junction

<table>
<thead>
<tr>
<th>XP Folder</th>
<th>Vista Equivalent</th>
<th>Junction/Symlink</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Documents and Settings\Default User</td>
<td>C:\Users\Default</td>
<td>Junction</td>
</tr>
</tbody>
</table>
Chapter 4

BitLocker

This chapter discusses the BitLocker encryption technology introduced in Vista. It starts by giving an introduction to BitLocker then provides details of the security features it provides. The chapter then describes some of the practical challenges when faced with a BitLocker encrypted system and, in particular, the importance of dealing with BitLocker systems in a live response scenario.

4.1 – Introduction

BitLocker is one of the most talked about features available in Vista. Its fundamental purpose is to protect the data on a laptop or PC even if it is stolen. Data loss has been thrust into the public eye over the last few years due to a number of high profile cases involving loss of personal data [19] [20]. The nonprofit consumer rights organization Privacy Rights Clearinghouse [21] lists instances of data loss since January 2005 which currently stands at a total of over 159 million data records of US residents being exposed. BitLocker is available in the Ultimate and Enterprise editions of Vista and is aimed primarily at commercial and government organizations looking to minimize the risks associated with data loss.

With previous versions of Windows, accessing data on the OS drive was as simple as removing the physical drive from the machine and then accessing it as a secondary drive on another machine. This made it possible to access any data on the drive by using the administrator privileges of the second machine. An even simpler solution was to use a software-based attack to gain access to an administrator account on the machine. This could be achieved by booting the computer from a CD or USB drive containing tools such as Offline NT Password & Registry Editor [22] or ophcrack [23]. BitLocker overcomes the limitations of its predecessors by providing two layers of protection:
- Secure Startup – Verifies the integrity of the pre-OS boot components and protects against offline attacks
- Full volume encryption – Encrypts the entire Windows OS volume on the hard drive (everything on the volume is encrypted including user data, system files, file slack, unallocated space etc.)

The aim of BitLocker is to provide this protection in an easy to use manner in which the encryption/decryption process is transparent to the user, with minimal disk and processor overhead, and with a simple recovery process in the event of hardware failure or changes.

4.2 – BitLocker Requirements

There are several hardware requirements that must be met in order for BitLocker to be enabled on a volume. The first requirements stem from BitLocker storing its encryption key in a hardware device that is separate from the hard disk that is being encrypted. Therefore, the following hardware components are necessary:

- If the user wishes to benefit from the secure startup component of BitLocker, there are two requirements:
  - A Trusted Platform Module (TPM) which is a tamper-proof microchip that can perform certain security-related functions. The TPM is typically installed on the motherboard of the PC or laptop. BitLocker requires a version 1.2 TPM to function.
  - A Trusted Computing Group (TGC)-compliant BIOS.
- If the user does not have a TPM (or wishes to use the TPM + Startup Key mode), BitLocker requires the following to store the startup key:
  - A removable USB device (e.g. a USB flash drive)
  - A system BIOS that supports reading from a USB device in the pre-OS environment
- The hard drive must be partitioned with at least two volumes:
- An NTFS-formatted OS volume which contains the Vista operating system and all of its supporting components. This volume will be encrypted by BitLocker.

- An NTFS-formatted system volume (at least 1.5GB in size) which contains the files necessary to boot the operating system as well as being required for OS upgrades. This volume is required to remain unencrypted.

### 4.3 – BitLocker Key Protectors

There are four different authentication modes that can be used with BitLocker. The choice of which authentication mode to use depends on the hardware available and the users intended level of involvement in the boot process.

#### 4.3.1 – TPM-Only

The TPM-only mode is the easiest to use of the four authentication modes as it does not require any user interaction. The TPM simply checks the pre-OS boot components and then boots into Windows if the check has been successful. This protects against any software attacks but may be susceptible to specialized hardware attacks.

#### 4.3.2 – USB Drive-Only

BitLocker provides protection in the absence of a TPM by allowing the user to store a startup key on a removable USB drive. In this scenario, the user must enter the USB drive every time they wish to boot into Windows. BitLocker verifies the USB drive contains the startup key (pre-OS) and will then boot into the operating system. This method also protects against hardware attacks but suffers from the risk of the pre-OS attacks or the USB drive being lost or stolen. There is also the risk of the startup key being copied from the USB drive without the owner’s knowledge.

#### 4.3.3 – TPM+PIN

In this authentication mode the TPM first validates the early boot components. If this check is successful, the user is requested to enter their PIN code (4-20 digits which were entered
when BitLocker is first enabled). Once the PIN is verified, the OS will start. The PIN is stored within a secure storage area of the TPM and is protected by anti-hammering techniques. This protects against brute-force attacks on the PIN by making the TPM inaccessible for a period of time whenever the wrong PIN is entered. The more times the PIN is entered incorrectly, the longer the period of inaccessibility [24]. This mode protects against many hardware attacks but may still be vulnerable to specialized TPM attacks.

4.3.4 – TPM+USB

This mode validates the early boot components and then requires the user to enter a USB drive before they are allowed to boot to Windows. This mode protects against many hardware attacks and TPM attacks but suffers from the same risks associated with the USB-only mode.

4.4 – Boot Integrity Validation (Secure Startup)

The secure startup feature of BitLocker uses the TPM to verify the integrity of the boot components. The TPM contains a number of Platform Configuration Registers (PCRs) which, upon power-up, are all initialized to zero. The value of each PCR can only be modified by calling it’s `extend` function which sets the value of the PCR to be the hash of its old value and a supplied data string [25]. What this effectively means is that the final value of a particular PCR will be the hash of all the data strings supplied to it’s `extend` function. The only way this value could ever be repeated is to supply identical data strings as before to the extend function (in the identical order) after power-up of the computer.

When BitLocker is first enabled in one of the TPM authentication modes, the computer will reboot and during the boot process the PCRs will keep track of any boot code that runs (including BIOS code and the MBR). The TPM will then seal the BitLocker encryption key based on the values of the PCRs. The seal function also ensures that only the TPM that sealed the key is able to unseal it. Subsequent boots should produce exactly the same values in the PCRs and the TPM will unseal the encryption key. Any changes made to the BIOS or boot code will result in the PCRs producing different values. The TPM is then
unable to unseal the encryption key and Windows cannot boot. This is what ensures that the encrypted drive cannot be unlocked simply by installing it in a different system.

4.5 – BitLocker Encryption Algorithm

The designers of BitLocker looked at a number of different encryption algorithms [25] before deciding to use the Advanced Encryption Standard (AES) [26]. AES has been subject to extensive analysis and is considered to be one of the most secure algorithms available and also managed to fulfill the performance requirements of BitLocker. AES is used in cipher-block chaining (CBC) mode with the addition of an “elephant” diffuser algorithm in which two diffusers work in opposite directions. The purpose of the diffuser is that “it ensures that even minute changes to the plaintext result in the entire sector changing in the encrypted ciphertext” [27]. Encryption is performed on a per-sector basis where the sector number is used as an extra parameter in the encryption. This ensures that even if identical data is written to different sectors of the drive, those sectors will end up containing different ciphertexts. Figure 4.1 shows an overview of the AES-CBC + diffuser encryption process:

![AES-CBC + Diffuser Encryption Process](image)
The steps of the process are as follows:

1. The unencrypted sector (i.e. the plaintext) is exclusive-ored with the sector key
2. This is then run through the diffuser (there are actually two diffusers – one for each direction)
3. The result is then encrypted with AES in CBC mode producing the encrypted sector

The full volume encryption key (FVEK) is the 512-bit key which is used to encrypt the drive sectors. It is actually a split key consisting of two 256-bit keys – one key for deriving the sector key and the other key to use in the AES-CBC component of the encryption. Whilst the technical details of the encryption process are outside the scope of this project, it is important to note that as the sector key component and the AES-CBC component are independently keyed, BitLocker is at least as secure as AES-CBC. Full details of the encryption process are available in a paper by Microsoft’s Niels Ferguson [25]. BitLocker can be used in the following modes:

- AES-128 with diffuser (default mode for BitLocker)
- AES-256 with diffuser
- AES-128 no diffuser
- AES-256 no diffuser

4.6– BitLocker Keys

As mentioned previously, the FVEK is the most important key in the BitLocker key hierarchy as it is the key that is used in the encryption and decryption process for every sector on the drive. BitLocker stores the FVEK on the disk as part of the volume metadata. Three copies of the volume metadata are written to a BitLocker drive in the interest of redundancy [27]. The FVEK can obviously never be written to the disk in an unencrypted form. Instead, it is first encrypted with the volume master key (VMK) before being stored. The VMK is also encrypted and can only be decrypted by one of the “key protectors” described in Section 4.3. Microsoft decided to use the two keys (FVEK and VMK) so that the BitLocker installation could be re-keyed without having to re-encrypt all the data on the
drive. Changing the BitLocker key effectively means changing the VMK that protects the FVEK.

### 4.7– Enabling BitLocker

Vista Ultimate and Enterprise do not come with BitLocker enabled by default. Access to enabling BitLocker can be found BitLocker Drive Encryption entry in the Control Panel. A default Vista installation will not allow BitLocker to be enabled unless the computer has a TPM. To allow the option to enable BitLocker with the USB-only authentication mode, the user must first perform the following steps:

1. Press the Start button and in the “Start Search” box type ‘gpedit.msc’ (without quotes) to open the group policy object editor.
3. Double-click the “Control Panel Setup: Enable advanced startup options” entry in the right-hand pane.
4. Click the “enabled” radio button. Ensure “Allow BitLocker without a compatible TPM” is checked and press Apply.
5. Close the group policy editor and open a command prompt by pressing the Start button and in the “Start Search” box type ‘cmd’ (without quotes).
6. In the command prompt use the command `gpupdate /force` to update the local group policy settings.

After the above steps have been completed, the user may enable BitLocker in the USB-only mode. When enabling BitLocker for the first time, the user must either save a copy of the recovery password, or print a copy (discussed later). Figure 4.2 shows the screen prompting the user to either save or print the recovery password. Once this has been completed, the computer then restarts and begins the encryption process in the background.
In addition to providing the BitLocker Drive Encryption entry in the Control Panel, Microsoft has also included a WSF script `manage-bde.wsf` which resides in `C:\Windows\System32`. This is the BitLocker Command-line Interface (CLI) allows access to enabling BitLocker not only on the OS volume, but also on any other volumes the user wishes to encrypt (except the system volume holding the unencrypted files used to start the OS). As well as providing access to enabling BitLocker, this script has many other useful features and will probably prove to be an essential part of any live response toolkit. A list of useful commands for the script can be found in Appendix D.

4.8–BitLocker Startup Keys

Many users of Vista Ultimate/Enterprise will not have the TPM hardware required to run BitLocker in any of the TPM authentication modes. In this case, they can only run BitLocker in the USB-only mode. The startup key is a binary key held in a 124-byte hidden file on the USB drive which has the filename format `XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX.BEK` where each `X` is a hexadecimal digit. If BitLocker is enabled in the USB-only mode, this key can be used to unlock the drive on another computer.
4.9-BitLocker Recovery Passwords and Keys

The security features implemented in BitLocker mean that it is currently not possible to decrypt an encrypted drive without the presence of the correct key protector. However, during the process of enabling BitLocker, the user is required to either save and/or print a recovery password. The recovery password is a 48-digit number that can be used in the event that something goes wrong during the BitLocker startup process. Microsoft has identified a number of recovery scenarios [28] including:

- Lost USB key
- Forgotten PIN
- Change to pre-OS files (e.g. BIOS upgrade)
- Broken hardware (e.g. TPM breaks and hard drive has to be moved to a new system)
- Deliberate attacks (e.g. hacked BIOS, modified MBR)

In the event of a recovery scenario, the user is presented with the following screen where they can enter their 48-digit recover password in order to unlock the drive:

![Windows BitLocker Drive Encryption Password Entry](image_url)

Figure 4.3 - Recovery Password Entry
In addition to this method, the user can also transfer the drive to another Vista Ultimate or Vista Enterprise system and unlock the drive via either the BitLocker control panel or using the `manage-bde.wsf` script (discussed later).

48-digit recovery passwords saved to disk are saved in a text file with the filename format `XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX.txt` where each `X` is a hexadecimal digit. This 32-digit hexadecimal string corresponds to the administrative password ID that is assigned by BitLocker when the drive is enabled. The text file contains the recovery password. The contents of a typical recovery password file are shown in Figure 4.4.

![Figure 4.4 - Contents of a Typical Recovery Password File](image)

The relationship between the administrative password ID, recovery password filename, and 48-digit recovery password are shown in Figure 4.5:

![Figure 1.5 - Relationship between BitLocker Password ID and Recovery Password](image)
As well as being able to identify a recovery password file for BitLocker, an investigator must also be aware that the user may have a printed copy of the recovery password. An example recovery password printout created using the BitLocker control panel is shown in Figure 4.6:

<table>
<thead>
<tr>
<th>BitLocker Drive Encryption Recovery Password, store in a safe place.</th>
</tr>
</thead>
<tbody>
<tr>
<td>142879 - 517825 - 291302 - 449383 - 025234 - 110880 - 031526 - 521323</td>
</tr>
</tbody>
</table>

Figure 4.6 - Example Printed Recovery Password

BitLocker recovery keys can also be created (see Section 4.11). BitLocker recovery keys are “cryptographically equivalent” [29] to startup keys and can be used to unlock the drive.

It is clear that recovery passwords/keys are important for when something goes wrong with a BitLocker installation. From a forensic point of view, they are even more important. A physical image of a drive with BitLocker can be created without any problems. However, a physical image of the drive will simply hold the encrypted data. This is of very little use from an investigative standpoint. In order for the investigator to view the unencrypted data, a logical image of the volume must be taken and this requires the use of the recovery password/key to unlock the drive. If the investigator cannot gain access to the system in a live response scenario, they must be aware that if BitLocker is enabled and the computer is then switched off, it may prove impossible to recover the unencrypted data. For this reason, it is vitally important in live response scenarios that removable media and printouts are seized and later searched for BitLocker keys or passwords.

4.10–Identifying the Presence of BitLocker

In a live response scenario, an investigator may approach a machine that is running Vista and may have BitLocker enabled. If the user account(s) on the machine have a password,
the following scenarios may require a password is entered before the investigator can gain access to the desktop:

1. The machine is currently logged off (All user accounts are password protected).
2. The machine is currently locked (All user accounts are password protected).
3. The machine is in sleep mode (By default, Vista has the “Require a password on wakeup” option set to Yes).
4. The machine is in hibernate mode (By default, Vista has the “Require a password on wakeup” option set to Yes).
5. The screensaver for the desktop has been activated (By default, Vista does not use the “On resume, display logon screen” under the screensaver settings).

In these scenarios, without obtaining a password from the suspect or system administrator it is currently not possible to gain access to the desktop. If the system is currently logged on, care must be taken that any of these scenarios do not occur. It is also important to note that if investigators approach an unlocked system that is only running a standard user account, UAC will make it impossible to identify the presence of BitLocker and/or create recovery keys using any of the methods discussed in this chapter. In this scenario, the investigator would need to obtain the password for one of the administrator accounts on the machine to elevate to administrator status.

4.10.1–Identification via BitLocker Command-Line Interface (CLI)

The best technique to determine if BitLocker is enabled is to use the manage-bde.wsf script mentioned in Section 4.9. Using the \-status parameter provides the encryption status of all volumes. The steps are as follows:

1. Navigate to Start->All Programs->Accessories. Right-click on the “command prompt” icon and select “Run as administrator” when prompted. Press “continue” in the UAC popup window.
2. In the command prompt, navigate to code:\windows\system32
3. Run the command cscript manage-bde.wsf -status
This will provide an output similar to the following:

![Image of BitLocker CLI output]

**Figure 4.7 - Using BitLocker CLI to Determine Encryption Status**

This will list the encryption status of any volumes on the system. Note: this script will only work when run with administrator credentials and using the “Run as administrator” action. When run with standard privileges, the BitLocker CLI produces the following error:

![Image of BitLocker CLI output with error]

**Figure 4.8 - Running BitLocker CLI with Standard Privileges**

### 4.10.2 Identification via BitLocker Control Panel

The encryption status of any drives can also be determined by checking the BitLocker control panel as follows:

2. Double-click the entry to open the BitLocker control panel.
Figure 4.9 shows the BitLocker control panel with two encrypted drives – the encrypted OS volume (C:) and an encrypted data volume (G):

![BitLocker Drive Encryption](image)

Note: The BitLocker control panel is only accessible with administrator privileges.

### 4.10.3–Identification via Disk Management Snap-In

The disk management snap-in can also be used to identify a BitLocker volume. This can be accessed using the following steps:

1. Press the Start button then right click on "Computer" and select "Manage".
2. In the Computer Management window select Computer Management (Local) > Storage > Disk Management in the left pane.
3. The disk configuration will then load in the centre pane.

Figure 4.10 shows a typical BitLocker disk configuration with the small, unencrypted system partition and the larger encrypted OS volume:

![Disk Management Snap-In with BitLocker Drive](image)
4.10.3—Identification Using a Hex Editor

A blog entry [30] posted by Jamie Hunter of Microsoft’s System Integrity Team describes how to identify the presence of BitLocker by looking directly at the contents of the physical disk. In particular, the method involves looking at the Bios Parameter Block (BPB) which is located at the first 0x54 bytes of the first sector of the volume. Jamie provides the following table to check for the existence of BitLocker:

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Field</th>
<th>Required Value for BitLocker</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x003</td>
<td>8</td>
<td>Signature</td>
<td>&quot;F:V:1:F:S:&quot;</td>
</tr>
<tr>
<td>0x00B</td>
<td>2</td>
<td>BytesPerSector</td>
<td></td>
</tr>
<tr>
<td>0x00D</td>
<td>1</td>
<td>SectorsPerCluster</td>
<td>One of 0x01, 0x02, 0x04, 0x05, 0x10, 0x20, 0x40 or 0x80</td>
</tr>
<tr>
<td>0x00E</td>
<td>2</td>
<td>ReservedClusters</td>
<td>0x0000</td>
</tr>
<tr>
<td>0x010</td>
<td>1</td>
<td>FatCount</td>
<td>0x00</td>
</tr>
<tr>
<td>0x011</td>
<td>2</td>
<td>RootEntries</td>
<td>0x0000</td>
</tr>
<tr>
<td>0x013</td>
<td>2</td>
<td>Sectors</td>
<td>0x0000</td>
</tr>
<tr>
<td>0x016</td>
<td>2</td>
<td>SectorsPerFat</td>
<td>0x0000</td>
</tr>
<tr>
<td>0x020</td>
<td>4</td>
<td>LargeSectors</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x038</td>
<td>8</td>
<td>MetadataLin</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1 - BPB Characteristics of a BitLocker Volume

By examining the first sector of the volume and checking all fields with a “required value”, it is possible to identify BitLocker. This can be achieved using WinHex by using the following procedure:

1. Start WinHex. Select Tools->Open Disk
2. Under “Physical Media” select the disk suspected of containing a BitLocker volume and press OK
3. The disk will now have its own tab in the WinHex main window. Double click the partition suspected of containing the BitLocker volume.
4. Inspect the disk using Table 4.1 as a guide.

Figure 4.11 shows each of the values from Table 4.1 highlighted in WinHex. Red and blue have been used for highlighting in an alternating fashion for ease of reading. The byte string “2D 46 56 45 2D 46 53 2D” corresponds to the “Signature” field of the BPB. The string “02 02” corresponds to the “BytesPerSector” field and so on.
4.11–Viewing and Creating BitLocker Recovery Passwords/Keys

Section 4.9 described the importance of the recovery passwords/keys for a drive. During a live response, if the investigator has access to the Vista desktop with an account that has administrator privileges, it is possible to view the recovery password for a drive or create a backup of the recovery keys.

4.11.1–Viewing Key Protectors via BitLocker CLI

To view the current key protectors for a drive, use the following procedure:

1. Navigate to Start->All Programs->Accessories. Right-click on the "command prompt" icon and select “Run as administrator" when prompted. Press “continue” in the UAC popup window.
2. In the command prompt, navigate to `c:\windows\system32`
3. To get the protectors for the `c:` drive, we run the command
   ```plaintext
cscript manage-bde.wsf -protectors -get C:
   ```
4. Protectors for other drives can be obtained by simply changing the drive letter.

This will provide an output similar to the following:
4.11.2–Obtaining Recovery Keys via BitLocker CLI

Copies of any recovery keys can be created using the following procedure:

1. Navigate to Start->All Programs->Accessories. Right-click on the “command prompt” icon and select “Run as administrator” when prompted. Press “continue” in the UAC popup window.

2. In the command prompt, navigate to c:\windows\system32

3. To copy the protectors for the C: drive, we run the command cscript manage-bde.wsf -protectors -get C: -sek G:\KeyBackup

This will copy the recovery keys for drive C: to the location G:\KeyBackup. The drive letter and save path can be amended as necessary. A typical output from running the command is shown in Figure 4.13. Investigators should note that this command does not make a copy of the numerical password. This must be noted manually.
4.11.3–Obtaining Recovery Keys via BitLocker Control Panel

Copies of the recovery keys can also be obtained by using the BitLocker control panel (Control Panel–>BitLocker Drive Encryption). By clicking “Manage BitLocker Keys” beneath the encrypted OS entry, it is possible to:

- Duplicate the recovery key to a USB drive
- Duplicate the recovery key to a folder
- Print the recovery password
- Reset the PIN (for TPM + PIN mode)

4.11.4–Other Sources of Recovery Keys

Windows Active Directory can be configured to remotely store recovery keys. This feature is used to allow admins to control access to recovery keys and silently backup any created recovery keys to Active Directory. Investigators should be aware that in enterprise environments with Active Directory, the IT administrator may have access to recovery keys. Full details are available from [31].

Windows Ultimate Extras are a series of free add-ons available for Vista Ultimate via Windows Update. One of these is the “BitLocker and EFS Enhancements” extra. This
provides a new “Secure Online Key Backup” feature (see Figure 4.14) allowing the user to store a copy of their recovery password and Encrypting File System (EFS) recovery certificates using Windows online Digital Locker [32]. The recovery keys are obtained by logging into Windows Marketplace using the user’s Windows Live ID.

Figure 4.14 - Secure Online Key Backup

4.12–Working with BitLocker Images

A BitLocker drive should be imaged by connecting the drive to a forensically-secure workstation (running either Vista Ultimate or Enterprise) and using a write-blocker. An physical image of the disk can be taken if necessary. However, it should be noted that this is an image of the encrypted data. To provide an image that an examiner can actually work with, a logical image of the volume must be taken. Before creating a logical image, the drive must first be unlocked which requires the recovery keys for the drive. Unlocking the drive can be achieved by using either the BitLocker CLI or using the BitLocker control panel. Once unlocked, the logical image can be created using tools such as FTK Imager [6]. It should be noted that such tools must be run with administrator privileges.

4.12.1–Unlocking via the BitLocker Control Panel

The BitLocker control panel allows unlocking of an encrypted drive using either the recovery password or the recovery key. This is accessed using Start->Control Panel->BitLocker Drive Encryption. The examiner then has the option to unlock any BitLocker encrypted volumes:
Windows then gives the option to “Load password from removable media” or “Manually input the password”. Loading the password from removable media requires the recovery key file (*.BEK) to be contained in the root of a removable USB device. It should be noted that the .TXT file containing the 48-digit recovery password will not unlock the drive using this option. However, the password contained in the file will unlock the drive by selecting the “Manually input the password” option and typing the 48-digit code into the text box.

4.12.2–Unlocking via the BitLocker CLI

The BitLocker command-line interface can be used to unlock an encrypted drive by using either the recovery password or the recovery key. Unlocking using the recovery password is as follows:

1. Navigate to Start->All Programs->Accessories. Right-click on the “command prompt” icon and select “Run as administrator” when prompted. Press “continue” in the UAC popup window.

2. In the command prompt, navigate to `c:\windows\system32`
3. To unlock the E: drive, we run the command `cscript manage-bde.wsf -unlock E: -RecoveryPassword XXXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX`

4. The drive letter can be amended to suit the examiners needs. The last argument to the command should obviously be replaced with the 48-digit recovery password specific to the drive being unlocked.

Figure 4.16 shows a typical output from running the command:

![Figure 4.16 - Unlocking using BitLocker CLI & Password](image)

The following method is used to unlock using a recovery key:

1. Navigate to Start->All Programs->Accessories. Right-click on the “command prompt” icon and select “Run as administrator” when prompted. Press “continue” in the UAC popup window.

2. In the command prompt, navigate to `c:\windows\system32`

3. To unlock the E: drive, we run the command `cscript manage-bde.wsf -unlock E: -RecoveryKey <file>`

4. The drive letter can be amended to suit the examiners needs. The last argument to the command should be the filename (including full path) of the recovery key. This part should be enclosed in inverted commas if there are any spaces in the path/filename.

Figure 4.17 shows a typical output from running the command
4.13–Attacking BitLocker

Douglas MacIver of the Microsoft Penetration Team gave a presentation [33] at the 2006 Hack in the Box security conference. This outlined a number of possible attack vectors for gaining access to an encrypted BitLocker volume. During extensive testing, the team did not manage successfully unlock the drive through an attack. Nitin and Vipin Kumar, owners of the Indian security firm NV Labs [34], were recently scheduled to give a presentation at the Black Hat security conference regarding their TPMkit software (abstract available at [35]). They claim to have developed a method to circumvent the TPM checks during the secure startup of a BitLocker system. The presentation was later controversially withdrawn at the request of the Kumar brothers [36]. As it stands, there have been no known successful attacks on a BitLocker system.
Chapter 5

Thumbnails

This chapter begins by outlining the caching system that was used in Windows XP. It then describes how this has changed in Vista and how an examiner can extract thumbnails from the new cache. The chapter concludes by detailing part of the file format used for storing thumbnails in the Vista thumbcache files and how this can be used to find extra information about the original file.

5.1 – Thumbnails in Windows XP

Thumbnail views in Windows show a miniature view of a larger image, allowing users to quickly scan through a large number of images. Some of the file formats XP stores as thumbnails include the following [37]: JPEG, BMP, GIF, TIF, PDF and HTM. Thumbnail views could be accessed in an XP Explorer window by using View->Thumbnails. This would display any graphics in that folder as thumbnails rather than the icon views normally associated with the files. When a folder had the thumbnail view enabled (and contained a “thumbnailable” format), a hidden file thumbs.db was automatically created. This file is a database of the thumbnails for that folder with each thumbnail being stored in the database as a JPEG file, regardless of the file’s original format [38]. In addition to storing the thumbnail image of the parent file, XP also stores the filename. When an entry is added to the thumbs.db, it stays there indefinitely even if the original file is deleted. This means that an XP user has to either delete the thumbs.db file itself (or the folder that holds it) in order to remove traces of any graphics that have been stored in the file. Tools such as FTK [39] are able to display the thumbnails contained in the thumbs.db file as well as the parent filename and modified date of the parent file. Full details of thumbs.db files and working with them in FTK are available at [38].
5.2 – Thumbnails in Windows Vista

Thumbnails (sometimes referred to as “Live Icons”) in Vista now allow scalable thumbnails of a file and show a preview of the actual contents of a file or folder, rather than the generic program icons used in XP. Figure 5.1 shows two views of a folder containing another folder and four files: two jpeg images and two PDF files. The left window shows the “List” view in Vista and uses the traditional program icons. The right window shows the “Large Icons” view and shows a preview of each of the files as well as a preview of the contents of the folder. Users change the size of the icons used by clicking the Views button on the folder toolbar and moving a slider up and down to increase/decrease the size of the icons used. Selecting any size from “Medium Icons” and above will preview the contents of the file, as will using the “Tiles” view.

![Figure 5.1 - Different Views in Windows Vista](image)

Vista has removed the need to create a thumbs.db file on a per-folder basis. Instead, it uses a centralized thumbnail cache for each user located at:

```
C:\Users\<username>\AppData\Local\Microsoft\Windows\Explorer
```

There are four thumbnail caches as Vista stores thumbnail images in a set of discrete sizes [40]. The sizes (in pixels) are 32x32, 96x96, 256x256, and 1024x1024. The corresponding cache files are named:

- thumbcache_32.db
- thumbcache_96.db
During testing, the files were found to contain embedded JPG, BMP, and PNG files. These were extracted using the “Data Carve” feature of FTK v1.71. Table 5.1 explains the content of the files found during testing. Figure 5.2 shows FTK after performing a data carve on thumbcache_1024.db.

<table>
<thead>
<tr>
<th>File</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>thumbcache_32.db</td>
<td>Stores thumbnails up to 32x32 pixels. During testing, all thumbnails were stored in BMP format. The BMP files held previews of both files and folders.</td>
</tr>
<tr>
<td>thumbcache_96.db</td>
<td>Stores any thumbnails larger than 32x32 pixels with maximum dimensions of 96x96 pixels. During testing, all thumbnails were stored in BMP format. The BMP files held previews of both files and folders.</td>
</tr>
<tr>
<td>thumbcache_256.db</td>
<td>Stores any thumbnails larger than 96x96 pixels with maximum dimensions of 256x256 pixels. During testing, thumbnails were stored in a mixture of JPG and PNG formats. The JPG files stored previews of files and the PNG files held previews of folders.</td>
</tr>
<tr>
<td>thumbcache_1024.db</td>
<td>Stores any thumbnails larger than 256x256 pixels with maximum dimensions of 1024x1024 pixels. During testing, all thumbnails were stored in JPG format. The JPG files stored previews of files. No previews of folders were stored.</td>
</tr>
</tbody>
</table>

Table 5.2 - Contents of thumbs.db Files During Testing

Figure 5.2 - FTK Preview from thumbcache_1024.db
The test data used during experimentation was obtained from three different Vista installations – two Ultimate installations which had been used for the majority of testing in this document and a Home Premium installation which had been used for general everyday use for approximately three months. Several specific test cases were also set up. These test cases were used to check for caching of thumbnails for files on removable devices and networked devices. It was found that Vista caches thumbnails in both these cases.

5.3 – Thumbcache File Format

During testing, the vast majority of thumbnails held in the thumbcache files had a filename of the format \texttt{XXXXXXXXXXXXXX.<ext>} where each \texttt{X} is a hexadecimal digit and \texttt{<ext>} is either \texttt{BMP}, \texttt{JPG} or \texttt{PNG}. There were a few cases where the filename had stayed the same as the parent file with the \texttt{.<ext>} part then appended.

Inspection of the thumbcache files with a hex editor revealed some interesting features of the caching. The embedded JPG, BMP, and PNG files could be found by searching for the appropriate file headers for each format (e.g. \texttt{0xFFD8FFE000104A464946} for a JPG). Before the header for the file there is a block of metadata. The start of the metadata is identified by the 4-byte hex string \texttt{0x434D4D4D}. Following this there are another 4 bytes, the purpose of which are unknown. The 8 bytes after that contain a unique identifier for the file. The last 32 bytes correspond to the hexadecimal name given to the thumbnail file (except in the rare case the thumbnail has retained the filename of the parent file). Every 2\textsuperscript{nd} byte of this 32-byte filename section is padded with \texttt{0x00} effectively leaving 16 bytes for the filename. The size of this metadata block varies, even between cases where thumbnails are being stored in the same image format. Figure 5.3 shows which parts of the thumbnail metadata remain constant. Any sections which have varying lengths are denoted by “\texttt{?bytes}”. 

<table>
<thead>
<tr>
<th>4bytes</th>
<th>4bytes</th>
<th>8bytes</th>
<th>?bytes</th>
<th>32bytes/?bytes</th>
<th>?bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>434D4D4D</td>
<td>UNKNOWN</td>
<td>&lt;UNIQUE ID&gt;</td>
<td>UNKNOWN</td>
<td>&lt;FILENAME&gt;</td>
<td>&lt;FILE CONTENT&gt;</td>
</tr>
</tbody>
</table>

Figure 5.3 - Thumbnail Metadata Format
Inside the \C:\Users\<username>\AppData\Local\Microsoft\Windows\Explorer\ folder there is also an index file named thumbcache_idx.db. If we search the index file for the unique ID of one of the thumbnails contained in the other four files, we should find a match. The 8 bytes that follow the matching unique ID give us the 8-byte Windows FILETIME value for when the parent file was created (i.e., the file the thumbnail was created from). During testing, a thumbnail view for the image in Figure 5.4 was created. Figure 5.5 on page 38 shows the thumbnail in FTK then the corresponding entries in thumbcache_256.db and thumbcache_idx.db.

![Figure 5.4 – File Properties of Parent Image](image-url)
Figure 5.5 - Thumbnail Entry in Hex Editor

<table>
<thead>
<tr>
<th>Windows FILETIME</th>
<th>Modified Data</th>
<th>Unique ID</th>
<th>Metadata Block</th>
<th>Thumbnail Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>19820260800158704</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 6

User File Activity in Vista

This chapter describes the challenges facing investigators when examining file activity in a Windows Vista system. It then goes on to describe a number of methods which can be used in helping to gain an insight into a user’s file activity.

6.1 – Windows XP and Last Access Times

Files on an NTFS volume contain a Last Access Time attribute showing when that file was last accessed. This attribute is updated by default on Windows XP. When a file is accessed, the NTFS filesystem updates the last accessed attribute in memory and later (with a delay of up to one hour) writes the attribute to disk [41][42]. Even with this potential delay of up to one hour, the Last Access Time attribute has been a valuable source of evidence by allowing investigators to build approximate timelines of file activity. This is useful in helping to build a profile of an offender’s activity and give an idea of the order events may have occurred.

6.2 – Vista and Last Access Times

During the development of Vista, Microsoft decided to disable the updating of the Last Access Time file attribute by default to improve NTFS performance [43]. While this may improve performance for the end-user, it provides a headache for forensic examiners as creating timelines of file activity is no longer possible. Examiners must now employ other methods to build a profile of user activity.

6.3 – Recent Items Folder

Windows XP kept a record of files and folders recently opened by the user. These were stored as shortcuts in the C:\Documents and Settings\<username>\Recent folder. Vista continues this behaviour by storing shortcuts to recently accessed files and folders in the C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Recent Items folder.
When a recently accessed file/folder is added to the recent items list for the first time, a shortcut is created. Whenever that file/folder is subsequently opened, the created date for the corresponding shortcut stays the same but the modified date of the shortcut is updated. By examining the shortcuts and their modified dates, an investigator can determine some of the files that have been recently accessed and at what time they were last accessed.

6.4 – Recently Executed Programs

Vista maintains a record of recently executed programs in the following registry key:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count

When a program is executed for the first time, a value is created in the registry key. The value name is stored in ROT13 format (each letter rotated by 13 places). The name consists of a string identifying the type of command followed by the name of the executable. During testing, the following command identifiers were found:

<table>
<thead>
<tr>
<th>Command Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UEME_CTLCUACount</td>
<td>Unknown.</td>
</tr>
<tr>
<td>UEME_CTLSESSION</td>
<td>Unknown.</td>
</tr>
<tr>
<td>UEME_RUNPATH</td>
<td>Executables run on the computer.</td>
</tr>
<tr>
<td>UEME_RUNPIDL</td>
<td>Executables accessed via the start menu and web pages accessed using the Start-&gt;Start Search entry dialog.</td>
</tr>
</tbody>
</table>

Table 6.3- Command Identifiers and Descriptions

An example value name for the UEME_RUNPATH command identifier is shown below:

UEME_RUNPATH:C:\Program Files\AccessData\AccessData FTK Imager\FTK Imager.exe

The value data then contains a counter of how many times the program has been executed and the date and time of when it was last executed. Any executables that have been launched by using Start->Start Search and entering the executable name take the format of:

UEME_RUNPATH:%csid10%<executable>" "

- 40 -
UEME_RUNPIDL entries appear to be programs which have been launched via links in the start menu. These entries are also used to maintain a record of URLs typed into the Start-Start Search dialog (this then launches the computer’s default web client and navigates to the page). Reference [44] outlines some work performed on examining the user assist keys on a Windows XP system. Viewing the user assist keys is made easier by using a tool which automatically retrieves and decrypts the entries such as UserAssist v2.3.0 [45].

6.5 – Vista RecentDocs Key

In addition to maintaining a list of previously executed programs in the user assist keys, Vista also stores a list of recently used documents in the registry in the following key:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

Recently accessed documents are added to this key as binary values and are numbered sequentially in order of when they were first accessed. If a file is accessed again, it retains its original value name. The RecentDocs key also contains a number of subkeys. They are named after the filename extensions of any files added to the recent documents list. In effect, each entry in RecentDocs will have another entry in its corresponding extension subkey. This is shown in Figure 6.1:
6.6 – The Windows Search Indexer

Vista has uses the Search Indexer to speed up the results of user searches. By default, Vista indexes the contents of the Users folder and the Start Menu folder. The activity of the search indexer process (SearchIndexer.exe) was monitored using Process Monitor [4]. The process performed writes to files located in the following folder (and its subfolders):

C:\ProgramData\Microsoft\Search\Data\Applications\Windows\Projects\SystemIndex\n
Of the files examined, only one appeared to contain any useful information. The file is contained in the folder above and has a filename in the format SystemIndex.NtfyX.gthr where the $x$ is replaced by a decimal number. When viewed with a hex editor, the file contains a 67-byte file header which shows the file is a “Microsoft Search Gatherer Transaction Log. Format Version 4.9”. After this header, the log file is then split up into entries. Each entry appears to contain details of files and folders that have passed through the indexer. During testing with the log file, all entries were regarding files or folders from indexed locations. Experimentation showed that a new entry was often appended to the end of the log file whenever folders or files were opened. This held true for the majority of files and folders used in indexed locations. No documentation for the log format could be found online. Examination of sample log files revealed that each entry is split into three blocks. These blocks take the following format:

![Figure 6.2 - Format of Entries in Gatherer Log File](image)

Figure 6.2 - Format of Entries in Gatherer Log File
Each entry starts with a 48-byte header block. The header block contains the following information:

- **<HEADER_ID>** – Header block identifier 0x4D444D44 (4 bytes)
- **UNKNOWN** - (12 bytes)
- **<TIME_1>** – Windows FILETIME Entry (8 bytes)
- **<TIME_2>** – Windows FILETIME Entry (8 bytes)
- **<CONST_1>** – Constant 0x0200000000000000 (8 bytes)
- **UNKNOWN** - (8 bytes)

After the header, there is a block holding the path and name of the file/folder the log entry was created for (note this block is not fixed-length):

- **<FILENAME>** – The name and full path of the file/folder (varying size)

After the filename block, there is an 89-byte footer block containing the following information:

- **<FOOTER_ID>** - Footer block identifier 0x000000000000000000FFFFFFFF (4 bytes)
- **<TIME_3>** – Windows FILETIME Entry (8 bytes)
- **<CONST_2>** – Constant 0xFFFFFFFF (4 bytes)
- **<TIME_4>** – Windows FILETIME Entry (8 bytes)
- **UNKNOWN** - (20 bytes)
- **<TIME_5>** – Windows FILETIME Entry (8 bytes)
- **<TIME_6>** – Windows FILETIME Entry (8 bytes)
- **UNKNOWN** - (20 bytes)

To test how entries go into the file, a mixture of file and folder operations were performed (created/opened/modified/closed). The order in which the files/folders were used as well...
as the approximate time was recorded. The majority of files used in testing had new entries added to the log file (often multiple entries) and the time fields in the header and footer blocks for the entries reflected the approximate time each file was used in testing. In the test cases, the fields `<TIME_1>` and `<TIME_2>` always contained valid Windows FILETIME entries. The remaining time fields were populated with valid filetimes in the majority of test cases, although in some instances the fields contained either `0x0000000000000000` or `0x0100000000000000`.

Inspection of this file may give examiners an idea of an offender's file activity and approximate times for when files were accessed. Due to time constraints, no additional testing could be performed on the log file.

### 6.7 – SuperFetch

The Windows XP Prefetcher was used to speed up boot and application launch times. Vista has introduced SuperFetch which takes this idea further by analysing usage patterns over a period of time. As a user uses Vista, SuperFetch builds a history of their usage pattern and keeps track of information such as the foreground application being used, time of day, and day of the week [46]. This allows Vista to derive the optimal memory content for a user based on their usage history and attempts to pro-actively populate the memory with this content. While XP was only used for speeding up boot and application launch times, SuperFetch takes a more broad approach and will also monitor what documents a user frequently accesses [47].

SuperFetch stores its files in the `C:\Windows\Prefetch` folder. The folder contains a number of prefetch files with a `.pf` extension. The filenames of the prefetch files are composed of [48]:

- The name of the application
- A dash
- An eight character hash of the location from which the application was run
- The `.pf` extension
Whenever a new application is executed for the first time, a corresponding .pf file will be created in the prefetch folder. Any time thereafter that same program is executed, the modified date for the .pf file is updated. If the application is then uninstalled, the corresponding prefetch entry still remains in the prefetch folder. This is useful to an investigator as it lets them infer:

1. What applications have been executed on the computer (by checking the filenames of the prefetch files)

2. When the application was first executed (the created date of the corresponding prefetch file)

3. When the application was last executed (the modified date of the corresponding prefetch file)

In addition to holding the .pf prefetch files, the prefetch folder also contains a number of usage scenario files [49] of the form Ag<X>.db (where <X> can be a number of strings depending on the system). Vista installations seem create the following scenario files by default: AgAppLaunch.db, AgGlFaultHistory.db, AgGlFgAppHistory.db, AgGlGlobalHistory.db, and AgRobust.db. Each user account on the system then appears to have its own scenario files created which take the form AgGlUAD_P_<SID>.db and AgGlUAD_<SID>.db where <SID> is the user's Security Identifier. The format of these files is unknown and there is currently no Microsoft documentation that gives any clues to describing them. If these files contain the usage patterns for a user, they could prove to be useful in investigations. Further investigation would however be needed.
Chapter 7

Backup Features in Vista

This chapter describes how Vista uses shadow copies to allow users to restore previous versions of files and the positive impact this may have on investigations. It then gives a brief outline of some of the other backup features in Vista.

7.1 – Restore Previous Versions

Restore Previous Versions is available in the Business, Enterprise, and Ultimate editions of Windows Vista [50]. When an inadvertent change or deletion of a file or folder occurs, a user can use Restore Previous Versions to revert the file or folder to a previous state. It works by using the Volume Shadow Copy Service (VSS) to take snapshots of changes made to files (up to 64 versions of the file can be stored in the snapshot database [51]). VSS uses a copy-on-write (differential copy) method to maintain the backups used for previous versions [12]. Instead of having to store a complete copy of a file every time it is changed, VSS tracks only the changes made. This allows VSS to create shadow copies of files quickly as it only has to write the changes to disk [52].

Shadow copies are enabled by default in Vista and are automatically created for the system volume (all files are monitored), although other volumes can be enabled using the Backup and Restore Centre in the Control Panel. The user can access previous versions by right-clicking on a file or folder and selecting the “Restore previous versions” option. This then brings up a dialog box showing any previous versions of the file. Figure 7.1 on page 47 shows the previous versions for a folder “Research Material”. Each previous version of the folder contents will have the date the snapshot was taken displayed next to it and contains previous versions of the entire contents of the folder. The user can browse a read-only version of the contents of any previous version by using the “Open” button. Similarly, the “Copy” button allows the user to specify a destination to store a copy of the selected previous version and the “Restore” button restores the previous version and replaces the current version of the folder.
Vista uses 15% of the available disk space on a volume to store shadow copies, although this can be adjusted using the C:\Windows\System32\vssadmin.exe tool. Once the 15% of reserved space has been used up, older shadow copies are deleted to make space [53]. The data for the shadow copies is stored in the System Volume Information folder. Researchers from Cranfield University conducted experimentation with the shadow copy container files [14] in the hope of creating a tool to extract the shadow copies. While this tool is still in development, they have reported success in viewing shadow copies of a suspect system by booting an image of the system using VMware and the methods described in [54].

As shadow copies are made by default on Vista machines, they may prove to be of great evidentiary value. If the investigator believes certain files/folders were present on a drive but does not know their previous location, they can simply use “restore previous versions” on the entire suspect volume. This will then show any shadow copies that have been created of the volume and the investigator can navigate through the older version of the filesystem and even perform searches.
It was mentioned at the beginning of this section that “restore previous versions” is only available in the Business, Enterprise, and Ultimate editions of Vista. However, users of the Home Basic and Home Premium versions have the option of using the Windows Anytime Upgrade tool to upgrade to Vista Ultimate. Users who have chosen this upgrade path have found that once they have upgraded, Ultimate will let them restore previous files from snapshots taken while they were running their Home Basic or Home Premium installations [55]. While these editions of Vista do not provide the front-end for restoring previous versions of files, all of the work of taking snapshots of the volume is still happening in the background.

For this report, a number of test cases were set up in order to evaluate Vista’s ability at restoring files. It was found that Vista can restore previous versions of files and folders even if the files have been deleted and the recycle bin emptied. In addition, SDelete v1.51 [56] by Mark Russinovich and Eraser v5.82 [57] were used to “securely” erase a number of files from the Vista test machine. SDelete was used to perform a 20-pass overwrite of one set of test files and Eraser was used to perform a 35-pass Gutmann [58] method overwrite of another set of test files. Both sets of test files were then successfully recovered using “restore previous versions”.

Investigators can use the vssadmin tool (C:\Windows\System32\vssadmin.exe) to obtain information about the shadow copies for a volume. One of the most useful commands is vssadmin list shadows which shows all volume snapshots on a system and displays how far back the snapshots for each volume go. For full details see [59].

While the current investigation technique of restoring previous versions using a virtual machine is not ideal, it does provide examiners with a valuable source of evidence which would not be available in Windows XP. If the forensic research community manages to produce a tool that can restore previous versions of files without booting the suspect image on a virtual machine, evidence of this nature may open investigative avenues in cases which would have otherwise produced no results.
7.2 – Other Vista Backup Options

The Business, Enterprise, and Ultimate editions of Vista allow users to access to Windows Complete PC Backup. This creates a complete block-level copy of the volume which the user can later use to restore their entire system. Backups are saved in virtual hard drive (.vhd) format and can either be saved to another hard disk or split between multiple DVDs. These .vhd images can later be used to restore a configuration and can also be mounted in Virtual PC 2007 [60] or be converted for use in VMWare. Investigators should also be aware that all versions of Vista (except for Home Basic) allow users to perform scheduled file backups to network locations. Although most previous versions of Windows allowed a user to manually backup files to a networked location, the automation of the process may lead to an increase in the number of home users transferring files to network devices.
Chapter 8

Windows Mail

This chapter discusses the Windows Mail email and newsgroup client for Vista. It starts by giving a brief overview of Windows Mail and its predecessor Outlook Express. The chapter then details the methods used for storing email accounts, email files, newsgroup accounts, and newsgroup posts.

8.1 – Windows Mail and Outlook Express

Windows Mail is an email and newsgroup client included in all versions of Vista. It is very similar in both appearance and functionality to Outlook Express which was included in previous versions of Windows. Each mail folder in Outlook Express had a corresponding .dbx that held all of the messages for that folder. Forensic analysis of these files typically involved the use of third-party tools to extract the individual email files, or simply importing the .dbx files to another Outlook Express configuration. Windows Mail has discarded the .dbx container format and now simply stores email messages as individual files.

8.2 – Windows Mail Files and Folders

Each email account added to Windows Mail will have its own account folder created. These folders are created in C:\Users\<username>\AppData\Local\Microsoft\Windows Mail\ by default (although this location can be changed via the registry). By default, Vista creates a Local Folders account folder which holds the messages and account information for any POP3 accounts [61]. Each non-POP3 email account the user adds will have its own folder created. The default naming convention is to name the folder after the incoming mail server (e.g. imaphost.cis.strath.ac.uk) although the user can change this later. Each account folder will then contain the following:

- A .oeaccount file which contains the account property data
8.2.1 – Account Property Data (.oeaccount) Files

Account details are held in XML format and are stored in filenames with the format account{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}.oeaccount where each X is a hexadecimal digit. A comprehensive list of the data fields contained in the account property files is available from reference [61]. Investigators should be aware that there are three account data files set up by default on a Vista system. The first two are used for Active Directory and the VeriSign Internet Directory Service. They are located in C:\Users\<username>\AppData\Local\Microsoft\Windows Mail\ and have sizes of 1,736 bytes and 1,508 bytes respectively. The third file is 672 bytes in length and is located in C:\Users\<username>\AppData\Local\Microsoft\Windows Mail\Microsoft Communities\.

8.2.2 – Mail Folders

Each email folder contains a winmail.fol file which tracks the folder location and the email items in the folder [62]. Any other files present in the folder will be the constituent .eml email files.

8.2.3 – Email (.eml) Files

Email filenames take the format XXXXXXXX-XXXXXXXX.eml where each X is a hexadecimal digit. The email data is stored as plaintext and can be viewed with any text editor. The .eml files contain the full header of the email, the body of the message, and any attachments. Attachments can either be extracted from the .eml files or a forensic workstation running Windows Mail can be used to open the .eml file and view the attachments. Investigators should be aware that Windows Mail supports encryption of outgoing mail using digital signatures [63]. Email messages which are sent with encryption enabled are encrypted before being placed in the Outbox therefore the body of the email will be unreadable but the email header is still in plaintext [11].
8.3 – Windows Contacts

Contacts for a user are contained in the C:\Users\<username>\Contacts\ folder. Contact files take the filename format of <contact name>.contact (where <contact name> is the contact’s full name) and are simple XML files containing information about the contact. The full list of properties that can be added to a contact file are listed in [64]. The Contacts folder is also used to store the list of contacts for any Windows Live accounts that have been accessed from the user’s account (e.g. using Windows Live Messenger). The contacts for a Windows Live account are placed into a folder named after the account. Windows Live Messenger encrypts the contacts by default and stores them in files with the filename format XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX.WindowsLiveContact where each X is a hexadecimal digit. The folder for the Windows Live account will also contain a file named contactcoll.cache. This is a cache of all the contacts associated with the account but is also encrypted by default. These files can be unencrypted by disabling encryption of contacts via the options in Windows Live Messenger. During testing it was observed that even if encryption is turned back on, plaintext copies of the contact names are available from the contactcoll.cache file. Any further contacts added when encryption is re-enabled are appended as encrypted entries to contactcoll.cache while the older contacts remain in plaintext.

8.4 – Windows Mail Log File

Windows Mail maintains a log file relating to sent and received email. It is contained in the file C:\Users\<username>\AppData\Local\Microsoft\Windows Mail\edb.log. By opening the file with a hex editor, email headers for a number of sent and received emails were clearly visible. A test set of 24 emails was used to send and receive email to/from an account. Of the test set, header information for 20 of the emails was retrieved from the log file including subject, date send, sender, and recipient. Examiners should note that outgoing email logged in the file will reference the sender by the .oeaccount filename for the account, rather than the email address. The information in this log file persists even after users clear all email folders through Windows Mail (Tools->Options->Advanced Tab-
>'Maintenance'->'Clean Up Now'->Reset). Testing showed that (as expected) 'deleted' email can also be recovered using Restore Previous Versions (see Section 7.1).

8.5 – Newsgroups

The storage system for newsgroup accounts is much the same as for email accounts. They reside in C:\Users\<username>\AppData\Local\Microsoft\Windows Mail\ as newsgroup account folders named after the news server (e.g. nntphost.cis.strath.ac.uk). The newsgroup account folders then contain an .oeaccount file for holding the account properties. They will also contain a number of subfolders which are named after any newsgroups which have been subscribed to under that newsgroup account. These, in turn, contain a winmail.fol file (discussed earlier) and individual files for each post which take the filename format XXXXXXXX-XXXXXXXX.nws where each X is a hexadecimal digit. The header and body information is stored in plaintext and can be viewed with a text editor.
Chapter 9

Internet Explorer 7 for Vista

This chapter provides investigators with an introduction to Internet Explorer 7 (IE7) for Vista and how it differs from its XP counterpart. It then discusses the new folders used in IE7 and the updated locations for the index.dat files. The chapter finishes by describing password storage and the new features for deleting browsing history.

9.1 – Introductory Note

Internet Explorer 7 is the web browser included with Windows Vista. While investigators may have come across IE7 when examining Windows XP machines, they should realise that the version included in Vista is significantly different. The two are built from the same code base [65] and look practically identical, but the added security features in IE7 for Vista make it considerably different from an investigative point of view.

9.2 – Protected Mode for IE7

All versions of Windows Vista come with the Internet Explorer 7 (IE7) web browser. Microsoft is well aware that there is a higher security risk associated with web browsers than there is with the majority of Windows applications available to end-users. For this reason, IE7 for Vista uses a defence-in-depth approach to security known as Protected Mode, with multiple layers of defence placed throughout the system. Protected Mode utilises three new technologies from the Windows Vista security model [66]:

- **User Account Control (UAC)** – A least privilege to access control where all users on the system run as a standard user. If a task requires higher privileges, the user is prompted with an elevation request that gives temporary rights to perform the task on the condition that the user’s account has the necessary credentials (see Chapter 2 for full details).
• **Mandatory Integrity Control (MIC)** – A model to prevent lower integrity processes from performing writes or deletes Windows securable objects [67] such as files or registry keys. Vista uses four primary integrity levels: Low, Medium, High, and System. Processes are assigned an integrity level and can only access securable objects that have an equal or lower integrity level.

• **User Interface Privilege Isolation (UIPI)** – Used to block lower integrity processes from accessing higher integrity processes.

IE7 for Vista runs in Protected Mode by default, during which time the Internet Explorer process (ieexplore.exe) runs as a low integrity process. Any add-ins available for IE7 such as toolbars and ActiveX controls all run within the IE process and therefore also run as Low integrity [68]. As a result, IE7 and its add-ins running in Protected Mode can only write to low integrity locations. Therefore, instead of IE7 using single folders for each of cache, cookies, history, and temp files, a pair of folders is used. Each has a folder for when IE7 is running outside of Protected Mode which, in turn, holds a Low subfolder to hold content for when IE7 is running in Protected Mode. The locations of these folders are shown below:

• **Cache:**
  C:\Users\<username>\AppData\Local\Microsoft\Windows\Temporary Internet Files\n  C:\Users\<username>\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low

• **Cookies:**
  C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Cookies\n  C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Cookies\Low

• **History:**
  C:\Users\<username>\AppData\Local\Microsoft\Windows\History\n  C:\Users\<username>\AppData\Local\Microsoft\Windows\History\Low

• **Temp:**
  C:\Users\<username>\AppData\Local\Temp\n  C:\Users\<username>\AppData\Local\Temp\Low
Another consequence of running in Protected Mode is that Internet Explorer extensions attempting to write to system locations such as Program Files and the HKCU registry hive are automatically redirected to the following low integrity locations [69]:

- **Virtualized folder:**
  C:\Users\<username>\AppData\Local\Microsoft\Windows\Temporary Internet Files\Virtualized

- **Virtualized registry key:**
  HKCU\Software\Microsoft\Internet Explorer\InternetRegistry

The favorites folder does not require a low integrity version. It resides in
C:\Users\<username>\Favorites.

### 9.3– Internet Explorer 7 Cache and index.dat Files

IE7 appears to cache web content in the same manner as Internet Explorer 6. Temporary Internet Files contains a Content.IE5 folder which, in turn, contains folders of the format `xxxxxxxxx` where each `x` is an alphanumeric character (note: remember that there will also be the Low version of the cache). These folders then contain any web content that has been cached by IE7.

IE7 also uses the same format for index.dat files and they are still readable using applications that read IE6 index.dat files. The following is a list of index.dat file locations on a typical Vista machine and a brief description of each (descriptions for each index file are based on information from [70]):

- **C:\Users\<username>\AppData\Local\Microsoft\Feeds Cache**
  - Holds an index of RSS feeds added to IE7.

- **C:\Users\<username>\AppData\Local\Microsoft\Windows\History\History.IE5**
  - Contains a list of the URLs that have been clicked when browsing. Allows IE7 to autocomplete URLs and mark links which have been visited in a different colour.
As above except holds data when IE7 is in Protected Mode.

These are history containers used for the different range dates that IE7 displays (e.g. today, yesterday, last week etc). The first <date> field specifies the start date for that particular history container and the second <date> filed specifies the end date.

As above except holds data when IE7 is in Protected Mode.

URLs of web content such as pages, images, and JavaScript are cacheable are placed into this index file until they expire.

As above except holds data when IE7 is in Protected Mode.

Holds URLs for userData entries which are essentially sophisticated cookies. See [71] for more details of userData objects. (Note: This folder and its low-integrity counterpart will only be created when IE7 first encounters a web page utilising userData objects).

As above except holds data when IE7 is in Protected Mode.
C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Cookies

- Used to map cookie URLs to individual cookie files.

C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Cookies\Low

- As above except holds data when IE7 is in Protected Mode.

9.4– Passwords

Internet Explorer 7 includes password management facilities in order to reduce the burden of trying to remember numerous passwords. In addition to helping the user, these can also be valuable for investigators. Comprehensive details of the storage mechanisms used in IE7 are available from [8].

9.4.1– Internet Credentials

The user will be prompted for their internet credentials password whenever they attempt to access certain websites that require the wininet.dll library [72]. This will automatically launch the Credential Manager User Interface (see Figure 9.1) which prompts the user for their username and password. The password will be saved in an encrypted file under C:\Users\<username>\AppData\Roaming\Microsoft\Credentials (assuming “Remember my password” is ticked).

Figure 9.1 - Prompt for Internet Credentials
9.4.2– AutoComplete Information

The AutoComplete of Internet Explorer is intended to save users time by recording information the user types to the browser such as website addresses, form information (including username/password combinations), and search queries. Previous versions of Internet Explorer encrypted AutoComplete data by using the Windows Protected Storage API (PStore) [73]. The data is encrypted using the Triple DES algorithm from the CryptoAPI [74]. Although this algorithm is considered to be secure, access to the encrypted data is tied to a user's Windows logon credential [75]. In a typical system, this is provided by calculating the hash of the user's password. Once the user is logged in, any program can access the unencrypted data by using the PStore API. This poses a serious security risk and so Microsoft has opted to now store encrypted AutoComplete data using the Data Protection API (DPAPI) [76].

DPAPI provides software developers with OS-level data protection services. This allows developers to create applications that can secure data by simply using function calls to DPAPI rather than having to write their own application-specific cryptographic code. As with PStore, the password used in the encryption/decryption process is the user’s logon credential. However, DPAPI allows applications to use an additional password when protecting data, thereby overcoming the vulnerabilities of PStore by hindering the ability of one application to compromise another application’s encryption key.

All of the data stored for AutoComplete fields (i.e. AutoComplete strings for data forms) is stored in the HKCU\Software\Microsoft\Internet Explorer\IntelliForms\Storage1 registry key. It is held as an encrypted list of HTML form field names and the corresponding data the user entered in that field. Unfortunately, the name or URL of the Web page is not cached. For example, if a user visits http://www.google.co.uk/ and types the search query “TEST1”, Internet Explorer will only cache the AutoComplete entry “TEST1” (assuming the corresponding AutoComplete option is enabled) as well as the form field name “q” (as this is the form field name assigned to the input box on the Google homepage).
AutoComplete for storing website passwords works in a similar fashion as it does for fields. The main difference is that each password is linked to a specific webpage (whereas AutoComplete data for fields is just linked to a form field name). The encrypted data is held in the `HKCU\Software\Microsoft\Internet Explorer\IntelliForms\Storage2` registry key. The encryption and decryption process for AutoComplete passwords is as follows [72]:

**Encryption:**

1. Save the URL of the webpage. This is then used as the encryption key:
   
   ```plaintext
   EncryptionKey = URL
   ```

2. Calculate the hash of the encryption key and store this as the record key:
   
   ```plaintext
   RecordKey = SHA(EncryptionKey)
   ```

3. Calculate the checksum for `RecordKey` to ensure integrity:
   
   ```plaintext
   RecordKeyCRC = CRC(RecordKey)
   ```

4. Encrypt the data (passwords) using the encryption key:
   
   ```plaintext
   EncryptedData = DPAPI_Encrypt(Data, EncryptionKey)
   ```

5. Save `RecordKeyCRC`, `RecordKey`, and `EncryptedData` in the registry.

6. Discard `EncryptionKey`.

**Decryption:**

1. When the original page is open, use its URL (should match original `EncryptionKey`) to calculate the record key:
   
   ```plaintext
   RecordKey = SHA(EncryptionKey)
   ```

2. Scan through the list of record keys in the registry trying to find `RecordKey`.

3. If `RecordKey` is found in the registry list, decrypt the data stored alongside it using `EncryptionKey`:
   
   ```plaintext
   Data = DPAPIDecrypt(EncryptedData, EncryptionKey)
   ```

From the decryption process outlined above, it should be obvious that without the appropriate URL, a password stored in the `Storage2` registry key cannot be decrypted.
Software that automates the decryption of these registry keys (such as Passcape Internet Explorer Password Recovery [77]) works by using the URL that is stored in the browser history. If the history has been cleared, the only way an examiner can retrieve the password is by manually trying URLs in the hope that they find the correct one to unlock the data.

9.5 – Deleting Browsing History

By default, Internet Explorer 7 uses 50mb for the disk cache and keeps visited pages in the history for 20 days. However, it has now made the process of deleting the browser history much easier. The user now uses the “Delete Browsing History” option from the Tools drop-down menu on the IE7 toolbar. This opens a dialog where the user can either clear individual components of the browsing history (Temporary Internet Files, Cookies, History, Form Data, Passwords) or they have a “Delete All” option. This will delete all the data mentioned above but also now wipes the index.dat files. Instead of simply deleting the old file, it is overwritten with blank spaces. Using a hex editor, the version information for each index file was still visible (“Client UrlCache MMF Ver 5.2”) and the rest of the file was filled with 0x20.
Chapter 10

Additional Points of Interest

This chapter explains a variety of features in Vista that may have significance during an investigation.

10.1 – The Recycle Bin

The XP Recycle Bin was held in the RECYCLER folder at the root of a drive. This held any files that had been deleted and also contained an INFO2 that mapped the files in the RECYCLER folder to their original path (comprehensive coverage of this subject is available from [78]). Vista’s Recycle Bin is held in the $Recycle.Bin folder at the root of a drive. This folder then contains subfolders named after the SID of the user who deleted the file. These subfolders then contain any files that have been deleted by the user.

Vista no longer uses an INFO2 file to track the original location of files. Instead, when a file or folder is deleted (i.e. moved to the Recycle Bin) two files are created. The first file will be of the form $I<fileID>.<ext> where <fileID> is a unique six-character alphanumeric string. <ext> is the original extension of the file that has been deleted (no extension if a folder was deleted). These $I files contain an 8-byte FILETIME structure at offset 0x10 that holds the time the file/folder was deleted. They then contain the full path and filename of the original file/folder at offset 0x18. Each of these $I entries will have a corresponding file or folder present alongside them. For folders that have been deleted, a folder is created named $R<fileID> where <fileID> will match the <fileID> used in the corresponding $I file. Any contents of the deleted folder are moved to the $R folder and retain their original names. Files that have been moved to the Recycle Bin are renamed $R<fileID>.<ext> where <fileID> and <ext> will match those used in the corresponding $I file. As $R files are just the original files with a different name, they contain all the actual content of the original file.
10.2 – Acquisition of Physical Memory

Acquisition of physical memory in Windows XP involves using a tool such as DD from the Forensic Acquisition Utilities [79]. It relies on taking a dump of the physical memory by using the \Device\PhysicalMemory (see [80]) object to access physical memory from a user-mode application. This is not possible in Vista as access to the \Device\PhysicalMemory object has been restricted to kernel-mode drivers [80]. Physical memory acquisition in Vista may still be possible in systems that support either USB or FireWire. Both the USB and FireWire specifications allow access to physical memory via Direct Memory Access (DMA). This has been successfully demoed using FireWire [81] where access to the physical memory is initiated from the device side. As the access is DMA, the device (e.g. a Linux machine) can directly access the memory on the host machine and circumvent any OS protection in the process [82].

10.3 – ReadyBoost

ReadyBoost is a technology available in Vista that is used for caching disk data. The idea is that an external USB flash device is used to supplement the main memory of a PC and reduce the need to access the hard disk for random I/O reads. This initially seems like an interesting prospect from an investigative point of view. The problem is that Vista encrypts the data written to the USB device with AES-128 using a per-boot encryption key [49]. This effectively makes a meaningful examination of the device impossible.

10.4 – Transactional NTFS and Transactional Registry

Transactional NTFS (TxF) and Transactional Registry (TxR) are two features in Vista which are built on top of the Vista Kernel Transaction Manager (KTM) [83]. The Microsoft definition of TxF is “Transactional NTFS allows an application to group multiple operations on multiple files as a single unit” [84]. Basically it is a method of ensuring a set of file operations occur as required (i.e. as a set) or are rolled back. TxR is a similar concept except it ensures a transaction for a set of registry operations occurs properly. These features will be useful when applied to scenarios such as software upgrades, but at the moment they do not have much significance from a forensic perspective.
10.5 – Disk Defragmenter

The disk defragmenter in Vista works in much the same way as it did in XP (except with a simplified GUI). The main concern for investigators is that, by default, the defragmenter is scheduled to run automatically at 1am every Wednesday on Vista systems. Scheduled defragmentation that is enabled by default has obvious implications with regards to recovery of deleted files. Recovery of files from shadow copies (see Section 7.1) may still be possible although defragmentation is known to affect shadow copies in systems with a cluster size of less than 16KB [85]. Examiners should be aware that the scheduled disk defragmenter operates on all volumes [86].
Chapter 11

Conclusion

This report has attempted to detail some of the issues surrounding the forensic investigation of the Windows Vista platform. Much of the content is the result of the research conducted at QinetiQ. In addition to this content, information gathered from a number of other sources, including the papers mentioned in Section 1.3, has been verified and documented.

Despite initial worry from the forensic community, Windows Vista does not appear to be significantly different to Windows XP in terms of performing a forensic investigation. The BitLocker encryption system will certainly start to be encountered more frequently in live response situations and this may lead to an emphasis on live analysis of a machine. The other major change in Vista that may change part of an investigative strategy is the absence of last access times. Traditional means of creating a profile of file activity based on last access times will therefore be made redundant and examiners will have to utilise other methods of investigation. The other changes documented in the report will simply necessitate a slight re-education on a particular aspect of the operating system or, what is probably the more likely scenario, the tweaking of a particular tool that automates that part of the investigation.

While the last paragraph has focused on some of the challenges created by Vista, there are other points that work in an examiner's favour. For instance, the moving of the thumbnail cache to a centralised location should lead to an increase in the amount of evidence that can be accumulated from thumbnails. The major new feature that works in an examiner's favour is the Volume Shadow Copy material that resides on Vista volumes. This could prove to be one of the most significant features used in an investigation of a Vista Machine.

There are a number of points covered in the report but it is by no means an exhaustive look at the forensic issues surrounding Windows Vista. Some of the points, in particular BitLocker and Volume Shadow Copy, would merit an entire report alone. Due to time
constraints, an exhaustive look was not possible. Instead, the report has shed some new light on Vista and attempted to collate information scattered amongst a number of other sources. Hopefully the information can be put to use in a working forensic lab and perhaps to contribute to further research.
Chapter 12

Suggestions for Further Work

As explained in the conclusion, there is a lot of research still to be conducted on Windows Vista. There are already attempts being made at circumventing parts of the BitLocker encryption (see Section 4.13) and it will be interesting to see how these develop. With the current situation of BitLocker never having been successfully attacked, an emphasis will be placed on live response of systems running BitLocker. Live analysis of machines has always been a point of contention in digital forensics. However, as disk encryption becomes more prevalent the issue of live analysis becomes more important as it may well be the only opportunity to gather evidence from a system. The forensic community needs to collectively work towards an agreed strategy to be used in live system analysis scenarios as attempts to break the encryption systems may ultimately prove futile.

The development of a tool that could restore previous versions of files using shadow copies would be a major coup for the forensic community. This would open up a wealth of information residing on a volume that could be of great evidentiary value. The current technique involves running live systems and only works with certain editions of Vista (see Section 7.1) which is far from ideal.

Another major limitation with Windows Vista is the inability to take a dump of the physical memory using traditional tools (see Section 10.2). The hardware requirements of Vista will be likely to force many users into upgrading their computers which may lead to a rise in the number of FireWire-enabled computers encountered during live incident response. This provides a viable avenue for the acquisition of physical memory although this cannot be relied on as the sole method of doing so. Acquisition of memory based on a software approach would be more desirable in the sense that it could be more widely utilised. Even better would be a reliable DMA-based method using USB as this would require no in-memory artifact.
Research involving the forensic investigation of Vista systems is still in its infancy. However, the forensic research community has a loyal and dedicated core which should, in time, overcome the majority of issues facing them regarding the examination of Vista.
Appendix A

References

This section contains the references used throughout the report.

   (http://www.vmware.com/products/ws/)

   (http://www.x-ways.net/winhex/)

   (http://notepad-plus.sourceforge.net/uk/about.php)

   01-09-2007)  
   (http://www.microsoft.com/technet/sysinternals/utilities/processmonitor.mspx)

   (http://regshot.blog.googlepages.com/regshot)


   (http://www.securityfocus.com/infocus/1889/2)

8. Jamie Morris, "Notes on Vista Forensics, Part Two". *Security Focus Article.* April  
   (http://www.securityfocus.com/infocus/1890)

9. Website: Forensic Focus. (Accessed 01-09-2007)  
   (http://www.forensicfocus.com/)


24. Microsoft, “Error message when you try to start a Windows Vista-based computer that is configured to use BitLocker: “The PIN has been entered incorrectly too many times”. Microsoft KB Article. November 2006. (Accessed 01-09-2007) (http://support.microsoft.com/kb/926187)


34. Website: NV Labs. (Accessed 01-09-2007) (http://www.nvlabs.in/)


37. Website: “Thumbs.db in imaging systems”. Alliance Group UK. (Accessed 01-09-2007) (http://www.alliancegroup.co.uk/thumbs.htm)


(http://msdn2.microsoft.com/en-us/library/ms724290.aspx)

(http://blogs.msdn.com/embedded/archive/2006/02/01/522053.aspx)

(http://blogs.technet.com/filecab/archive/2006/11/07/disabling‐last‐access‐time‐in‐windows‐vista‐to‐improve‐ntfs‐performance.aspx)

(http://www.geocities.com/TimesSquare/Maze/1125/articles/explorer_spy.txt)

45. Software: UserAssist v2.3.0. Didier Stevens. (Accessed 01-09-2007)  
(http://blog.didierstevens.com/programs/userassist/)

(http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/perfaccel.doc)

(http://windowsir.blogspot.com/2005/03/what‐is‐prefetcher.html)

(http://www.microsoft.com/technet/technetmag/issues/2007/03/VistaKernel/)

(http://windowshelp.microsoft.com/Windows/en-US/Help/afbd89-ed63-4e07-a482-d7004a8f40121033.mspx)


(http://technet2.microsoft.com/WindowsServer/en/library/2b0d2457-b7d8-42c3-b6c9-59c145b776f1033.mspx)

(http://blogs.technet.com/filecab/archive/2007/03/03/why‐do‐restore‐points‐in‐windows‐vista‐use‐so‐much‐disk‐space.aspx)


    (Accessed 01-09-2007)
    (http://www.passcape.com/internet_explorer_passwords.htm)
    (http://www.libdirectory.idsc.gov.eg/cas/articles/computer&internet/computer2.pdf)
    (http://www.securityfocus.com/infocus/1882/2)
    (Accessed 01-09-2007)
77. Software: “Passcape Internet Explorer Password Recovery (beta 7) v2.4.0.237”. Passcape Software. (Accessed 01-09-2007)
    (http://www.passcape.com/internet_explorer_password_recovery.htm)
    (http://www.e-fense.com/helix/Docs/Recycler_Bin_Record_Reconstruction.pdf)
    (http://www.gmgsystemsinc.com/fau/)
81. M. Becher, M. Dornseif, C. Klein, “FireWire – All Your Memory Are Belong To Us”. 
*CanSecWest Conference 2005*. (Accessed 01-09-2007) 
(http://www.cansecwest.com/core05/2005-firewire-cansecwest.pdf)

(http://www.wolthusen.com/publications/ISTR2006.pdf)

(Accessed 01-09-2007) 


(http://support.microsoft.com/kb/312067)

(Accessed 01-09-2007) 


## Appendix B

### Features Comparison

This section shows a table which documents the features available in each edition of Windows Vista (Table reproduced from [87]).

<table>
<thead>
<tr>
<th>Key Features for All Users</th>
<th>Home Basic</th>
<th>Home Premium</th>
<th>Business</th>
<th>Enterprise</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome Center</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>User Account Control</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Security Center</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Defender</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Firewall</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Explorer 7 Protected Mode</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Explorer 7 Fix My Settings</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Anti-phishing in Internet Explorer 7</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Anti-phishing in Windows Mail</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Update</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Service Hardening</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Performance self-tuning and hardware diagnostics</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Experience Index</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Next-generation TCP/IP stack</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv6 and IPv4 support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows ReadyDrive</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Display Driver Model (WDDM)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Ad hoc backup and recovery of user files and folders</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scheduled backup of user files</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Backup of user files to a networked PC or device</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Home Basic</td>
<td>Home Premium</td>
<td>Business</td>
<td>Enterprise</td>
<td>Ultimate</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------</td>
<td>--------------</td>
<td>----------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>Incremental backup</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Automatic backup scheduling</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Complete PC Backup and Restore (image based)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Shadow Copy (restore previous versions of your documents)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Vista Basic user interface</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Aero user experience (glass, dynamic windows, and a smoother-performing desktop)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Instant Search</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>File tagging</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Stacking and Group By View.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Explorer 7 with RSS feed support, tabbed browsing, and integrated search</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>.NET Framework 3.0</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows CardSpace</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows SuperFetch</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows ReadyBoost</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>I/O prioritization</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Automatic hard disk defragmentation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>64-bit processor support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Maximum RAM supported with 32-bit system</td>
<td>4 GB</td>
<td>4 GB</td>
<td>4 GB</td>
<td>4 GB</td>
<td>4 GB</td>
</tr>
<tr>
<td>Maximum RAM supported with 64-bit system</td>
<td>8 GB</td>
<td>16 GB</td>
<td>128+ GB</td>
<td>128+ GB</td>
<td>128+ GB</td>
</tr>
<tr>
<td>Two processors (two processor sockets) support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Years of product support</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

**Key Features for Home Users**

<table>
<thead>
<tr>
<th></th>
<th>Home Basic</th>
<th>Home Premium</th>
<th>Business</th>
<th>Enterprise</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Mail</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Calendar</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Sidebar</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Photo Gallery—for organizing, editing, printing, and sharing photos and videos</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Home Basic</td>
<td>Home Premium</td>
<td>Business</td>
<td>Enterprise</td>
<td>Ultimate</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------------</td>
<td>--------------</td>
<td>----------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>Themed slide shows</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Media Player 11</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Native DVD playback</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Media Center—for music, photos, videos, live and recorded TV, and online entertainment</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Media Center—for recording and watching high-definition TV (U.S. and South Korea only)</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Media Center—CableCard support (U.S. only)</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Support for Media Center Extenders, including Xbox 360</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Movie Maker</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Movie Maker HD</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Windows DVD Maker (Video DVD Authoring and Burning)</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Games Explorer</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Updated games</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>New premium games</td>
<td>Yes</td>
<td></td>
<td>Yes*</td>
<td>Yes*</td>
<td>Yes</td>
</tr>
<tr>
<td>Parental Controls</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes*</td>
<td>Yes</td>
</tr>
<tr>
<td>Universal game controller support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes*</td>
<td>Yes*</td>
<td>Yes</td>
</tr>
<tr>
<td>Speech Recognition</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Accessibility Settings and Ease of Access Center</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Key Features for Business Users**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Home Basic</th>
<th>Home Premium</th>
<th>Business</th>
<th>Enterprise</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPS Document support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Small Business Resources</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes*</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Fax and Scan</td>
<td>Yes</td>
<td></td>
<td>Yes*</td>
<td>Yes*</td>
<td>Yes</td>
</tr>
<tr>
<td>Network and Sharing Center</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Network Diagnostics and troubleshooting</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Improved wireless networking</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Wireless network provisioning</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Home Basic</td>
<td>Home Premium</td>
<td>Business</td>
<td>Enterprise</td>
<td>Ultimate</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------------</td>
<td>--------------</td>
<td>----------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>Improved peer networking</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Improved VPN support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Improved power management</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Simultaneous SMB peer network connections</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Windows HotStart</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Mobility Center</td>
<td>Partial</td>
<td>Partial</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Sync Center</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offline Folder support</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Tablet PC with integrated pen/digital ink input</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Tablet PC touch screen support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Tablet PC handwriting recognition improvements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Tablet PC usability and navigation improvements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows SideShow</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Meeting Space</td>
<td>View only</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Improved file and folder sharing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Ad hoc backup and recovery of user files and folders</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scheduled backup of user files</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Backup of user files to a network device</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Key Features for IT Professionals</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System image–based backup and recovery</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Encrypting File System</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Desktop deployment tools for managed networks</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy-based quality of service for networking</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Rights Management Services (RMS) Client</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Control over installation of device drivers</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Network Access Protection Client Agent</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Home Basic</td>
<td>Home Premium</td>
<td>Business</td>
<td>Enterprise</td>
<td>Ultimate</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>------------</td>
<td>--------------</td>
<td>----------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>Pluggable logon authentication architecture</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows BitLocker Drive Encryption</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Support for simultaneous installation of multiple user interface languages</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>All worldwide user interface languages (36 languages total) available</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Subsystem for UNIX-based applications</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Application Compatibility features</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>File-based image format (WIM)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Appendix C
Registry Keys

This chapter provides the locations of the registry hives in Vista. It then details a number of registry keys which are of interest to an examiner. The lists are based on the “Registry Quick Find Chart” by AccessData [88]. Entries in the list were examined in a Vista system and to check for any changes. Other registry entries discovered during the course of the project research have also been added.

C.1 – Registry Hive Locations

Table C.1 shows a list of registry hives in Vista and the corresponding file locations. Full details of each hive are available from [89]. A number of other hives are also present on a default Vista configuration [11] but appear to hold no major significance from an investigative standpoint. Backups of the registry hives are now stored in \C:\Windows\system32\config\RegBack.

<table>
<thead>
<tr>
<th>Registry Hive</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKEY_LOCAL_MACHINE\SAM</td>
<td>C:\Windows\system32\config\SAM</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SECURITY</td>
<td>C:\Windows\system32\config\SECURITY</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SOFTWARE</td>
<td>C:\Windows\system32\config\SOFTWARE</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SYSTEM</td>
<td>C:\Windows\system32\config\SYSTEM</td>
</tr>
<tr>
<td>HKEY_USERS.DEFAULT</td>
<td>C:\Windows\system32\config\DEFAULT</td>
</tr>
<tr>
<td>HKEY_USERS&lt;user_SID&gt;</td>
<td>C:\Users&lt;username&gt;\NTUSER.DAT</td>
</tr>
</tbody>
</table>

Table C.1 - Registry Hives of Interest

C.2 – Internet Explorer

<table>
<thead>
<tr>
<th>Information</th>
<th>File</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
</table>

- 85 -
<table>
<thead>
<tr>
<th>Information</th>
<th>File</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE7 AutoComplete Passwords</td>
<td>NTUSER.DAT</td>
<td>\Software\Microsoft\Internet Explorer\IntelliForms\Storage2</td>
<td>Encrypted list of AutoComplete passwords (see Section XX)</td>
</tr>
<tr>
<td>IE7 AutoComplete Strings</td>
<td>NTUSER.DAT</td>
<td>\Software\Microsoft\Internet Explorer\IntelliForms\Storage1</td>
<td>Encrypted list of AutoComplete strings (see Section XX)</td>
</tr>
<tr>
<td>IE URL History – Days Saved</td>
<td>SOFTWARE</td>
<td>\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Url History - DaysToKeep</td>
<td>The number of days IE7 stores visited URLs</td>
</tr>
<tr>
<td>Default Download Directory</td>
<td>NTUSER.DAT</td>
<td>\Software\Microsoft\Internet Explorer - Download Directory</td>
<td>The default download directory</td>
</tr>
<tr>
<td>IE7 Settings</td>
<td>NTUSER.DAT</td>
<td>\Software\Microsoft\Internet Explorer\Main</td>
<td>List of settings for IE7</td>
</tr>
</tbody>
</table>

Table C.2 - Internet Explorer Keys

**C.3 – System Information**

<table>
<thead>
<tr>
<th>Information</th>
<th>File</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Name</td>
<td>SYSTEM</td>
<td>\ControlSet###\Control\ComputerName\ComputerName</td>
<td>Computer name from system properties</td>
</tr>
<tr>
<td>Current Control Set</td>
<td>SYSTEM</td>
<td>\Select - Current</td>
<td>Identifies the current control set</td>
</tr>
<tr>
<td>Event Logs</td>
<td>SYSTEM</td>
<td>\ControlSet###\Services\Eventlog\</td>
<td>Identifies the location of the event logs</td>
</tr>
<tr>
<td>Install Date</td>
<td>SOFTWARE</td>
<td>\Microsoft\Windows NT\CurrentVersion</td>
<td>Lists the date the operating system was installed</td>
</tr>
<tr>
<td>Product Name</td>
<td>SOFTWARE</td>
<td>\Microsoft\Windows NT\CurrentVersion</td>
<td>Lists the name of the operating system</td>
</tr>
<tr>
<td>Information</td>
<td>File</td>
<td>Location</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Registered Owner</td>
<td>SOFTWARE</td>
<td><code>\Microsoft\Windows NT\CurrentVersion</code></td>
<td>Identifies the registered owner entered during installation (can be modified later)</td>
</tr>
<tr>
<td>Registered Organization</td>
<td>SOFTWARE</td>
<td><code>\Microsoft\Windows NT\CurrentVersion</code></td>
<td>Identifies the registered organization entered during installation (can be modified later)</td>
</tr>
<tr>
<td>System Root</td>
<td>SOFTWARE</td>
<td><code>\Microsoft\Windows NT\CurrentVersion</code></td>
<td>Identifies the system directory</td>
</tr>
<tr>
<td>Last User Logged In</td>
<td>SOFTWARE</td>
<td><code>\Microsoft\Windows\CurrentVersion\Authentication\LogonUI - LastLoggedOnUser</code></td>
<td>Lists the last user that logged into the system</td>
</tr>
<tr>
<td>Mounted Devices</td>
<td>SYSTEM</td>
<td><code>\MountedDevices</code></td>
<td>Lists current and prior mounted devices that use a drive letter</td>
</tr>
<tr>
<td>Pagefile</td>
<td>SYSTEM</td>
<td><code>\ControlSet###\Control\Session Manager\Memory Management</code></td>
<td>Contains the pagefile settings such as location, size, set to wipe etc.</td>
</tr>
<tr>
<td>Run</td>
<td>SOFTWARE</td>
<td><code>\Microsoft\Windows\CurrentVersion\Run</code></td>
<td>Lists programs that run automatically when the system boots</td>
</tr>
<tr>
<td>Time Zone</td>
<td>SYSTEM</td>
<td><code>\ControlSet###\Control\TimeZoneInformation</code></td>
<td>Identifies the time zone entered during installation (can be modified later)</td>
</tr>
<tr>
<td>USB Devices</td>
<td>SYSTEM</td>
<td><code>\ControlSet###\Enum\USBSTOR</code></td>
<td>Lists the system’s USB devices</td>
</tr>
</tbody>
</table>

Table C.3 – System Information Keys
## C.4 – Network Information

<table>
<thead>
<tr>
<th>Information</th>
<th>File</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Groups</td>
<td>SAM</td>
<td>\Domains\Builtin\Aliases\Names</td>
<td>Lists local account security identifiers</td>
</tr>
<tr>
<td>Local Users</td>
<td>SAM</td>
<td>\Domains\Account\Users\Names</td>
<td>Lists local account security identifiers</td>
</tr>
<tr>
<td>Printers – Currently Defined</td>
<td>SYSTEM</td>
<td>\ControlSet###\Control\Print\Printers</td>
<td>Lists all printers that are configured on the current system</td>
</tr>
<tr>
<td>Printer - Default</td>
<td>NTUSER.DAT</td>
<td>\Software\Microsoft\Windows NT\CurrentVersion\Windows - Device</td>
<td>Identifies the current default printer</td>
</tr>
<tr>
<td>Printer Information</td>
<td>SYSTEM</td>
<td>\ControlSet###\Control\Print\Environments\WindowsNTx86\Drivers\Version...</td>
<td>Contains information about the current printer</td>
</tr>
<tr>
<td>Profile List</td>
<td>SOFTWARE</td>
<td>\Microsoft\Windows NT\CurrentVersion\ProfileList</td>
<td>Contains the user security identifier for users with a profile on the system</td>
</tr>
<tr>
<td>TCP/IP Data</td>
<td>SYSTEM</td>
<td>\ControlSet###\Services\TCPIP\Parameters</td>
<td>Lists the current system’s domain and hostname data</td>
</tr>
<tr>
<td>TCP/IP Settings of a Network Adapter</td>
<td>SYSTEM</td>
<td>\ControlSet###\Services\TCPIP\Parameters\Interfaces&lt;ServiceName&gt;</td>
<td>Lists the current system’s IP address and gateway information</td>
</tr>
</tbody>
</table>

### Table C.4 – Network Information Keys

**Note:** For the last entry in the above table, `<ServiceName>` can be found by finding the appropriate key for a network adapter in `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkCards\` and then checking the ServiceName value for the adapter.
## C.5 – User Data

<table>
<thead>
<tr>
<th>Information</th>
<th>File</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Extensions\Program Associations</td>
<td>NTUSER.DAT</td>
<td>Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts</td>
<td>Identifies associated programs with file extensions</td>
</tr>
<tr>
<td>Last Logon Time</td>
<td>SAM</td>
<td>\Domains\Account\Users\F Key</td>
<td>Bytes 9–16 store the last logon time</td>
</tr>
<tr>
<td>Last Time Password Changed</td>
<td>SAM</td>
<td>\Domains\Account\Users\F Key</td>
<td>Bytes 25-32 store the last time the password was changed</td>
</tr>
<tr>
<td>Account Expiration</td>
<td>SAM</td>
<td>\Domains\Account\Users\F Key</td>
<td>Bytes 33-40 store the account expiration. If not expiration is set 0xFFFFFFFF will be shown</td>
</tr>
<tr>
<td>Last Failed Login</td>
<td>SAM</td>
<td>\Domains\Account\Users\F Key</td>
<td>Bytes 41-48 store the last unsuccessful logon</td>
</tr>
<tr>
<td>MRU – Open Saved</td>
<td>NTUSER.DAT</td>
<td>\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU</td>
<td>Lists the filename and path of the most recent files saved or copied to a specific location in Windows</td>
</tr>
<tr>
<td>MRU – Recent Documents</td>
<td>NTUSER.DAT</td>
<td>\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs</td>
<td>Identifies recently opened documents</td>
</tr>
<tr>
<td>MRU – Run MRU</td>
<td>NTUSER.DAT</td>
<td>\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU</td>
<td>Lists the most recent commands entered in the Windows Run box (note programs started using Start- &gt;Start Search will not appear in this key)</td>
</tr>
<tr>
<td>Information</td>
<td>File</td>
<td>Location</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Executed Programs</td>
<td>NTUSER.DAT</td>
<td><code>\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count</code></td>
<td>List of programs executed on the computer</td>
</tr>
</tbody>
</table>

*Table C.5 – User Data Keys*
## Appendix D

### The BitLocker Command Line Interface (CLI)

This appendix lists some useful commands when using the BitLocker CLI.

#### D.1 – Parameter List for BitLocker CLI

The CLI for BitLocker must be run from an administrative command prompt:

1. Navigate to Start->All Programs->Accessories. Right-click on the “command prompt” icon and select “Run as administrator” when prompted. Press “continue” in the UAC popup window.
2. In the command prompt, navigate to `c:\windows\system32`.
3. Run the command `cscript manage-bde.wsf`

Running the script with the `/?` parameter shows the following options:

```
manage-bde[.wsf] -parameter [arguments]
```

**Description:**
Configures BitLocker Drive Encryption on disk volumes.

**Parameter List:**
- `-status` Provides information about BitLocker-capable volumes.
- `-on` Encrypts the volume and turns BitLocker protection on.
- `-off` Decrypts the volume and turns BitLocker protection off.
- `-pause` Pauses encryption or decryption.
- `-resume` Resumes encryption or decryption.
- `-lock` Prevents access to BitLocker-encrypted data.
- `-unlock` Allows access to BitLocker-encrypted data.
- `-autounlock` Manages automatic unlocking of data volumes.
- `-protectors` Manages protection methods for the encryption key.
- `-tpm` Configures the computer's Trusted Platform Module (TPM).
- `-ForceRecovery` or `-fr`
  Forces a BitLocker-protected OS to recover on restarts.
- `-ComputerName` or `-cn`
  Runs on another computer. Examples: "ComputerX", "127.0.0.1"
- `?-` or `/?`
  Displays brief help. Example: "-ParameterSet -?"
- `-Help` or `-h`
  Displays complete help. Example: "-ParameterSet -h"

**Examples:**
- `manage-bde -status`
- `manage-bde -on C: -RecoveryPassword -RecoveryKey F:\`
- `manage-bde -unlock E: -RecoveryKey F:\84E151C1...7A62067A512.bek`
The most useful commands to an examiner involve the `status`, `protectors`, and `unlock` parameters. The parameters to be passed to `manage-bde.wsf` are shown in Table D.1:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-status</code></td>
<td>Provides the encryption status of all volumes.</td>
</tr>
<tr>
<td><code>-status &lt;driveletter&gt;</code>:</td>
<td>Provides the encryption status of a particular volume. Replace <code>&lt;driveletter&gt;</code> with the letter of the volume to be queried.</td>
</tr>
<tr>
<td><code>-protectors -get &lt;driveletter&gt;</code>:</td>
<td>Shows the key protectors for a particular volume. Replace <code>&lt;driveletter&gt;</code> with the letter of the volume to be queried.</td>
</tr>
<tr>
<td><code>-protectors -get &lt;driveletter&gt;</code>: -sek &lt;savepath&gt;</td>
<td>Shows the key protectors for a particular volume. Replace <code>&lt;driveletter&gt;</code> with the letter of the volume to be queried. Replace <code>&lt;savepath&gt;</code> with the path where the files are to be saved. Care should be taken to manually note the numerical recovery key.</td>
</tr>
<tr>
<td><code>-protectors -get &lt;driveletter&gt;</code>: -Type RecoveryPassword &gt; &lt;filename&gt;</td>
<td>Pipes a copy of the numerical password for a particular volume to a file. Replace <code>&lt;driveletter&gt;</code> with the letter of the volume to be queried. Replace <code>&lt;filename&gt;</code> with the path and filename of where the output is to be saved.</td>
</tr>
<tr>
<td><code>-unlock &lt;driveletter&gt;</code>: -RecoveryPassword &lt;pass&gt;</td>
<td>Unlocks a volume using the numerical recovery password. Replace <code>&lt;driveletter&gt;</code> with the letter of the volume to be unlocked. Replace <code>&lt;pass&gt;</code> with the 48-digit numerical password. The password can be entered with or without dashes.</td>
</tr>
<tr>
<td><code>-unlock &lt;driveletter&gt;</code>: -RecoveryKey &lt;file&gt;</td>
<td>Unlocks a volume using a recovery key. Replace <code>&lt;driveletter&gt;</code> with the letter of the volume to be unlocked. Replace <code>&lt;file&gt;</code> with the filename (including full path) of the recovery key. This part should be enclosed in inverted commas if there are any spaces in the path/filename.</td>
</tr>
</tbody>
</table>

Table D.1 – Useful Commands for BitLocker CLI