Guide to Computer Forensics and Investigations Fifth Edition

Chapter 9

Digital Forensics Analysis and Validation

Objectives

- Determine what data to analyze in a digital forensics investigation
- Explain tools used to validate data
- Explain common data-hiding techniques

Determining What Data to Collect and Analyze

- Examining and analyzing digital evidence depend on the nature of the investigation
  - And the amount of data to process
- **Scope creep** - when an investigation expands beyond the original description
  - Because of unexpected evidence found
  - Attorneys may ask investigators to examine other areas to recover more evidence
  - Increases the time and resources needed to extract, analyze, and present evidence

Determining What Data to Collect and Analyze

- Scope creep has become more common
  - Criminal investigations require more detailed examination of evidence just before trial
  - To help prosecutors fend off attacks from defense attorneys
- New evidence often isn't revealed to prosecution
  - It's become more important for prosecution teams to ensure they have analyzed the evidence exhaustively before trial

Approaching Digital Forensics Cases

- Begin a case by creating an investigation plan that defines the:
  - Goal and scope of investigation
  - Materials needed
  - Tasks to perform
- The approach you take depends largely on the type of case you're investigating
  - Corporate, civil, or criminal

Approaching Digital Forensics Cases

- Follow these basic steps for all digital forensics investigations:
  - 1. For target drives, use recently wiped media that have been reformatted and inspected for viruses
  - 2. Inventory the hardware on the suspect's computer, and note condition of seized computer
  - 3. For static acquisitions, remove original drive and check the date and time values in system's CMOS
  - 4. Record how you acquired data from the suspect drive

Approaching Digital Forensics Cases

- Follow these basic steps for all digital forensics investigations (cont'd):
  - 5. Process drive's contents methodically and logically
  - 6. List all folders and files on the image or drive
  - 7. Examine contents of all data files in all folders
  - 8. Recover file contents for all password-protected files
  - 9. Identify function of every executable file that doesn't match hash values

Approaching Digital Forensics Cases

- Follow these basic steps for all digital forensics investigations (cont'd):
  - 10. Maintain control of all evidence and findings
- Refining and Modifying the Investigation Plan
  - Even if initial plan is sound, at times you may need to deviate from it and follow evidence
  - Knowing the types of data to look for helps you make the best use of your time
  - The key is to start with a plan but remain flexible in the face of new evidence

Using OSForensics to Analyze Data

- OSForensics can perform forensics analysis on the following file systems:
  - Microsoft FAT12, FAT16, and FAT32
  - Microsoft NTFS
  - Mac HFS+ and HFSX
  - Linux Ext2fs, and Ext4fs
- OSForensics can analyze data from several sources
  - Including image files from other vendors

Using OSForensics to Analyze Data

- Includes OSFMount utility which can access many formats, including:
  - Raw, Expert Witness, and Advanced Forensics Format (AFF)
  - Can also mount and examine VMware images (.vmdk), SMART images (.s01), and VHD images (.vhd)
- Can use the NIST National Software Reference Library (NSRL)
  - Enables you to mount the NSRL ISO image

Using OSForensics to Analyze Data

- Using the Index Feature in OS Forensics
  - OSForensics indexes text data so that you can perform searches immediately
  - Follow steps starting on page 5 to learn how to index a case

Using OSForensics to Analyze Data

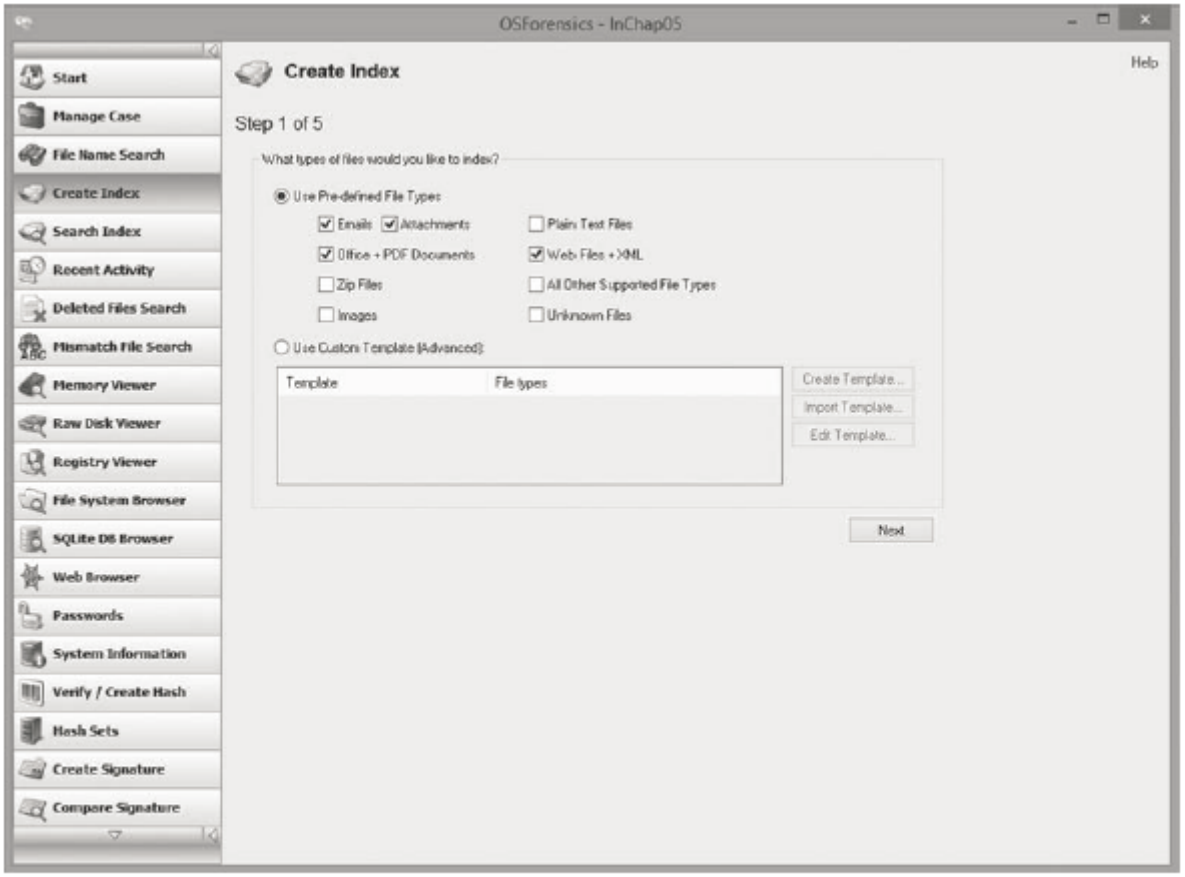

Figure 9-1 Specifying file types to index
Source: PassMark Software, *www.osforensics.com*

Using OSForensics to Analyze Data

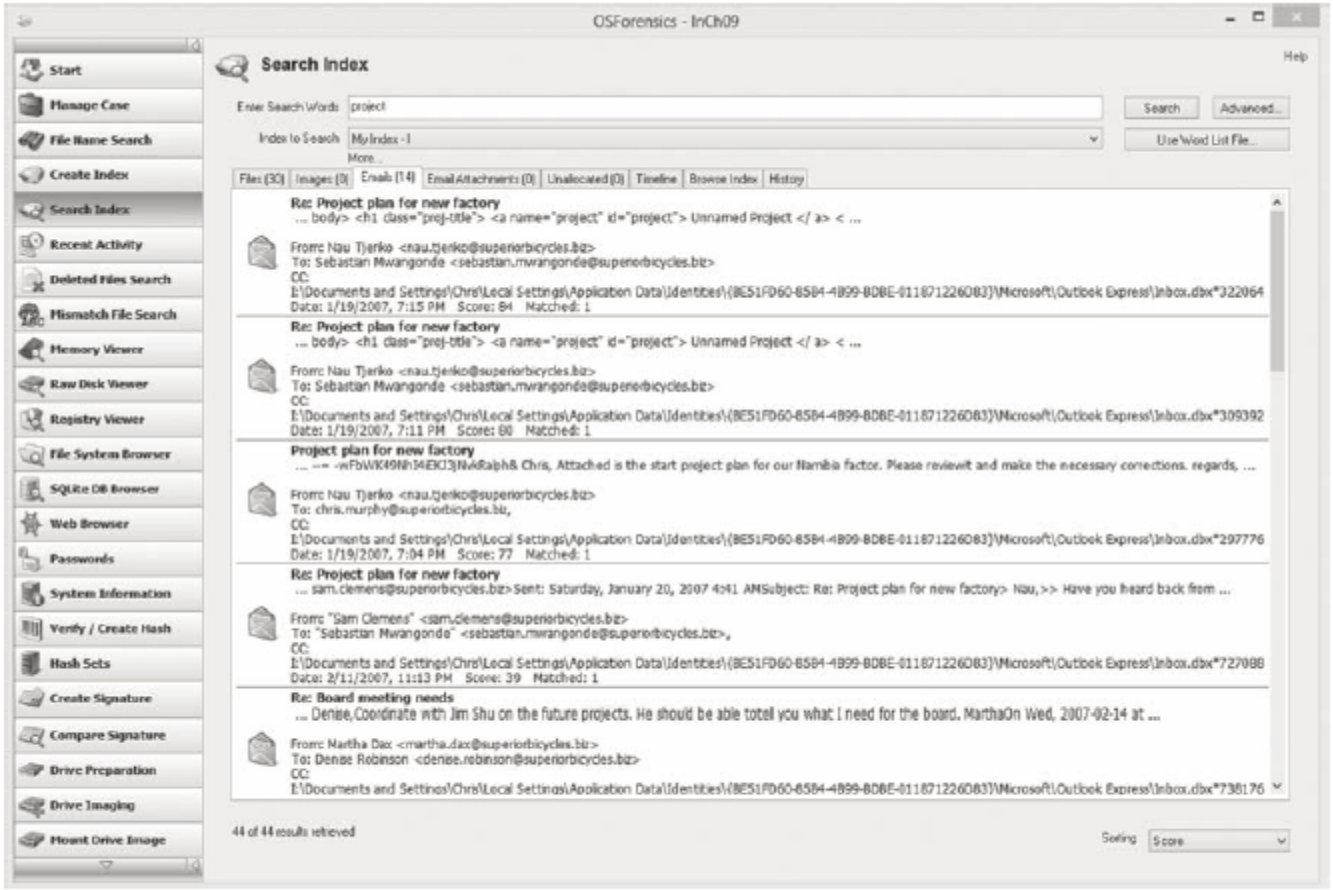

Figure 9-3 Entering a search term
Source: PassMark Software, *www.osforensics.com*

Using OSForensics to Analyze Data

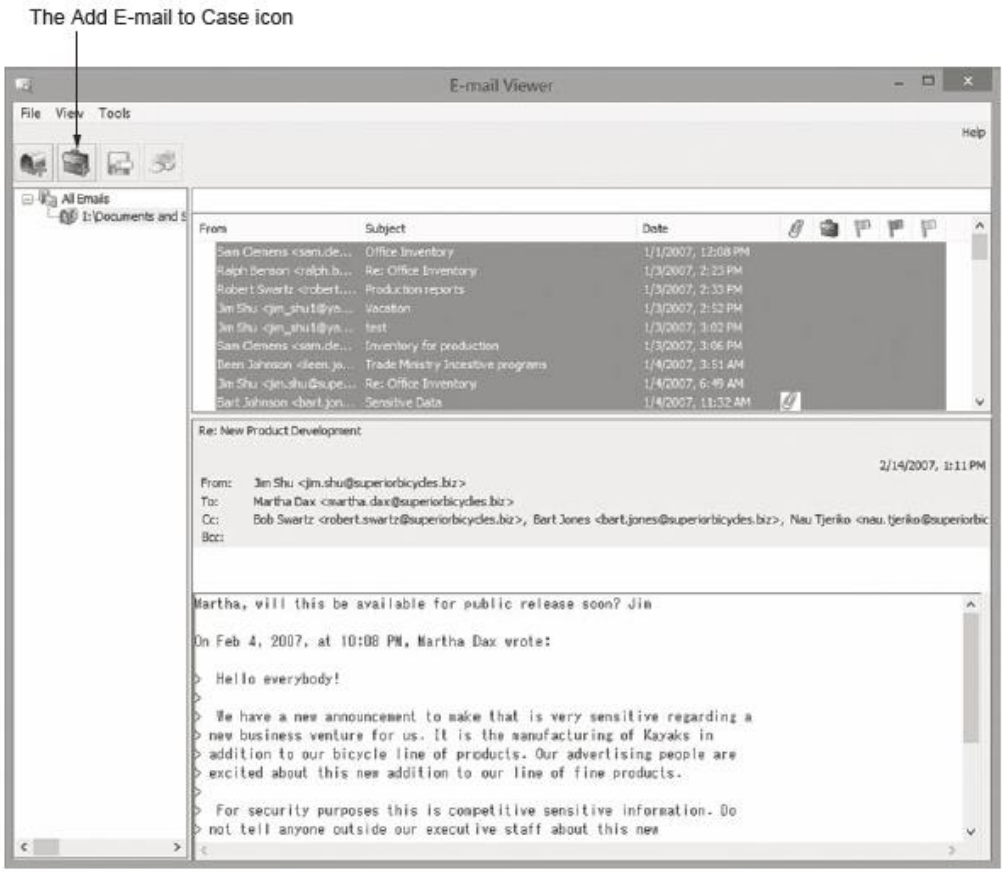

Figure 9-4 The E-mail Viewer window
Source: PassMark Software, *www.osforensics.com*

Validating Forensic Data

- Ensuring the integrity of data collected is essential for presenting evidence in court
- Most forensic tools offer hashing of image files
- Example - when ProDiscover loads an image file:
  - It runs a hash and compares the value with the original hash calculated when the image was first acquired
- Using advanced hexadecimal editors ensures data integrity

Validating with Hexadecimal Editors

- Advanced hex editors offer features not available in digital forensics tools, such as:
  - Hashing specific files or sectors
- With the hash value in hand
  - You can use a forensics tool to search for a suspicious file that might have had its name changed to look like an innocuous file
- WinHex provides MD5 and SHA-1 hashing algorithms

Validating with Hexadecimal Editors

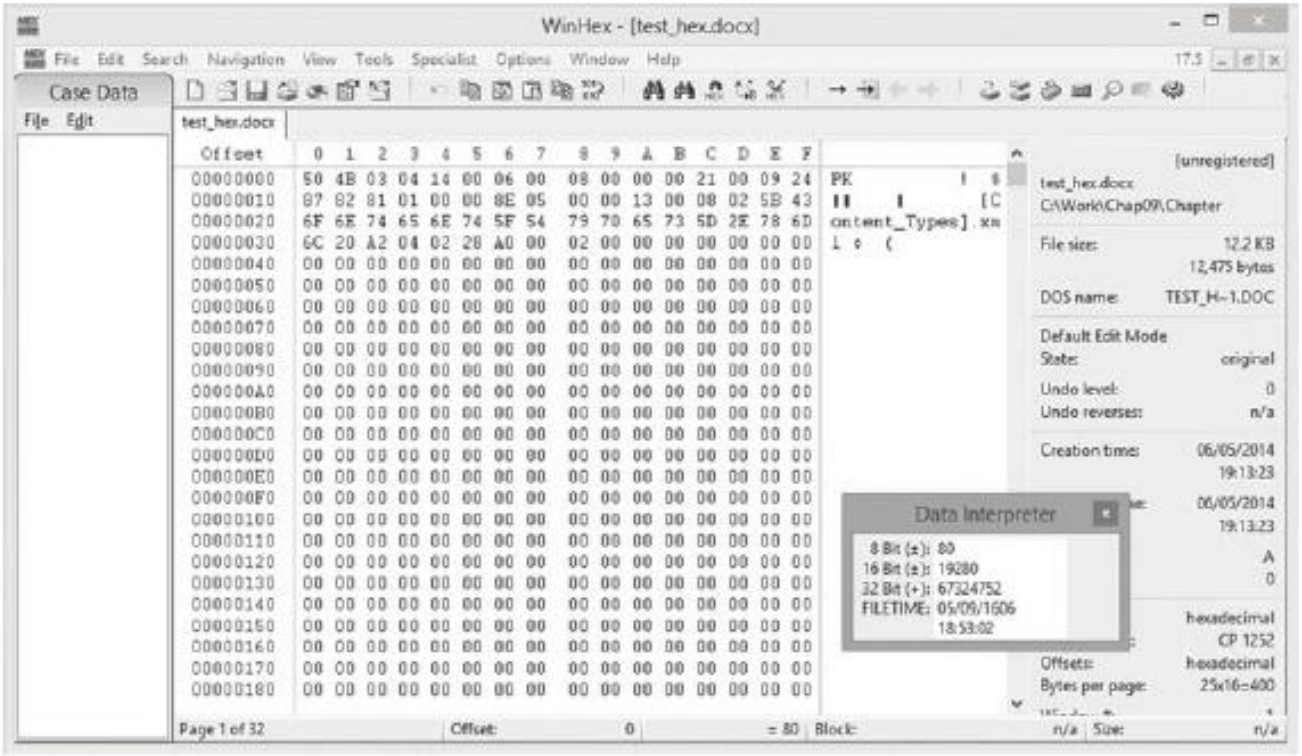

Figure 9-5 Viewing a file opened in WinHex
Courtesy of X-Ways AG, *www.x-ways.net*

Validating with Hexadecimal Editors

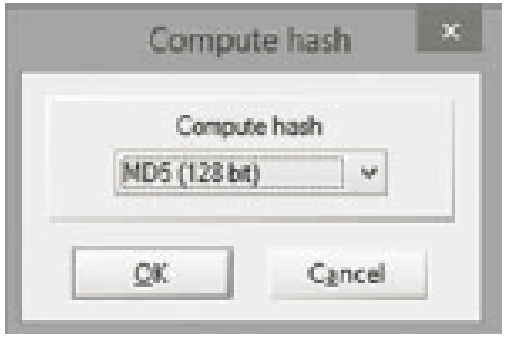

Figure 9-6 The Compute hash dialog box

Validating with Hexadecimal Editors

Figure 9-7 MD5 hash results
Courtesy of X-Ways AG, *www.x-ways.net*

Validating with Hexadecimal Editors

- Advantage of recording hash values
  - You can determine whether data has changed
- **Block-wise hashing**
  - A process that builds a data set of hashes of sectors from the original file
  - Then examines sectors on the suspect's drive to see whether any other sectors match
  - If an identical hash value is found, you have confirmed that the file was stored on the suspect's drive

Validating with Hexadecimal Editors

- Using Hash Values to Discriminate Data
  - AccessData has its own hashing database, **Known File Filter (KFF)**
  - KFF filters known program files from view and contains has values of known illegal files
  - It compares known file hash values with files on your evidence drive to see if they contain suspicious data
  - Other digital forensics tools can import the NSRL database and run hash comparisons

Validating with Digital Forensics Tools

- ProDiscover
  - .eve files contain metadata that includes hash value
  - Has a preference you can enable for using the Auto Verify Image Checksum feature when image files are loaded
  - If the Auto Verify Image Checksum and the hashes in the .eve file's metadata don't match
    - ProDiscover will notify that the acquisition is corrupt and can't be considered reliable evidence

Validating with Digital Forensics Tools

- Raw format image files don't contain metadata
  - You must validate them manually to ensure integrity
- In AccessData FTK Imager, when selecting the Expert Witness (.e01) or SMART (.s01) format:
  - Additional options for validating the acquisition are available
  - Validation report lists MD5 and SHA-1 hash values
- Follow steps starting on page 12 to see how ProDiscover's built-in validation feature works

Validating with Digital Forensics Tools

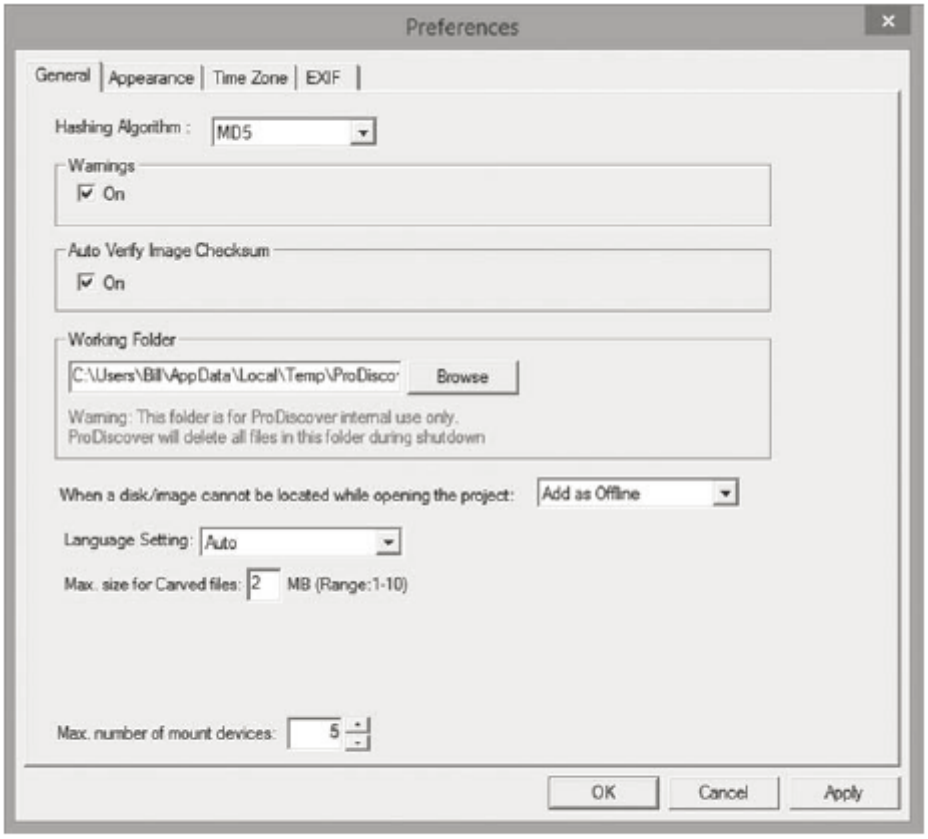

Figure 9-8 Enabling the Auto Verify Image Checksum feature
Courtesy of Technology Pathways, LLC

Validating with Digital Forensics Tools

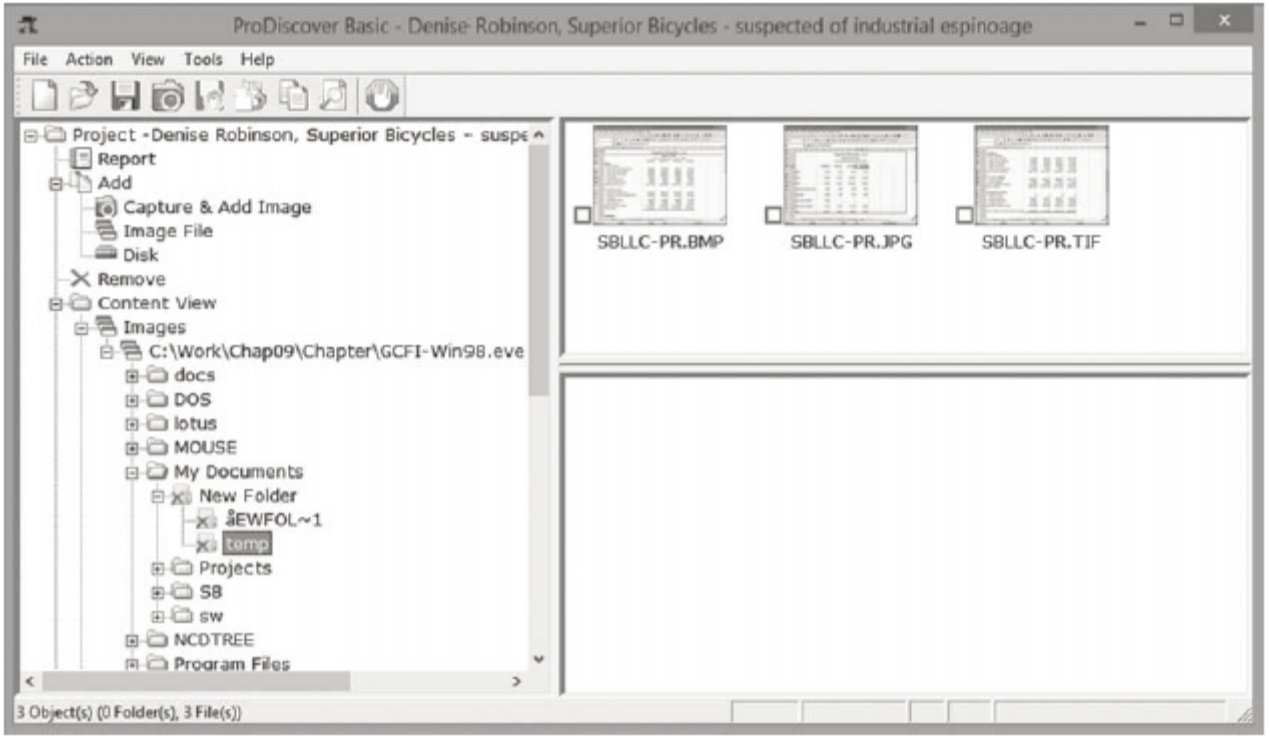

Figure 9-9 The Gallery view
Courtesy of Technology Pathways, LLC

Addressing Data-Hiding Techniques

- Data hiding - changing or manipulating a file to conceal information
- Techniques:
  - Hiding entire partitions
  - Changing file extensions
  - Setting file attributes to hidden
  - Bit-shifting
  - Using encryption
  - Setting up password protection

Hiding Files by Using the OS

- One of the first techniques to hide data:
  - Changing file extensions
- Advanced digital forensics tools check file headers
  - Compare the file extension to verify that it's correct
  - If there's a discrepancy, the tool flags the file as a possible altered file
- Another hiding technique
  - Selecting the Hidden attribute in a file's Properties dialog box

Hiding Partitions

- By using the Windows `diskpart remove letter` command
  - You can unassign the partition's letter, which hides it from view in File Explorer
- To unhide, use the `diskpart assign letter` command
- Other disk management tools:
  - Partition Magic, Partition Master, and Linux Grand Unified Bootloader (GRUB)

Hiding Partitions

- To detect whether a partition has been hidden
  - Account for all disk space when examining an evidence drive
  - Analyze any disk areas containing space you can't account for
- In ProDiscover, a hidden partition appears as the highest available drive letter set in the BIOS
  - Other forensics tools have their own methods of assigning drive letters to hidden partitions

Hiding Partitions

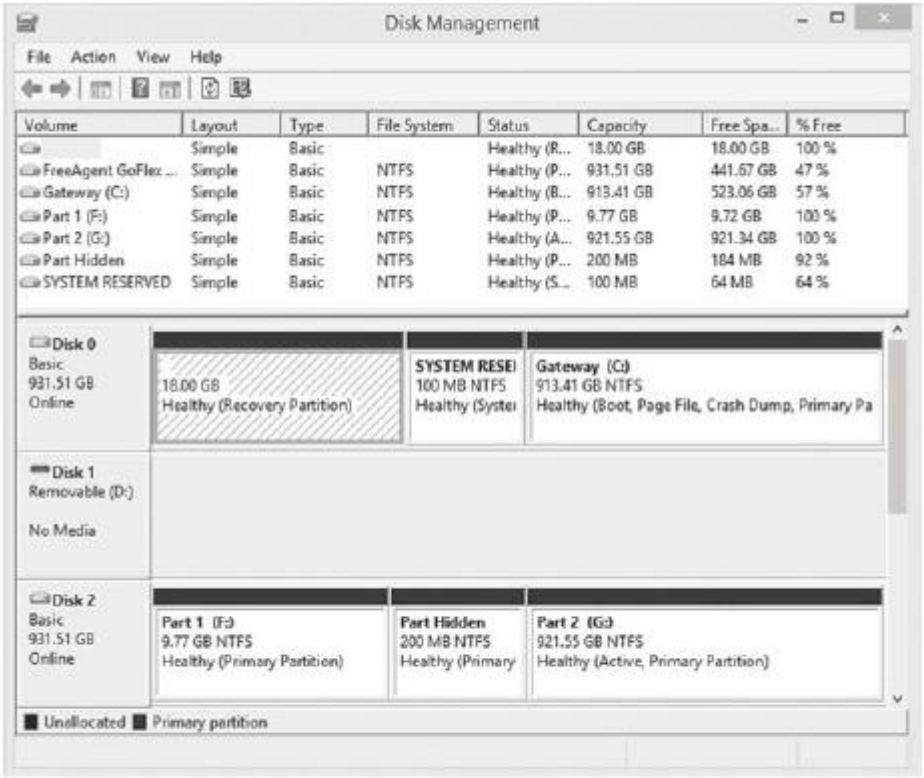

Figure 9-10 The Disk Management window
Courtesy of Microsoft Corporation

Hiding Partitions

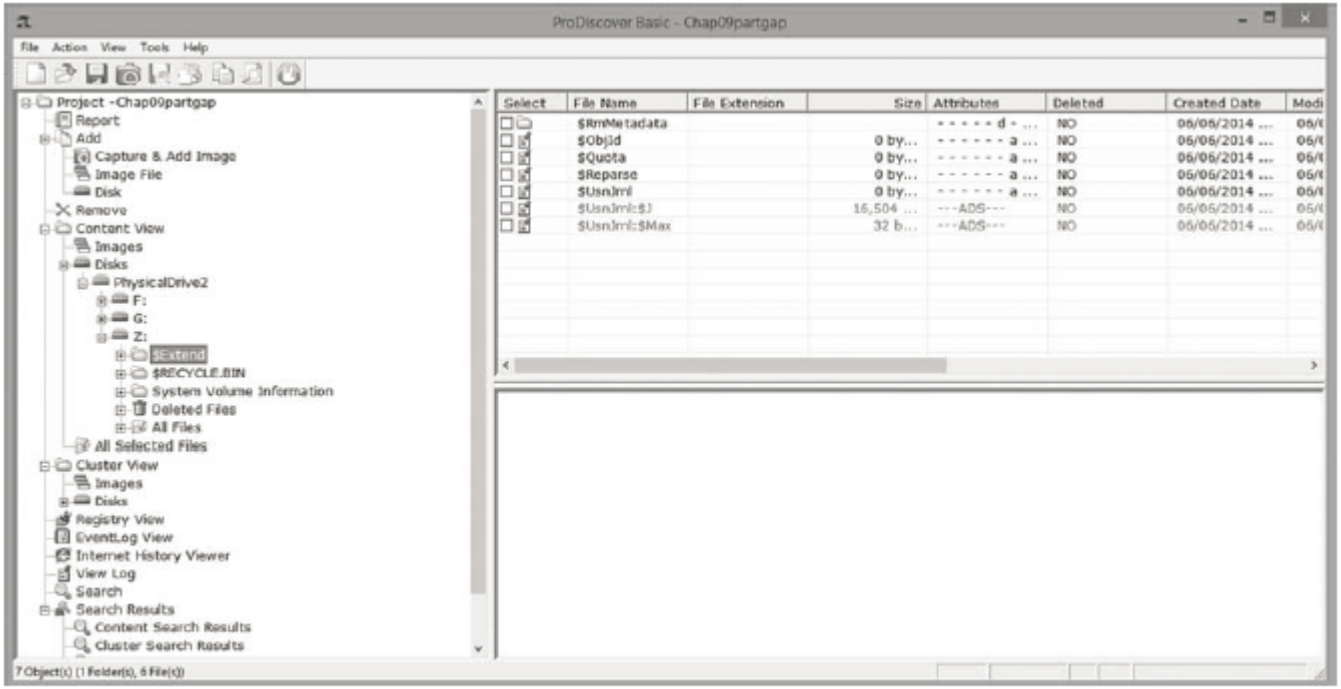

Figure 9-11 Viewing a hidden partition in ProDiscover
Courtesy of Technology Pathways, LLC

Marking Bad Clusters

- A data-hiding technique used in FAT file systems is placing sensitive or incriminating data in free or slack space on disk partition clusters
  - Involves using old utilities such as Norton DiskEdit
- Can mark good clusters as bad clusters in the FAT table so the OS considers them unusable
  - Only way they can be accessed from the OS is by changing them to good clusters with a disk editor
- DiskEdit runs only in MS-DOS and can access only FAT-formatted disk media

Bit-Shifting

- Some users use a low-level encryption program that changes the order of binary data
  - Makes altered data unreadable To secure a file, users run an assembler program (also called a “macro”) to scramble bits
  - Run another program to restore the scrambled bits to their original order
- Bit shifting changes data from readable code to data that looks like binary executable code
- WinHex includes a feature for shifting bits

Bit-Shifting

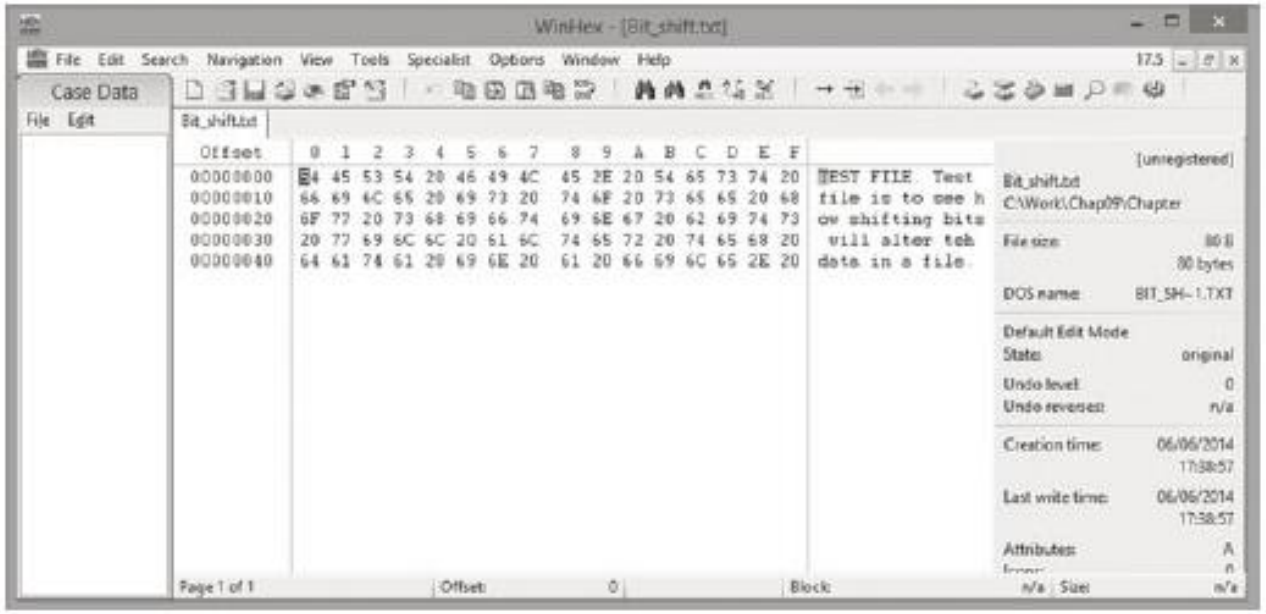

Figure 9-12 `Bit_shift.txt` open in WinHex
Courtesy of X-Ways AG, *www.x-ways.net*

Bit-Shifting

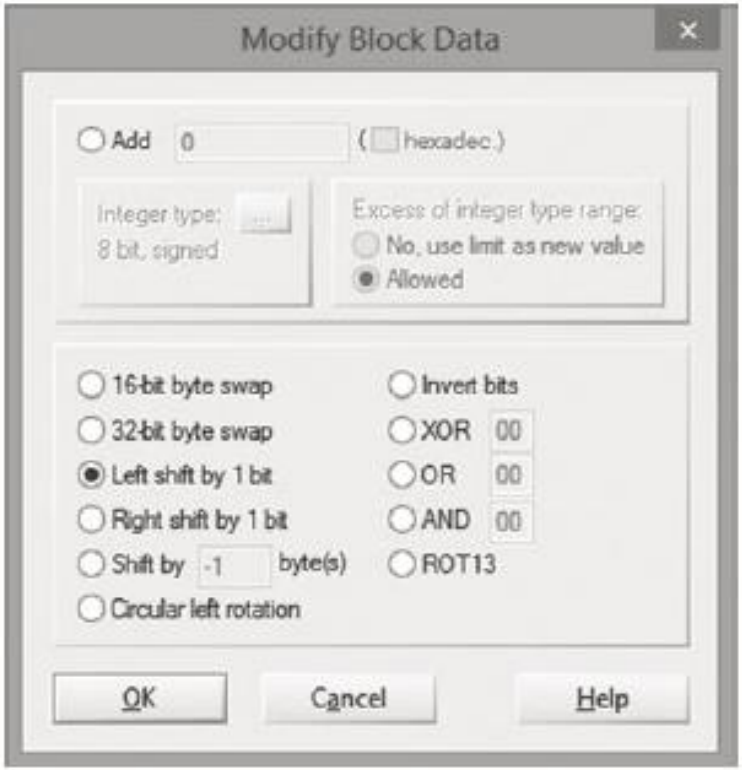

Figure 9-13 The Modify Block Data dialog box
Courtesy of X-Ways AG, *www.x-ways.net*

Bit-Shifting

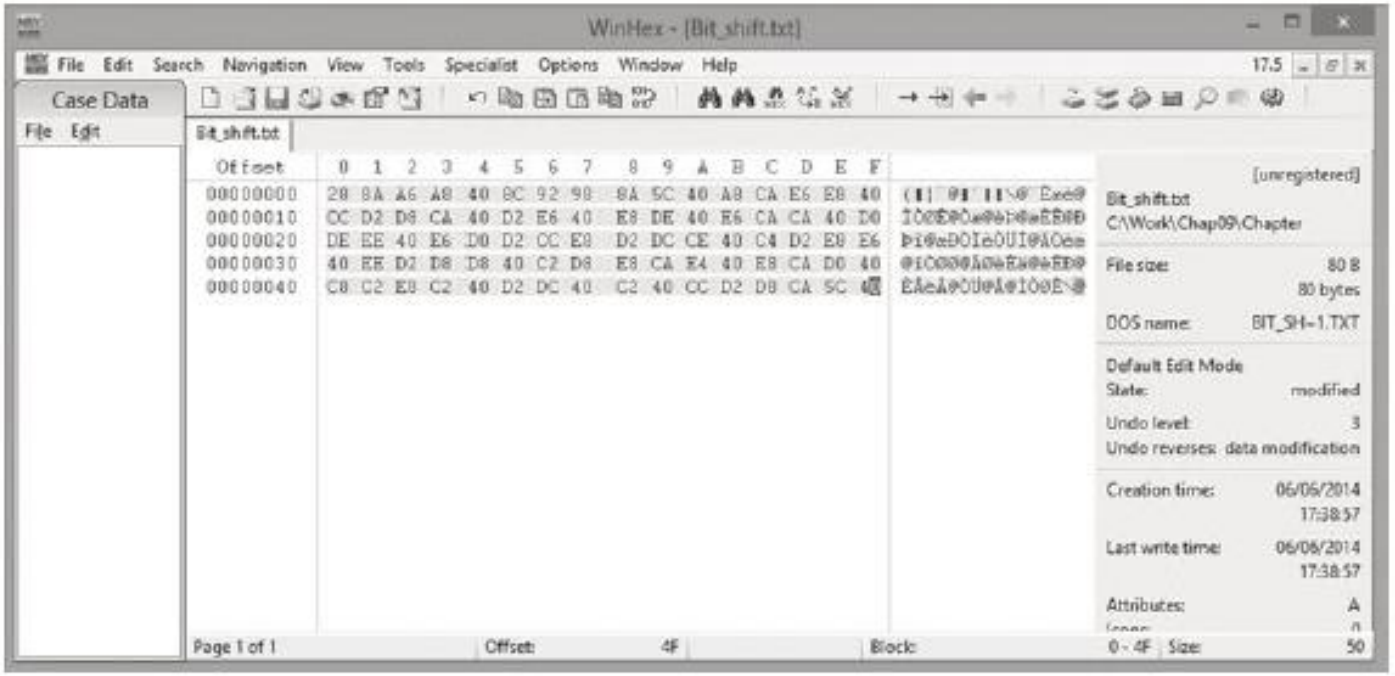

Figure 9-14 Viewing the shifted bits
Courtesy of X-Ways AG, *www.x-ways.net*

Understanding Steganalysis Methods

- Steganography - comes from the Greek word for "hidden writing"
  - Hiding messages in such a way that only the intended recipient knows the message is there
- Steganalysis - term for detecting and analyzing steganography files
- Digital watermarking - developed as a way to protect file ownership
  - Usually not visible when used for steganography

Understanding Steganalysis Methods

- A way to hide data is to use steganography tools
  - Many are freeware or shareware
  - Insert information into a variety of files
- If you encrypt a plaintext file with PGP and insert the encrypted text into a steganography file
  - Cracking the encrypted message is extremely difficult

Understanding Steganalysis Methods

- Steganalysis methods
  - Stego-only attack
  - Known cover attack
  - Known message attack
  - Chosen stego attack
  - Chosen message attack

Examining Encrypted Files

- To decode an encrypted file
  - Users supply a password or passphrase
- Many encryption programs use a technology called "key escrow"
  - Designed to recover encrypted data if users forget their passphrases or if the user key is corrupted after a system failure
- Key sizes of 128 bits to 4096 bits make breaking them nearly impossible with current technology

Recovering Passwords

- Password-cracking tools are available for handling password-protected data or systems
  - Some are integrated into digital forensics tools
- Stand-alone tools:
  - Last Bit
  - AccessData PRTK
  - ophcrack
  - John the Ripper
  - Passware

Recovering Passwords

- Brute-force attacks
  - Use every possible letter, number, and character found on a keyboard
  - This method can require a lot of time and processing power
- Dictionary attack
  - Uses common words found in the dictionary and tries them as passwords
  - Most use a variety of languages

Recovering Passwords

- With many programs, you can build profiles of a suspect to help determine his or her password
- Many password-protected OSs and application store passwords in the form of MD5 or SHA hash values
- A brute-force attack requires converting a dictionary password from plaintext to a hash value
  - Requires additional CPU cycle time

Recovering Passwords

- Rainbow table
  - A file containing the hash values for every possible password that can be generated from a computer's keyboard
  - No conversion necessary, so it is faster than a brute-force or dictionary attack
- Salting passwords
  - Alters hash values and makes cracking passwords more difficult

Summary

- Examining and analyzing digital evidence depend on the nature of the investigation and the amount of data to process
- General procedures:
  - Wipe and prepare target drives, document all hardware components on the suspect's computer, check date and time values in the suspect's computer's CMOS, acquire data and document steps, list all folders and files, attempt to open password-protected files, determine function of executable files, and document steps

Summary

- Advanced digital forensics tools have features such as indexing text data, making keyword searches faster
- A critical aspect of digital forensics is validating digital evidence
  - ensuring the integrity of data you collect is essential for presenting evidence in court
- Data hiding involves changing or manipulating a file to conceal information

Summary

- Three ways to recover passwords:
  - Dictionary attacks
  - Brute-force attacks
  - Rainbows tables