Guide to Computer Forensics and Investigations
Sixth Edition

Chapter 9

Digital Forensics Analysis and Investigation

Objectives

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

Approaching Digital Forensics Cases (1 of 4)

- 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 (2 of 4)

- 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 (whether on, off etc),
  - 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 (3 of 4)

- 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,
- 10. Maintain control of all evidence and findings,

Using Autopsy to Validate Data

- In previous chapters we used Autopsy for windows to perform forensics analysis for the following file systems:
  - MS FAT, NTFS ExFAT
- But did not do the MC and Linux files (illustrated in Chapter 7)
- In addition to all Autopsy can analyze data from several sources, include image files from other vendors. Autopsy can handle many formats, including raw, Expert Witness and virtual machine image files (.vdi and vhd).
- To enhance this process Autopsy has an indexed version of NIST-National Software Reference Library (NSRL) of MD5 hashes and you can import NSRL reference hashes into Autopsy.

Using Autopsy to Validate Data

- Do the following Activities from the textbook:
- Installing NSRL Hashes in Autopsy:
  - pages 381-383
- Collecting Hash Values in Autopsy:
  - Pages 383-388

Validating Forensic Data

- Ensuring the integrity of data collected is essential for presenting evidence in court
- Most forensic tools offer **hashing** of image files (the concept and procedure that you used in Lab2: imaging)
- Using advanced hexadecimal editors ensures data integrity
- Common hashing algorithms are MD5 and SHA1
- AccessData has its own hashing database, **Known File Filter (KFF)**

Validating with Hexadecimal Editors (3 of 6)

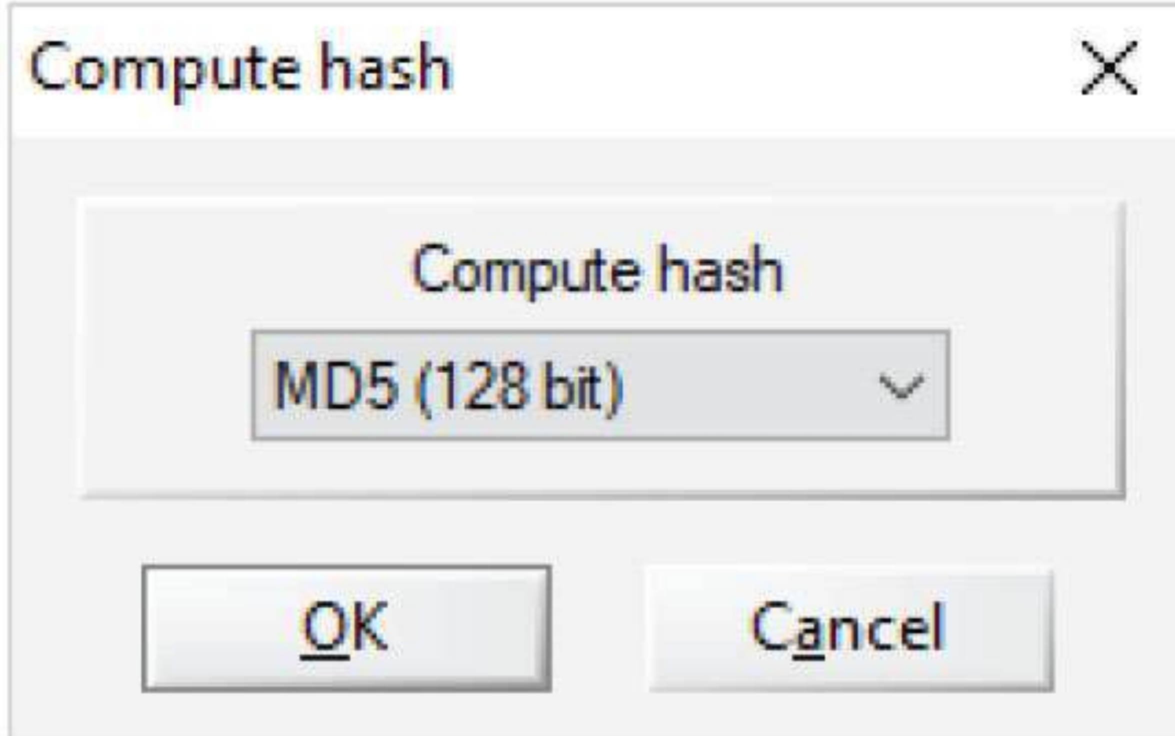

Figure 9-11 The Compute hash dialog box

Source: X-Ways AG, *www.x-ways.net*

Validating with Hexadecimal Editors (4 of 6)

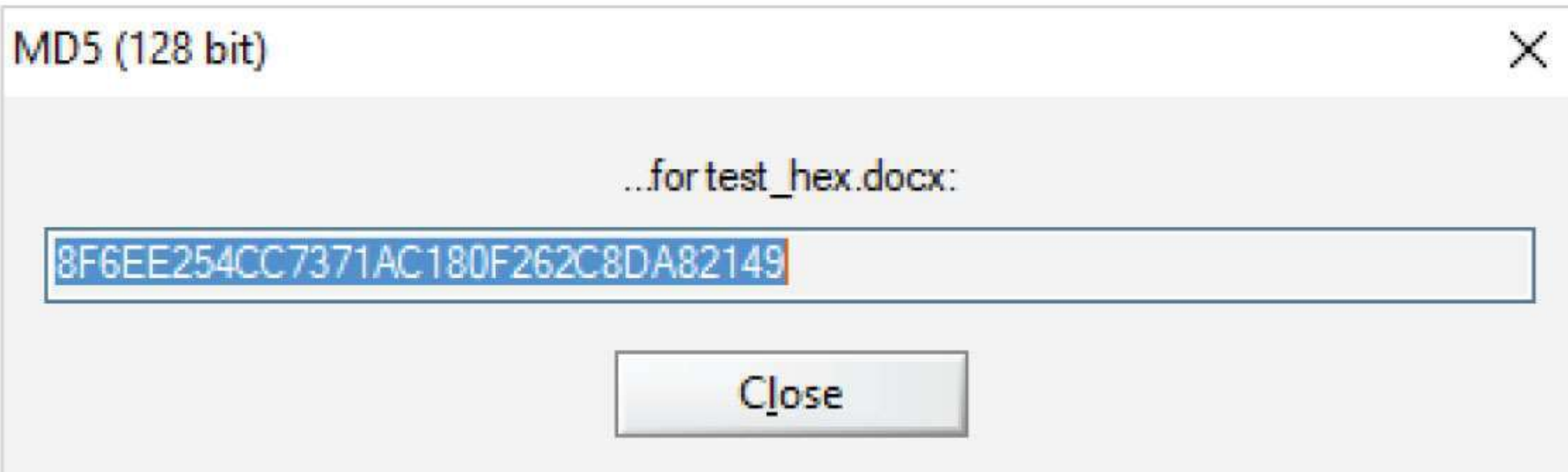

Figure 9-12 **MD5 hash results**

Source: X-Ways AG, *www.x-ways.net*

Validating with Digital Forensics Tools (1 of 3)

- In AccessData FTK Imager, when selecting the Expert Witness (.e01) or SMART (.s01) format:
  - Additional options for hashing all the data are available.
  - Validation report lists MD5 and SHA-1 hash values.
- Follow steps starting on
  - page 383-393
- to see how to use WinHex to hash an image file and then compare it with the original hash value FTK Imager calculated.

Validating with Digital Forensics Tools (2 of 3)

InChap09.dd.txt - Notepad

File Edit Format View Help

```
Created By AccessData® FTK® Imager 3.1.1.8

Case Information:
Acquired using: ADI3.1.1.8
Case Number: InChap09
Evidence Number: InChap09
Unique description: In chapter exercise
Examiner: Joe Friday
Notes: In chapter exercise on hashing raw image files

--------------------------------------------------------------

Information for C:\Work\Chap09\InChap09:

Physical Evidentiary Item (Source) Information:
[Device Info]
 Source Type: Logical
[Drive Geometry]
 Bytes per Sector: 512
 Sector Count: 3,074,048
[Physical Drive Information]
 Removable drive: False
 Source data size: 1501 MB
 Sector count:    3074048

ATTENTION:
The following sector(s) on the source drive could not be read:
        1960096 through 1960101
        2061632 through 2061635
The contents of these sectors were replaced with zeros in the image.

[Computed Hashes]
 MD5 checksum:    db945a7e3589743923237c0518ababe1
 SHA1 checksum:   6d87a3665d756b7e22de3d0b087c6ab9ec3f8bf7

Image Information:
 Acquisition started:   Thu Jul 27 15:36:30 2017
 Acquisition finished:  Thu Jul 27 15:38:03 2017
 Segment list:
  C:\Work\Chap09\InChap09.001
```

Figure 9-14 The FTK Imager case information file

Validating with Digital Forensics Tools (3 of 3)

```
[Computed Hashes]
 MD5 checksum:     db945a7e3589743923237c0518ababe1
 Verified MD5:     DB945A7E3589743923237C0518ABABE1
 SHA1 checksum:    6d87a3665d756b7e22de3d0b087c6ab9ec3f8bf7
```

Figure 9-15 Recording the MD5 hash value

Addressing Data-Hiding Techniques

- Data hiding - changing or manipulating a file to conceal information
- Techniques:
  - Hiding entire partitions
  - Changing file extensions (.png to .doc)
  - Setting file attributes to hidden
  - Using encryption
  - Setting up password protection

Hiding Files by Using the OS

- One of the first techniques to hide data:
  - Changing file extensions (changing .png to .doc to avoid detection say in a child pornography case)
- 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 (2 of 4)

- 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
- Many digital forensics tools can detect and view a hidden partition

Understanding Steganalysis Methods (1 of 3)

- **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 (2 of 3)

- 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

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 (1 of 4)

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

Recovering Passwords (2 of 4)

- 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 (3 of 4)

- 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 (4 of 4)

- **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