CHAMPLAIN
COLLEGE


# The Power of Simple Hands-On Cyberforensics Exercises: A Guide for Faculty

**12th Colloquium for Information Systems Security Education**

Dallas, TX

June 3, 2008

**Gary C. Kessler & Jim Hoag**

Champlain College

Burlington, Vermont, U.S.A.

## Overview

- Introduction
- Pedagogy issues and hands-on exercises
- The role of hands-on exercises
  - » Detailed view of sample exercise
  - » Other exercises
- Conclusion

# What Did I Learn From This?

- Lectures are well-suited to the transfer and exchange of information
- Hands-on tasks cement the concepts
  - » Students do not need an instructor or proctor hovering while they exercise hands-on skills
  - » Students need someone to go to for questions, problems, etc. but that resource does not need to be geographically or temporally local
    - In fact, students are not so quick to ask questions when the resource is *not* right there
  - » Students do not need to practice hands-on skills in a central facility
    - Not everything in computer forensics needs to be done on a FRED

# Why Hands-On Exercises Are So Important

- **Active learning** is a critical component in technology education
  - » Constructivism
    - Cognitive structures are the building block of learning
  - » Resource-based learning
    - Engages students in active learning, enables their curiosity and investigative skills, and demands critical thinking
  - » Problem-based learning
    - Ill-defined problems allow students to synthesize information and build their own problem-solving skills

## Program Philosophy

- We are trying to educate students about the digital investigations process rather than train them about any individual product
  - » They will build their own tool kits
    - They will determine which tools excel at which functions
  - » Education = life-long learning
    - Product training expires with new versions

## Burning Issue

- *Critique:* "You can't teach computer forensics online because you have to have students do hands-on exercises."
  - » *Corollary:* "You need to teach computer forensics in a laboratory environment."

## Our Response...

- *Yes* -- computer forensics education requires hands-on tasks
- *No* -- you don't need a lab; students can do the work on their own systems
  - » Not only is this possible, it probably enhances their (current and future) education
  - » Not limited to college's tools
  - » Not limited to lab hours

## The (abbreviated) Computer Forensics Process

1. Identification -- An incident occurred
2. Preservation -- Freeze the state of the scene
3. Collection -- Create the examination media
4. Examination -- Gather pertinent data
5. Analysis -- Interpret findings
6. Presentation -- Report the results

***Digital Forensics Research Workshop, 2001***

## Hands-On Forensics Exercises

- Focus of exercises is on *process* rather than specific tools
  - » Homework exercises emphasize *examination*, *analysis*, and *presentation*
  - » Exercises employ many tools
    - Shareware, freeware, demo versions
    - EnCase, FTK, Helix, ProDiscover, WinHex, ...
  - » Evidence images come from a variety of sources
  - » Network-based analysis uses Internet tools

## Sample Hands-On Exercise

Analysis of a Floppy Disk That Contains Deleted Files

# Project Outline

- Students start with a floppy disk image file
  - » There are reportedly three files on the floppy
- Examine floppy to find an address, an image, some passwords, some other data, and verify the MD5 hash of the distribution medium
- This project is based on a forensic challenge posted on the Internet, October 2002

# DOS Floppies Use FAT12

- Sector 0 is the Boot Sector
  - » B/S points to FATs and Root Directory
- Primary and Secondary FATs follow Boot Sector
  - » FATs contain linked lists indicating clusters assigned to files
- Root Directory follows Secondary FAT
  - » Contains names of files (and directories), starting cluster, length, MAC times, etc.
  - » The starting cluster points to a cluster of data on the medium **and** an entry in the FATs
- Data Area follows Root Directory
  - » Data area always starts in *logical* cluster 2
  - » In FAT12 on a floppy, data area starts in *physical* cluster 33

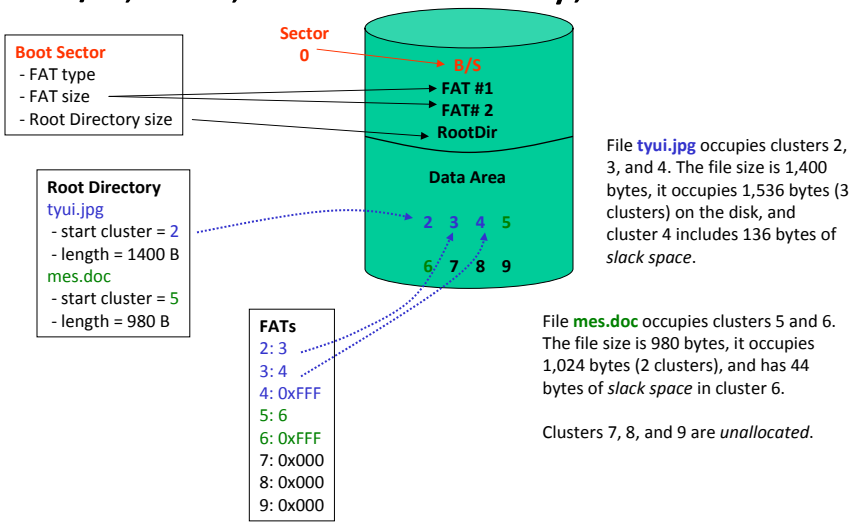
 

# Capacity of this Medium

- FAT12 allocates 12 bits per FAT entry
  - » Limits addressing to 4,096 ($2^{12}$) clusters
- This floppy is configured so that:
  - » 1 cluster = 1 sector
  - » 1 sector = 512 B
- This FAT12 table is limited in capacity to 2,097,152 bytes (2 MB)
  - » The device itself is only 2880 sectors (1.44 MB)




# B/S, FAT, Root Directory, & Files




File **tyui.jpg** occupies clusters 2, 3, and 4. The file size is 1,400 bytes, it occupies 1,536 bytes (3 clusters) on the disk, and cluster 4 includes 136 bytes of *slack space*.

File **mes.doc** occupies clusters 5 and 6. The file size is 980 bytes, it occupies 1,024 bytes (2 clusters), and has 44 bytes of *slack space* in cluster 6.

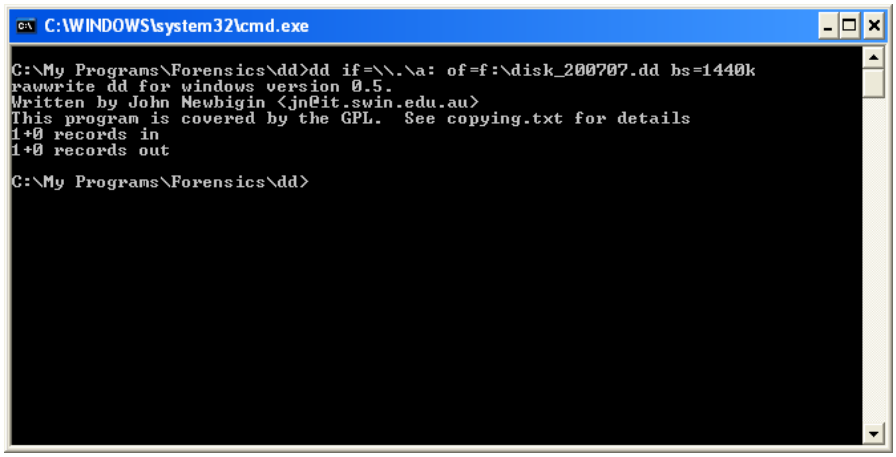
Clusters 7, 8, and 9 are *unallocated*.

# Forensic Process Redux

- It is **imperative** that the integrity of the original evidence medium be maintained
  1. Make a forensically correct image (copy) of the evidence medium
  2. Put the evidence medium in a safe place
  3. Do the analysis on the image *or* on a restored version of the original
  4. Verify that the examination copy is the same as the original evidence

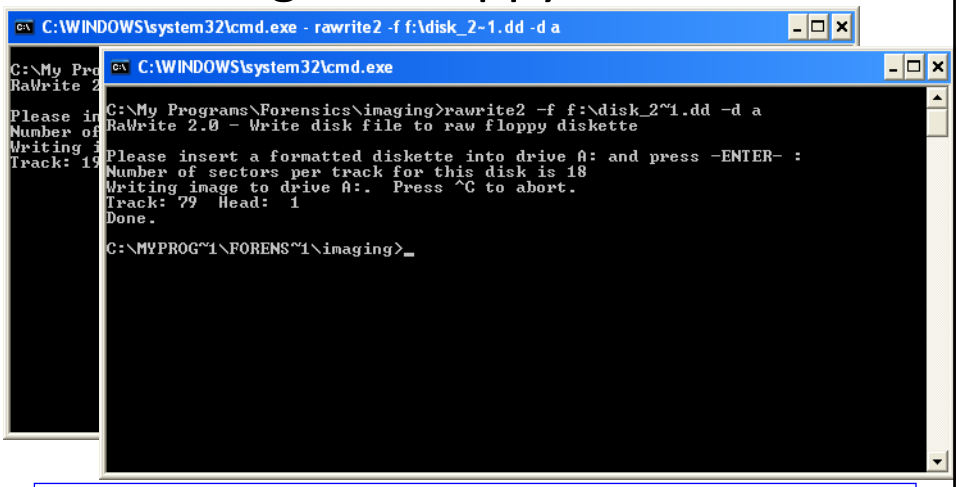
# Imaging a Floppy With dd




This use of dd creates an image file called disk_200707.dd on the f: drive.

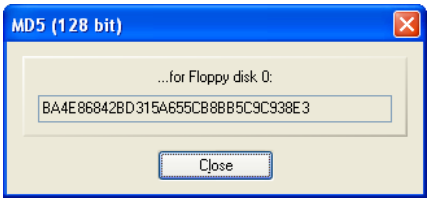
# Restoring the Floppy With rawrite




**This use of rawrite2 writes the image file to a floppy. *NOTE* that rawrite2 requires use of short file names.**

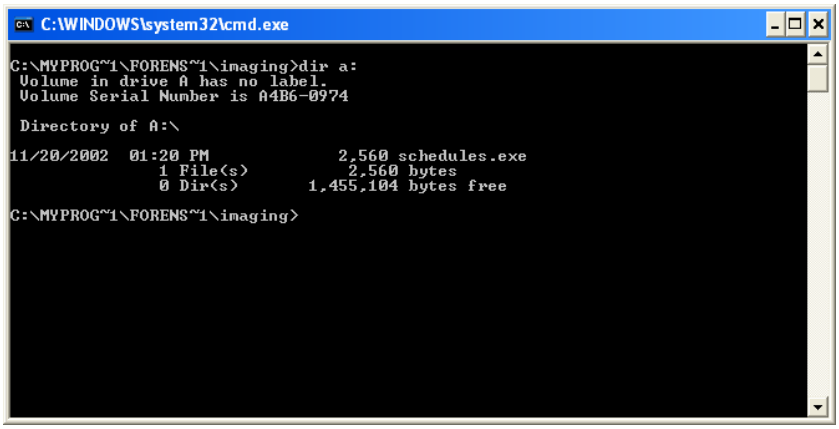


# Verifying the Copy




**WinHex calculated the same MD5 has for both the original evidence and examination media.**

# A DOS View of the Floppy




**Note that DOS sees only one file although we have been led to believe that there are three files on the floppy...**

# Let the Examination Begin...

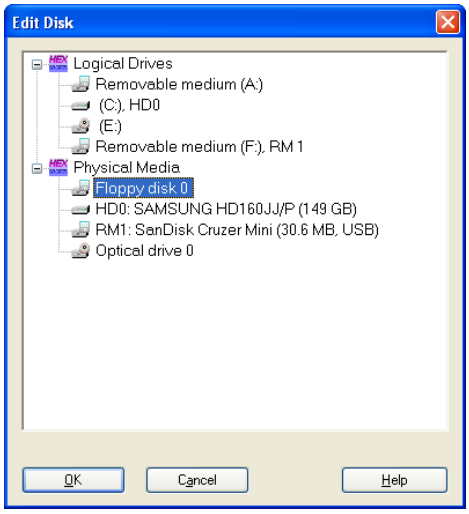
 

# Examination/Analysis Software

- The examination of this floppy could employ any number of tools, including FTK, EnCase, or any hex editor
  - » FTK or Encase could work with the dd image
  - » Editor could be used with the examination floppy
- Examples that follow employ WinHex

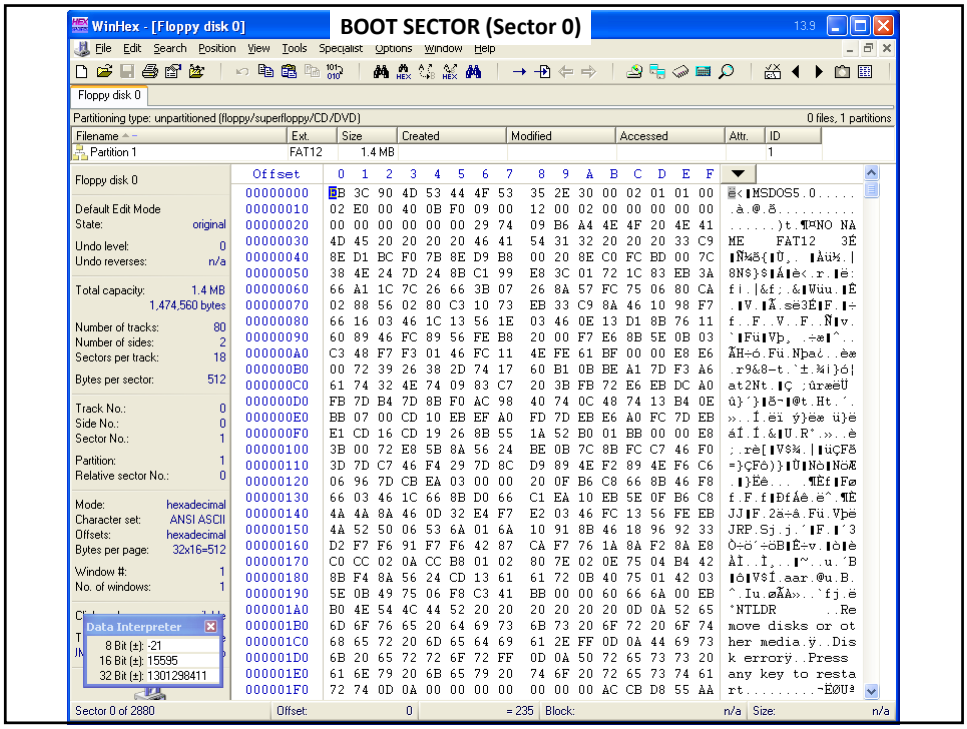
 

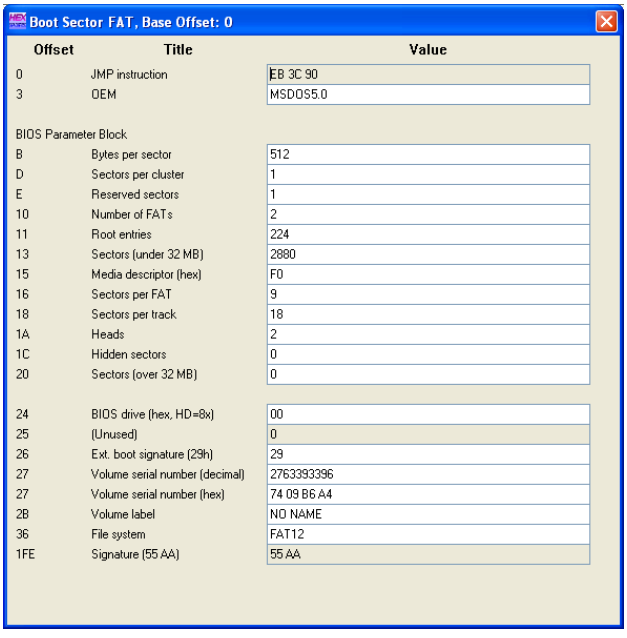
# Physical vs. Logical




- Logical view
  - » Examiner sees what the file system sees
  - » I.e., we won't see deleted files, and slack and unallocated space
- Physical view
  - » Examiner sees everything on the medium
- We want the **physical** view!!




# Boot Sector Template (WinHex)

# Sector Assignments

| Sector(s) | Offset | Function |
|---|---|---|
| 0 | 0x0000-0x01FF | Boot Sector |
| 1-9 | 0x0200-0x13FF | File Allocation Table (primary) |
| 10-18 | 0x1400-0x25FF | File Allocation Table (secondary) |
| 19-32 | 0x2600-0x41FF | Root Directory |
| 33-2879 | 0x4200-0x167FFF | Data area (**cluster 2**) |

**NOTES:**
- **Boot Sector is 1 sector (0x200 bytes)**
- **There are two FATs, each 9 sectors (0x1200 bytes)**
- **The Root Directory can contain 224 entries, each 32 bytes (7168, or 0x1c00, bytes; 14 sectors)**
- **File storage starts at sector #33 (1+9+9+14), byte #0x4200 (0x200+0x1200+0x1200+0x1c00)**




# Root Directory (1)

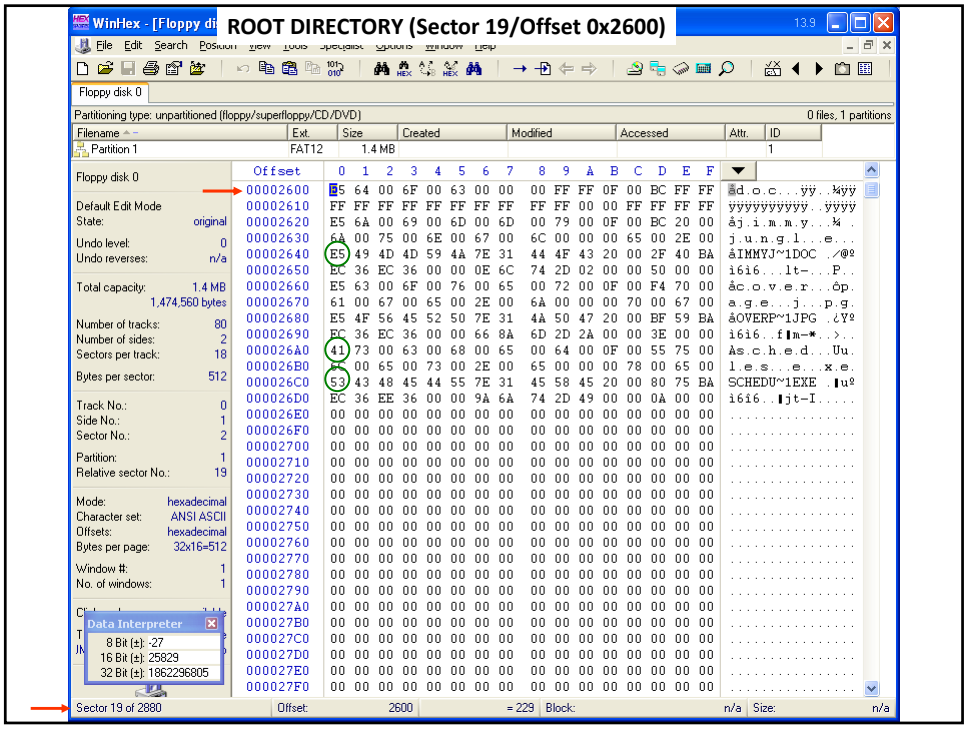
- The Root Directory starts at sector 19, byte offset 0x2600.
- Each entry is 32 B and contains the short (8.3) filename, attributes, MAC times, start cluster, and file size.
  - » Short file names (SFN) are preceded by the file's long filename (LFN), if appropriate.
  - » LFN entries have a sequence number in the first byte of the name and are stored in reverse order, 13 characters/entry
  - » SFNs use ASCII (8-bit) characters; LFNs use Unicode (16-bit)
  - » Example.... file named *ThisIsAReallyVeryLongName.html* would be stored in four entries:

```
html            (LFN entry, seq. 0x43)
VeryLongName.   (LFN entry, seq. 0x02)
ThisIsAReally   (LFN entry, seq. 0x01)
THISIS~1HTM     (SFN entry, 8.3 format)
```

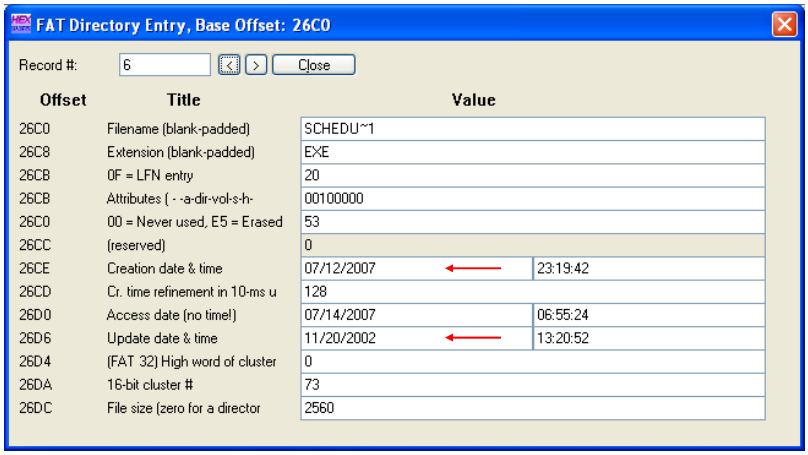
# Root Directory (2)

- The first byte of the SFN entry is either the first character of the file name or a usage indicator:
  - » 0x00 - entry has never been used
  - » 0xE5 - file has been deleted
- The start cluster in the directory entry points to the first cluster of the data on the medium **and** acts as a pointer to the cluster linked list in the FAT.



# Root Directory Template (WinHex)



# An Aside on MAC Times

- Modified (M) time is the actual time that the file was last written
- Access (A) time is the time that the file was last accessed
  - » This time is set by the application, not all of which do this accurately
  - » FAT file systems only maintains the **date** with A-time
- Creation (C) time is the time when the file was created *in this directory*

# What the Root Directory Tells Us

| Short File Name | ?IMMY~1.DOC | ?OVERP~1.JPG | SCHEDU~1.EXE |
|---|---|---|---|
| Long File Name | Jimmy Jungle.doc | cover page.jpg | schedules.exe |
| Starting Cluster (logical) | 0x0002 (2) | 0x002A (42) | 0x0049 (73) |
| Starting Cluster (physical) | 0x0021 (33) | 0x0049 (73) | 0x0068 (104) |
| Byte Offset (physical) | 0x4200 | 0x9200 | 0xD000 |
| File Size (bytes) | 20,480 | 15,072 | 2,560 |
| File Size (clusters) | 40 | 31 | 5 |
| Offset Into FAT | 0x004 (4) | 0x03F (64) | 0x06E (110) |

**Offset into FAT is (starting_cluster*3/2 + 1). This is where the link to the next cluster in the file is located.**



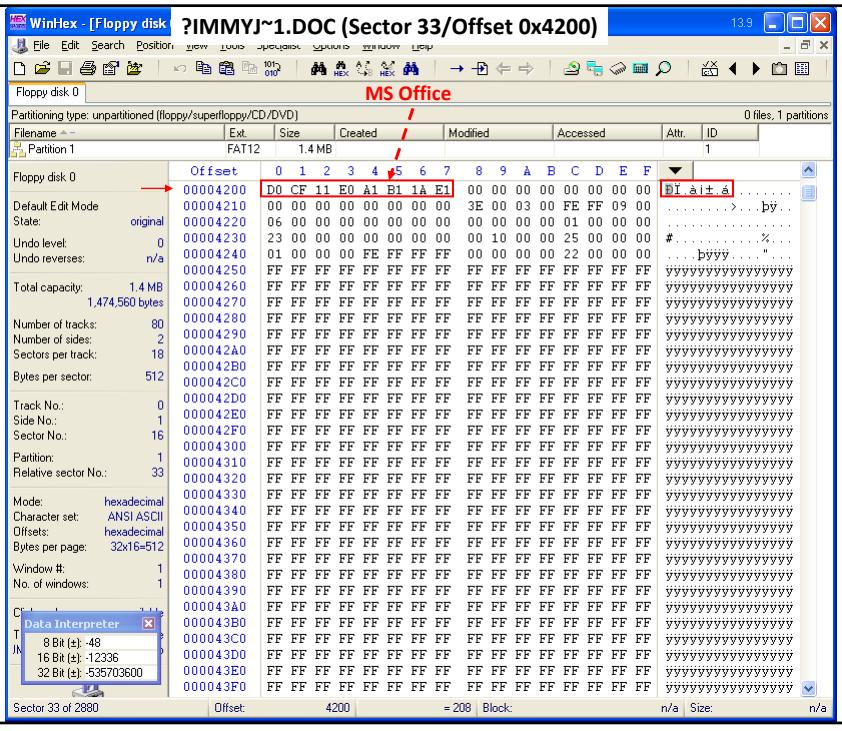

# The Actual Analysis

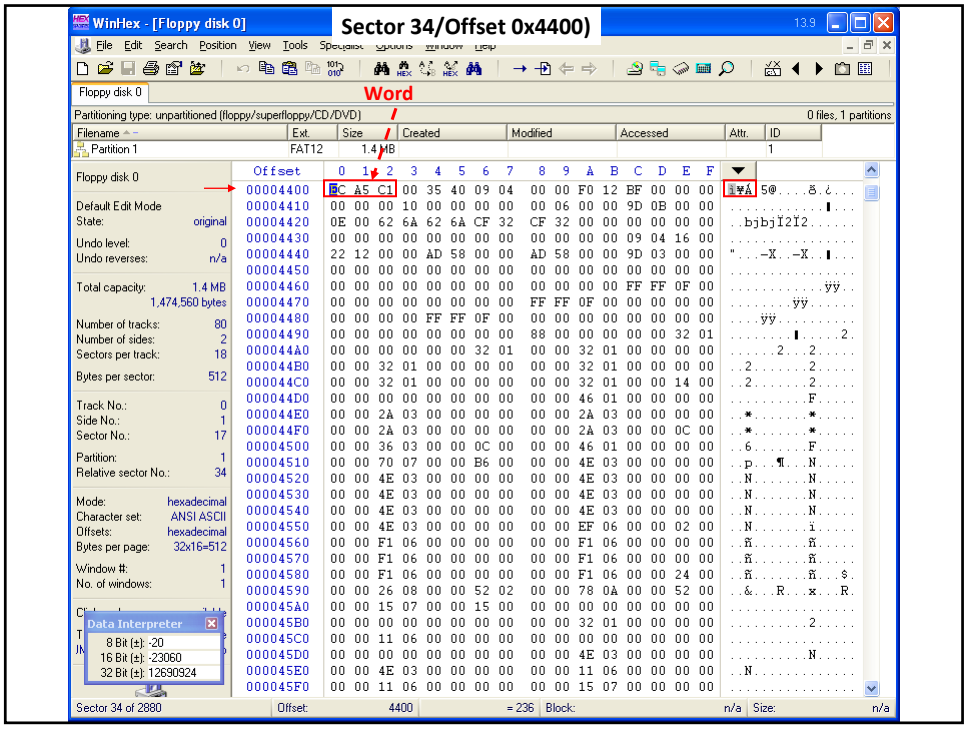
- We know that there are at least three files on the floppy. How do we get to them?
  - ?IMMYJ~1.DOC
  - ?OVERP~1.JPG
  - SCHEDU~1.EXE

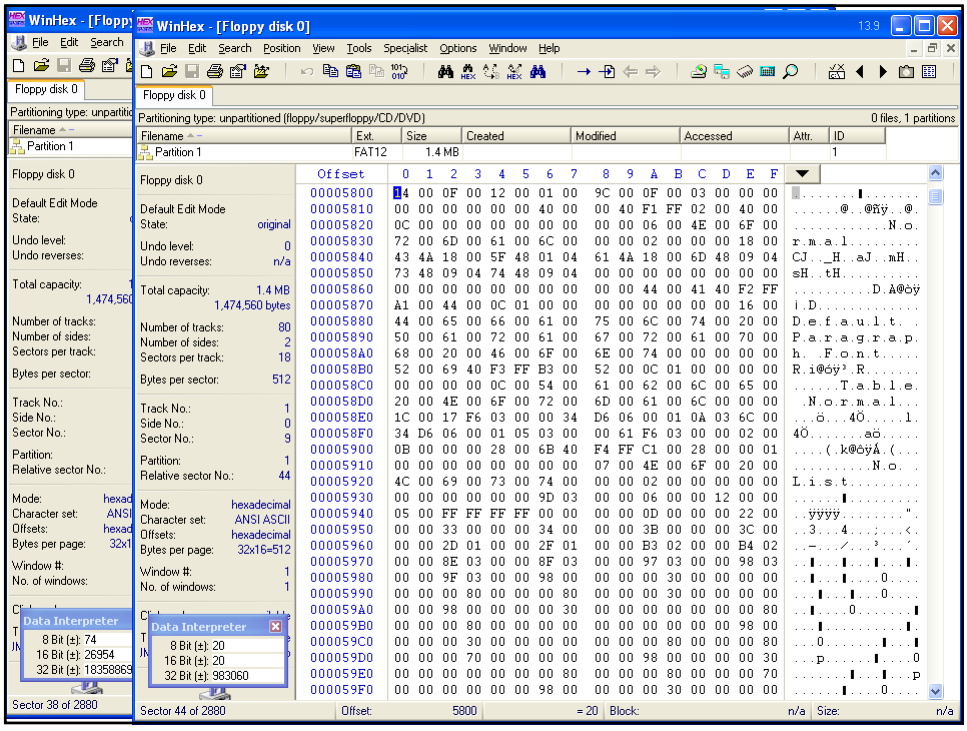
# ?IMMYJ~1.DOC

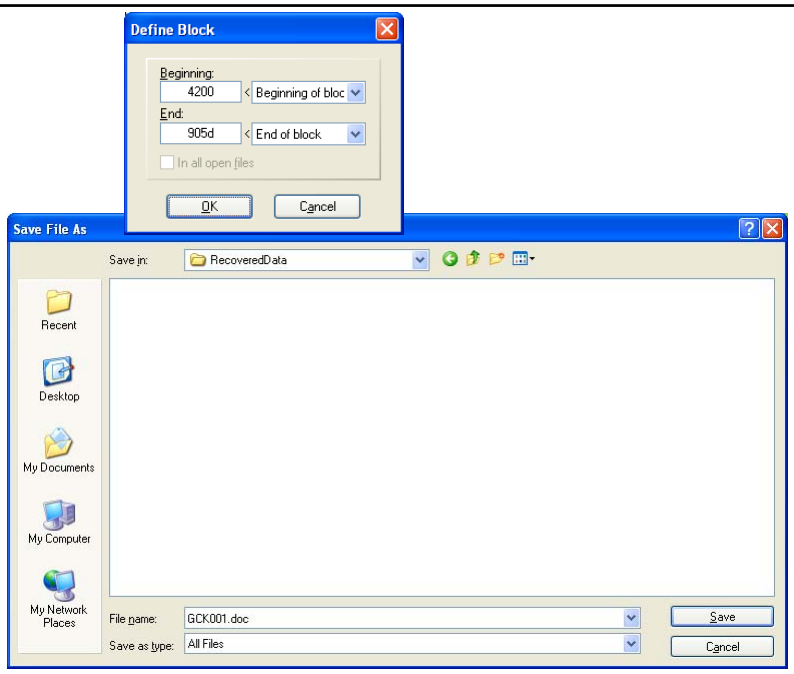
- The Root Directory tells us that the file starts at cluster 2 (i.e., sector 33, offset 0x4200)
  - » Scrolling down through the disk or image file would tell us the save thing
- Pertinent information:
  - » 0x4200: File header for MS Office document
  - » 0x4400: File subheader for Word file
  - » 0x905D: End of data
- Some text and a lot of metadata found!




Sector 34/Offset 0x4400)
Word
WinHex - [Floppy disk 0]
Floppy disk 0
Partitioning type: unpartitioned (floppy/superfloppy/CD/DVD)
Partition 1
FAT12
1.4 MB
Data Interpreter
8 Bit (±): -20
16 Bit (±): -23060
32 Bit (±): 12690924
Sector 34 of 2880


WinHex - [Floppy disk 0]
Floppy disk 0
Partitioning type: unpartitioned (floppy/superfloppy/CD/DVD)
Partition 1
FAT12
1.4 MB
Data Interpreter
8 Bit (±): 20
16 Bit (±): 20
32 Bit (±): 983060
Sector 44 of 2880



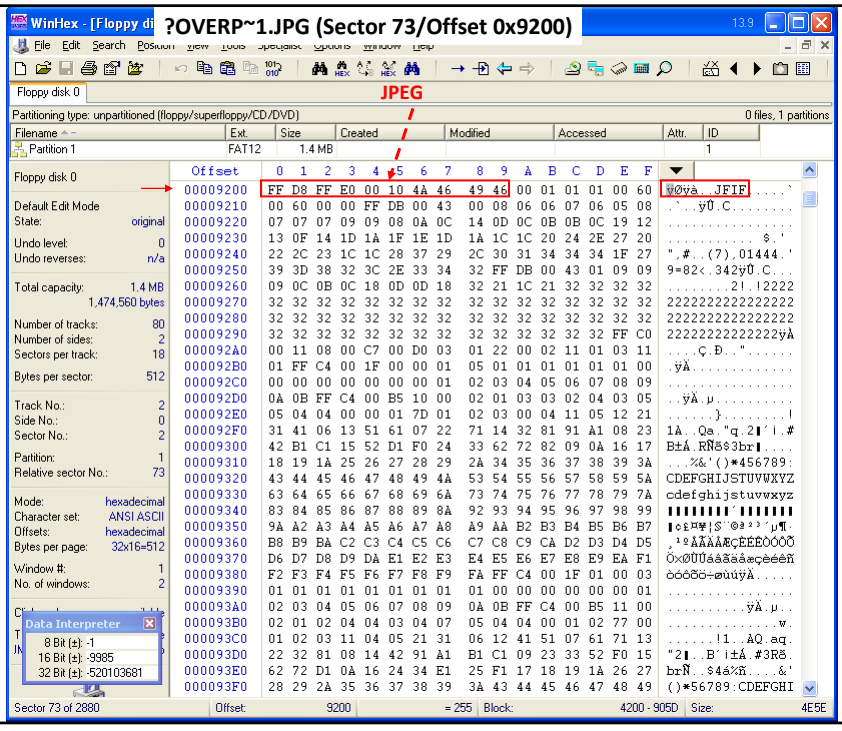
# The Incriminating Letter

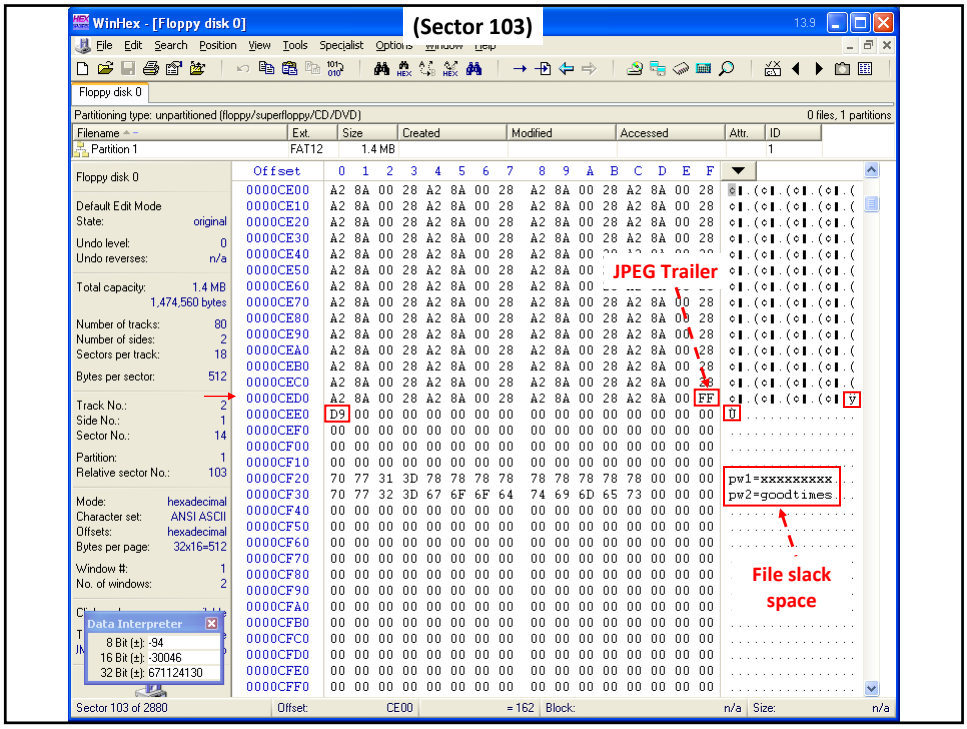

 

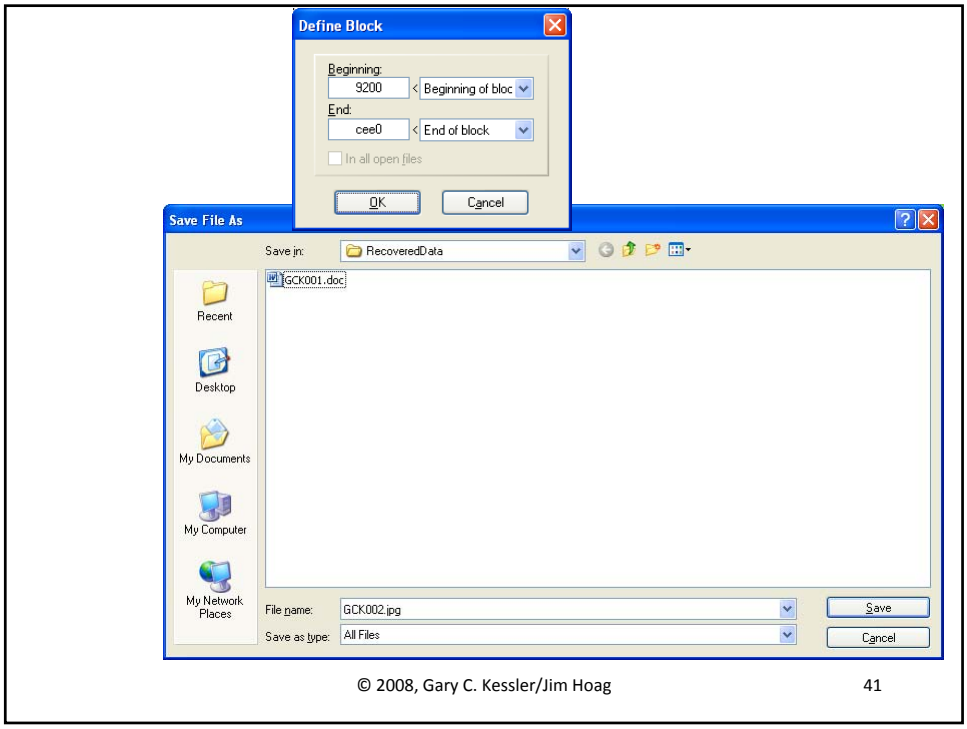
# ?OVERP~1.JPG

- The Root Directory tells us that the file starts at cluster 42 (i.e., sector 73, offset 0x9200)
  - » Scrolling down through the disk or image file would tell us the save thing
- Pertinent information:
  - » 0x9200: File header for JPEG file
  - » 0xCEDF: Two-byte trailer for JPEG file
- Character strings found in file's slack space!!



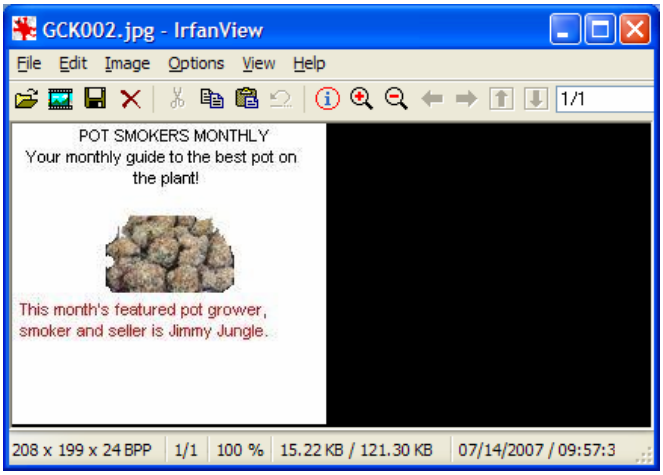





# The Incriminating Graphic



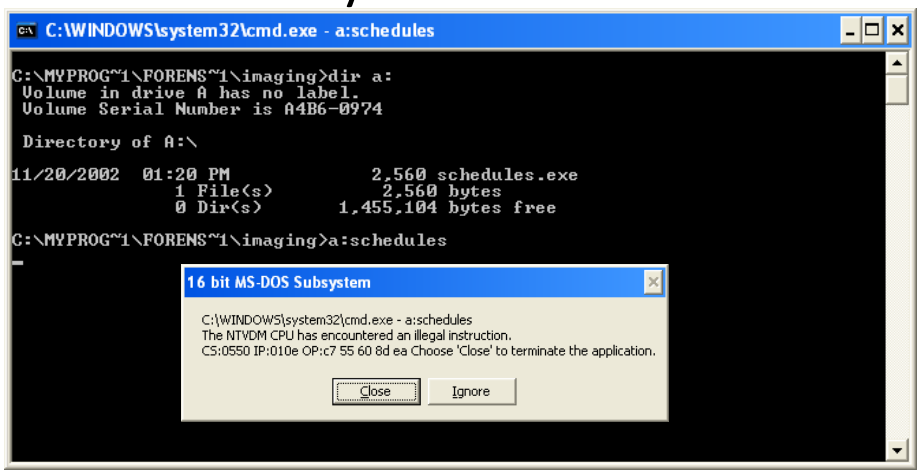

 

# SCHEDU~1.EXE

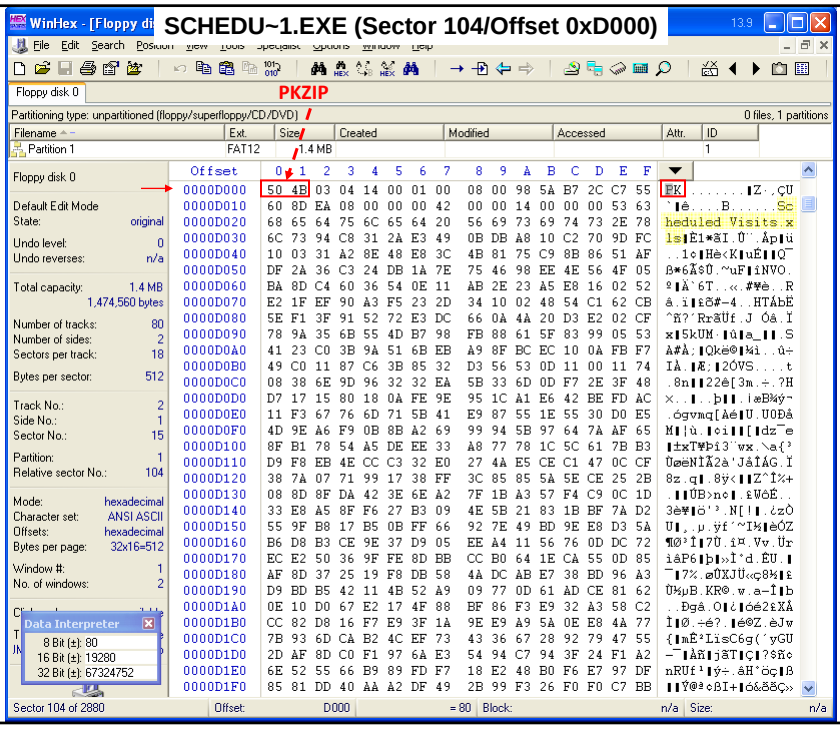
- The Root Directory tells us that the file starts at cluster 73 (i.e., sector 104, offset 0xD000)
  - » Scrolling down through the disk or image file would tell us the save thing
- Pertinent information:
  - » 0xD000: File header for ZIP file (!!!)
- Why is there a .EXE extension?
  - » Because anyone can rename a file!

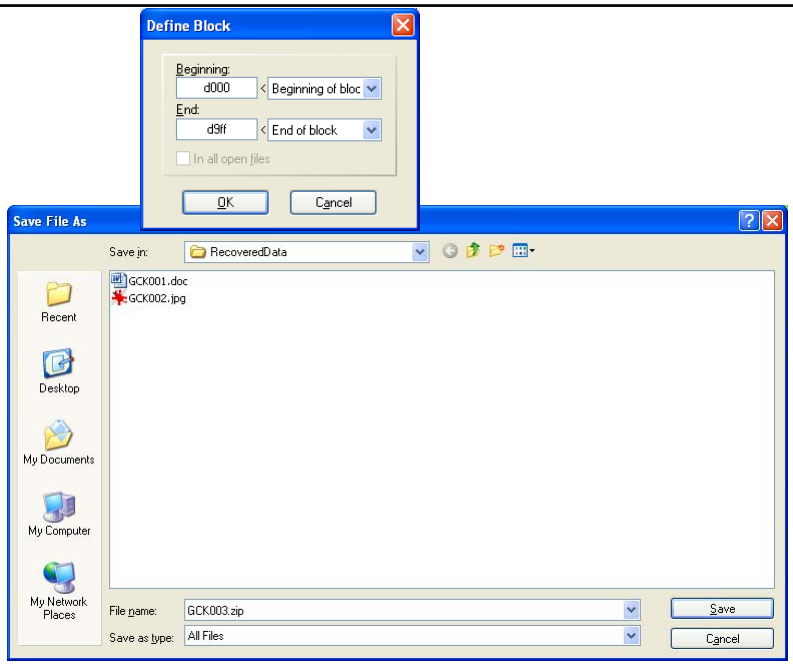
 

# Definitely Not Executable...



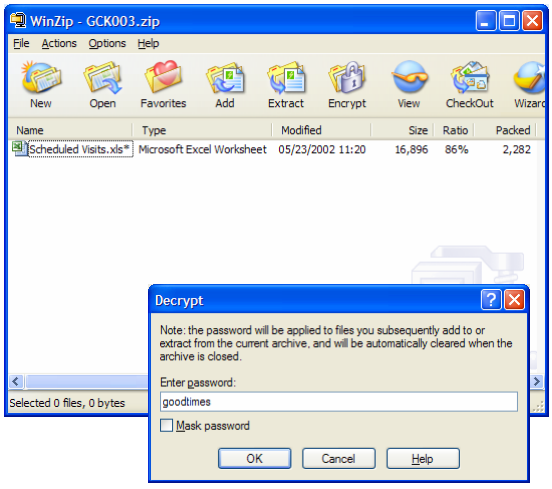

 






 46

# Opening With WinZIP




Opening the file with WinZIP shows the name of the compressed file, "Scheduled visits.xls". The "*" indicates that the file is password protected.

The recovered letter alludes to a previously used password:

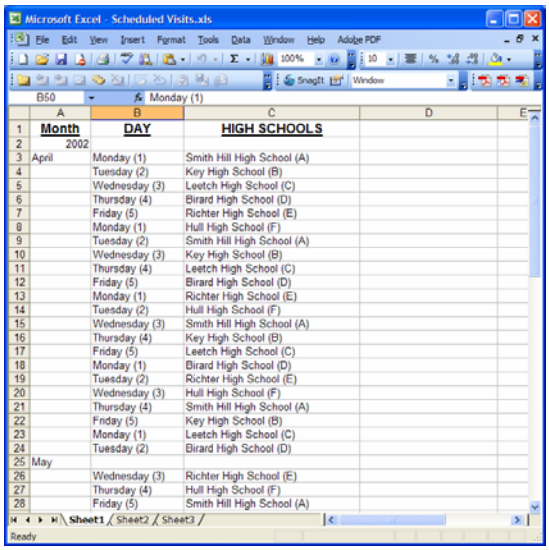
"I emailed you the schedule... To open it, use the same password that you sent me before with that file."

Recall our two character strings in the JPG file's slack space...

```
pw1=xxxxxxxxx
pw2=goodtimes
```

 47

# The Incriminating Spreadsheet




| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | **Month** | **DAY** | **HIGH SCHOOLS** | | |
| 2 | 2002 | | | | |
| 3 | April | Monday (1) | Smith Hill High School (A) | | |
| 4 | | Tuesday (2) | Key High School (B) | | |
| 5 | | Wednesday (3) | Leetch High School (C) | | |
| 6 | | Thursday (4) | Birard High School (D) | | |
| 7 | | Friday (5) | Richter High School (E) | | |
| 8 | | Monday (1) | Hull High School (F) | | |
| 9 | | Tuesday (2) | Smith Hill High School (A) | | |
| 10 | | Wednesday (3) | Key High School (B) | | |
| 11 | | Thursday (4) | Leetch High School (C) | | |
| 12 | | Friday (5) | Birard High School (D) | | |
| 13 | | Monday (1) | Richter High School (E) | | |
| 14 | | Tuesday (2) | Hull High School (F) | | |
| 15 | | Wednesday (3) | Smith Hill High School (A) | | |
| 16 | | Thursday (4) | Key High School (B) | | |
| 17 | | Friday (5) | Leetch High School (C) | | |
| 18 | | Monday (1) | Birard High School (D) | | |
| 19 | | Tuesday (2) | Richter High School (E) | | |
| 20 | | Wednesday (3) | Hull High School (F) | | |
| 21 | | Thursday (4) | Smith Hill High School (A) | | |
| 22 | | Friday (5) | Key High School (B) | | |
| 23 | | Monday (1) | Leetch High School (C) | | |
| 24 | | Tuesday (2) | Birard High School (D) | | |
| 25 | May | | | | |
| 26 | | Wednesday (3) | Richter High School (E) | | |
| 27 | | Thursday (4) | Hull High School (F) | | |
| 28 | | Friday (5) | Smith Hill High School (A) | | |




# Sector Assignments Finale

| Sector(s) | Address | Function |
|---|---|---|
| 0 | 0x0000-0x01FF | Boot Sector |
| 1-9 | 0x0200-0x13FF | File Allocation Table (primary) |
| 10-18 | 0x1400-0x25FF | File Allocation Table (secondary) |
| 19-32 | 0x2600-0x41FF | Root Directory |
| 33-72 | 0x4200-0x91FF | DATA: ?IMMYJ~1.DOC (deleted) |
| 73-103 | 0x9200-0xCFFF | DATA: ?OVERP~1.JPG (deleted) |
| 104-108 | 0xD000-0xD9FF | DATA: SCHEDU~1.EXE |
| 109-2878 | 0xDA00-0x167DFF | DATA: All bytes set to 0xF6 |
| 2879 | 0x167E00-0x167FFF | DATA: All bytes set to 0x00 |

## Lessons Learned

- In-depth analysis of a floppy can be done manually... just barely
  - » File carving
- Analysis of a large hard drive requires automated tools

## Project Files

- Digital Forensics Download Center
  - » `http://digitalforensics.champlain.edu/download/`
  - » dd
    - `dcfldd-1.3.4.x86win32.zip`
    - `dd-0.5.zip`
  - » rawrite: `rawrite2.zip`
  - » WinHex: `winhex.zip`
  - » Floppy disk image: `disk25.zip`
    - MD5: `391D4A30C29DA81D131DED2B1837B1A8`
- `http://www.garykessler.net/library/file_sigs.html`

# Other Hands-On Exercises

# Additional Hand-On Exercises

- Computer Forensics I
  - » Cell phone image analysis
  - » Web site and domain name information analysis
- Anti-Forensics and Network Forensics
  - » Photoshop and morphing images
  - » Cryptography/Steganography
  - » Packet sniffing
- Cybercrime
  - » Internet investigations
  - » E-mail, IRC, IM, and chat log analysis

# Hands-On Exercises Are Key

- Computer Forensics II
  - » Emphasis on reporting and building one's own tool kit
  - » Imaging and restoration exercises
    - Use of dd (and dcfldd), FTK Imager, and WinHex
    - Imaging CDs (multi-session) as well as "traditional" drives
  - » Analysis exercises using a variety of tools
    - Students build their own matrix of what software works best when doing what functions
  - » Live acquisition using Helix
    - Students image their own RAM to view contents

# Conclusion

- Non-lab hands-on exercises can augment live classes and enable online classes
- Properly designed hands-on execrises -- with appropriate technology support -- can work as well as or better than in-lab exercises
  - » Most "hands-on" activities can be done as easily in a dorm room or at home as in a lab setting

# Author Contact Information

**Gary C. Kessler, Ed.S., CCE, CISSP**
Associate Professor
Dir., Information Assurance Education & Research
Dir., Center for Digital Investigation
Champlain College
163 South Willard Street
Burlington, VT 05401

Adjunct Associate Professor
School of Computer and Information Science
Edith Cowan University
Joondalup, Western Australia

**office:** +1 802-865-6460
**cell:** +1 802-238-8913
**fax:** +1 802-865-6446
**e-mail:** gary.*kessler@champlain.edu*

*http://digitalforensics.champlain.edu*
*http://c3di.champlain.edu*
*http://www.garykessler.net*

**Jim Hoag, M.S.**
Assistant Professor
Prog. Dir., Computer Networking & Information Security
Champlain College
163 South Willard Street
Burlington, VT 05401

**office:** +1 802-865-6459
**fax:** +1 802-865-6446
**e-mail:** jhoag@*champlain.edu*

*http://cnis.champlain.edu*

**This work was partially supported by Grant No. 2006-DD-BX-0282 awarded by the Bureau of Justice Assistance. The Bureau of Justice Assistance is a component of the Office of Justice Programs, which also includes the Bureau of Justice Statistics, the National Institute of Justice, the Office of Juvenile Justice and Delinquency Prevention, and the Office for Victims of Crime. Points of view in this document are those of the author and do not represent the official position of the U.S. Department of Justice.**

# Acronyms and Abbreviations

| | |
|---|---|
| **ASCII** | **American Standard Code for Information Interchange** |
| **B** | **Byte** |
| **B/S** | **Boot sector** |
| **CD-ROM** | **Compact disc read-only memory** |
| **FAT** | **File allocation table** |
| **FRED** | **Forensic Recovery of Evidence Device** |
| **FTK** | **Forensic Toolkit (AccessData)** |
| **IM** | **Instant messaging** |
| **IRC** | **Internet Relay Chat** |
| **JPEG** | **Joint Photographic Experts Group** |
| **LFN** | **Long file name** |
| **MAC** | **Modified, access, and creation times** |
| **MB** | **Megabyte (millions of bytes, $2^{20}$)** |
| **MD5** | **Message Digest 5** |
| **MS** | **Microsoft** |
| **RAM** | **Random access memory** |
| **SFN** | **Short file name** |