Teaching IT Hardware Concepts Using Computer Forensics as a Motivator

Megan S. Conklin
Elon University
Campus Box 2126
Elon, NC 27244
(336) 229-4362
mconklin@elon.edu

ABSTRACT
This paper details experiences using computer forensics as a teaching tool to improve student performance and engagement in the "Introductory Hardware and Systems Software" course at a small, liberal arts institution. Students majoring in information systems often approach the hardware course expecting the standard lecture and textbook readings supplemented by an occasional hardware lab. Computer forensics, on the other hand, captures the interest of most students immediately. How to use hardware, software, and networking tools to help solve computer crimes is a problem most students can be easily motivated to investigate. This paper addresses whether the same hardware concepts can be taught using computer forensics as a motivator as would be taught in a "traditional" information technology (IT) hardware class, and whether there are any additional benefits to doing so. Metrics will include: student performance on examinations and an analysis of the breadth of subjects covered and the associated depth of the covered subjects in both scenarios. The discussion includes strategies for matching traditional IT hardware course content to related computer forensics techniques. Additionally, ideas for forensic lab activities in each subject area are outlined.

Categories and Subject Descriptors
B.0 [**Hardware**]: General.

K.3.2 [**Computers and Education**]: Computer and Information Science Education – *Computer science education, Information systems education.*

K.4.2 [**Computers and Society**]: Social Issues – *Abuse and crime involving computers.*

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – *unauthorized access, invasive software*.

Keywords
Computer forensics, IT education, curriculum development, hardware.

ISSN 1550-1469
SIGITE Newsletter, Volume 1, no. 2, 2004. Copyright 2004.

1. INTRODUCTION
The idea to use computer forensics as a motivator for teaching the traditional IT hardware course came after students expressed interest in additional lab content in the introductory-level hardware course for the Computer Information Systems (CIS) major at our small, liberal arts institution. Specifically, students expressed in one-on-one conversations and during informal labs how much they enjoyed problem-solving and being able to put their skills to use. At the same time, formal evaluations were starting to indicate a growing malaise with respect to the textbook readings and the format of lectures ("Powerpoints are boring" was a common complaint.)

The proposed solution of teaching the identical course material but using an integrated set of labs and problems from a single domain mirrors the use of case studies or final projects in other courses. Having a single, overarching *raison d'etre* in a course can provide focus and structure, obviously, but it can also have the additional benefit of providing constant and compelling motivation for why the material is important. In the case of this experiment, the question is whether using computer forensics (collecting and analyzing evidence of digital crimes) will prove to be a useful and motivating focal point for an IT hardware course.

This paper outlines the decision-making process and the planning that went into the execution of this experiment.

2. WHY COMPUTER FORENSICS?
Computer forensics refers to the discipline of collecting and analyzing the evidence needed to investigate or prosecute digital crimes. (Network or Internet forensics [1] is a related discipline.) As such, of great interest to computer forensics investigations is the developing body of knowledge surrounding how to keep evidence from being contaminated, how to serve as an expert witness, how to write reports, and how to interface with law enforcement officials. Many of the computer forensics textbooks deal extensively with such procedures [2] [6] [8] [9] [12].

However, the technical coverage of IT hardware issues in the textbooks varies. Conceptually, computer forensics touches every area of computer hardware from number systems to file management. Despite this, few textbooks had specific guidance on different ways to extract evidence from a hard drive, for instance. Berghel notes in [1] that the practitioners of computer forensics are often law enforcement officials themselves, rather than IT professionals. So perhaps the textbooks are being used predominately in criminal justice classrooms, not IT hardware

classrooms. Since the focus of this class was to use only forensics as a motivator and a framework for problem solving, it was very important not to dilute the hardware content with interesting, but ultimately extraneous, material covering digital evidence preservation and serving as an expert witness, for example.

3. COURSE DEVELOPMENT

3.1 Traditional IT Hardware Course

The traditional course was taught a full year before the forensics-based course in the same twice-weekly 100-minute time slot, and used a combination of lecture, textbook readings, and lab work to deliver material covering eight core IT hardware concepts:

- General Computing Concepts
- Number Systems
- Data Formats
- CPU and Memory
- Storage Devices
- Input-Output and Peripheral Devices
- Operating Systems Concepts
- File Management Concepts

The textbook [5] used in this course included 18 chapters, of which 13 were chosen to be included in the course. Labs were designed by the instructor, and drew heavily from such general resources such as [7] and [10]. Students displayed an alarming unfamiliarity with the command line, so [10] was particularly helpful in designing labs that introduced the command line. Examples of lab (or other non-lecture content) follow:

Table 1

Content Area	Example of Lab Content
General Concepts	Watching *Triumph of the Nerds* [3] documentary in class and had discussion of major themes (PC revolution, "killer app", "great artists steal", etc)
Number Systems	Team-based problem sets: binary-hex-octal-decimal conversion; binary addition.
Data Formats	Looking at different file headers in hex editor; buying used punch cards on auction web site and deciphering the encoded data.
CPU / Memory	Role playing the fetch-execute cycle (students play parts of ALU, control unit, program counter, registers, IO interface).
Storage Devices	Using DOS debug to investigate MBR, DET. Formatting and partitioning hard drives. Taking apart and comparing the inside of hard drives, floppy disks and drives, differences between Winchester and Bernoulli drives.
IO/Peripherals	Inventing a role-play example showing the difference between interrupts and polling.
OS Concepts	Professor institutes "Command line only" day where students have a variety of tasks to perform using only the command line (no GUIs).
File Mgt Concepts	Testing file fragmentation for various block sizes.

3.2 IT Course Based on Computer Forensics

The intention of the forensics-based course was to teach the same eight course concepts in the same time slot, but using computer forensics to tie the majority of labs together and to give an overall "theme" to the class.

The textbooks [6] [8] chosen for the forensics class reflected this change in focus. Table 2 shows that hands-on lab content also changed substantially in order to concentrate on the forensics material (as did the numbers of contact hours – see Section 4 for a breakdown of contact hours for each content area).

Table 2

Content Area	Example of Lab Content
General Concepts	Watching *Triumph of the Nerds* documentary outside of class and writing short paper on one theme.
Number Systems	Same team-based problem sets, but with fewer problems. Additional coverage of using a hex editor.
Data Formats	Same labs, plus additional coverage of Unicode, the digitization process.
CPU / Memory	Same labs, plus additional lab coverage includes using debug to look at memory addresses, using forensic tools to save and examine RAM slack.
Storage Devices	Same labs, plus additional coverage of restoring free space and slack space, using forensic tools to examine disk at the byte level. Making an image of a disk (fixed or removable).
IO/Peripherals	No labs. Coverage is lecture-only.
OS Concepts	No labs. Coverage of command line is moved to Storage and File Mgt units.
File Mgt Concepts	Using hex editor to repair (intentionally) damaged file headers. Finding the start and end cluster for a file in free space.

The primary rationale for moving or removing lab content in key areas like IO and OS was either because it was hypothesized that the material was covered elsewhere in the forensics labs (OS), or because the lab wasn't critical to student understanding of the material (IO). In most cases, the lecture content was simply reduced in favor of lab content and additional readings (Number Formats, Data Formats). In other cases, the in-class contact time was used more carefully: the documentary [3] was moved to an out-of-class assignment, for example.

In addition to the change in textbooks and the composition of planned contact hours, six forensics PCs and a small lab space were procured for this experiment. The forensics PCs were taken apart and rebuilt by the students on the second day of class, and outfitted with Windows 98. Working in teams, the students then configured these PCs with a start-up menu so they could boot to DOS and install and run the forensics software [4] that came with the textbook [9] and other required packages (hex editor, etc).

4. RESULTS

Primary metrics for judging success in this experiment will be measures of student performance on exams and the instructor's

assessment of the breadth and depth of material covered. Secondary measures of success will include comparisons of student course evaluations, course enrollment figures, and student attendance. These metrics were harder to evaluate, potentially less reliable, and more difficult to ensure an "apples to apples" comparison.

4.1 Primary Measures of Success

4.1.1 Exams

A substantial section of the exams in both courses included the same objective questions, so that student performance on a consistent set of learning objectives could be easily measured from year to year. However, some of the questions were worded slightly differently and answer choices were modified slightly so as to discourage students from attempting to memorize a previous year's exam, should they manage to procure one. (Hard copies of exam questions were also collected at the end of each exam to discourage test banks. Instead, students were given a practice exam and a list of topic areas to study for the exam.)

Table 3 shows the concepts that were tested in both exams. There were 65 questions in common over these eight subject areas. The expectation with the forensics class was that students would perform better in the area of Storage Devices, File Management, and Data Formats, and would stay about the same in all other categories.

Table 3. Comparative Performance on Exams: Objective Questions Only

Content Area	Traditional % Correct	Forensics-Based % Correct	Delta
General Concepts	87%	85%	-2%
Number Systems	66%	62%	-4%
Data Formats	74%	75%	+1%
CPU / Memory	58%	63%	5%
Storage Devices	60%	82%	22%
IO/Peripheral	80%	77%	-3%
OS Concepts	71%	70%	-1%
File Mgt Concepts	71%	79%	8%

The result of the common questions experiment showed that indeed the students taking the forensics-based course answered correctly a higher percentage of questions about Storage Devices and File Management. The improvement in the Storage Devices section was particularly encouraging. Their performance on General Computing Concepts actually showed a decline, however, and their performance on the CPU and Memory section was actually much better than expected. The lack of correct answers in the General Computing Concepts category could be attributed to the difference in textbooks, or it could be attributed to the fewer labs in this area at the beginning of the semester (to make room for additional forensics-based labs in the middle and end of the semester). The improvements in CPU and Memory could be attributed to the use of forensics labs, which covered how topics such as how to access RAM slack space, and how to use DOS debug for debugging memory.

4.1.2 Depth/breadth

The comparative performance of students on exams also yielded important insights into how the course content changed with the additional forensics-based material. Although the course textbook changed and some of the lab-lecture content changed for logistical reasons (i.e. the need to add some basic information about the discipline and practice of computer forensics), the intention of the experiment was to teach the identical core concepts of an IT hardware course, but use a different lab delivery methodology. In practice, this seems to have not been entirely possible. Table 4 shows in minutes how the course lecture and lab content changed from the traditional course to the forensics-based course. (Each class period is exactly 100 minutes, so a lecture-lab combination class would typically be 40 minutes of lecture and 60 minutes of lab work in the same class period. 50 minutes would constitute a half-period of class.) This table only shows the change in minutes between the two instances of the course.

Table 4: Comparative Use of Course Contact Hours

Content Area	Delta: Lecture Time (minutes)	Delta: Lab Time (minutes)
General Concepts	0	-150
Number Systems	-50	0
Data Formats	-25	+25
CPU / Memory	0	+50
Storage Devices	0	+200
IO/Peripheral	-40	-60
OS Concepts	-40	-60
File Mgt Concepts	+40	+60
Forensics Basics	+50	0

As shown in Table 4, the addition of time-intensive hands-on labs at the middle and end of the semester actually reduced the amount of time spent at the beginning of the semester in the General Concepts, Number Systems, and Data Formats topic areas. These are important foundational concepts that were unfortunately reduced in scope in order to make room for the additional forensics-based labs at the end of the semester. The original intention of the course was to teach the same concepts but in a different format – for example, instead of learning about hexadecimal notation by listening to a lecture, reading a few pages, and answering a problem set, students would learn through the use of a hex editor to repair broken image file headers or through finding and extracting particular bytes from a file on disk using DOS debug. However, because the forensics labs started later in the semester (students had to know something about storage devices and file management in order to be able to do even the basic forensics labs), there was no clear intersection between the Number Systems concepts taught in lecture and the forensics lab that happened five weeks later. The results were

discouraging: a 4% decline in performance on Number Systems in the objective portions of the exams, perhaps due to fewer contact hours used for discussion and instructor-led problem-solving in this important area, and perhaps due to the lack of timely, well-integrated labs.

The student performance on the CPU and Memory portion of the class seems to bear this out: because of an increased number of hands-on lab minutes in this content area, performance on exam questions seems to have risen. This assumes the same number of lecture hours.

As shown in Table 4, the biggest change in course content came through the addition of the forensics-based labs covering magnetic drive storage and file management topics. These labs covered concepts such as restoring deleted files, carving out deleted or corrupted files from free space or slack space on the drive, examining the data stored in particular clusters on the drive, repairing damaged file headers, and using extra bytes in a file to store secret messages (steganography). The data in Table 4 suggests that the additional time spent in the areas of disk storage and file management - specifically in the hands-on labs - was critical to the students' improved performance on the objective examination questions in these two areas.

The original intention of the experiment was not to detract from other course content areas in order to increase the student understanding in these two areas. However, judging from the slightly worsened performance in four of the eight content areas, this may have actually happened despite best intentions.

4.2 Secondary Measures of Success

Using official university student course evaluations to judge the success of an experimental course can be a problematic undertaking, since the primary goal of these surveys is often not the evaluation of the course but rather the evaluation of the professor. Similar problems exist with using course enrollment figures. These can be affected by time of day that the course is offered, professor teaching the course, and the number of currently enrolled majors in a particular discipline. For this particular comparison, the course was offered in the same time slot in both the traditional and forensics-based course. The same professor offered the courses in the same semester, but one year apart. Course enrollment figures were slightly higher in the forensics-based hardware course than in the previous traditional hardware course. Unfortunately, course evaluations for the forensics-based course were still unavailable at the time of this publication.

Lacking these quantitative measurements of student satisfaction (no matter how dubious their worth in any particular case), it is tempting to rely on empirical, observed factors such as "perceived student engagement" or "excitement surrounding the class". In this case, attendance figures could be useful: student attendance was markedly improved in the forensics-based course as compared to the traditional hardware course. Student attendance is 1.2 absences per student in the forensics course, as compared to 2.4 absences per student in the traditional course. This is despite having an attendance policy and "participation points" in the traditional course and *no attendance policy* and *no participation points* in the forensics course (professor takes attendance, but it is not used in any way for grading). Since both courses were taught in the same semester in the same time slot, it is possible that the increased attendance is due to the informality of the forensics lab environment and the general student "buzz" over the subject material. In order to measure this effect more precisely, perhaps a survey of the student affective domain would have been appropriate.

5. CONCLUSIONS

This experiment could be classified as a qualified success. There were certain aspects of the forensics-based course that were clearly victories: the improvements in student performance on the Storage Devices and File Management, for instance. However, it could be argued that these victories came at the expense of four other equally important concept areas.

While significant data was gathered for comparing student performance on objective examinations, the experiment did not adequately measure student satisfaction before and after the changes. Similarly, there is no data on instructor satisfaction with the forensics-based course. This could be measured in hours spent preparing for labs, assessment of software efficacy, and assessment of student engagement in labs, as well as performance of students in un-graded labs. These would be useful metrics to have in a future experiment.

6. REFERENCES

[1] Berghel, H. The Discipline of Internet Forensics. *Communications of the ACM, 46, 8* (August 2003), 15-20.

[2] Casey, E. *Handbook of Computer Crime Investigation: Forensic Tools & Technology*. Academic Press, New York, NY, 2001.

[3] Cringely, Robert X. *Triumph of the Nerds*. Ambrose Video, New York, NY, 1996.

[4] Digital Intelligence. DriveSpy and Image software (evaluation versions). http://www.digitalintel.com

[5] Englander, I. *The Architecture of Computer Hardware and Systems Software: An Information Technology Approach, 3rd Ed*. John Wiley & Sons, 2003.

[6] Kruse, W., Heiser, J. *Computer Forensics: Incident Response Essentials*. Addison-Wesley, Boston, MA, 2001.

[7] Meyers, M. *Introduction to PC Hardware and Troubleshooting.* McGraw-Hill/Osborne, New York, NY, 2003.

[8] Mohay, G. *Computer and Intrusion Forensics*. Artech House, 2003.

[9] Nelson, B., Phillips, A., Enfinger, F., Steuart, C. *Guide to Computer Forensics and Investigations*. Course Technology, Boston, MA, 2004.

[10] Simrin, S. *MS-DOS Bible, 4th Ed*. Sams Publishing, Indianapolis, IN, 1991.

[11] Troell, L. Pan, Y., Stackpole, B. Forensic Course Development. In *Proceedings of the 4th Conference on Information Technology Education (CITC4'03)* (Lafayette, Indiana, USA, 2003). ACM Press, NY, 2003, 265-269.

[12] Vacca, J.R. *Computer Forensics: Computer Crime Scene Investigation*. Charles River Media, Hingham, MA, 2002.