Integration of Computer Security Laboratories into Computer Architecture Courses to Enhance Undergraduate Education

Jayantha Herath, Susantha Herath, Ajantha Herath*
St. Cloud State University, St. Cloud, MN 56301
*University of Dubuque, Dubuque, IA 52807
jherath@stcloudstate.edu

Abstract

Most computer science and engineering programs have two or more required computer architecture courses but lack suitable interfacing laboratory experience for other upper-level classes. Information assurance and network security tracks have been developed over the recent years without providing necessary and sufficient background knowledge in logic, storages and processor architecture. Integration of real-world applications is always a better approach to not only to excite the passive student body but also to explore the computer architecture subject area. At the intermediate level, architecture knowledge can be extended to provide information and network security experiences to students. Such extensions to the course will provide proper interfacing to networking, operating systems, databases and other senior level security related courses. This paper describes possible integration of security and privacy concepts into computer architecture course sequence with hands-on classroom activities, laboratories and web-based assignments.

1. Introduction

One day a tenured Professor of Medicine received an e-mail from a close friend via Yahoo e-mail, with an attachment file. He opened this e-mail with other messages, and continued to work as usual. The attachment contained a worm that causes automatic transmission of more e-mail messages. A few days later his computer was confiscated, his supervisors accused him of creating and transmitting a virus from his computer, and the FBI was called in. His entire address list had received e-mails with a worm, automatically. One and half years later he found himself indicted by a grand jury for violating Federal law 18 USC 1030 a 5 (a). The grand Jury charged that this Professor "knowingly caused the transmission of a program, information, code or command, and as a result of such conduct, did intentionally cause damage without authorization to a protected computer, which is used in interstate and foreign commerce and communication, and, by such conduct, caused loss to one or more persons during a one-year period aggregating at least $5000.00 in value". The Professor had to find an attorney to represent himself. Many attorneys asked him to pay $150,000 and another asked for $60,000 upfront and $450 per hour to represent him in this case.

The case above illustrates the complexity of conflicting technical and legal issues. Interestingly, most lawyers and judges do not understand the technologies involved and need help from technical expert witnesses to find out what happened in the computer using the forensic evidence available in the storage. And often the technical experts do not know much about the legal issues involved. Finding the hidden evidence in storage systems and presenting it in an acceptable form to the court is a hard problem for a computer architect to solve. For this reason, it is important to provide basic computer forensic techniques in a computer architecture setting, to provide a way to recover information in storage systems that ensures its integrity.

In general, knowledge gained in computer architecture courses serves as the gateway for upper level undergraduate computer science courses. The main objective of this study is to develop computer architecture course modules with computer security applications for undergraduate students. One of the challenges facing us in the classroom is finding experiments to engage the interests of students while improving the quality of computer architecture courses [1]. The goal has been and continues to be helping them become good computer scientists in a relatively short period of time with a solid grounding in both theoretical understanding and practical skills so that they can enter the profession and make valuable contributions to the society. The proposed active learning modules aim to provide students with an exciting learning environment and the necessary tools and training to become proficient in the computer architecture subject matter with applications in security and privacy. The following sections outline the details of course plan, examples, assessment plan, future work and summary.

2. Detailed Course Plan

To help master computer architectures, our curriculum provides three semester courses. The first course in this sequence covers the fundamentals of digital logic circuit design [2]. This foundation course helps the students develop component integration skills from gate-level to register-transfer-level, when designing a circuit to perform a particular task. The laboratories for this course consist of hardware and VHDL software simulations of combinational and sequential digital logic circuits. The intermediate level course introduces both complex instruction set and reduced instruction set processor architectures. The laboratories for this course consist of hardware and software simulations of basic programming constructs in CISC and RISC architectures. The third course focuses on advanced concepts in special purpose architectures to provide both depth and breadth to the subject matter. One could, conceivably, introduce security protocols for storage and system-on-a-chip related laboratories at this level.

Integrating Computer Architecture with Security

The focus of the intermediate-level course so far has been on implementing techniques of basic programming constructs such as I/O, arithmetic expressions, memory operations, register operations, if-else and switch conditional operations, for-while iterative computation controls, simple functions and recursive functions in several different instruction set processor architectures. This approach provides interfacing for cs-1 and cs-2 courses taken in the previous semesters. Increasing the performance of the processor by reducing program execution time is considered at the gate, register and functional levels of the processor design [3] [4]. At the end of the semester, students in this course design a pipeline processor using VHDL as their final project.

It is observed that learning processor architecture alone is insufficient at this level. Students should start to understand the importance of storage systems and applying classroom knowledge to solve real-life problems. A course sequence with extensions to security applications would help students develop such skills using several different architectures before their graduation. The following examples constitute a way to integrate real-life problem solving to this level of students. Example 1 illustrates a closed laboratory designed to help students understand the changes in the content of the memory. The open laboratory followed by this lab, a packet-sniffing example, would provide necessary interfacing with a network security course module. Example 3 shows an application of the knowledge acquired in logic-design class to provide interfacing with database security course module. It describes a simple logic extension used to break into database systems. Example 4 describes the application of the clock values in a legal issue. This could provide interface to secure operating system course module.

```
TUTOR 1.32> MS 2000 'ABCDEFGHIJKLMNOPWRSTUVWXYZ'

TUTOR  1.32> MS 2020 'abcdefghijklmnopwrstuvwxyz'

TUTOR  1.32> MS 2040 '0123456789'

MEMORY DISPLAY

TUTOR  1.32> MD 2000 256
002000    41 42 43 44 45 46 47 48  49 4A 4B 4C 4D 4E 4F 50  ABCDEFGHIJKLMNOP
002010    57 52 53 54 55 56 57 58  59 5A FF FF FF FF FF FF  WRSTUVWXYZ......
002020    61 62 63 64 65 66 67 68  69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
002030    77 72 73 74 75 76 77 78  79 7A FF FF FF FF FF FF  wrstuvwxyz......
002040    30 31 32 33 34 35 36 37  38 39 FF FF FF FF FF FF  0123456789......
....
002090    12 EB 00 13 12 EB 00 0E  12 FC 00 20 12 EA 00 02  .k...k...|. .j..
0020A0    12 EA 00 12 12 EA 00 03  12 EA 00 02 12 FC 00 21  .j...j...j...|.!
....
0020F0    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
```

Figure 1. Setting and Displaying the Content of Storage

Example-1

To understand how the processor operates, students need to recognize the contents of registers and memory, learn the limitations of instruction sets and how the basic programming constructs are implemented in processor architecture. Figure 1 depicts the first classroom activity used to help students understand the addresses and content of the memory. In this example, memory is set by using MS and the content of the memory is displayed using MD. In addition, students also perform translations of arithmetic expressions, data transfers, if-else-for-while control and recursive functions as in-class activity. They perform traces of registers and memory as an in-class activity using Motorola 68000 instruction set architecture.

Example -2

The objective of this experiment is to apply the knowledge acquired in interpreting content of memory displays in the previous experiment into network security applications. Network traffic is easy to capture and analyze using the tools available in the web. Network protocol analyzers, such as Ethereal Packet Sniffer, can be used to accumulate both incoming and outgoing network data [11] [12]. Most packet analyzers assemble all the packets in a TCP conversation and represent the data using tcpdump format. Students were asked to capture the user-id and password of their own email accounts using a packet analyzer. Also, they were instructed about the legal issues involved in packet capturing. After doing this experiment, one student observed that rediff login is not secure. However, he noted that hotmail.com is not only using secure login, but also encrypts the traffic. Hence, students could not identify their user-Id, password or message in that communication. Moreover, they found that encryption will significantly increase network traffic by observing the amount of data captured. This experiment would be a good interfacing for a Network-security class. Figure 2 shows the memory display of the captured data.

```
00000000  50 4f 53 54 20 2f 63 67  69 2d 62 69 6e 2f 6c 6f POST /cg i-bin/lo
00000010  67 69 6e 2e 63 67 69 20  48 54 54 50 2f 31 2e 31 gin.cgi  HTTP/1.1
00000020  0d 0a 41 63 63 65 70 74  3a 20 61 70 70 6c 69 63 ..Accept : applic
00000030  61 74 69 6f 6e 2f 76 6e  64 2e 6d 73 2d 65 78 63 ation/vn d.ms-exc
00000040  65 6c 2c 20 61 70 70 6c  69 63 61 74 69 6f 6e 2f el, appl ication/
00000050  6d 73 77 6f 72 64 2c 20  61 70 70 6c 69 63 61 74 msword,  applicat
```

Figure 2(a) Packet Sniffer Output - www.rediff.com

```
00000000  16 03 00 04 79 02 00 00  46 03 00 2f ed 29 44 2a ....y... F../.)D*
00000020  7a b2 b5 95 40 08 c3 74  ae 70 98 20 49 08 00 00 z...@..t .p. I...
00000030  82 32 61 be ad eb b1 27  ee 5e 93 e6 b3 1e ac 79 .2a....' .^.....y
00000040  7e 80 31 0b d2 2e b9 70  3b e5 55 b3 00 03 00 0b ~.1....p ;.U.....
00000050  00 03 5a 00 03 57 00 03  54 30 82 03 50 30 82 02 ..Z..W.. T0..P0..
00000060  bd a0 03 02 01 02 02 10  3c f4 4e cc 7b c3 e6 34 ........ <.N.{..4
00000070  b0 3f 2d 8e b8 78 41 27  30 0d 06 09 2a 86 48 86 .?-..xA' 0...*.H.
00000080  f7 0d 01 01 05 05 00 30  5f 31 0b 30 09 06 03 55 .......0 _1.0...U
```

Figure 2 (b) Packet Sniffer Output - www.hotmail.com

Example -3

In general, database systems track their own users before allowing the access. Access to database systems is controlled using a user-id and a password for legitimate users. However, some databases can be attacked without a valid user-id and password. Such attacks can be performed by applying the knowledge of Boolean logic expressions learned in an introductory digital-logic design class. In one such successful database intrusions, the attacker entered the system by converting user-id and password expressions combined with one AND operation into two arbitrary expressions combined with an AND operation followed by an OR operation with an expression that always evaluated to a True value [8][9].

Example -4

Often legal experts have difficulties interpreting the forensic data available in the memories. Utah vs. Payne presents an interesting application of clock values. In this case, the prosecution presented data to the courts without analyzing its content. The defense team accurately analyzed the data column that represented clock values to illustrate the time line of events. The defendant was found not guilty [10].

Other Architecture-Security Applications

In addition to the network-database-security experiments described above, it is possible to search, develop and integrate introductory experiments for intrusion detection, forensic analysis of storages, sanitizing storages, web security and network security. Providing hardware and software support, and specifying protocols to make the processor and storage secure can also be considered at any stage of the course sequence. The most time-consuming task in solving security issues, similar to the one described in the introductory part of this paper, will be the postmortem analysis. There is a need to perform forensic analysis of the data in the storage to determine the source of e-mail transmissions and decode other communications. A computer architecture course with security and privacy

related applications, such as the ones described above, and protocols to provide security and privacy to storage will enhance students' higher level skills: teamwork, analysis, synthesis and active participation in the classroom. These course modules will help students learn architectural concepts in an active learning environment, thus providing students an opportunity to function well in an increasingly competitive society in which security is highest priority.

Difficulties

Incorporating security and privacy related issues in many subject areas into the architecture course may overload the students and faculty. Selecting a series of projects that increase enthusiasm among a diverse body of students will not be an easy task. To overcome this difficulty, we plan to work with information security faculty to develop suitable laboratory experiments.

Course Assessment

Once developed, the course material can be evaluated by soliciting criticism from faculty and students. Student learning can be evaluated in many different ways. Background knowledge can be performed in the form of a simple questionnaire/worksheet that the students fill out prior to completing the lab assignments. Students will be asked to explain the concepts they learned. Recording experiences from laboratory assignments is an essential part of student work. Group-work evaluations will also be used to assess the course. The faculty and teaching assistants regularly observe the team work. There are opportunities to test course materials within a large university system that could possibly extend use to other faculty and students.

Information Security Symposium

Two computer security symposiums were organized at the end of the Spring 2003 semester to stimulate our students, computer science, information systems and engineering faculty in five neighboring states as well as businesses and industry. Invited speakers from West Point Military Academy, Carnegie Melon University, University of Idaho, the University of Minnesota, the University of Iowa, the University of Wisconsin and the University of North Dakota delivered lectures based on their work. The symposium was well attended by students, faculty and industry representatives. These symposiums helped our efforts to develop a curriculum emphasizing secure storages, forensics, network-database security that presents an integrated view of hardware, software and security issues to the undergraduate students [6][7].

3. Summary and Future Work

Traditionally, computer architecture courses are presented to a less-than-enthusiastic student body and often delivered without indicating real world applications in a relatively passive classroom environment. One of the reasons for diminished student interest of learning the subject is poor interfacing with other courses in the curriculum. In general, learning takes place if the student can both integrate what he is learning in the classroom into real-life applications and understand how the subject pertains to learning other subjects in the degree program. To promote this in the classroom and to overcome the above-mentioned deficiencies, an intermediate computer architecture course can be developed with hands-on classroom activities and laboratories involving architecture and security. Computer security issues are important for businesses, industries and government. There is a need for increased computer architecture education with security and privacy emphasis through undergraduate level computer science curricula. This need can be met through several stages. Computer architecture laboratories should have application interfaces to other courses in the curriculum such as operating systems, network, databse security. And further training of students can be accomplished by developing mathematics, operating systems, networking and database courses with an emphasis on security. Regular regional symposiums are very helpful to share the laboratory and curriculum development efforts with other colleges and universities. The availability of properly designed and developed course materials, with a series of hands-on laboratories as well as classroom activities, will reduce both instructors' preparation time and multiple copying stages, and increased students' ability to absorb the subject matter. Developing a series of system-on-chip experiments and/or security protocols for storages would be useful for the laboratories in the third course of the sequence. These objectives can be accomplished through the optimal use of available resources. Through such platforms, students will learn to appreciate instruction set architecture.

Acknowledgments

Our computer architecture project has been supported by the MnSCU Center for Teaching and Learning through the Bush/MnSCU Learning by Doing Program. Many students helped to make our extensions to the courses successful. We would like to specially thank to Professor Larry Grover, Dr. Splittgerber, Erik Cramer, Mark Ebersole, WeiShin, Mohammad Khan, Amit Parnerkar for helping us throughout the years to accumulate experimental data.

4. References

1. Gehringer, E.F. A Web-Based Computer Architecture Course Database,

http://www.csc.ncsu.edu/eos/users/e/efg/archdb/FIE/2000CACDPaper.pdf
2. Computer Architecture I http://web.stcloudstate.edu/jherath/CompArch-1
3. Hennessy, J. L., Patterson, D. A. Computer Organization and Design: Hardware/Software Interface, Second Edition, 1997 http://www.mkp.com
4. Computer Architecture II http://web.stcloudstate.edu/jherath/CompArch-2
5. Hardware/Software Interfacing for High Performance Symposium -02 http://web.stcloudstate.edu/jherath/Conference.htm
6. Symposium on Information Assurance and Security -2003 http://web.stcloudstate.edu/sherath/SIAS2003
7. Information and Network Security Workshop-2003 http://web.stcloudstate.edu/sherath/INSW2003
8. Pfleeger, Security in Computing, http://www.prenhall.com
9. http://www.cs.uwec.edu/~wagnerpj/security/
10. http://www.simson.net/2002-11-Forensics.ppt
11. http://winpcap.mirror.ethereal.com/install/default.htm
12. http://www.ethereal.com/distribution/win32/