Approaches to Undergraduate Instruction in Computer Security

Luiz Felipe Perrone[†], Maurice Aburdene[‡], and Xiannong Meng[†]
[†]Dept. of Computer Science / [‡]Dept. of Electrical Engineering,
Bucknell University

Abstract

Although economies of scale have turned the networked computer into a commodity, its usability at large is determined by the levels of security and privacy the technology can offer. This phenomenon has created a new landscape in which the demand for trained professionals in computer security is extremely high. Colleges and universities are still adapting to this reality and different approaches to computer security instruction are being used throughout the world. Our main contributions in this paper are the identification and the analyses of three main categories of approaches to instruction in computer security: *single-course*, *track*, and *thread*. The single-course approach, which is highly popular, is one in which the student is offered a survey of several different topics in computer security in one course in the curriculum, often an elective. Although it provides considerable breadth of topics, it cannot provide depth since it is only introductory by design. In the track, concentration, or program approach the student takes a sequence of courses specialized in security and information assurance. The resources required to implement this approach are numerous and therefore it is not applicable to a wide variety of schools and departments. To illustrate the discussion of these first two approaches, we present an informal survey of courses and programs in computer security throughout the U.S. The thread approach is seldom advertised or implemented and is a compromise which bridges the gap between the single-course and the track approaches. This approach uses security and privacy as a unifying theme across the standard core Computer Science or Computer Engineering curricula. We argue that this approach can effectively meet the educational needs of the computer professional of today using a minimum of resources.

1. Introduction

The world is fast becoming a very large, inter-networked collection of computing devices. Your personal computer connects to the Internet; it may even do so wirelessly. Your contact information and your family pictures are shared with family and friends on a web page that resides in a server shared with a number of other users. Your car uses an anti-theft system that reports to a cellular network its current location obtained with a GPS device. Your satellite television system downloads software updates autonomously from up above. On-line merchants keep your credit card number on file, in a networked computer. You haven't been to a brick-and-

"Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2005, American Society for Engineering Education"

mortar bank in two years since all your transactions are done by phone, ATM, or the World-Wide-Web. Even though you manage to avoid intense feelings of paranoia most of the time, there are moments when you just have to stop and wonder how much this technology has made you vulnerable to the evil that man can do. As you spend time worrying, scientists and engineers, like those that made all this exciting technology possible, are hard at work creating mechanisms that may not make you safe in an absolute sense, but perhaps as safe as it can be managed. Some of these people have terminal degrees in their fields, Ph.D.'s and D.Sc.'s, though not all. Many more of them, in fact, never went beyond a bachelor's degree and may not have developed the level of expertise in computer security offered in graduate programs. Are you confident that these professionals with no more than an undergraduate degree have received in school all the training they need to keep you safe in this networked world? Perhaps you don't have that many reasons to be.

There currently exists an undeniable, compelling need for strong undergraduate instruction in computer security, information assurance, and privacy. Although the current standard curricula in Computer Science and Engineering is rife with recommendations for providing undergraduates with classroom instruction in topics in security, the emphasis in this area is arguably not finely attuned to the needs of today's reality. It is expected that the professionals in industry, in government, and in academia are trained to address, should have the capabilities to confront, and to mitigate the risks, the threats, and the vulnerabilities in software and hardware components in computer systems.

Education in computer security has historically matured much more rapidly in graduate programs than in undergraduate programs due to the formers' natural involvement in cutting-edge research and due to the nature of their mission. Undergraduate education in Computer Science and Engineering has first attempted to bridge this gap by offering specialized, elective courses in computer security in their degree programs. These courses often come later in a course sequence and, since they are not compulsory, not all students graduate taking with them the essential concepts and skills that the job market needs of them. Alternatively, select institutions have chosen to provide specialized undergraduate degree programs or tracks in computer security and information assurance. Although this approach serves well to educate the students with interests focused on the area, it doesn't satisfy the security learning needs of the broader student population in Computer Science and Engineering.

In this paper we discuss the two approaches to undergraduate instruction in computer security described above, to which we refer as the *single-course approach* and the *track approach*, respectively. We contrast these approaches with a third model of initiative, the *thread approach*, which emphasizes computer security across the core curriculum. The main thesis of this paper is the argument that the thread approach, while lacking the depth of learning provided by the track approach, can be more easily transferred across different kinds of institutions and can be more effective than the single-course approach.

The remainder of this paper is organized as follows. In Section 2, we establish the context for this paper by presenting a survey of the relevant literature. The *single-course approach*, the first of the three models of undergraduate education in computer security is discussed in Section 3. Next, the *track approach* is discussed in Section 4. The discussion of these first two approaches

Copyright © 2005, American Society for Engineering Education"

presents the results of preliminary surveys that expose the status quo of computer security education in institutions across the U.S. The *thread approach* is presented, discussed and analyzed in Section 5. Finally, in Section 6, we present our conclusions.

2. Background and Related Work

An extensive body of literature has been developed since the pioneering work in computer security education and many authors have written on the subject of curriculum development in the area. Frincke and Bishop [3] present an interesting overview of venues that serve as primary resources for one interested in joining this community. Arguably one of the most relevant forums in this area is National Colloquium for Information Systems Security Education (NCISSE) created in 1997 and later renamed as CISSE. The production presented at CISSE illustrates that academia, industry, and government are making orchestrated efforts to work together in defining the minimal skill set required of the Computer Science and Engineering professionals of today.

Professional societies in Computer Science and Engineering, which have historically made strong contributions in the development of curricular standards and recommendations, have also been emphasizing security education. The main forces in this field are the Association of Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS). Their *Joint Taskforce in Computing Curricula* identified in CC2001 [11] thirteen distinct areas of knowledge which define the core of Computer Science and Engineering programs: Discrete Structures (DS), Programming Fundamentals (PF), Algorithms and Complexity (AL), Architecture and Organization (AR), Operating Systems (OS), Net-Centric Computing (NC), Programming Languages (PL), Human-Computer Interaction (HC), Graphics and Visual Computing (GV), Intelligent Systems (IS), Information Management (IM), Social and Professional Issues (SP), Software Engineering (SE). It is fair to say that, except for DS, all of the identified core areas have an intersection with some topic in computer security, information assurance, and/or privacy. (In fact, CC2001 documents that CC1991 [6] identifies security as a recurring concept that is pervasive and persistent throughout Computer Science.)

Close inspection of CC2001 reveals recommendations that instruction in security be presented across the core undergraduate curriculum in Computer Science and Engineering. The descriptions of several core units have direct relationships with topics in security. The list below illustrates the most salient of these core units presented in the CC2001 document. Along with each unit's code and title, we indicate in parenthesis the minimum number of classroom hours recommended and the relationship of that unit to security, privacy, or information assurance:

- OS1 Overview of Operating Systems (2): The identification of potential threats to operating systems and potential threats and the security features design to guard against them.
- OS4 Operating Systems Principles (2): Mutual exclusion as a mechanism for the implementation of access control in trusted operating systems.
- OS5 Memory Management (5): Memory protection as a fundamental mechanism in the design of a trusted operating system.

Copyright © 2005, American Society for Engineering Education"

- NC3 Network Security (3): Fundamentals of cryptography, public-key and secret-key algorithms, authentication protocols, and digital signatures.
- PL2 Virtual Machines (1): Security issues arising from the execution of mobile code.
- PL4 Declarations and Types (3): Type checking as a tool to enhance the safety and the security of a computer program.
- IS2 Search and Constraint Satisfaction (5): Search heuristics as essential components in intelligent intrusion detection systems.
- IM1 Information Models and Systems (3): Information privacy, integrity, security, and preservation.
- SP4 Professional and Ethical Responsibilities (3): Computer usage policies and enforcement mechanisms.
- SP5 Risks and Liabilities of Computer Based Systems (2): Implications of software complexity, and risk assessment and management.
- SP7 Privacy and Civil Liberties: Study of computer based threats to privacy.
- SE6 Software Validation (3): Validation and testing of software systems.
- SE8 Software Project Management (3): Risk analysis and software quality assurance.

This list clearly indicates the taskforce's resolve to use security in CC2001 as a recurring theme *across the curriculum*, much in the same way that concepts such as layers of abstraction, efficiency, and complexity are used. The proposed curriculum model recommends a broad coverage of topics in security in several units and courses, but several questions on the efficacy of this approach remain and incite additional discussions and investigations:

1) If security is integrated throughout the curriculum as proposed, how would the measured learning outcomes in these topics fare?
2) If it is determined that the units in the core cannot live up to the learning objectives on their own, should a compulsory "capstone" course in security be recommended?
3) How much would this model benefit from a publicly declared institutional commitment to emphasize security across the curriculum?

As we discuss later, in Section 5, the thread approach we propose for undergraduate programs attempts to address these questions. Its underlying goal is to show that security is a multifaceted *process*, which is present at all stages of a system's lifecycle, by bringing it up in a variety of different contexts.

Whitman and Mattord identify the five academic approaches to curricular development in security [9]:

1) Add elements to existing courses.
2) Add elements to a capstone course or courses.
3) Create independent information security courses.
4) Create information security certificates or minors.
5) Create information security degree programs.

They indicate that an institution might find it beneficial to start with one of the first two approaches and later move toward the other three as its resources mature. The stated goal of this model is to allow undergraduate majors in Information Systems and Computer Science to assume positions in careers that evolve through technical knowledge areas and into management of information security. The resulting curriculum draft defines programs of one to four courses and presents a linear spectrum of options that has in one extreme the single-course approach and

Copyright © 2005, American Society for Engineering Education"

the track approach in the other. The approach an institution selects is dictated by the level of mastery it sets out to achieve. On one hand, as we discuss later in this paper, the track approach can arguably produce optimal learning outcomes, especially with higher number of courses. On the other hand, it may not be a viable option for many smaller colleges and universities.

An additional noteworthy example in the computer security education literature is Vaughn [8], which postulates three models for academic instruction in security:

1) The integration of computer security training into existing computer science programs,
2) The integration of computer security into software engineering degree programs,
3) The creation of a degree program on computer security.

Although, these models do not have a one-to-one mapping to the three models we discuss in this paper, we must underline the fact that (1) reflects the *security across the curriculum* initiative at the heart of the thread approach, which is also proposed by Yang [10]. Vaughn's work in this area is particularly relevant; he presents insightful arguments on how curricula can be adapted to incorporate instruction in computer security [7][8]. He identifies that there is currently no requirement of Computer Science and Engineering programs to ensure that graduating students take with them a solid appreciation for security issues and the understanding to develop solutions to address them. His papers identify the need to highlight and address computer security topics in courses such as Operating Systems, Computer Networks, Software Engineering, Databases, and Artificial Intelligence. He also reports experiences in following these courses with a required Information Security Capstone course.

Finally, the recognition of security as a process leads us to evaluate its role in Systems Engineering (SE) education, which brings and ties together strands developed throughout several courses of the Computer Science and Engineering curriculum. Hansche [4] argues that since Information Systems Security Engineering (ISSE) is an essential element of SE, it should be featured as early as possible in the curriculum. Preliminary evidence reported in this paper indicates that security learning objectives are not included in most introductory SE courses. The paper argues that only by including security learning objectives in the student's early stages of academic development will the curriculum stimulate the desired level of awareness of security issues.

We close this section with a statement closely related to the question posed in the final paragraph of [4]: The time has come for the academic community to start working towards integrating security into the core learning objectives of Computer Science and Engineering so that every undergraduate that successfully completes our degree programs will have a level of understanding of security and its importance to the design and the development of information systems.

3. Single-Course in Computer Security or Information Assurance

We consider three approaches to bring computer security and information assurance into an undergraduate degree program in this paper. Of these three, the single-course approach is by far the simplest and the least costly to implement. It can be implemented with the expertise of a single faculty, if it can be found and if it cannot, it requires investment in the training of only one instructor.

Copyright © 2005, American Society for Engineering Education"

The idea is to create one (and only one) additional, regularly offered course that attempts to complement the curriculum by giving students a concentrated exposure to the most relevant topics. The course is offered either as a requirement of the degree program or as an elective. When the course is established as a requirement for graduation, all students in the degree program have a good opportunity to obtain a broad perspective of the issues and the challenges in security.

There is little uniformity on the prerequisites for this course across different schools and, judging only by a preliminary survey that collected course information posted on the web, it is not clear what institutions offer it really as an insolated or terminal course on security since it may always be followed by an ad hoc topics course. Table 1 shows a sample of schools offering a single course in computer security.

Table 1: Sample of schools offering a single security course

Institution	Department	Course	Prerequisites
Bucknell University	Computer Science	CSCI 379 Topics in Computer Science	CSCI 315 Operating Systems or permission
Dartmouth College	Computer Science	CS38 Security and Privacy	CS23 Software Design and Implementation CS37 Computer Architecture
Denison University	Math and Computer Science	CS 402 Advanced Topics in Computer Science	CS-272 Data Structures and Algorithm Analysis II
Oberlin	Computer Science	CSCI 343 Secure Computing Systems	An introductory programming course or permission
Old Dominion University	Computer Science	CS 472 Network and Systems Security	CS 361 Advanced Data Structures and Algorithms
Richmond University	Math and Computer Science	CMSC 395 Special Topics	CMSC 301 Computer Architecture
Rose-Hulman Institute of Technology	Computer Science and Software Engineering	CSSE 442 Computer Security	CSSE 332 Operating Systems MA 275 Discrete and Combinatorial Algebra I

4. Track in Computer Security or Information Assurance

This approach is highly effective; it is arguably the optimal solution to educate undergraduates with the necessary computer security and information assurance skills. It is, however, more

Copyright © 2005, American Society for Engineering Education"

difficult to implement in small undergraduate programs where faculty is small in numbers or when they lack expertise in the topics and require additional training .

The track approach fares well in following Bruce Schneier's recommendation to treat "security as a process, not a product" [5], especially when compared to the single-course approach. Azadegan et al. [1] present a very strong plan for an undergraduate track in computer security that has been in use at Towson University since 2001. While this track keeps the standard core courses in the general Computer Science curriculum, it adds as many as five new Computer Science electives (Introduction to Information Security, Network Security, Application Software Security, Operating Systems Security, and Case Studies in CS) and one Math elective (Introduction to Cryptography). Judging by this list of courses and their coverage of topics, the learning objectives and outcomes of the program are certainly impressive. Undoubtedly the level of preparedness of their graduates must make them prime assets to their potential employers in industry and promising candidates for graduate programs. The number of courses in the track, as well as the range of topics they cover, plays to the expertise and the strengths of their faculty, but also to their numbers. The number of faculty in the Department of Computer Science at Towson is over 30 and growing. This number is over three times as large as that found in many smaller colleges and universities, if not larger. Clearly, under these more restrictive conditions, this plan for a track in computer security is not transferable.

The strength of the track approach is recognized by the National Security Agency (NSA). This governmental agency has created stringent, 10-point criteria for the evaluation of programs in Computer Science and Engineering and in Information Assurance. If these criteria are fully satisfied, the institution is awarded the designation of National Center of Academic Excellence in Information Assurance Education (NCAEIAE). Criteria 8 in the list in specifies that a qualifying program must have declared concentrations in information assurance [12]:

> "Academic program, within a nationally or regionally accredited 4-year college or graduate-level university, has declared concentrations in IA. Identify the courses required for each concentration, provide syllabus, enrollment data for current academic year (not projected) and actual graduation data (not projected) for the past two academic years."

Perhaps even more so than other requirements, this particular one makes it hard for smaller institutions to aspire to the NCAEIAE designation. The fact that this seal of national approval may be out of reach for some institutions does not imply that they cannot seek alternative approaches to provide their undergraduate students with the educational goals to support the Nation's cybersecurity needs.

The body of knowledge Computer Science and Engineering spreads out across a number of different areas of knowledge, as discussed above in Section 2 and in drafts for curricula [6] [11]. Students will invariably gravitate towards one or another area of interest in their choices of elective courses. While tracks or concentrations in Computer Security and Information Assurance will serve to satisfy the needs and the interests of a group of students, they will not be able to reach out to *all students in these majors*, similarly to the approach of single or even multiple elective courses. We argue that in order for education in computer security to have this broad coverage *it is the core curriculum that must be adapted*. The importance of security in computing is high today and will tend to get much higher as networking technologies develop further. Security education can only hope to live up to the needs of the present and the

Copyright © 2005, American Society for Engineering Education"

expectations of the future if it is widely and repeatedly emphasized across the core curriculum. We propose to address these needs with the thread approach described next.

5. Thread in Computer Security: A Unifying Theme for the CSE Curriculum

This approach makes extensive use of all the opportunities created by the breadth of topics in undergraduate Computer Science and Engineering curricula to start education in computer security early and to follow it through across the entire sequence of core courses. It represents an alternative to the track approach which also emphasizes security as a process rather than a product.

As Bishop states "the advantage of a good undergraduate education is the breadth of application of principles taught." [2] The thread in Computer Security approach takes advantage of opportunities already present in standard curricula such as CC1991 and CC2001 to expose and to emphasize elements which are part of instruction in security right when they first appear in the sequence of core courses.

Training the faculty to join the computer security education community, however, comes at the cost of a high investment. As pointed out by Yang [10], training the existing faculty may require institutional support for travel to conferences, workshops, and courses, reduced teaching loads, summer grants for curricular development and/or research. (The average cost per day of courses with the SANS Institute, for instance, can be roughly from US $530 to US $740.)

The NSA CAEIAE program certification requires the satisfaction of 10 criteria. Each criterion is assigned a minimum and a maximum number of points. Many undergraduate programs cannot aspire to this certification since several of the 10 criteria are beyond the reach of these institutions. This, however, does not mean that these programs should not strive to satisfy all the criteria in the list that is within their possibilities. We take as a particularly relevant example to the thread approach Criteria 9, entitled "Declared center for IA Education or Research" [12]:

> "The university has a declared center for IA education or a center for IA research from which IA curriculum is emerging. The center may be school or university-based. Provide documentation of the designation. (Example: The Computer Science Department has an officially designated "Center for IA Studies" with a clear link to and sponsorship by the College of Engineering Sciences, with a charter signed at least at the College of Engineering level.)"s

Although this approach speaks to educators' common-sense and goals in computer security, there is little evidence that it has been widely adopted. Our preliminary web survey has indicated that Rose-Hulman Institute of Technology and Old Dominion University may be following this approach. In their course catalog description these two institutions contain references to security topics in several core courses in addition to providing later in the sequence a dedicated computer security course.

Assuming that the thread approach is indeed effective, a claim which we cannot make for lack of concrete evidence, the question to ask next is how a smaller university or college can employ its

Copyright © 2005, American Society for Engineering Education"

resources in the implementation of a Computer Security thread in its curriculum. We propose that much of the foundation for creating this thread may already be in place for a number of institutions. When that is the case, with small additional efforts, it may be possible to expand the syllabi in the core courses to provide a strong education in Computer Security without incurring in the costs of creating a special track or degree program. There should be little or no need to put faculty through specific training in security as long as they continue to be assigned courses with subjects that they have mastered: the implementation of the thread would lead to small, incremental changes in syllabi of their courses and localized to the scope of their courses.

Using Computer Security throughout the Computer Science curriculum has the added benefit of creating for the students a bigger scope that contains many subject areas in this discipline. We argue that with the use of the thread approach, students would be reminded (in every course of the curriculum) that each area of focus within Computer Science fits within a bigger picture, that of working toward or working with a trusted system that requires parties to be authenticated, that preserves confidentiality, and which can be ultimately trusted. In the absence of efforts to unify the curriculum, students have a strong tendency to compartimentalize their knowledge. The Computer Security thread as unifying theme would work against this tendency and would likely bring out the desirable synergy between focus areas that not all students normally experience.

Arguably the first hurdle in the implementation of a thread in Computer Security in our curricula is time. If the idea is to use topics in security and privacy in the syllabi of multiple courses, we would need to identify the opportunities, that is, classroom time and assignments, which allow us to cover these topics. While at first blush this would seem to be a difficult goal to achieve, in many curricula much of the material may already be embedded in several different courses. To illustrate this point, we take a critical look at the recommendations in CC2001 [11] and attempt to indicate the coverage of topics for the implementation of a thread in computer security.

The introductory course sequence, typically referred to as CS1 and CS2, covers a wealth of concepts in Computer Science divided into main areas such as algorithmic thinking, programming fundamentals, and computing environments. Although the programming language used in these courses varies widely, the courses present ideal opportunities to present students with notions of *secure programming*. Programming language choices will dictate the extent to which students' code may be insecure, but practices like input validation, failing securely, and adhering to the principle of least privilege. If the programming language of choice is unlike Java, which enforces range checking on array indices, students can be taught to do it on their own as a matter of habit. Even if the consequences of these practices are not within the grasp of the students' level of understanding, they can be taught to follow the established best practices in secure programming. Typically CS1 and CS2 are followed by core courses in Computer Organization, Operating Systems, and Programming Languages. The opportunities for continuing with the Computer Security are plentiful in all these three courses.

In the syllabi of Computer Organization courses, it is customary to find a combination of topics in hardware and software. The hardware modules discuss, among many other topics, concepts important in the implementation of trusted operating systems. The presence of the Computer Security thread in this course would, at the very least, lead to emphasizing mechanisms of memory protection and the notion of *modes* (user/supervisor) in the design of the instruction set.

Copyright © 2005, American Society for Engineering Education"

The software modules in this course, which are usually centered on assembly programming set the context for the discussion of procedure calling conventions. This discussion sets the most appropriate stage for an introduction to buffer overflow attacks. Depending on the resources used in this course, students can be shown how machine code can be injected into a program that doesn't validate its inputs and executed to give an attacker special privileges in a computer system.

Operating Systems courses, which build upon Computer Organization concepts, continue to explore topics in protection. Since they typically revisit issues presented in the previous course, Operating Systems courses can be used to reinforce important concepts and place them in a larger context. Of particular interest to Computer Security, these courses can emphasize concepts such as virtualization, access control policies and mechanisms. This is perhaps the most natural opportunity to introduce notions of authentication and data confidentiality in the broad context of the computer *system*. Time permitting, models of security can also be introduced in this course; at this point, it may be useful to establish a conceptual understanding of mandatory access control, discretionary access control, and even of role-based access control.

More recently, security also established a strong presence in the design of programming languages. Consequently, in the context of a Programming Languages course, there are several opportunities to continue with the Computer Security thread. These courses can discuss type-safety, the integration of security models with a programming language, the use of virtual machines, the execution of mobile code, and again protection and access control. It can be argued that of the courses we discussed so far, this one poses the greatest challenges for the implementation of a Computer Security thread. Traditionally, the syllabi for Programming Language courses may not have included many topics in security, however, this landscape changed with the introduction of network-centric languages such as Java and C#. Similarly to what happens in other core courses, the amount of work in the implementation of the Computer Security thread in the Programming Languages course is determined by how up-to-date its current syllabus is.

Other important opportunities in the Computer Science curriculum are offered by courses that explore the impact of this discipline in a societal context. These courses, which cover requirements for the ABET certification, "analyze the impact of technology on individuals, organizations, and society, including ethical, legal, security, and global policy issues." [13] Typically, the syllabi in this type of course already address computer crime, computer related risks, and data intellectual property. Since this coverage includes important topics in Computer Security and Privacy, these courses are likely to already incorporate many elements of the thread.

We have presented above first recommendations toward the implementation of the thread approach to instruction in Computer Security in the core Computer Science curriculum. When resources allow, it would be extremely beneficial to follow the core courses with an elective in Computer Security that builds upon and ties together the concepts presented across the curriculum. This elective would serve as a focused course, a capstone of sorts, which would create an opportunity for students with strong interest in Computer Security to explore the inter-relationships between several key topics and perhaps even attempt to apply the knowledge they acquired previously to some kind of project. With the support of the thread, an existing course in

Copyright © 2005, American Society for Engineering Education"

Computer Security can be made much more effective. Students who would opt for not taking this course would still have been taught the basics concepts required by the current landscape in Computer Science and technology. These students would be prepared for our current reality in computing even when they don't express the interest in pursuing a career in Computer Security.

To conclude this section, we summarize the strengths of the thread approach in a few points:

- It can be integrated into a degree program incrementally without drastically changing the sequence in the core courses.
- It does not require that all faculty be trained simultaneously. Individual instructors can develop material at their own pace and change their syllabus gradually.
- It does not necessarily require additional courses since much of the material on security can be embedded in existing courses.
- It allows students to appreciate the importance of security as an underlying theme across the curriculum, which can help to avoid the isolation of knowledge units.
- It provides exposure in smaller units over a longer period of time allowing students to reflect and better assimilate the basic concepts of security.

7. Conclusion

In this paper we discussed three different approaches to undergraduate instruction in computer security. We made the distinction between programs in which security is presented in a single-course (most often a higher-level elective), in a dedicated track or concentration, or finally in a comprehensive effort that exposes issues in computer security across several courses in the core curriculum, which we called the thread approach.

We suggested that the single-course approach is of limited effectiveness in forming the professionals that the colleges and the universities produce, and the track approach demands extensive resources that most undergraduate programs cannot afford.

Acknowledgement

This project was supported under Award No. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security. Points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security.

References

[1] AZADEGAN, S., M. LAVINE, M. O'LEARY, A. WIJESINHA, and M. ZIMAND. "*An Undergraduate Track in Computer Security"*.ACM SIGCSE Bulletin , Proceedings of the 8th annual conference on Innovation and technology in computer science education, Vol. 35, No. 3, June 2003.

Copyright © 2005, American Society for Engineering Education"

[2] BISHOP, MATT. *Education in Information Security.* IEEE Concurrency Vol. 8, No. 4, pp. 4-8, October-December, 2000.

[3] FRINKE, DEBORAH and MATT BISHOP. *Joining the Security Education Community.* IEEE Security & Privacy, pp. 61-63, September/October, 2004.

[4] HANSCHE, SUSAN. *Preparing the Next Generation of SE Students for a Brave New World: Making the Case for an Early Introduction of ISSE.* Proceedings of the 8th Colloquium for Information Systems Security Education, pp. 21-30.

[5] SCHNEIER, BRUCE. *Secrets and Lies.* John Wiley & Sons, Inc., August 2000.

[6] TUCKER, ALLEN B., BRUCE H. BARNES, ROBERT M. AIKEN, KEITH BARKER, KIM B. BRUCE, J. THOMAS CAIN, SUSAN E. CONRY, GERALD L. ENGEL, RICHARD G. EPSTEIN, DORIS K. LIDTKE, MICHAEL C. MULDER, JEAN B. ROGERS, EUGENE H. SPAFFORD, AND A. JOE TURNER. *Computing Curricula '91*. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, 1991.

[7] VAUGHN, RAYFORD. *Application of Security to the Computing Science Classroom*. Proceedings of the Thirty-First SIGCSE Technical Symposium in Computer Science Education, pp. 90-94, 2000.

[8] VAUGHN, RAYFORD. *Building a Computer Security Emphasis in Academic Programs*. Proceedings of the Fourth Annual National Colloquium for Information Systems Security Education (NCISSE), May 2000.

[9] WHITMAN, MICHAEL E. and HERBERT J. MATTORD. *A Draft Model Curriculum for Programs of Study in Information Security and Assurance.* Proceedings of the 8th Colloquium for Information Systems Security Education, pp. 77-83, 2004.

[10] YANG, T. Andrew. *Computer Security and Impact on Computer Science Education.* Proceedings of the 6th Annual CCSC Northeastern Conference on the Journal of Computing in Small Colleges, Vol. 16, No. 4, pp. 233-246, May, 2001.

[11] ACM/IEEE-CS JOINT TASK FORCE ON COMPUTING CURRICULA. *Computing Curricula 2001 in Computer Science.* ACM Journal of Educational Resources in Computing, Vol. 1, No. 3, Fall 2001.

[12] NATIONAL SECURITY AGENCY – CENTRAL SECURITY SERVICE. *Centers of Academic Excellence.* http://www.nsa.gov/ia/academia/caeiae.cfm?MenuID=10.1.1.2 [Accessed January 1st, 2005]

[13] ABET. *Criteria for Accrediting Computing Programs*. November 1st, 2004. http://www.abet.org/images/Criteria/C001%2005-06%20CAC%20Criteria%2011-29-04.pdf [Accessed January 1st, 2005]

Author Biographies

LUIZ FELIPE PERRONE is Assistant Professor of Computer Science, at Bucknell University. He has been developing an elective in Computer Security since the spring of 2003. His research on the application of computer simulation to the study of the security properties of wireless networks is supported by the Office for Domestic Preparedness, U.S. Department of Homeland Security, via the Institute for Security Technology Studies at Dartmouth College.

Copyright © 2005, American Society for Engineering Education"

MAURICE F. ABURDENE is the T. Jefferson Miers Professor of Electrical Engineering and Professor of Computer Science at Bucknell University. He has taught at Swarthmore College, the State University of New York at Oswego, and the University of Connecticut. His research areas include, parallel algorithms, simulation of dynamic systems, distributed algorithms, computer communication networks, control systems, computer-assisted laboratories, and signal processing.

XIANNONG MENG is an Associate Professor in the Department of Computer Science at Bucknell University in Lewisburg, Pennsylvania, U.S.A. His research interests include distributed computing, data mining, intelligent Web search, operating systems, and computer networks. He received his Ph.D. in Computer Science from Worcester Polytechnic Institute in Worcester, Massachusetts, U.S.A.

"Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition Copyright © 2005, American Society for Engineering Education"