1998-12-25

Integrating Security into the Curriculum

Irvine, Cynthia E.; Chin, Shiu-Kai; Frincke, Deborah

IEEE

Cybersquare, IEEE December 1998 pp. 25-30, December 1998
https://hdl.handle.net/10945/7189

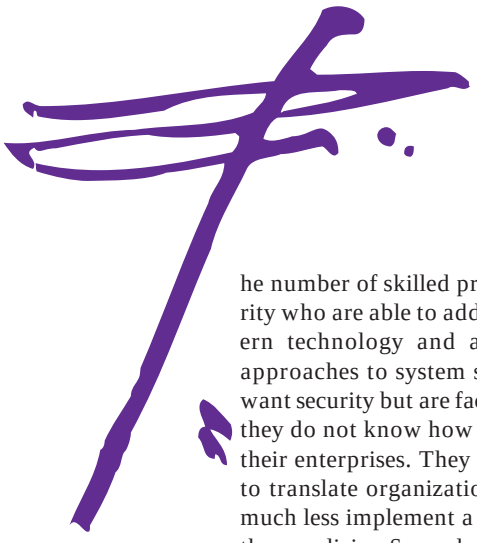
Cynthia E. Irvine
Naval Postgraduate School

Shiu-Kai Chin
Syracuse University

Deborah Frincke
University of Idaho

Integrating Security into the Curriculum

Computer security can be used as a vehicle to achieve accreditation goals for computer science and engineering programs, while at the same time engaging students with relevant, exciting topics. The authors' approach, based on educational outcomes, illustrates that security topics can contribute to an engineering program by fostering all skills required to produce graduates capable of critical thinking.



The number of skilled practitioners of computer security who are able to address the complexities of modern technology and are familiar with successful approaches to system security is very small. People want security but are faced with two difficulties. First, they do not know how to achieve it in the context of their enterprises. They may not even know of a way to translate organizational procedures into policies, much less implement a set of mechanisms to enforce those policies. Second, they have no way of knowing whether their chosen mechanisms are effective.

The recent US Presidential Commission on Critical Infrastructure Protection recommends developing education on methods of "reducing vulnerabilities and responding to attacks on critical infrastructures." The commission recognizes the need to make the "required skill set much broader and deeper in educational level [for] computer scientists, network engineers, electronics engineers, [and] business process engineers."[1]

Broadly speaking, the engineering discipline is fundamentally designed to assure results using techniques based on scientific principles. In terms of information assurance and security, the goal of engineering is to build secure systems from the outset rather than to discover that what we have built is inadequate. By moving to an educational system that cultivates an appropriate knowledge of security, we can increase the likelihood that our next generation of IT workers will have the background needed to design and develop systems that are engineered to be reliable and secure.

WHAT IS COMPUTER SECURITY?

The field of computer security focuses on designing systems that can enforce security policies even in the presence of malicious code. One of the great difficulties of security engineering is that subverted systems may appear to behave normally and a lack of security may not be evident. Systems must be engineered to be

0018-9162/98/$10.00 © 1998 IEEE

secure as part of their conception, design, and implementation.

The challenge is to design, develop, and deploy complex systems with confidence in their ability to satisfy security requirements. A theory of computer security has emerged that offers a formal method for security engineering.[2] This theory has three components: policy, mechanism, and assurance. Charles Pfleeger and Deborah Cooper[3] expand on these components by listing broad classifications of security concepts:

- *Policy*. Understanding the threats from which information needs to be protected to ensure confidentiality, integrity, and availability.
- *Privilege*. Creating mechanisms to distinguish and control the ability of active system entities to access and affect system resources.
- *Identification and authorization*. Associating the activities of the executing computer with individual users who may be held accountable for the activities undertaken on their behalf.
- *Correctness*. Providing assurance that the hardware, software, and systems for security policy enforcement are not susceptible to tampering or bypass.
- *Audit*. Creating traces and the ability to interpret them.

The implication here is that to achieve a coherent security architecture, security must be considered from the outset—not as an afterthought. Also, competence in design for security policy enforcement, testing for security, and assessment of security must be part of the education of system implementers.

COMPUTER SECURITY IN EDUCATION

Two important criteria for selecting outcomes for information security education are as follows:

- The education must result in graduates prepared for the security challenges they will encounter in their professional roles.
- The specific educational outcomes for security in a given educational program must be consistent with those of the larger engineering context.

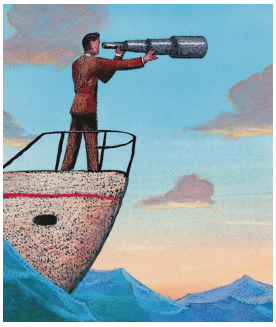
It is unreasonable, however, to suggest that everybody should know everything about security. Instead, we propose matching appropriate knowledge and skills with typical roles in the information society. Cynthia Irvine identifies ten such roles,[4] including software and hardware developers, system architects, system certifiers, CERT members, and security researchers.

In the skill set specified for engineering programs by the Accreditation Board for Engineering and Technology, engineering programs must demonstrate that their graduates have:

1. An ability to apply knowledge of math, science, and engineering.
2. An ability to design and conduct experiments, as well as to analyze and interpret data.
3. An ability to design a system, component, or process to meet desired needs.
4. An ability to function on multidisciplinary teams.
5. An ability to identify, formulate, and solve engineering problems.
6. An understanding of professional and ethical responsibility.
7. An ability to communicate effectively.
8. The broad education necessary to understand the impact of engineering solutions in a global and societal context.
9. A recognition of the need for—and an ability to engage in—lifelong learning.
10. A knowledge of contemporary issues.
11. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

To understand how security topics could be used to distinguish a computer science or computer engineering curriculum and to achieve ABET Criterion 3 goals, we decided to focus on a set of high-level educational objectives as the basis of a security-enhanced curriculum that would prepare students for the five roles listed above.

Understanding objectives

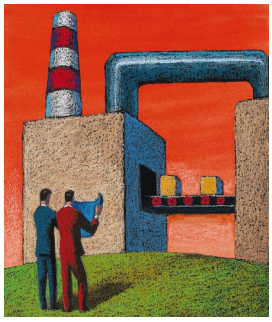


A major goal in computer engineering and computer science is to construct computer systems or processes that meet a desired end or requirement. To do so, students must learn how to express overall system objectives.

A general goal of security is to develop computing systems that can ensure security policy enforcement in the presence of malicious software and abusive user behavior. This particular security goal may encompass policy objectives for information confidentiality, integrity, or availability. In addition, the system must provide a mechanism to hold its users accountable for their actions through identification, authentication, and audit. Users must have confidence that their information will, in fact, be protected within the system.

By understanding objectives, students will have the ability to state a requirement's purpose, significance, and achievability and to determine the consistency of requirements and purposes.

Defining problems

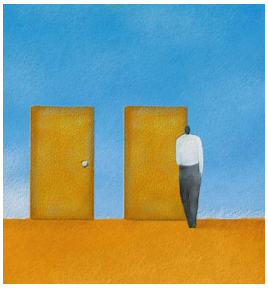


The fundamental characteristic of engineering is the ability to answer the question, "Does this structure of components have the properties required for the purpose for which they are designed?" Engineers ask this question at all levels of design, from the level of transistors to the systems themselves.

We can define security requirements as the properties of a system that must hold during system operation. The question at each level of design becomes, "Does this structure of components map to a security mechanism about which we can have confidence even in the presence of malicious code?" Using formal security policy models and specifications to provide a chain of evidence that the implementation corresponds to policy can demonstrate the feasibility of an actual implementation.

The formulation of appropriate questions and problems to be solved in the context of the discipline yields the ability to formulate questions of significance relative to the overall purpose; to state the problem to be solved and how it can be decomposed; and to determine the feasibility of the problem's solution.

Understanding context



Concerns at the architectural level require describing a combination of computer and network security mechanisms to ensure a coherent system for policy enforcement. The software developer must be concerned with the use of hardware mechanisms to support these security objectives. The hardware designer, meanwhile, must attempt to construct devices that support protection objectives while keeping in mind a wide variety of software implementations.

An appreciation of the contexts brought to bear upon a problem through various roles permits students to internalize the ability to design and analyze solutions to meet requirements and specifications at multiple levels of abstraction and from several viewpoints; to understand the impact actions on one level or in one viewpoint have on other levels or viewpoints; and to trade off several requirements from different viewpoints in order to achieve the maximum benefit.

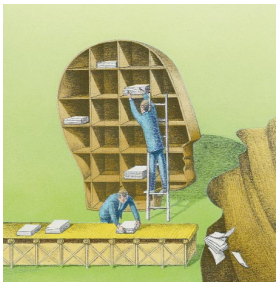
Reasoning empirically

Engineers often obtain empirical results at the lab bench by building prototypes and measuring their performance. These methods are also applicable to security. Functional interface testing, internal engineering tests of selected subsystems, system generation and recovery tests, and unit and module testing are all parts of the development process for a secure system.

In the lab, prototype systems can be built and examined for security flaws. Performance issues may also be examined by balancing expected decreases in vulnerability against user convenience and system efficiency. Techniques for assessing system vulnerability can be used to examine real systems for genuine flaws.

The educational outcome for cultivating a focus on empirical reasoning skills includes the ability to construct experiments or prototypes to demonstrate some purpose or facilitate some meaningful exploration and the ability to observe, collect, analyze, and interpret data from experiments.

Synthesizing knowledge



The fundamental theories and the reasoning techniques that allow those theories to be expressed define a discipline. In computer science and engineering, the fundamental theoretical concepts are based on mathematics, logic, and physics. These theoretical concepts form the principles of construction and analysis. In electrical and computer engineering, linear systems theory is based on sinusoidal signal composition and on superposition, which gives rise to the classical treatments of networks, controls, and communications theory.

The construction of computer hardware—and to a lesser extent software—is based on logic, predicate calculus, discrete math, and finite-state machine theory. In addition to applying standard mathematical foundations to constructing hardware and software, security includes theoretical concepts to support the development and use of cryptography and cryptographic functions, protocols, policy models, specifications, and the use of formal methods for verification and covert channel analysis. The means for analysis is based on discrete math, information theory, and

mathematical logic such as standard predicate calculus, modal logic, and specialized belief logics.

The educational outcomes achieved through a study of reasoning tools and theoretical results include a clear understanding of the mathematical, logical, and physical concepts that form the analytical basis and principles of construction. They also include the ability to apply analytical concepts and principles of construction to the analysis and construction of real systems.

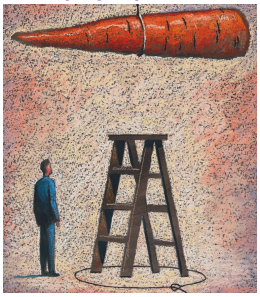
Identifying assumptions

Security is never the principal objective of a system. Systems are designed to provide services. In this sense, security engineering must always take place within the context of the overall system objectives. Each engineering discipline makes assumptions regarding the various objectives of the components, services, and properties available at each level of design. In fact, design levels and levels of abstraction are defined by these assumptions as well as by the particular rules of composition used for creating structures of components.

For example, designers of authentication protocols assume the presence of encryption functions of suitable strength. Designers of software assume the correctness of the hardware platform supporting the instruction-set architecture. Secure system designers may assume that the system security administrator is trustworthy and that the compiler—placed under configuration management—does not contain artifices to create trapdoors.

It is essential that the consistency between the assumptions of security engineers and other engineering concerns be checked. Mismatches in design levels, frames of reference, or applications cause inconsistent assumptions. By learning to compare differing assumption sets, students can increase their ability to state, justify, and check the consistency of their design assumptions.

Identifying implications and consequences



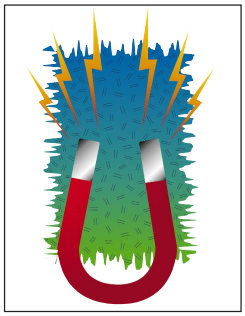
In all engineering processes, the implications of design decisions and system behaviors affect risk analysis, cost, ease of manufacture, ease of maintenance, reliability, and ethics. Determining implications and consequences means relying on all the previous elements of ABET's Criterion 3. Correctly balancing consequences is sometimes called business sense, which successful system architects and designers find to be based on experience, empirical reasoning, and conceptual reasoning coupled with a deep understanding of intended purposes or goals.

Determining the ethical consequences of computer use is complex but may be based on the following three ABET Engineering 2000 Criteria:

- an understanding of professional and ethical responsibility;
- the broad education necessary to understand the impact of engineering solutions in a global and societal context; and
- a knowledge of contemporary issues.

By cultivating an understanding of consequences and implications, students can anticipate and articulate positive and negative consequences, and can more easily judge their likelihood.

Drawing inferences



The ability to draw inferences from a great amount of information is a skill that must be refined through years of practice. While at school, students should be encouraged to attempt inferences. And in terms of learning about security, students should be introduced to inference techniques by being asked to draw conclusions about systems security.

For example, security engineers make inferences in the areas of fail-secure and secure-system recovery; systematic penetration testing; and detection of evidence proving abusive behavior based on profiling and audit data. These concerns are common to both security and engineering.

By focusing on inference techniques, students can cultivate an ability to draw correct inferences based on principles, observations, concepts, and data; to justify conclusions; and to draw conclusions that are relevant and consistent.

INTEGRATING SECURITY INTO THE CURRICULUM

Security insights must be integrated within the existing information systems programs, rather than be treated separately. The technical aspects of security are closely related to computer science and engineering. And many of the goals, concepts, and reasoning techniques are similar too. Thus, two approaches are possible:

- Computer security could be the focus of the curriculum, which would investigate the foundations and technical approaches to security in considerable depth.
- A computer science or computer engineering curriculum could choose to use computer security as an important property to be addressed in all coursework.

Greg White and Greg Nordstrom[5] have already demonstrated the feasibility of successfully integrating security concerns into traditional courses. This approach has the advantage of viewing security as an integral part of computer engineering and science.

Topics appropriate to a security-oriented curriculum include security policy models; formal methods applied to system specification, development, and analysis; hardware and software protection mechanisms; secure system design, implementation, and testing; database security; modern cryptography; cryptographic protocols; key management and key distribution; auditing; identification and authentication; and coherent network security architectures.[6,7]

Ideally, computer science and computer engineering texts, course materials, and laboratory exercises would have computer security completely integrated into appropriate topics. Unfortunately, such materials do not yet exist. In the interim, however, security-related supplements can be used.

AN EXAMPLE OF INTEGRATION

In a curriculum that places considerable emphasis on visual simulation, for example, the possibility of including security topics may seem rather far-fetched. Even so, security requirements rest at the heart of every system, and visual simulation systems are no exception. They will depend upon the security of OSs, databases, and networks to operate properly. Core requirements in the visual simulation curriculum might include stochastic modeling, system simulation, physically based modeling, and image synthesis. Students could be required to take courses in other basic topics, such as networking, OSs, programming languages, and software engineering. More advanced topics might include the use of networking for virtual environments and systems for creating virtual worlds.

Examples, exercises, and special topics could be used to add security concepts to both introductory and advanced courses. For instance, in beginning programming courses, students might be given an exercise to enter information into certificates or to check passwords. As part of a course in discrete mathematics, students could discuss the use of Boolean arithmetic in cryptography. A presentation on lattices could be enlivened by showing how they can be used for formally expressing mandatory security policies. Software engineering students could take advantage of existing cryptographic libraries while building a larger system.

While learning the basics of computer architecture, students could hone their assembly language skills by experimenting with the privileged instructions essential to building protection mechanisms. In networking, they could examine not only traditional communications protocols but also algorithms and protocols for secure communications. Through experiments in system configuration and management, students could learn the value of well-defined procedures to maintain the security of systems once they are operational. The notion of certified code for upgrades and patches could illustrate concepts in authentication and distribution of software for critical systems.

More advanced courses could ask students to consider security when designing software and networks to support distributed simulations. Here, systems issues relating to the protection of databases and algorithms essential to creating a secure network could be addressed. Students could be asked questions such as, "Where are the cryptographic keys stored and why do you believe they are protected?" and "How will company-proprietary information be separated from public information and how will it be protected?" By asking tough questions such as these, students will appreciate the fact that security must be an integral part of system design.

Security insights must be integrated within the existing information systems programs, rather than be treated separately.

WILL GRADUATES BE QUALIFIED?

How adequate is our plan? To answer this question, we compare our educational objectives with remarks made by employers in the computer security field at the 1996 IEEE CS Symposium on Security and Privacy,[8] the 1997 ACM Workshop on Education in Computer Security (WECS 97),[9] and the 1997 National Colloquium for Information Systems Security Education.[10]

All who made remarks specified that security is not an isolated discipline but rather is part of the larger context of engineering and computer science. Some indicated that ethics should be part of security education. The study of ethics is already part of an engineering education and falls into the major objective of implications and consequences. WECS 97 attendees concluded that information responsibility should be taught well before students enter institutions of higher education.

Several of the participants suggested that operational expertise be applicable to industry—an idea that is covered by several elements of our approach. Many expressed specific concerns over linking security to several engineering activities spanning requirements, specification, design, implementation, testing, and val-

idation. And there were requests that theory inform practice and practice inform theory.

How well does the proposed framework meet the accreditation requirements for engineering? The accreditation criteria for electrical and computer engineering programs proposed by the IEEE refer to ABET's Criterion 3 listed before. Programs must demonstrate that graduates can

- achieve the outcomes listed in Criterion 3 in three or more areas of electrical and/or computer engineering;
- have the ability to apply the math and science necessary to analyze and design complex devices and systems containing hardware and software; and
- have knowledge of discrete mathematics.

There are broad similarities between the educational outcomes we expect to emerge from our approach and ABET's Criterion 3. Thus, including computer and network security topics within a computer science or engineering curriculum will provide two benefits. First, the topics will contribute to the educational outcomes required for ABET accreditation. Second, integrating the topics into a computer science or engineering curriculum will add a highly relevant dimension to the program—a feature prized by prospective employers.

The increasing use, reliance upon, and vulnerability of current large-scale information systems demands that more resilient, reliable, and secure systems be built and deployed. But too few computer science and engineering programs today pay adequate attention to security, even though security concepts are fundamental and apply to all levels of system and application design.

It is reasonable to ask that technically meaningful ways be sought to integrate security into the engineering and computer science curricula charged with the education of the majority of system designers and implementers. An approach based on educational outcomes illustrates that security topics can contribute to an engineering program by fostering all skills required to produce graduates capable of critical thinking. ❖

References

1. Defense Science Board, "Report of the Defense Science Board Task Force on Information Warfare—Defense IW-D," Office of the Secretary of Defense, Washington, D.C., 1996.
2. D.L. Brinkley and R.R. Schell, "Concepts and Terminology for Computer Security," *Information Security: An Integrated Collection of Essays*, IEEE CS Press, Los Alamitos, Calif., 1995, pp. 40-97.
3. C. Pfleeger and D. Cooper, "Security and Privacy: Promising Advances," *IEEE Software*, Sept./Oct. 1997, pp. 27-32.
4. C.E. Irvine, "Challenges in Computer Security Education," *IEEE Software*, Sept./Oct. 1997, pp. 110-111.
5. G. White and G. Nordstrom, "Security Across the Curriculum: Using Computer Security to Teach Computer Science Principles," *Proc. 19th Nat'l Information Systems Security Conf.*, National Inst. of Standards and Technology, Baltimore, Md., 1996, pp. 483-488.
6. C.E. Irvine, "Goals for Computer Security Education," *Proc. IEEE Symp. Security and Privacy*, IEEE CS Press, Los Alamitos, Calif., 1996, pp. 24-25.
7. C.E. Irvine, D.F. Warren, and P.C. Clark, "The NPS CISR Graduate Program in INFOSEC: Six Years of Experience," *Proc. 20th Nat'l Information Systems Security Conf.*, National Inst. of Standards and Technology, Baltimore, Md., 1997, pp. 22-30.
8. C.L. Schuba and M.E. Zurko, "IEEE CS Symposium on Security and Privacy," *Electronic Cipher*, June 1996, http://www.itd.nrl.navy.mil/ITD/5540/ieee/cipher/issue9606.
9. H. Hinton, "Review of First Annual Workshop on Education in Computer Security," *Electronic Cipher*, Mar. 1997, http://www.itd.nrl.navy.mil/ITD/5540/ieee/cipher/issue9703.
10. J. Kauza, "Industrial Perspective on INFOSEC Education Requirements," *Proc. Nat'l Colloquium for Information Systems Security Education*, Maritime Inst. of Technology, Linthicum, Md., 1997, pp. 76-80.

Cynthia E. Irvine holds a joint appointment as assistant professor in the Department of Computer Science and the Department of Electrical and Computer Engineering at the Naval Postgraduate School. She is director of the Naval Postgraduate School Center for INFOSEC Studies and Research. Her research interests include network security architectures and high-assurance multilevel distributed systems. She received a PhD from Case Western Reserve University. Contact her at irvine@cs.nps.navy.mil.

Shiu-Kai Chin is an associate professor in the Department of Electrical Engineering and Computer Science at Syracuse University. He is also director of the New York State Center for Advanced Technology in Computer Applications and Software Engineering. His research interests include applying mathematical logic to the engineering of highly assured systems. He received a PhD in computer science from Syracuse University. Contact him at skchin@syr.edu.

Deborah Frincke is an assistant professor in the Department of Computer Science at the University of Idaho. Her research interests include computer security and software testing. She received a PhD in computer science from the University of California, Davis. Contact her at frincke@cs.uidaho.edu.