An Undergraduate Track in Computer Security

S. Azadegan, M. Lavine, M. O'Leary, A. Wijesinha, M. Zimand
Towson University
8000 York Road
Towson, Maryland 21252

azadegan@towson.edu, mlavine@towson.edu moleary@towson.edu,
wijesinha@towson.edu, mzimand@towson.edu

ABSTRACT

To better prepare our graduates to face the challenges in computer and information security, in Fall 2002, Towson University launched an undergraduate track in computer security for the computer science majors. This paper describes the motivation behind this track and discusses its structure and requirements.

Categories and Subject Descriptors
K.3.2 Computer and Information Science Education

General Terms
Security

Keywords
Undergraduate Computer Security Education, Cryptography, Network Security, Operating Systems Security, Application Software Security, Computer Security Case Studies

1. INTRODUCTION

The Report of the Presidential Commission on Critical Infrastructure Protection (PCCIP) recommends "education on methods of reducing vulnerabilities and responding to attacks on critical infrastructures", and "programs for curriculum development at the undergraduate and graduate levels" and efforts to make the "required skill set much broader and deeper in education level for computer scientists, network engineers, ...". Clearly, all aspects of the national infrastructure depend upon the correct operation of computers and networks and the security of these systems is imperative to the health and protection of our national infrastructure and information assets. The "Call to Action" document [1], produced by top security experts, identified lack of security skills as one of the top ten trends impacting security, while investing in training and developing a sound educational program as one of the six action items viewed as most critical by the group. The computer security track would address the needs of skilled people in the Computer Security field and provide an opportunity for our undergraduate students to be educated in this field. Students graduating with this track will have a strong background in the fundamental principles of computer security and their applications, plus hands-on experience with security tools commonly used in industry and they will be better prepared to join the 21st century workforce.

2. GOALS AND OBJECTIVES

The main objective was to develop a high-quality computer security track that uses and builds upon the courses that are already part of the computer science curriculum and allows students to finish their degree in four years without having to take additional courses. The courses in this track expose students to a wide range of security problems and vulnerabilities that exist in operating systems, application systems and networking protocols and show them how these vulnerabilities might be exploited by potential adversaries. The students will be given opportunities to work with malicious codes, to use port scanners and network sniffers, to crack passwords and to attack servers that were configured by their classmates as part of their projects. The goal is not to make them hackers, rather, provide them with the knowledge and skills that firstly they can recognize these security loopholes and secondly they can handle these problems. At the present, the developed track is only available to our computer science majors and computer science and mathematics double majors. Both programs are accredited by Computer Science Accreditation Commission. The track allows students to join the program at any time before the start of their third year.

3. THE TRACK

We have identified seven courses as the key components of this track:

1) Computer Ethics
2) Introduction to Information Security
3) Introduction to Cryptography
4) Network Security
5) Application Software Security
6) Operating Systems Security

ITiCSE'03, June 30–July 2, 2003, Thessaloniki, Greece.
Copyright 2003 ACM 1-58113-672-2/03/0006...$5.00.

7) Case Studies in Computer Security

The table below contains the Computer Science (COSC) program and also illustrates the changes that were made to incorporate the security track.

Table 1: Computer Science Program Changes

Computer Science	Computer Science with a Track in Computer Security
Required Core courses Computer Science I and II, Computer Architecture, Data & File Structure, Computer Organization, Operating Systems, Programming Languages: Design & Implementation, Database Systems, **Computer Ethics**	The same as COSC
Required Math courses Calculus I, Calculus II Discrete Math, and Statistics	Same as COSC
One upper-level math elective course	**Intro to Cryptography**
Science Requirement (12 credits)	Same as COSC
Elective COSC courses (12-14 credits)	**Intro to Info. Security** **Network Security** **Application Software Security** **Oper. Systems Security** **Case Studies in CS**

For the track to be successful, it is necessary to provide an environment that facilitates active learning and allows maximum opportunity for hands-on experiences for the students. Moreover, the nature of the experiments and projects does not allow the use of a general-purpose computer laboratory. PCs used for computer security experiments can never be considered to be in a "safe" configuration for general use and the disk image configuration of these systems will be constantly changed, based on the needs of a particular laboratory exercise Therefore, such systems cannot be maintained in a consistently configured state for general use. As part of our grant, we did receive funding for an isolated security laboratory. Access to this lab will be limited to the students who are taking the course and they will be closely supervised. Security measures will be implemented to prevent any unauthorized and inappropriate access. Students taking the security track, will be given a document describing the code of conduct and general responsibilities of the students [2] and they will be asked to sign an agreement acknowledging that they have read and understood the code of conduct of computer security track and will act at all times with accordance with that code. We are in process of creating the laboratory and will be able to report on its status at the conference.

Out of the seven courses identified for our security track, the Computer Ethics course is a required course for all students majoring in Computer Science. This course prepares students to deal as professionals with ethical questions and societal concerns related to the widespread uses of computers and resulting responsibilities of computer scientists. Below we describe the remaining six courses that the students in the computer security track have to complete.

3.1 Introduction to Information Security

This course provides students with a broad understanding of technical and human components of information systems security. The course focuses on information systems security threats, technologies and business requirements. Emphasis is placed on the human and technological aspects of IT security problems and issues relevant to the risks in which information systems are exposed and methods of dealing with such risks. The prerequisite for this course is junior standing, and students are advised to take this course as the first course for the track.

Detailed Topics Covered:

- Identification and Authentication, Access Control
- Security Threats and Vulnerabilities
- Security Models, Security Requirements and Standards
- The Security Kernel
- Network and Distributed Systems Security
- Internet Security and Cryptography
- Operating System Security
- Database Security
- Legal and Ethical Issues

We selected *Computer Security* by Gollmann [3], and *Security in Computing* by Pfleeger [4] as the textbooks for this course. The homework for this class consists of two or three small hands-on projects, written assignment and a research paper. This course provides students with fundamental principles of computer and security and lays the foundation for the other courses.

3.2 Cryptography

The course will give a broad overview of the mathematical basis of modern cryptography and to the main cryptosystems currently in use. Students taking the course will be exposed to relevant chapters of number theory and computational number theory at a level appropriate for undergraduates. The course will cover the most important cryptosystems (e.g. DES, AES, RSA) and the basic tools used in building security mechanisms. Some important methods for cryptanalysis will be presented as well. The course will provide an overview of some important protocols having a strong cryptographic flavor. At the end of the course, students will have a good understanding of the theoretical foundations of cryptography and of the basic techniques in achieving different cryptographic services. Discrete math and junior standing are the prerequisites for this course. We chose *Introduction to Cryptography* by Wade Trappe, Larry Washington [5], as the textbook for this course.

Detailed Topics Covered:

- **Basic Concepts of Cryptology:** (Historical ciphers, Cryptanalysis of historical ciphers, One-time pads)

- **Modern Symmetric Cryptographic Systems**: (DES, Differential Cryptanalysis, Triple DES, Modes of Operation (ECB, CBC, CFB, OFB), Advanced Encryption Standard - Rijndael)
- **Basic Number Theory:** (Euclidean Algorithm, Modular Arithmetic, Chinese Remainder Theorem, Fermat's Little Theorem and Euler's Theorem, Primitive Roots, Quadratic Residues, Finite Fields)
- **Public Key Cryptography:** (RSA, Attacks on RSA, Factoring and Primality Testing, Discrete Logarithms, ElGamal Public Key Cryptosystem)
- **Data integrity and authentication:** (Digital signature schemes, Hash Function, Pseudo-random generators, Security of Hash Functions and MACs)
- **Protocols**: (Digital Cash, Secret Sharing Schemes, Bit Commitment Schemes, 2-party and multy party protocols for private distributed computation)
- **Zero-Knowledge Techniques**: (Basic Schemes, Feige-Fiat-Shamir Identification Scheme)

In addition to homework assignments, the students will be asked to develop either a software project or an analytical project based on cryptographic techniques. Software projects typically involve writing a program in a high-level language (C, C++, Java, etc.). Analytical projects involve comparative analysis of competing algorithms, protocols, or implementations. They also may involve reviewing/surveying issues related to cryptology and some other fields such as number theory, physics, law, etc.

3.3 Computer Networks

The course will provide the students with a thorough understanding of the concepts underlying all aspects of network security with an emphasis on applications. This course covers network security principles and applications. Topics include: authentication applications, IP security, Web security, network management security, wireless security, and system security. The prerequisite for this course is computer networks, a required course for our computer science majors and cryptography. We have chosen *Network Security Essentials* [6] by William Stallings, as the textbook for this course.

Detailed Topics Covered:

- **Introduction to network security**
- **Review of cryptography**: (Encryption, Public-key cryptography and Message authentication)
- **Authentication applications**: (Kerberos, X.509 authentication service)
- **E-mail security**: (PGP, S/MIME)
- **IP security**: (IPSec, Virtual private networks, IPv6 security, Mobile IP security)
- **Web security:** (Secure Sockets Layer and Transport Layer Security, Secure Electronic Transaction)
- **Network management security**: (SNMP security)
- **Wireless security**: GSM security;
- **System security Overview**: Intrusion and intrusion detection, Viruses and worms, Firewalls

The students in this course will have bi-weekly programming projects dealing with network security applications and tools.

3.4 Operating Systems Security

This course allows students to gain an in-depth knowledge of security threats, different types of malicious codes, and access control problems in the context of operating systems. We will discuss intrusion detection techniques and the design of trusted operating systems along with their cost and performance analysis. The prerequisite for this course is operating Systems, a required course for our computer science. The prerequisite provides the theoretical foundations and this course will be more applied and topics will be discussed in the context of Linux and Windows NT operating systems. We have selected *Maximum Linux Security* [7], Anonymous, and *Window NT Security* [8] by Michael Mclnerney, as the textbooks for this course.
Detailed Topics Covered:

- **Overview of the Linux Operating System**
- **Linux Security Basics**: (User accounts, Discretionary Access Control, Network Access Control, Intrusion Detection)
- **Linux User Security**: (Password Attacks, Data Attacks)
- **Linux Network Security**: (Malicious Code, Sniffers and electronic eavesdropping, Scanners, Spoofing)
- **Overview of the Windows NT Operating Systems**
- **Windows NT Security Architecture Overview** (Layered approach to securing your network, Modules of NT Security Architecture, Security implementation overview)
- **File and Directory Security**: (Disk Partition, File & directory permission, File & directory security, Share permission)
- **User profile**: (Overview, Profile permissions, Default profile)
- **Registry:** (Registry Structure, Registry Tree permission, Registry editing tools)

Similar to the Network Security course, students will work on small bi-weekly programming projects to gain hands-on experience with the security issues covered in class.

3.5 Application Software Security

This course studies the security concepts in developing software applications. It discusses design principles for secure software development, and some of the security issues in current programming and scripting languages, database systems and web servers. The *Survey of Programming Languages* course is the prerequisite and the *Database Systems* course is the co-requisite for this course. We have chosen *"Building Secure Software: How to avoid Security Problems the Right Way* [9], by John Viega and Gary McGraw as the textbook for this course.

Detailed Topics Covered:

- **Software Security:** (Security Goals, Common Software Security Pitfalls, Overview of Software Risk Management for Security, Software Security Principles, Auditing Software, Selecting a language)

- **Java Security:** (Java Virtual Machine, Byte code Verifier, Java Sandbox, Java Language security constructs, The Class loader, Class accessibility, Java Cryptography architecture)
- **Secure CGI/API Programming**
- **Buffer Overflows:** (Overview, Defending against Buffer Overflow, Internal Buffer Overflows, Heap Overflows, Stack Overflows, Attack Code)
- **Database Security:** (Security Problems in Databases, Secure DBMS Design, Security Controls, Using Views for Access Control, Field Protection, Statistical Database Protection, Statistics Concepts and Definitions, Security against Statistical Attacks)
- **Client-side Security:** (Traditional Threats, Using SSL, Browser as a security hole)
- **Server-side Security:** (Current Major Host Security Problems, Minimizing Risk by Minimizing Services Secure Content Updating, Physical Security, Access Control Strategies)
- **Firewall:** (Basic Architecture, Client Proxies Server Proxies)

In addition to small projects, students will work on a team projects, which allow them to integrate and use their knowledge about Java security, database security, client-side and server-side security in a real-world project.

3.6 Computer Security Case Studies

This is a capstone course that allows students to work on comprehensive security-related projects. It provides students with as in-depth study of the practical aspects of computer security vulnerabilities in a hands-on laboratory setting. Course work will consist primarily of computer laboratory projects. There will be no exams. Course assignments will consist of 6 to 8 projects, some smaller homework assignments, and a final paper. Towards the end of the semester, speakers from industry and government will be invited to present topics of particular interest to them in the areas of computer and network security, computer ethics, public policy for computing and security. This interaction between students and industry and government representatives would allow them to get first-hand knowledge about real projects and problems. It will provide students the opportunity to establish connections with their potential future employers.

Case Studies:

- **Case Study 1: Services** (FTP , Mail, Telnet / SSH , Web Servers , File Sharing, Finger, WebDAV)
- **Case Study 2: Hardening a Server** (Linux / UNIX, NT / 2000 / XP)
- **Case Study 3: Sniffers & Spoofing** (Hubs vs. Switches, Detecting Sniffers, IP Spoofing, ARP Spoofing, DNS Spoofing, EMail Spoofing, Web Spoofing)
- **Case Study 4: Session Hijacking & Anonymity**
- **Case Study 5: Firewalls & Scanners**
- **Case Study 6: Intrusion Detection Systems** (SNORT, TripWire, Logging and Audit Tools, Web Server, Tools for analyzing Log Files)
- **Case Study 7: Password Attacks & Encryption** (Password Management, NT / 2000 Password Implementation, UNIX / LINUX Password Implementation, Web Server Passwords, Application Passwords, PGP, Steganography, Password Cracking Tools)
- **Case Study 8: DoS Attacks** (Email floods, Network DoS, Network DDoS)
- **Case Study 9: Malicious Code** (Viruses, & Worms, Detection & Removal, Policies and Procedures)

4. CONCLUSION

In this paper, we presented a track in computer security that can be easily incorporated into any computer science program and described in detail its courses. At Towson University, the track was made available to our students, as of Fall 2002. During the fall semester the Introduction to Information Security and the Introduction to Cryptography courses were offered. The Network Security course is scheduled for the Spring 2003 semester and the remaining 3 courses will be offered next academic year. The track has already attracted many students, and we are getting requests from our Computer Information Systems (CIS) majors to either allowed them to take this computer science track or to offer a similar track for them.

5. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grant No. DUE-0113783. We express our thanks to the academic and industry consultants and evaluators Dr. Deborah Frincke and Dr. Ronald Gove.

6. REFERENCES

[1]www.cerias.purdue.edu/events/accenture_cta_1q2001.pdf

[2] Julie J. C. H. Ryan, Daniel J. Ryan, "Institutional and Professional Liability in Information Assurance Education", George Washington University.

[3] Dieter Gollmann, "*Computer Security*", John Wiley & Sons, 1999.

[4] Charles P. Pfleeger, "*Security in Computing*", Second Edition, Prentice Hall PTR, 1997.

[5] Wade Trappe, Larry Washington, "*Introduction to Cryptography*", Prentice Hall, 2002.

[6] William Stallings, "*Network Security Essentials*", Prentice Hall, 2000.

[7] Anonymous, "*Maximum Linux Security*", Second Edition, SAMS, 2001.

[8] Michael McInerney, "*Window NT Security*", Prentice Hall, 2000

[9] John Viega and Gary McGraw, "*Building Secure Software: How to Avoid Security Problems the Right Way*", Addison Wesley, 2002.