Proceedings of the 14th Colloquium for Information Systems Security Education
Baltimore Marriott Inner Harbor
Baltimore, Maryland June 7 - 9, 2010

Experiences in Designing and Implementing an Undergraduate Certificate in Information Security

Prem Uppuluri, Jeff Pittges, and Robert Phillips, *Radford University*

Abstract – This paper describes an undergraduate certificate in Information Security that supplements the Baccalaureate degrees in Computer Science and Technology and Information Systems at Radford University with comprehensive coverage of information security. The paper presents the rationale behind our decision to develop a certificate and discusses the issues we encountered while developing and implementing the curriculum for the certificate.

Index terms – Information Security Education, Projects

I. Introduction

To incorporate information security into the curriculum in the Department of Information Technology at Radford University, we saw a need to develop a new set of courses. Our main goals were to:

- Provide a security education accessible to all the students in the department regardless of their specific focus in information technology, such as web development or networking, and,
- Provide a comprehensive education in security to prepare undergraduate students for a career in information security or graduate study.

While these goals seemed easy enough to achieve, especially given the large amount of work done in the field of security education, the actual development and implementation ended up being challenging for a variety of reasons, primarily:

- The structure of the IT department where students have the choice to concentrate in one of six focus areas, thus making it difficult for us to assume that all the students have had the same preparation to take the information security courses, and,
- Achieving the right balance between covering security topics in-depth and working with resource limitations in terms of faculty and labs, when developing the curriculum.

P.Uppuluri, J. Pittges and R. Phillips are with the Department of Information Technology, Radford University, Radford, VA, 24141, e-mails: (puppuluri, jpittges, rphillips)@radford.edu

We developed an undergraduate certificate in information security to meet our goals. In this paper we discuss our experiences in terms of the issues we faced and addressed in the development and implementation of the certificate.

II. Certificate Development: Motivation

The Department of Information Technology offers Bachelor of Science degrees in Computer Science (CS) and Information Systems (IS). Both degrees require students to complete at least one concentration. The CS program offers four concentrations: Computer Science, Databases, Networks, and Software Engineering, and the IS program offers two: Information Systems and Web Development. Each concentration includes around three courses in the area of specialization.

Our challenge was to create a comprehensive study of security across all six concentrations. The University did not offer undergraduate certificates (unlike other Universities[1]). Therefore, we considered three options that fit the structure of our curriculum:

1. Develop a security concentration.
2. Add a security component to the existing concentrations.
3. Create security courses to be taken as technical electives.

These options were considered with respect to two objectives:

- Provide a comprehensive security education that compliments all six concentrations and is accessible to students in both degree programs, and,
- Offer a credential to recognize the breadth and depth of study and motivate students to complete the security program.

None of these three options satisfy both objectives. A separate concentration for security would not be accessible to all of the students in our department. The concentrations within the Computer Science degree overlap enough that many students complete multiple

[1] Examples: http://infosec.kennesaw.edu/education.html; http://www.cs.wcupa.edu/isc/curricula.html.

concentrations, but the requirements across the two degree programs have little in common. The highest-level course required by all the concentrations in both degrees is *Principles of Computer Science II* (CS2). Adding a concentration to either degree would make it impractical for students in the other degree program to complete the security concentration. A separate concentration would also cannibalize the existing concentrations. The second and third options would both produce fragmented coverage. Adding security components to the existing concentrations was attractive in theory because all six concentrations would produce students with extensive security training. In practice, however, many of the concentrations are already overloaded. Consequently, it is not feasible to create a meaningful standard of security education across all six concentrations.

We believe, creating elective courses without a credential would not provide enough incentive for students to choose security courses over other electives. Moreover, employers would have difficulty evaluating the security skills of individual students based on the student's concentration and elective courses. As we evaluated these three options it became clear that the ideal solution was to follow the trend followed by other Universities such as Kennesaw State and West Chester University (*see footnote 1*) and create an undergraduate certificate; in our case, one that cuts across all six concentrations. With a certificate we could develop a comprehensive education in security that would enhance each concentration, motivate our students to invest any additional time to complete the certificate, and attract prospective employers. A goal here was to minimize the additional time over a regular bachelor's program that a student would have to invest in the certificate program.

Although the University did not offer undergraduate certificates, several departments wanted to create certificates for their undergraduates. With strong support from faculty and industry partners our proposal for a certificate in security compelled the University to adopt undergraduate certificates. However, several issues remain due to limitations in the systems and processes used to manage undergraduate programs.

The primary challenge is the administration's ability to track certificates. The process of awarding a certificate at the time a student graduates is fairly straightforward. However, the University is unable to award a certificate after graduation and it is unclear if the University is capable of awarding certificates prior to graduation. The certificate would be especially valuable to rising seniors looking for internships that require security training. It is also likely that the security certificate will be attractive to professionals, but the current process does not support non degree-seeking students.

III. Developing the Curriculum for the Certificate

Lot of work has been done in developing curriculum for information security at an undergraduate level ([1], [2], [3], [4], [5], [6], [7], [8]). Some educators, e.g., Bishop [1], suggest teaching broad principles and applications at an undergraduate level while leaving the depth to graduate level courses. Others, such as Yang [2], Azdegan et al. and [4], Mattord et al. [6] and more recently Sexton [8] have proposed comprehensive coverage through undergraduate tracks that either require four or more courses with in-depth coverage of specific areas such as network security, operating systems security, database security, computer forensics and cryptography, or, require changes to existing courses.

Our challenge was to customize this vast base of extensive knowledge on creating an undergraduate information assurance curriculum towards the best interests of our University students while at the same time considering our constraints. Specifically, after graduation, undergraduate students at Radford University typically tend to pursue professions such as database, network and system administrators, software testers, web developers, software engineers and some pursue graduate studies. To be successful in these professions, an extensive knowledge of best practices in information security relevant to that profession is important. Hence, we decided that our curriculum must provide in-depth coverage of the topics relevant to these professions.

Ideally, we could have followed the model of others such as [2], [4], [5], [6], and offered several security courses. However, as our previous offerings of security courses were very limited (and offered as special topics), we were constrained in terms of the number of faculty as well as lab resources. Furthermore, requiring additional courses over the standard concentration curriculum requirements, or, adding many pre-requisites would have increased the time students would invest beyond their regular programs – making the certificate less attractive to those students who wish to graduate within strict time-lines. This was unacceptable given the infancy of the certificate.

Consequently, we decided to focus on topics that would directly supplement the various concentrations, and reduce (but not eliminate) our focus on topics such as formal verification, forensics, security audit, risk management, legal issues, and deep mathematical aspects of cryptography.

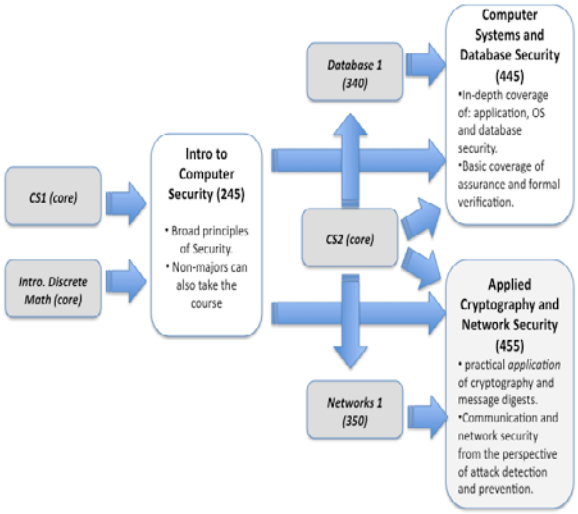

Figure 1: Summary of the Certificate. The arrows show the pre-requisite chain. 245, 445 and 455 are the three security courses.

Figure 1 gives an overview of the courses that form the certificate along with their pre-requisites. We developed a standard introductory course covering a broad range of security principles in hopes of attracting both majors and non-majors early in their undergraduate studies. This course serves as a starting point for other courses that go into depth. It also introduces students to certain topics in physical and administrative aspects of security such as risk assessment and management and legal issues, which we don't delve on in the more advanced courses. We limited the pre-requisites for this course to introductory programming (CS1 level) and discrete math – both freshman level courses. We also developed, two advanced senior-level security courses, reflecting topics (primarily technological) across the concentrations. In keeping with our goals discussed earlier, we decided that the following core topics should be covered as part of the advanced level courses:

- *Application security*: which gives students substantial insight into principles of secure design and implementation of software applications. As all students in the department have to take CS2 (Principles of programming II course in Java) as a core course, they have sufficient background for us to get into depth in covering secure design principles and coding guidelines.
- *Operating System and Database security*: which gives students examples of threats and controls on large software systems. While OS security is a large topic worthy of its own course, we decided to limit our coverage on OS security to user-level security issues. We made this decision for this reason:
    - To keep the pre-requisite chain small so that students have the flexibility of taking the security courses even in their junior year. Covering more topics in OS security requires that the student have a background in OS fundamentals including concepts such as processes, process management, and, memory and file management. Currently, only one concentration in our program, *computer science*, requires students to take a junior level OS course. Prerequisites for this include computer organization (another junior level course) and several other courses. Therefore, making it a pre-requisite for a security course will require students to take other pre-requisites and introduce a long enough pre-requisite chain that could potentially delay the graduation date for the students, making our certificate less attractive.

    For databases, basic SQL knowledge is required to understand threats and moreover, some experience in using a DBMS is required to understand and implement prevention techniques. Hence, we decided to make the introductory database course a pre-requisite. Unlike the OS course, this course doesn't have additional pre-requisites beyond the core courses for the degree.

    A broad coverage of formal verification techniques is also discussed for benefit of students interested in graduate study.
- *Applied cryptography*: We designed this topic with focus on the practical usage of cryptography, such as the modes of operation and using them in authentication protocols (e.g., Kerberos), rather than going into deep details on the mathematics behind cryptography. As students do have background in discrete mathematics, the course does cover some of the cryptographic algorithms such as RSA in depth, but only provides an outline for more complex mathematics such as elliptic curves. We did this to motivate students who may be interested in graduate study.
- *Network Security*: This topic was designed to equip students not only with skills to handle network break-ins, disaster recovery and prevention techniques. In addition, given the importance of network security, we decided to cover issues related with security in protocols. As knowledge of basic network components, protocols and transmission techniques is necessary to understand network security, we decided to make the introductory course in networks a prerequisite for this topic.

These topics are covered using two courses:

- *445: Computer Systems and Database Security*: a

senior level course with focus on application, operating system, database security and a brief coverage of assurance and trust.
- *455: Applied cryptography and network security* a senior level course with focus on applied cryptography and communication security.

These two courses count towards technical electives and B.S requirements in all the concentrations. In addition the certificate has only two pre-requisites, 350: networking and 340: databases beyond the core courses. These two courses are either required in some of the concentrations or would count as technical electives or BS requirements. Therefore, a student in the Bachelor's program could plan on finishing the certificate by just investing additional time for one course (245: Introductory Security).

A. Developing Labs/Projects for these courses

In the information security education community, there seem to be two points of view: those who believe in the need to have live projects and labs that give students practice with attacking/hacking techniques, e.g., [10], and those who believe the focus should be less on attacks and more on detection/protection techniques, e.g. Logan et al. [11]. Our approach is similar to those of Whitman et al [6], Logan et al. [11], and Sexton [8]: incorporate both points of view, with a lot more emphasis on the latter. Regardless of the approach, one thing that many educators seem to agree on is the need for a dedicated lab [12], [13].

Developing a lab for the first course was the easiest due to its introductory nature. We could design assignments that did not require a dedicated lab: the homework was either theoretical and problems requiring use of software could be done without system administrator privileges. Hence, the department's standard computer labs were sufficient. However, we had to develop dedicated labs for the two senior level courses. The following subsections discuss our experience in developing the labs with limited resources and creating projects that balance hacking with detection and prevention techniques.

B. Computer System and Database Security Course

We offered this course in Fall 2009. To setup a dedicated lab without expending too many resources we used virtual machines, a very well researched and established option being used in many Universities as discussed in [9], [14], and [15]. We used the freely available VMPlayer™[2], installed on an old (reclaimed) Linux server to setup the lab for no additional expense. The main challenge was giving projects that would in some ways fit the three areas covered in-depth by the course: application, OS and database security. We used the following projects:

- Introductory project: attacking security servers. We installed the OWASP live CD[3]. This includes attacks ranging from: web based attacks (cross site scripting), parameter manipulation and database specific SQL injection attacks. We picked these specific attacks to motivate the need for defense in depth.
- Subsequent projects: In keeping with our goal of emphasizing protection and detection, we shifted the focus of subsequent projects to the defense in depth design principle for securing hosts.
    - Second project: Developing secure applications. Students in the department primarily learn Java. We gave the students an insecurely designed and implemented Java package. The package provided methods to manage a simple *employee payroll system*. Students had to identify the implementation and design loopholes of the package and fix them. To develop the package we started with the CERT Sun Microsystems Secure Coding Standard for Java[4]. This standard specifies over 14 categories of secure coding guidelines, which seem to cover several specific standards. We picked 10 standards we normally encounter in Java programming and purposefully violated those standards in the package code. Students had to manually go through the code and identify at least 6 of these violations and then fix some of the violations in the code.
    - Third project: This project built on the previous one with focus on database security. The payroll program from the second project was modified to now use a database to store the payroll using JDBC. The students had to build a front-end in Java that filtered SQL queries to prevent database inference attacks. Due to lack of time in the semester, we made this an extra credit project.
    - Fourth and subsequent projects: These projects were based on configuring security software on operating systems. Students worked on setting up a role based access control system (SELinux[5]), a one-time password system by configuring PAM, and performed other crucial and basic security administrative tasks. These included conducting security audit and evaluating audit records.

[2] http://www.vmware.com

[3] http://www.owasp.org
[4] http://www.securecoding.cert.org
[5] http://www.nsa.gov/research/selinux/index.shtml

C. Applied cryptography and Network Security

This course is being taught (Spring 2010) as of the time of writing of this paper. This is a standard network security course; the topics covered are in the areas of applied cryptography and communication security. Being a course common across all concentrations, we are placing more emphasis on the *applied* aspect of cryptography. This primarily includes discussing the practical usage of secret-key, message digests and public-key cryptography. In the second part of the course we discuss topics related to network and telecommunication security.

As many educators have discussed ([12], [10], [9], [16]), a network security course requires a lab with dedicated networks to provide students hands-on experience. The virtual machine setup we had for the *computer system* course is ineffective because of the limits in resources in setting up a virtual network. The server we have for running the virtual machines cannot execute more than 4 virtual machines at the same time. Moreover, as most students primarily have access to Windows™ based PCs, they had to run an Xwindows server to run the VMplayers™ (which run as Xclients on the Linux host). This proved to be painfully slow even when students were on-campus when we tried setting up a network of virtual machines. Other solutions such as developing a lab on the scale of Iwar [5] or even a portable network laboratory such as GW-PEN [16] were not realistic given limitations in our resources as we were offering these courses for the first time. For setting up a network, we were motivated by a graduate level security course in the University of Delaware[6], and just like them decided to use the publicly accessible *Emulab*[7] [17] created and maintained by the University of Utah. Briefly, Emulab allows users to create large networks by sharing several PCs. It also provides easy access to them through SSH. Emulab works by providing users with their requested number of computers (up to a certain limit) for the duration of their experiments from a pool of currently available PCs. Once the experiments are over, the systems are swapped out and returned to the general pool for everyone to use. A fresh image of the operating system is installed during each swap-in, thus wiping out the previous installation. Using Emulab, we were able to create several network configurations with commonly used operating systems such as Fedora Core™ Linux, Windows XP™ and FreeBSD. The one downside of Emulab we faced was that the software installed on the system is erased/lost when the systems are swapped out and returned to the general pool after a period of being idle. This requires that the students finish their project in one session – or come up with ways to install the software in their user directories. To alleviate this problem – we are providing each student with a desktop to finish and test his or her installation before deploying it on the emulab. Just like other Departments with similar constraints [18], we decided to rely on free-ware or open-source software to run on the networks created on Emulab.

Our goal was to provide students with a good overview of both hacks and attack prevention/detection using end-to-end security. Keeping in mind the limitations of emulab, we used the following set of network security projects:

- Project 1: Casing a network: This project is similar (and greatly influenced by) the first project in the Network security course at the University of Delaware[5] and based on the first chapter of the book *Hacking Exposed* [19]. We setup a network on emulab with computers running various operating systems (Windows XP™, different version of Linux and FreeBSD) and different servers on them (E.g., *SSH*, *telnet server*, *web server* etc.). Students had to identify the topology of the network as well as fingerprint each computer on the network to identify the servers (both TCP and UDP based) and vulnerabilities on them. This involved hands-on experience with *nmap*[8], *traceroute*, *ping* and *SARA*[9].
- Next projects: While the first project focused on how attackers hack a network, subsequent projects were chosen based on providing hands-on experience with attack detection and prevention.
    - Project 2: Detecting network scans. This project required students to repeat the first experiment but this time, the students were looking at how their actions could have been detected by an alert security professional using security tools. Students installed and used the Linux firewall *iptables* and the *port scan attack detector* (PSAD[10]) tool to detect scan attempts.
    - Project 3: Installing and using Kerberos. This project is the first of the many that will give the students hands-on experience in configuring secure network services. Students install Kerberos and then configure a client such as SSH to use Kerberos based tickets for authentication.
    - Project 4: Perimeter security. Students will provide perimeter security to a large network on Emulab that is running web and other servers. As part of the project students will install a firewall (using *iptables).* They will also provide legitimate users of the network a VPN (optional) to securely login

[6] http://www.cis.udel.edu/~sunshine/F03/CIS659
[7] Several papers discuss the design and implementation of emulab as listed on the following URL: http://www.emulab.net/pubs.php3
[8] http://www.nmap.org
[9] http://www-arc.com/sara
[10] http://cipherdyne.org/psad

to the network. Furthermore, they will (optionally) install and configure an intrusion detection system (our plan is to use *snort*[11]). To test their installations, students will be made to run the same scanning techniques they used in Project 1 and then see if they are able to prevent such scan attempts with the perimeter security. In addition, students will run Wireshark[12] to monitor network traffic and use it to debug their DMZ configuration. Students will optionally run other tools such as Cain and Abel[13].

We must note that while we have discussed only the hands on projects, students also worked on more theoretical homework such as analyzing security handshake protocols and cryptographic algorithms.

D. Conduct of the Certificate

As the Certificate is in its infancy, the first course is being offered every alternate spring semester. The next two courses are offered in the next two semesters following the introductory course.

IV. Conclusion

This paper discussed the development and preliminary implementation of an undergraduate certificate in information security at Radford University. We developed the curriculum by customizing the vast base of existing knowledge to satisfy the different constraints in the IT department at Radford University. Despite limitations in resources, by using publicly available network resources such as Emulab and the extensive existing knowledge on using virtual machines in security courses we were able to provide students with hands-on experience. Furthermore, we were able to provide skills needed for students in most of the concentrations in the department.

The courses in the certificate have been certified to meet CNSS[14] standards 4011 (INFOSEC professional) and 4013E (System administrator - entry level). The first set of students to complete the certificate will graduate in May 2010. As we go through more iterations of the certificate we plan to collect feedback and evaluate the outcomes of this certificate. If the certificate program is successful in attracting enough students, we are also interested in refining our projects as well as looking at developing a network similar to GW-PEN [16], using older (reclaimed) PCs and routers.

IV. References

[1] M. Bishop, “Academia and education in information security: Four years later,” in *Fourth National Colloquium on Information System Security Education*, 2000, pp. 30–32.

[2] T. A. Yang “Computer security and impact on computer science education”, in J. *Comput. Small Coll.* 16, 4 (Apr. 2001), 233-246.

[3] P. Logan, “Crafting an undergraduate information security emphasis within information technology”, in *Journal of Information Systems Education*, vol. 13(3), 2002.

[4] S. Azadegan, M. Lavine, M. O’Leary, A. Wijesinha, and M. Zimand, *A dedicated undergraduate track in computer security education*. Norwell, MA, USA: Kluwer Academic Publishers, 2003.

[5] G. Conti, J. Hill, S. Lathrop, K. Alford, and D. Ragsdale, “A comprehensive undergraduate information assurance program,” in *Security Education and Critical Infrastructures IFIP TC11/WG11.8 Third Annual World Conference on Information Security Education (WISE3)*. Kluwer Academic Publishers, 2003, pp. 243–260.

[6] M. Whitman and H. Mattord, “A model curriculum for programs of study in information security and assurance,”

[11] http://www.snort.org
[12] http://www.wireshark.org
[13] http://www.oxid.it/cain.html

[14] http://www.cnss.gov

in *http://infosec.kennesaw.edu/InfoSecCurriculumModel.pdf*, 2004.

[7] M. Hentia and H. Dhillon, "Towards changes in information security education," in *Journal of Information Technology Education*, vol. 5, 2006.

[8] J. Sexton, "Establishing an undergraduate information assurance (information security) program at a small liberal arts college," in *J. Comput. Small College*., Vol 24 (2), pp 234-240, Consortium for Computing Sciences in Colleges, 2008.

[9] H. J. Mattord and M. E. Whitman, "Planning, building and operating the information security and assurance laboratory," in *InfoSecCD '04: Proceedings of the 1st annual conference on Information security curriculum development*. New York, NY, USA: ACM, 2004, pp. 8-14.

[10] G. Vigna, "Teaching hands-on network security: Testbeds and live exercises," *Journal of Information Warfare*, vol. 3, no. 2, pp. 8–25, 2003.

[11] P. Y. Logan and A. Clarkson, "Teaching students to hack: curriculum issues in information security," in *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*. New York, NY, USA: ACM, 2005, pp. 157–161.

[12] C. Irvine, "Report on the first acm workshop on education in computer security," *ACM SIGSAC Review.*, vol. 15, no. 2, pp. 3–5, 1997.

[13] J.M. D. Hill, C. A. Carver, Jr., J.W. Humphries, and U.W. Pooch, "Using an isolated network laboratory to teach advanced networks and security," *SIGCSE Bull.*, vol. 33, no. 1, pp. 36–40, 2001.

[14] D. Collins, "Using VMware and Live CD's to configure a secure, flexible, easy to manage computer lab environment," in *J. Comput. Small Coll.*, vol. 21, no. 4, pp. 273–277, 2006.

[15] W. I. Bullers, Jr., S. Burd, and A. F. Seazzu, "Virtual machines - an idea whose time has returned: application to network, security, and database courses," in *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*. New York, NY, USA: ACM, 2006, pp. 102–106.

[16] T. Rosenberg and L. J. Hoffman, "Taking the network on the road: Portable network solutions for computer security educators," in *J. Educ. Resour. Comput.*, vol. 6, no. 4, p. 2, 2006.

[17] J. Duerig, R. Ricci, J. Zhang, D. Gebhardt, S. Kasera, and J. Lepreau, "Flexlab: A realistic, controlled, and friendly environment for evaluating networked systems," in *Record of the Fifth Workshop on Hot Topics in Networks (HotNets V)*, Irvine, CA, Nov. 2006, pp. 103–108.

[18] Bhagyavati, Agyei-Mensah, S. O., Shumba, Rose, Kearse, and I. B.C., "Teaching hands-on computer and information systems security despite limited resources," *SIGCSE Bull.*, vol. 37, no. 1, pp. 325–326, 2005.

[19] S.McClure, J. Scambray, and G. Kurtz, *Hacking Exposed: Network Security Secrets and Solutions, Sixth Edition*. McGraw-Hill Osborne Media, 2009.