SECURITY ACROSS THE CURRICULUM: USING COMPUTER SECURITY TO TEACH COMPUTER SCIENCE PRINCIPLES

Major Gregory White, Ph.D.
Captain Gregory Nordstrom (ret.)
2354 Fairchild Dr., Suite 6K41
HQ USAFA/DFCS
USAF Academy, CO, 80840
white@cs.usafa.af.mil
gnordstr@cs.usafa.af.mil

ABSTRACT

Insuring that individuals who obtain computer science degrees have a sound foundation in security principles is becoming increasingly important as the worldwide connectivity of our networks grows and the number of security incidences increases. Increasing the number of courses a computer science major is required to take by adding additional computer science courses dealing with security is not the solution, however. Instead, an organized approach to include security topics into already existing curricula (as was first proposed in ACM's Curricula '91 document) is the key. This paper describes the approach taken at the United States Air Force Academy in introducing security topics at numerous points in its computer science curriculum. This approach goes far beyond briefly mentioning security at various points, pioneering the concept of using security to actually teach core computer science principles. This paper focuses in particular on changes that have been made to the Networks course required of all computer science majors which has been modified to use security to help illustrate and teach the underlying network principles.

INTRODUCTION

An ever growing number of colleges and universities have introduced courses in computer security. While this increased attention to security in academia is a good sign, the courses are being offered as elective courses. As an elective course, a significant number of students will not take these security courses which means that a significant number of computer science majors at these institutions will graduate without a solid background and basic understanding of security.

The ACM Curricula '91 document, proposed that a basic amount of computer security and ethics education be covered in all computer science programs. While the option to offer an elective course was acknowledged, the document proposed that a certain amount be covered at appropriate times in the curriculum. With the increasing need for computer professionals who have a solid grounding in security principles, this rather passive approach to security education is not sufficient. At the same time, computer science programs do not have the luxury of adding additional required courses to what in many cases is an already full program.

The solution to this dilemma is to introduce an organized approach to teaching security across the curriculum. Instead of addressing security topics as separate issues, security should be woven into all courses that make up the fabric of the core computer science curriculum. Indeed, what is needed is to make security considerations and concerns part of every programming assignment given to computer science students. In a manner similar to questions about good coding practices, students should be taught to always consider the security implications of any program developed.

The introduction of computer security across the curriculum should not come at the expense of other topics. Instead, security should enhance the learning of these other topics. Indeed, in certain courses, because of their very nature, security can actually be used to help teach the course itself. An example of this is a course in networks and computer communications which has numerous opportunities to introduce security related projects.

SECURITY ACROSS THE CURRICULUM

In today's heavily internetworked computing environment it is imperative that all students of computer science have an understanding of computer security principles and practices. Consequently, any implementation of security across the curriculum should begin with the first introductory computer science course. Many other majors today require some exposure to computers, in their introductory courses security should also be addressed. At this most basic level the detail required is minimal. Exposure to the concept of viruses and how to protect against them, good password management techniques, and elementary encryption issues will serve to introduce the students to the idea that security should always be a concern. Most of the time at this level is better spent in addressing the ethical and legal issues surrounding 'hacking' and viruses. Discussion on subjects like the ease in which electronic mail can be spoofed, or the fact that an individuals password or credit card numbers can be discovered using 'sniffers' will alert both the computer science major and the non-major alike to the real dangers that are present in placing too much trust in insecure networks. Programming assignments at this level will probably allow for few opportunities to address security concerns but research papers on subjects like public key encryption, malicious software, and 'hacking/cracking' provide ample opportunities to raise student's level of security awareness.

An operating system course provides many opportunities to address security issues both from a practical and a design point of view. Issues such as access control are already part of almost all textbooks on operating systems. Other issues such as authentication, object reuse, auditing, and security kernels also lend themselves to this course. For those interested in introducing even more security, the issues of multi-level security and its many additional requirements as well as the writing and detection of viruses and other forms of malicious software provide ample opportunities for programming projects.

While entire books have been written on data base security, many general textbooks designed for introductory data base courses often spend only a few pages on this subject or ignore it entirely. Issues such as multilevel protection, polyinstantiation, access modes, auditing, and inference controls provide a rich opportunity to reintroduce security concepts to the students.

Second only to operating systems in its opportunity to introduce security topics, a course in networks provides some of the best possibilities to stress the importance of security. This can easily be reinforced through the use of the many articles that appear in the news media concerning lapses in security protections in networks and computer systems. There are numerous security topics which can be used to illustrate or emphasize various network principles. Among these are cryptography, intrusion detection, firewalls, "worms", and security among distributed systems.

Software engineering courses with their emphasis on the entire life cycle of software also present several opportunities to discuss security issues. The design phase of software development provides the chance to discuss the modeling of secure systems. Discussion of program testing provides similar opportunities to discuss verification and validation. Covert channel analysis can also be easily introduced into this course.

USING SECURITY TO TEACH COMPUTER SCIENCE

The first course in which we attempted to use security to teach the principles embodied in the course was our senior level networks course. In the past, we taught the course centered around the seven-layer OSI model familiar to all who have taken an undergraduate-level network course. Lab assignments involved such tasks as development of programs to perform remote file transfer. These assignments, while providing examples of what was seen in lectures did nothing to motivate or excite the students. The labs were completed, the lessons learned, and the entire experience was then most likely quickly forgotten.

The most immediate benefit we observed using security to teach networking principles was a renewed enthusiasm for the course and computer science in general. Individuals who had been exhibiting only mediocre interest in their coursework came alive when challenged with our security related lab assignments.

The specific assignments used in this course began with simply downloading and running programs such as *crack*. This allowed the students to become comfortable with downloading and working with a program to get it to run on their specific system. It also served to illustrate how vulnerable a system is if an intruder is able to gain access to the password file. The students next learned to use the program *tcpdump* to monitor the packets that are sent across the network. Their assignment forced them to use several different options for this program and to track and observe many different types of packets that are sent across the network. When the assignment was distributed, we conducted a discussion on how this specific program, and other programs called ‘sniffers’, can be used to obtain passwords. The isolated nature of the lab meant the students weren’t able to discover passwords to systems outside of their special subnet. While it would be absurd to assume that some student won’t take advantage of this program on the isolated systems for mischievous purpose, the amount of damage, intentional or unintentional, that an individual can cause is very limited. This assignment also served to illustrate the different types of packets and their formats used in the TCP/IP protocol suite.

The next series of assignments had the students exploiting well known holes in a variety of packages. Many of these holes have been fixed in later releases of system software (which actually caused some problems as we had to insure that we didn’t upgrade all of their systems). Examples of the types of holes/flaws they exploited include SMTP spoofing, the ***sendmail*** */etc/passwd* file hack, the TFTP */etc/passwd* file hack, and a ***uudecode*** spoof.

The culminating event for the course was the final project which was referred to as a ‘hack-off.’ For this assignment, the students were divided into teams which were further divided into two squads. Each team had an offensive and a defensive squad. The hack-off consisted of the teams attempting to break into their opponents systems while protecting their own. The systems they used were all on the isolated subnet and had been ‘cleaned’ prior to the event so they resembled their original, ‘out-of-the-box’ condition. The teams were provided a list of capabilities or functions their systems had to support at the start of the exercise. The instructors periodically checked the systems to insure the required capabilities still existed. This was done to insure that teams didn’t simply “unplug” their system from the net and added a level of realism to the exercise. At various points in the exercise additional requirements were added to simulate the ever-changing environment administrators face. Not only did the students enjoy this project, they had the opportunity to actually get hands-on experience in minor system administration and security protection. The lessons they learned in this exercise will undoubtedly provide big dividends as they leave the academic environment.

ADDRESSING THE ISSUE OF ‘HACKER’ TRAINING

At first glance it may appear that the approach that we have taken at the Air Force Academy results in nothing more than a basic primer for the training of computer hackers. Implementation of a program similar to ours at other institutions where even less control of the students is possible will undoubtedly result in abuses of the information presented. During the initial implementation of this program, as the students and instructors were setting the boundaries, there were indeed minor incidents which were quickly resolved. Since these minor infractions, no problems have been encountered. We believe that this is partly due to the laboratory environment we have set up. We have a series of machines that were separated from the rest of our academic network which allowed the students to experiment in a controlled environment. Indeed, we encouraged them to test the security boundaries on these machines. Doing so has allowed our students to satisfy their curiosity and to learn many valuable security lessons without fear of destroying other important work in progress. At the same time, they could feel secure in that they did not have to hide their actions because of a fear of potential criminal prosecution. This fostered an environment in which the students freely shared the ‘tricks’ they learned.

We have had some claim that what we are doing is unleashing a new generation of trained hackers on the Internet. We do not agree with this sentiment. There are scores of hackers operating throughout the Internet today. We believe that hiding their techniques from our students only leads to a generation of system administrators who are ‘sitting ducks’ for the hackers that are out there. We use a knowledge of security holes to teach our students what must be done in order to secure their own systems. By doing so, our graduates are better able to handle the attacks on their systems that will surely occur.

CONCLUSIONS

As we have implemented our security across the curriculum program, we have noticed a number of benefits. The first one was a new level of interest in computer science from those who had previously not considered registering for the computer science major. There is a certain “frontier mystique” surrounding hackers and those who protect computer systems and networks from this new breed of “outlaws.’” On several occasions we have been able to use this interest to capture a student’s interest long enough to explain the major to him/her which has resulted in an increase in the number of computer science majors.

Along with a new interest in the major, the introduction of security topics has renewed a number of the computer science majors interest in the program. A number of those, who had in the past shown less than total enthusiasm for the program, had a spark ignited in them with security and showed an improvement in their overall performance.

Using security to teach computer science principles did not detract from the other course material. We were able to use it to enhance the lessons being taught, to emphasize the points being made in a manner that the students found interesting. While this concept could be taken to the extreme and security forced upon all computer science courses, we did not take this approach, instead choosing to include it only in those programs for which we could see the course objectives easily applied to a security environment. This resulted in a well-balanced series of courses and an overall organized approach to applying the recommendations of the ACM Curricula ‘91 committee.

Finally, we entered into this experiment with a certain amount of apprehension surrounding the possibility that the things we taught could be used in an inappropriate manner. While we did indeed experience some minor incidents in the beginning, the students eventually settled down and did not push beyond the boundaries that were ultimately worked out. As a result, we do not believe that we have trained a corps of hackers, but rather have created a corps of “cyber defenders’” ready to leave academia and enter the work force prepared to defend their systems from the hackers that already, and will continue to, exist.