# Computer Security in Undergraduate Computing Curricula: The Challenge of Educating the Next Generation of Information Security Specialists

Andrew T. Phillips, University of Wisconsin–Eau Claire, Eau Claire, WI

I agree with the basic conclusion of participants in the Computing Research Association's conference on grand research challenges in information security and assurance (1). Here's what they said:

1. There are two possible futures in computing. The first and most likely future, if we as computing experts and educators do little beyond what we are doing now, is one in which spam, viruses, worms, ID theft, data theft, and network outages are rampant and unchecked. In such a future, the computer will cease to be a productive enabling tool but rather be a nuisance not worth the frustration of using. In such a future, users will require constant support from systems and networking experts who tweak the configurations and patch the holes in an ever more desperate attempt to fend off the attacks. In such a future, there will be no security of data, networks, and systems. Everything will be compromised or will be a potential target for compromise.
2. The second possible future is one in which computers remain the boon of productivity; a future in which there are no spam, no viruses, and no worms; a future in which data and personal privacy are protected, even to the extent that (finally) personal medical records can be transmitted electronically, both safely and securely; a future in which networks rarely "go down" and users have essentially unlimited and trouble-free access to required resources.

Who would choose future number 1 above? But as I sit and write this chapter, I am receiving spam at the rate of about one new message per hour. And yes, I am running a spam filter that does indeed filter quite well!

Are computing experts and educators adequately preparing this nation's undergraduate computer scientists to wage war on future number 1 and reposition us to move closer to future number 2? In some select instances, the answer is "yes." But, on the whole (and this is just an observation of mine), the answer is "no." Here is my claim: while almost all computer science faculty in almost all of this nation's computing programs would likely acknowledge the great need for technical coursework and experiences involving the current issues in computer security, very few such faculty have sufficient background and experiences in this field, and so very few are adequately prepared to deliver the necessary courses and curriculum. This is not in fact a criticism of anyone—certainly not the computing faculty. Attacks of the types I mentioned above are a fairly recent phenomenon. I would argue that it wasn't until the Robert Morris "worm" of 1989 (2) that anyone took much interest in this aspect of computer security. And it probably wasn't until about 2003 that spam really became a nuisance (does anyone know what happened in 2003 that accelerated this problem?). So, computing faculty across the nation are now challenged to respond to something that is both pretty recent and pretty urgent.

What's my point? It's simple. Computer science educators must prepare and train many more undergraduates in the field of computer security. In particular, students need extensive hands-on training and education on what I call "the art and science of attack and defense." That means providing coursework covering the concepts of computer security; but even more importantly, it means providing practical hands-on laboratory experiences in which students learn to use various security tools, defend and (yes) probe systems,

investigate and patch flaws, and learn through experience how to defend by a combination of system hardening and intrusion detection.

A discussion of specific course content that one could include in a computer security course is always a fun subject. There are many possible ways to do this. Some stress Linux-type systems; others stress Windows platforms. Some stress defensive measures exclusively; others include some attack strategies. And then you can argue the relative amount of time and detail that should be placed on social engineering, technology, and policy creation. But in my space here, I want to focus on something different. Regardless of what the computing faculty decide to cover in a computer security course, most faculty need some practical training to support their efforts. Specifically, computing faculty need guidance and hands-on experience with a variety of tools and techniques so that they can organize and teach an effective course on the subject, even if they don't choose to include all of the possible topics. The most critical aspect here is the hands-on experience with tools and techniques in defense and attack.

What I'm suggesting here, if we intend to head off future number 1 and move toward future number 2, is a series of faculty workshops offered through a variety of national venues (conferences typically). The workshops that I envision could be considered as faculty "professional development" and could be offered as a supplement to the normal activities of the conference.

What would be required to provide interested computing faculty with enough hands-on exposure to both tools and techniques so that after a single workshop series they can be expected to "make a difference" in teaching computer security? Certainly not theory or concepts that one can read about in textbooks or journals. I make the assumption that such faculty will always read relevant background materials covering theory and concepts and that workshops would not be required for that. What such faculty need is an idea of how the security tools and techniques can be explored and taught with practical hands-on exercises. What they need is an experience that demonstrates the concepts in action—in real time. What they need is to see that by carefully pre-configuring a system with a variety of security flaws, open ports, poor user passwords, etc., students can experience both attack and defense in an educational setting that links the theory and concepts with the tools and techniques of practical computer security. What they need is to experience this for themselves so they are better prepared to pass on the information and experiences to the students in whatever way they see fit in their own environment.

What I propose is that we package a subset of typical computer security exercises into something like a six-hour (three hours per day for two days) hands-on computer security workshop for computer science educators. I envision that the first three-hour block will provide directed instruction on the use of various tools commonly used for gathering information and assessing the vulnerability of other systems. One possible approach here is for participants to use a laptop running both a Windows XP image and a Linux image (through virtual software such as Microsoft Virtual PC) preconfigured with all the tools required in the workshop. In addition, participants will be asked to experiment with the tools throughout this first day of the workshop as the workshop facilitators guide them through typical tool use and scenarios. This first block would conclude with an information-gathering exercise on an isolated network. The system images will be preconfigured by the workshop facilitators with a variety of common security "holes" so that the participants may experience first-hand the process of information gathering and vulnerability detection. Some examples of tools and techniques to be discussed and practiced should be as follows: firewall configuration (example tool: ZoneAlarm), packet sniffing (example tool: ethereal), open ports/unneeded services left available (example tool: nmap), bad passwords/unsafe accounts (example tools: john the ripper and l0phtcrack), and general vulnerability assessment (example tool: nessus).

There is one key aspect that I think is crucial to the success of the first phase of such a workshop (at least as I have described it so far). The experience must be fruitful; it must present sufficiently interesting possibilities so that faculty will quickly see the ways in which the practical hands-on experience can really link the theory and concepts with a "real" situation. This is no easy task. If done poorly, the experience will only convince faculty of the complexity of the task and encourage them to ignore, in their own teaching, the practical side of security. In fact, many faculty might argue that there is little to be learned from such a cooked-up scenario in the first place. I disagree. In fact, I think one

of the most important lessons to be learned by our future information security professionals is that of constantly gathering information, testing, and hardening your system. It's all about process, not product. And one very common scenario is to be asked to do this on a system that is already in operation and has been administered, possibly for years, by others. It's very enlightening to conduct an information-gathering exercise and see the consequences of "flaws."

The second three-hour block of the workshop would provide directed instruction and experimentation on defensive techniques and an understanding of typical exploits for the purpose of better defending systems. Some examples of tools and techniques to be discussed and practiced here would be intrusion detection (example tool: snort), auditing and log analysis (example tool: logwatch), and checking for root kits (example tool: chkrootkit). This second day should conclude with a longer hands-on exercise giving the participants an opportunity to participate in a carefully constructed and monitored cyberwar laboratory scenario, i.e., the participants will further harden their systems, identify potential exploits and threats that can affect their system, and work to understand the mindset of the attacker by identifying weaknesses in their own and other systems on the isolated network.

Based on two years of experience with computer science undergraduates in a computer security course at my institution, the most significant educational experience the students get in computer security is when they apply their understanding of the concepts and theory to a controlled attack and defend exercise in an isolated network environment. We preconfigure a network topology and a variety of systems to include many different holes, flaws, and weaknesses. The students are given 24 hours to perform information gathering on the network and on their Linux system and to use that time to harden their systems by removing unneeded services, closing open ports, changing weak passwords, etc. There are rules of course: There is a list of services that must be provided, and there is to be no "attacking" other systems during this phase. But once the second 24 hours begins, it's war. The students are strongly encouraged to continue to monitor and harden their systems (those who don't, usually pay for it), but they are also permitted to probe other systems for weaknesses in a capture-the-flag style game. I'm suggesting a small-scale and modified version of this game be conducted at the tail end of the faculty workshop as well. The modification would be that the probing would be done by the workshop facilitators so that the participants can quickly see the areas they may have missed without having to guess what attacks to mount.

The last key aspect of my vision is that the workshop facilitators must provide all workshop computing and network equipment; workshop participants cannot be expected, nor permitted, to use their own computers. It is only in this way that the systems, security tools, and exploits can be provided, carefully controlled, and reconfigured throughout the workshop and that the threat to the host venue (i.e., conference or university) can be minimized. Far too many great ideas for faculty professional development fall short of expectations because of limitations and unforeseen obstacles in the computing environment. In short, the workshop materials and infrastructure must be completely isolated and independent of the workshop site. They must be pretested by the workshop facilitators and ready for "prime time" onsite.

In conclusion, I return to my main point. Computer science educators must prepare and train many more undergraduates in the field of computer security. Students need extensive hands-on training and education in defensive hardening of systems as well as the most common attacks and exploits. That means providing faculty with the same sort of practical hands-on experiences that we would wish them to use with students. If we are to rise up to meet the grand challenges in research in computer security, we'll first need to meet the challenge of educating the next generation of information security specialists.

## REFERENCES

1. http://www.cra.org/Activities/grand.challenges/security

2. http://world.std.com/~franl/worm.html