Teaching Statement

Brendan Dolan-Gavitt

Education in computer science serves two vital but sometimes opposing interests. It introduces students to the body of scholarship and knowledge in the field, giving them the understanding they need to eventually build on existing work and make their own contributions—in other words, it prepares them to be a new generation of scholars. At the same time, it is undeniable that most students in computer science will seek jobs in industry, and it is therefore the duty of an educator to inculcate practical skills and habits. I believe that successful instruction in computer science must balance these two interests, and seek out educational experiences that can serve both.

One way I have tried to accomplish this is by finding ways to integrate cutting-edge research into the classroom. In the summer of 2010, I was invited to teach a one-day tutorial session. In addition to teaching the basics of virtualization and its application to security (providing strong isolation, reducing attack surface, and so on), I wanted to give students a taste of how these ideas were being used in current research. I created a lab exercise based on an research system that had recently been published by a colleague. The paper described a virtualization-based application level firewall that could allow or deny outgoing connections based on the originating application without running in the same virtual machine as the applications themselves. I felt that having the students recreate a simplified version of this research in the lab would be an excellent way to give students an understanding of how virtualization security can be used.

To make the exercise doable in the four hours available for the lab, I structured the lesson as a progression in which the students steadily did more of the exercise on their own. First, they ran built-in commands from an open source virtual machine introspection and memory analysis framework that listed network connections and applications running in the virtual machine. Next, they were tasked with modifying and completing a simple example that extended the framework to handle a passive intrusion detection task (check-

ing for malicious tampering by a rootkit). Finally, drawing on some existing libraries, students built a new extension that monitored outgoing packets, matched them to applications running inside the guest virtual machine, and allowed or rejected each packet. The lab was successful: the students enjoyed the experience and were able to deduce what needed to be done at each step (albeit with some hints or troubleshooting required at times). Several of them corresponded with me via email after the course was over about how to extend what they had done in class. This was very personally rewarding, as it demonstrated that my teaching had nurtured a more lasting interest in the subject.

Both from this experience and reflecting on my own education, I have concluded that hands-on experience, in which students actually build something interesting, is an excellent way to foster a thorough understanding of a subject. Although lecture and theoretical knowledge are important for building the foundation on which these projects can succeed, the experience of creating something on your own is vital to making lessons from the classroom stick. Moreover, if students seek employment in industry, being able to point to non-trivial projects they've completed can put them head and shoulders above other candidates.

I believe that excellence in teaching is a requirement for any faculty member, on par with achievements in research. Whereas the latter helps push forward the frontiers of human knowledge today, the former trains the scientists of tomorrow. For those students who do not go on to do research in computer science, a firm grasp of its fundamentals will help them navigate an increasingly computerized world, thus fulfilling the promise of a classical liberal education: producing informed citizens.