

My goal as an instructor is to teach in a way that I would have enjoyed during my undergraduate and graduate studies. Student engagement is important for lessons to stick, and I made engagement a priority during my time as a teaching assistant and content creator for online and offline software security courses.

Course Structure

Working alongside my advisor, Dr. Laurie Williams, I helped “flip” a graduate-level software security course for both the graduate students at North Carolina State University and an online offering of the same course to the public (i.e. a MOOC). I have an appreciation for what “flipping” content brings to students:

- **Personal interaction with students.** I want students to view me as someone who wants them to understand the material, rather than a delivery device for the assigned reading. Using class time to discuss particularly difficult concepts, current events related to the course topic, and working through examples or problem sets is, in my view, a better use of class time. While lecturing is an important part of the learning process, making myself available as they work through a problem for the first time can help students grasp new concepts quickly.
- **Student-to-student interaction.** By creating an environment for students to work collaboratively, students can ask each other for help while working together. Students helping each other has several positive benefits: the student asking the question can have it explained to them in a different way than I have, which may help understanding. The student explaining the answer can check their own knowledge as they talk through the issue, embodying the expression “If you really want to know something, teach it.”
- **Learn by doing.** While developing content for the software security course, I realized how intimidating topics like security can be for some students. Having students work through basic, guided examples of how to perform certain attacks, such as cross-site scripting attacks or buffer overflow attacks, can help de-mystify software security for them. I have made use of open-source tools for these activities, like the Open Web Application Security Project’s WebGoat, so that students have a safe environment to practice their skills. Additionally, I am currently working with Dr. Jason King at NCSU on a grant for developing security materials for instructors to use in their coursework. Using class time for “doing” will be a core part of classes I teach, regardless of the subject.

Lectures

When I have had the opportunity to lecture, I focus on keeping my lecturing segments as short as possible. Keeping these segments short helps students stay focused on the material longer, as research has shown that students tend to lose focus after 8-10 minutes of lecturing. These breaks can be facilitated in several different ways, including group discussion, question and answer sessions, or examples and problem sets.

The goal of my lectures is to *provide content that students cannot get out of a book or video at home*. That means reacting dynamically to the class. For software security courses, talking about how a current event in cybersecurity news relates to the current topic is not something you can plan weeks in advance. Keeping flexible time in the lecture schedule for such events with backup material available if nothing interesting happens is one way to incorporate current events into course content.

Mentorship/Advising

During my graduate studies, I had the opportunity to mentor two undergraduate students working on projects in the area of software security. In both cases, the students had minimal experience with security topics, and tended towards the least experienced and youngest members of their internship cohort. The students were very different. One was confident in their abilities but unsure about doing research in computer science, while the other was less confident but explicitly interested in computer science research.

These differences made it important for me to *tailor mentorship to meet the needs of the individual student*. For example, the first student could be left on their own for several days to work through an activity, while the second needed a more hands-on approach so they knew that they were on the right track.

In my view, my role as a mentor is to **help my mentee succeed at both their current task and their future career**. While the task the students were working on each summer was important, it was more important to me that the mentees developed skills that would help them accomplish their long term goals. At the beginning of each summer, I sat down with each student and talked about what their goals were: what they wanted to accomplish that summer, where they wanted to be after graduation, and what drives them. Because the first student was certain they wanted to do research but unsure about computer science, I focused on developing skills that would help them in graduate school regardless of the field of study. The student wrote their own research plan, wrote a four page paper detailing their results, and gave a “conference” talk to the lab at the end of the summer. The second student hopes to work at a prestigious software company after graduation, so we built an open source tool to support the research project that she could use in her portfolio, along with help developing her personal site and resume. These experiences inform how I will mentor Masters and PhD students. Individual attention to the specific needs of each student is, in my mind, critical for their success.

Conclusion

When I left industry to return to academia, I was motivated by the idea of helping students achieve their goals. I have been lucky enough to have excellent teachers and mentors during my career, whose instruction and advice has been critical for my growth as both a computer scientist and a person. If I can help others achieve their goals as well, I’ll be able to look back on my career and call it a success.