Sarah E. Elder

Teaching Statement

A fellow graduate student once commented that the only time I seemed truly happy with my work was when I was teaching. To be fair, in my area of expertise, software security, teaching is one of the more peaceful jobs. Education is also one of the most difficult and critical challenges in security and computer science broadly.

Teaching-Relevant Experience

- Taught Discrete Mathematics during the summer session at NCSU (2018) as Instructor of Record. Prepared and presented lectures and assignments based on existing material for a 15 student course.
- Worked as a Teaching Assistant for an undergraduate software engineering course, an undergraduate senior capstone course, and a graduate software security course.
- Gave two lectures for graduate security courses based on existing material
- Gave two lectures for an introductory computer science course (Java) based on existing material
- Mentored four undergraduate researchers
- Assisted in the NCSU Cybersecurity Awareness Data Privacy Checkup leading a small group of students, faculty, and staff in reviewing and updating privacy settings on participants' phones

Teaching Philosophy

There are several principles that I find helpful when working with students. These principles are only the beginning. As I explore new opportunities, I am always looking for ways to improve as a communicator and educator.

Engage: Engagement through activities, even simply encouraging students to come up and write the answer to a question on a whiteboard, is key to encouraging student learning. Trends such as "active learning" have made it even easier to find ideas for how to integrate student interaction into lectures.

Explain Why It Matters: As with a good research presentation, a good lecture helps the students understand why the material is important. Expecting students to pay attention when they may not know why they should do so is unreasonable. Explaining the importance of concepts is important to ensure students understand why they need to learn the material presented, both in the classroom and in the lab.

Assume Little or No Background Knowledge: From classroom lectures to research mentoring, it is important to avoid making assumptions about background knowledge. Even when a prerequisite covers some background knowledge, highlighting key terms that students can review may be helpful, as some students may have forgotten or never fully understood the

concepts to begin with. No student should be left behind because they come from a different background, both in terms of their previous knowledge and more broadly.

Use Diverse Examples and Approaches: Each student is coming from a different perspective. Using a diverse range of examples and many types of activities improves the odds that most of the students in the class will be able to connect with the material. In my own early experiences in computer science, as an individual who had never seen herself as a “computer person”, I found examples from library science or history much more relatable than examples from video games. Recognizing my own biases, I research examples from video games to include in my own lectures.

Offer Multiple Perspectives: I do not know everything, and I will not always be able to convey information clearly to all students in a lecture. When possible, it is important to provide students opportunities to review the material from other perspectives. Background reading, asking guest speakers come in, linking to relevant youtube videos, and having students work in groups or do peer evaluations can help students better understand the material.

Connect with Industry: Bringing in examples and relevant material from industry helps prepare students for their careers and helps teachers engage with students. For example, blog posts from industry experts may be easier to read than academic papers and can help students understand how concepts are applied in the real world. In addition to bringing my own industry experiences to teaching, I hope to find ways to bring in other industry professionals with more recent and relevant experience to discuss suitable topics such as the role of requirements in practice.

On Diversity

In computer science, but particularly in security - we cannot afford to continue to be a largely homogenous group of individuals. Protecting against diverse and creative malicious actors is not possible without building a diverse group of defenders. Improving diversity begins with the “gates” to our profession, such as education.

Being an effective instructor inherently requires endeavoring to be an effective instructor for a diverse group of students. Many of the principles I follow such as Using Diverse Examples and Approaches apply as much for teaching a diverse group of students as for teaching generally. We should always be aware that we may not know what a student is going through. I also do not pretend to be an expert on all areas of diversity. I hope to learn how to be better at educating and communicating with diverse groups of people through whatever resources are available.

Looking Forward

Particularly in security, computer scientists often decry “the human factor” as the weakest link in computer systems. While I do not know the answer to this problem, I cannot think of a better place to start than with computer science education. I am interested in teaching computer science basics, as well as software engineering and software security courses at all levels.