When I was pursuing my PhD, I was invited to give a guest lecture on fuzzing, a dynamic analysis technique, for a software security class. Reflecting on past lectures that had the most impact on me, I realized that the ones I remembered best were those where the instructors provided intuitive explanations of complex concepts. Inspired by this, I aimed to do the same in my lecture by breaking down the core ideas of fuzzing in a way that was both accessible and engaging. As an example, to illustrate why target feedback is crucial in fuzzing, I used an example of a conditional branch comparing a user input against a hard-coded constant. I highlighted how a fuzzer, blindly sampling from a random distribution, would struggle to generate an input that meets this condition due to the vast size of the input space. This helped students appreciate the need for strategies like coverage guidance to narrow down the search space systematically.

Teaching Experience. As a PhD student and postdoctoral researcher, I've broadened my instructional experience by delivering guest lectures on fuzzing and fundamental security principles in software security courses at undergraduate and graduate levels. Teaching at varied levels has allowed me to hone my ability to adapt to the complexity of course material and examples to ensure students grasped the concepts effectively. At Purdue University, I served as a graduate teaching assistant for two semesters in the undergraduate Operating Systems course, which had an enrollment of 120 students. In one of these semesters, I took on the role of head teaching assistant, leading a team of three TAs to support the primary instructor in managing the class. I also designed lab assignments from the ground up to provide students with experience building functionality in real-world operating systems.

Mentorship. Mentoring the next generation of researchers has been a key focus throughout my academic career. While at Purdue, I mentored undergraduate students on projects related to fuzz testing. During this time, I guided them through the entire research process—from idea conceptualization and debugging implementations to testing hypotheses and effectively presenting their findings in final reports. As a postdoc at Columbia, I continued this mentorship, working with undergraduate, Master's, and PhD students. At Columbia, I developed clearly scoped research ideas tailored to the skill levels of undergraduate and Master's students, allowing them to gain research experience without needing to invest excessive time in refining project ideas or assessing feasibility.

Teaching Philosophy. One of the key skills I want my students to develop in my classes is critical thinking. In our modern age, where we are inundated with information, it's crucial for the next generation to be able to analyze multiple data points and draw logical conclusions based on the evidence. Equally important is the ability to revise those conclusions when presented with new information. I emphasize this skill through concrete exercises, such as teaching threat modeling in the software security class. I present students with a real-world scenario of an application in use and ask them to identify potential ways it could be exploited. Then, I introduce new features to the application and ask them to reassess their initial models, refining their strategies based on the updated context. This approach illustrates that securing applications is an evolving process and teaches students to adapt their models as new information becomes available.

Teaching Plan. Owing to my prior experience, I will be particularly suited to teaching systems programming in the form of C programming to and fundamentals of operating systems to undergraduate classes. In addition, I would be very interested in taking charge of the software security class at both undergraduate and graduate levels. A core focus of mine when delivering these classes is to keep the students engaged in the subject matter by making the lectures interactive. When teaching the concept of pointers in an introductory C programming class, I will use exemplary code snippets employing pointer operations and ask the class to vote on what they think is the correct result of a pointer operation. I will then use simple animations to show values being stored or dereferenced from memory to bolster the mental model they have built about pointers.

I am dedicated to enhancing my students' critical thinking skills, enabling them to navigate and reason through today's information-saturated world. I focus on equipping them to evaluate diverse sources, question assumptions, and adapt their perspectives as new information arises. Outside the classroom, I create research opportunities that foster creativity and intellectual curiosity. By mentoring students through these projects, I aim to inspire the next generation of scholars to embrace risk-taking and make meaningful contributions to advancing knowledge.