Fellowship Plan
Anitha Gollamudi

My PhD thesis employs techniques from the theory of programming languages (PL) to building secure systems using Trusted Execution Environments (TEEs). My expected graduation date is in Spring 2021. In about two years, I am seeking to enter the academic job market for tenure-track faculty positions in R1 institutions. My short term plan right after the graduation is to pursue a postdoctoral position in PL and security research groups that help expand the scope of my research. Toward this end, the UCSD CSE postdoc fellowship will specifically enable me better explore the research connections between PL and security areas.

1 Primary Research

A TEE is a hardware mechanism that contains security critical code and data. It aims to provide more security guarantees in the presence of powerful attackers. However, TEEs aren’t panacea. Their use require architectural specific expertise that a high-level programmer may not possess. A big part of my research involved designing programming language features that abstract away the low-level details of TEEs and yet be able to use them correctly to get application-specific security guarantees. Moreover, the security guarantees are proven to be mathematically correct.

My PhD dissertation lays more emphasis on the underlying theory of building a secure system. The next natural step is to build practical secure systems applying the theory. Toward this end, I would like to primarily focus on the following potential projects during the fellowship.

Secure Composition of Unsafe Components
Software bugs in untrusted components (e.g., third party libraries) can compromise the security of an otherwise correct application. For example, a buffer overflow in one component could leak the cryptographic keys stored in another component. Such effects should be contained to minimize the adversarial impact. Building on my previous research [1], I am proposing sandboxing, protected memory container offering isolation, as a first step towards enforcing the principle of least privilege. This is not entirely a new idea and has been explored in the past. Deian and Sorin’s latest research on RLBox [2] also uses similar technique to contain the effects of adversarial components inside a web browser, but without any formal guarantees. The crucial insight in my proposal is that the software sandbox can be given a very expressive type that could lead to mathematically proven security guarantees. Also, if naively done, software sandboxing is an expensive operation stifling its adoption in real world systems. More research is necessary to make this a practical technique.

Rethinking Syscall Interface to Operating Systems
Given the recent rise of various isolation mechanisms—such as TEEs, virtual machines and SFIs— for enforcing confidentiality and (or) integrity in an application, it is time to rethink some of the system call interfaces to an operating system (OS) for exporting these guarantees to applications. Such an interface would abstract away the low-level details from the application enabling it to focus on application level security specifications. This is an ambitious project that can immensely benefit from the expertise of system and networking research groups.

1 Given the page limit, I am only outlining the ideas here. For more details, please refer to the accompanying research statement.
2 Mentor and Collaborators

The theme of my research—building principled and practical secure systems—aligns well with that of Deian Stefan. Specifically, our work uses type systems from programming languages to enforce information flow control properties—such as confidentiality and integrity—in the software systems. Often, a junior researcher like me is at the risk of getting lost in the details. In such cases, a mentor’s guidance is thus of utmost importance. I believe that Deian’s expertise will help steer my research towards meaningful goals.

Collaboration is important to any successful research. Given the inter-disciplinary nature of my research, it is even more crucial. The UCSD CSE post-doctoral fellowship will specifically enable me better explore the research connections between PL and other areas.

Besides Deian, my research also crosses paths with Nadia Polikarpova, Ranjit Jhala and Sorin Lerner. One of the ideas that I am interested in exploring (with Nadia and Ranjit) is the expressiveness of an authorization logic in which the access control policies are specified using liquid types. Such an ability will enable specifying interesting policies about running mobile code (e.g. a browser can only run typesafe Javascript libraries) More broadly, the programming systems group is a perfect fit for nurturing my research agenda.

Besides participating in their group seminars, my plan for improving collaborations is by organizing new inter-disciplinary seminars. At Harvard, I founded a student-run seminar across PL, systems and security research groups. The main aim of the seminar is to promote interaction and awareness of research happening in related groups. Students either presented their research or collectively read a paper. Given the diverse background, the discussions are quite informative. Hosting a seminar with similar goals, if not already happening, may lead to interesting interactions.

3 Other Goals

Besides conducting research, my three other goals include gaining experience in teaching, mentoring and authoring a research grant proposal: all are important for advancing my academic career in the future. I would like to co-teach the graduate security course (CSE 227) with a focus on information flow control. By coordinating with Nadia Polikarpova, I am also interested in co-teaching (a module) in the undergraduate programming language course. I am also looking forward to mentor graduate students in their research. This will be yet another excellent way of building my collaborative network.

References
