Deployable and Impactful Security
SaTC 2019 PI meeting breakout group report
Co-leads: Daphne Yao and Michelle Mazurek

1. Problem/Domain Summary

We argue that NSF (and the security community broadly) should aim to increase security research that **targets real-world problems** and **produces actionable insights and tools that can be used by practitioners**.

- Incumbent tool vendors are missing large parts of attack space (that research has addressed)
    - Things that researchers sometimes consider old/solved are still problems in the industry
- Without a clear path to disseminate artifacts (and to practitioners), it's hard to maximize broader impacts
- We can work on important problems that industry might not have the motivation to solve (e.g., more transparency in personal data privacy)
- Developers may not know how to apply even "well known" security techniques
- We create things for which the market doesn't yet exist, and just handing over research may not be enough. We don't have to worry first about monetization, we can build it first.
- Better connections with practitioners can help to identify important problems academics might not otherwise see
- We can emphasize aspects (e.g. usability) that practitioners can sometimes overlook
- More deployable solutions can contribute to reproducible science

A quick survey shows that out of the 24 session participants, 17 of them think there are tensions between research and practice. 12 participants experienced obstacles when trying to publish work that bridges the gap, and 2 expressed difficulties in securing funding. 10 participants think there is a lack of diversity in defining what good security work is. When asked "Is it a good idea for a junior faculty to start deployment work?", only 2 participants agree.

2. Key Research Challenges

- Researchers have limited insight into real-world practice
- It can be difficult to identify/recognize novelty and frame key research questions about practice that also contribute to scientific knowledge
- Practical application is limited not just by technical issues but by organizational and behavioral factors
    - In practice there are many use cases, scenarios, priorities academics may not be aware of
    - Practitioners may need to focus on specific requirements/checklists/compliance rather than looking at security overall

- Researchers and practitioners often talk past each other (and are not at the same events)
- Making code that is deployable (and maintained) is hard work, over a lot of time, and doesn't align well with academic incentives
    - If you don't want to deploy it yourself, how far is "far enough" in deployable that practitioners will be able to use it?
    - In order to transition to practice, tools must fit in with existing stacks or processes
    - Dependencies can disappear
    - Needs good programmers to work on deployable/maintainable programs -- where can we find them? How can we pay them competitively?
- It can be hard to disseminate research results to practitioners (different language, different events) and hard to know whether they have been seen, thought about, implemented, and if not why not (not seen, not understood, not considered practical, etc.)
- Security spans so many areas (databases, hardware, web, etc.). This requires a lot of interdisciplinary work to make tools that make sense across these areas, and making the work accessible to many different communities is a challenge as well
- Not enough security faculty with enough time/expertise for improving education

3. Potential Approaches

- Building partnership and community with practitioners; not just swooping in for one item but building long-term relationships
- Forging relationships with university IT departments as large users of security technology
- Sending students as interns; they bring back knowledge of practical operations and provide practitioners with insight into researcher skills, can build relationships ($ to support this?)
- Adding more practitioner content at big security conferences (industry/deployment track, relevant keynotes, organized networking/matchmaking between industry and academics, etc.)
- More workshops or other events that bring academics and practitioners together to discuss these issues and build relationships, maybe especially smaller events
- Requiring more artifact evaluation (perhaps with an incentive from NSF?)
- Using class projects, or guest speakers in classes, as a seed for forming practitioner relationships
    - Classroom students who become software engineers as bridges to industry
- Promoting deployable research top-down via NSF/industry partnerships
- Promote more research into how to make sure things remain usable over time
- Promote more research into existing standards and processes and how they are used in practice
- Promote work with policymakers/regulators to ensure that industry requirements, regulations, etc. reflect the current state of the art

- Developing best practices for conducting and reporting on deployable research
- More interdisciplinary research to maximize deployability into many communities and to groundwork in real problems. More collaboration to bring different kinds of expertise.
- Community resources for finding good programmers to support this work (e.g., summer of code, campus “programmers for hire” who can visit different research groups). More $ to support this kind of effort (from industry, NSF, other?)
- More education research into how to produce software engineers with more security awareness; how to bring in real practitioner knowledge here

4. Long-Term (> 10 years) Significance

- In the next 10 years, bridging the gap from research to implementation will only become more important (the gap may only grow)
- Making things that are deployable / maintainable is only getting harder
- Incentives and priorities will remain challenging / conflicted