Some Experiences in Developing Security Technology That Actually Gets Used

Sean Peisert

February 15, 2019

Why do research? Researchers do research for all kinds of reasons – because we want to learn more about the subject matter, because we like working with the other people who do research, or the places where research is done, or just like the research process. Or perhaps because we want to figure out what makes something work, or to solve a problem. Many reasons exist, each just as valid as any other. In my own work, a few years ago, I noticed that I was gradually shifting toward wanting to do something that other people used and found useful. At the same time, though I didn't really notice it until much later, successfully making that change was much harder than I thought it would be at the outset. "Just pick a practical problem that other people need solved, and solve it, right?" The ability to just pick a practical problem and charge in has definitely not been true, at least for me.

Solving problems One of the most important things that I learned is that most of the practical problems that I wanted to solve involved expertise beyond what I had myself, and indeed often from outside my own domain of computer science. For example, in my work developing solutions for cybersecurity for the power grid, I realized I could read all I wanted about SCADA and the grid, but not being a power engineer, I didn't really understand the grid itself and the way it operates in real-world, practical terms. Rather, I tended to just view it through the lens of a computer scientist (e.g., control devices) and ignore the other details, like the electrical part itself, even though that's what the grid is all about!

So, I went and found a power engineer to partner with. Even then, however, although finding someone with power engineering background helped to understand the data and system we were looking at much better, it didn't help much with how a solution might actually end up being used. For that, it was necessary to understand who the people are that are in charge of operational security for the power grid. That question is more easily asked than answered — most environments have "grid operators" who work in a grid operations center, and who look for electrical stability within the grid itself. Those people are very distinct from "security operations" teams, who work in a security operations center (SOC) and look for security issues. Indeed, as I found out, more often than not, those two teams

typically work, at best, in loose coordination with each other, but physically sit in different places, have very different sets of expertise, tools they use, and vocabularies they use to describe things. Indeed, it was not uncommon for the power experts and the security experts to use the same word to mean different things. To uncover such miscommunications, we had to talk with actual grid operators and actual grid operational security members.

Not unlike the principles of "building security in" in which one begins designing security from the outset, rather than tacking it on later, starting with the end user of the technology would of course have been a good idea to begin with. Starting with the user is the fundamental precept of "user-centered design" [3], and indeed, one could argue that question also heavily underlies the fourth central tenet of the immortal "Heilmeier Catechism" [1], namely, "Who cares?" Well, the people who need to use the the technology definitely care!

My experiences in working with the healthcare field in areas of cybersecurity parallel those of my experiences in working with the power field in many ways. For example, prior to following surgeons on their hospital rounds, I never would have guessed that the primary interface for the attending surgeons to their electronic health record (EHR) system was not a computer, but rather the surgeon's medical interns and residents. It was the job of the interns and residents to interface with the EHR and report that information back to the surgeon. This is a very important detail about how the system is used, which is in turn a very important detail about how the system need be secured. For example, just considering access control alone, it is not merely the attending surgeon that needs access to a record (if that person even needs access at all), but rather, anyone else who will be reporting back to that surgeon, which explodes the size of the access control rights being granted to the EHRs for every patient.

Of course, even with this "lesson learned" of starting with the end user to understand their usability constraints, it is not sufficient to simply ask the person what they need, but to develop a process to intuit the nature of the problem they face. As Henry Ford is said to have stated when asked what improvements to transportation people needed, he is said to have indicated would have been told "a faster horse" rather than car that became his own trade. The late Apple co-founder and CEO Steve Jobs is said to have made similar remarks about the nature of focus groups, and what kind of answers he would have gotten had he asked people what they wanted in a phone, circa the era of the Motorla Razr "flip phone" rather than the iPhone that he and his team at Apple eventually came up with. The reality is that people can't always foresee what would truly be more

useful.

People On the other hand, asking end users what *problem* they are trying to solve may well lead to a two-way conversation that results in a useful understanding of an answer. But even there, even uncovering the real problem may still be confounding. Speaking from my own experience, worst of all for the end-user in these situations is when the researcher comes to the end user with a hammer, searching for a nail. Most often it takes the end user ten seconds to realize that the researcher isn't really trying to solve the end user's problem, but is simply looking for a use case for their technique that they can test against and publish in a paper, and then never to be seen again. The point at which the end user realizes they are a research subject, or simply the means to a research end is often when they also check out of the conversation. In the future, the researcher's chances of changing the perception of the end user and redeeming themself in the eyes of the end user can be very hard to do.

A key item that readers of this letter may also take away with them is the notion that understanding the problem doesn't mean just understanding the technological constraints. Understanding human issues is also vital. Yet another domain in which I've both experienced and observed challenges in applying research to practice is the field of elections. While there has been a great deal of wonderful work in vulnerability analysis of election systems, I've seen very little security research that isn't focused on attacks against existing systems translate successfully into practice. Here again, I believe one of the key challenges is often a disconnect between researchers and end users — consider the mathematically brilliant end-to-end cryptographic voting schemes that not only ignore the way that most elections are defined (e.g., in the U.S. Federal government, mostly at the state and local level) but presume that a voter is willing to trust pointy-headed mathematicians that the encryption scheme is actually counting their vote correctly. Additional solutions don't seem to always take into account the average age and level of technical sophistication of a typical poll worker, or may drastically overestimate the amount of time that a given county's elections staff might have to work with researchers to the bitter end of a supremely secure solution — a mistake I myself have made.

Along similar lines, it is very important not only to understand the end user and the *problems* they wish to solve but to understand their personal *motivations* [4]. I was recently at an industry research lab and spoke with a researcher who was lamenting about the fact that he had developed a technique to reduce false positives in static source code analysis, but couldn't get the company's software test-

ing team to adopt the technique, even though it would reduce their workload. I asked the researcher, "Is it at all possible that the software testing team is compensated for the number of bugs they fix each day, or views the volume of bugs to fix as some kind of job security?" This was just one possible hypothesis of why the researcher was struggling to engate with the test team, but regardless of the answer gives an example of why I believe deeply understanding the needs and motivations of the end user is vital.

Returning to a point that I made earlier in this letter about cross-disciplinary collaborations, that point is also not as simple as finding a partner in another department who has common interest, available graduate students, and a need for a sponsored research project to work on. For example, in the early days of my power grid work, I recall my own computer science graduate students looking at a programmable logic controller (PLC) and saying something to the effect of, "Well, that's not a computer." And similarly, my colleague's electrical engineering graduate students, while deeply versed in power systems and signal processing, asked questions to the effect of, "What's packet monitoring?" or "What's signature-based intrusion detection?"

Cross-disciplinary partnerships are again a time in which academics must remember that focusing on solving the problem is the real goal, and not obtaining publications in whatever happens to be their favorite conference. One reason for this in cross-disciplinary partnerships is that each discipline will have different "ideal" publication venues, or even mediums. For example, conferences are often the premier peer-reviewed publication venues for computer scientists (certainly in security), whereas journals are the premier venues for many in electrical engineering. It may be possible to determine a venue to publish that satisfies the professional needs of both disciplines, or also may be possible to figure out creative ways to divide publications up for publication in both disciplines. But at the end of the day, researchers should remind themselves that going down the path of doing something useful means that it's ultimately the impact of doing the useful thing that counts, not another publication on one's CV.

One final point I think is worth mentioning is how often researchers approach meeting operational personnel with the idea that something is broken to begin with and that "broken" equals "bad" and can always be fixed by the "right" smart person. As computer scientists, it's easy for us to think in ones and zeros, but it's important for us to remember that not every problem can be solved immediately, and that doesn't mean that the existing solution is necessarily bad. Further, characterizing it as such, even accidentally, can

make operational personnel feel like researchers simply aren't in touch with operational realities.

Bringing it together Researchers not focused on seeing their work actually get used — which is not only fine, but is often not even on the radar of people doing "basic" research — can certainly ignore what I've written here. Other researchers well may have innate intuitive knowledge of producing useful technology and techniques. For the rest of us, I believe there are many lessons that can be learned that can help make the process of doing "useful" things smoother.

Everything I've described here is not rocket science, and in reading through it, I think most of it seems intuitively obvious, in hindsight, even though it frequently wasn't, in my experience, at the time. But it also requires a true desire to understand one's collaborators, the problems one wishes to solve, the people who are affected by those problems, and the people who are affected by potential solutions, and empathy to the needs of those individuals.

In the case of understanding end users, one of the things I've found that has significantly helped my ability to understand my role as a researcher is to actually live and breathe the life of wearing an "operational security" hat. That's not an experience many researchers have, unfortunately, although I find I keep hearing more and more about students who spend some time doing internships in a campus SOC, actually sitting and working with operational security personnel. For additional demonstration of the value of such an experience, I refer the reader to Sundramurthy, et al.'s anthropological work, which was a wonderful example of transformative security research applied to practice [5].

Students doing internships in private industry can obtain similar experience by working side by side with developers or operational personnel, and may even have a chance to be on the other side of a research pitch, this time receiving the request to apply their tool to some part of an organization. Once this happens, I found that I've forever changed the way I look to solve security problems. Never again (or, almost never, unless I momentarily revert to bad habits), do I approach a problem wildly waving my particular security hammer *du jour*.

Wearing my operational hat, I recall a conversation with a researcher building a solution that required full packet capture, and pointing out that all the equipment to do this would be provided by the project or the project's sponsor, and in fact all that would be needed was to mirror the traffic of a key router. The part that the researcher had not anticipated was the fact that the racks in the co-location space where the network hardware was stored were often

full, or didn't have available power supplies, or that optical fiber simply can't be tapped because it's connected directly on router interfaces, rather than going through switches, which have their own compatibility problems.

Thus, to counter all of this, researchers must approach situations in a way that seeks to *solve the problem*, while being agnostic to the actual solution. The reality is that this may not always be possible — researchers are typically experts in specific technologies, and so it's both natural and beneficial that a researcher would look for opportunities to apply a technology they have expertise in. At the same time, it can also be beneficial for researchers to gain knowledge about technologies from other domains that help solve specific problems. Doing so can teach them more about the problem being solved, and also more about the domain whose technology is best suited to solving the problem. That openness can open even more future collaborative possibilities, and better insights into what the "right" technology or combination of technologies to solve a problem are.

At the same time, I don't intend to suggest that the roadblock between researchers and developers is entirely at the feet of researchers to address — I'd also love for operational personnel and policymakers to gain familiarity with research pertaining to their field, that might help them better understand the longer-term possibilities that research in a particular field can provide. I think that there's reason to be optimistic that a lot of this may end up happening – Ph.D. computer science students are now frequently spending time working in the technology industry before or during the pursuit of their advanced degrees, from startups to large technology firms, and there is also a healthy flow of Ph.D. computer science students to industry jobs, as well. Another reason for optimism is that funding agencies are also now encouraging such opportunities — consider the NSF "Cybersecurity Innovation for Cyberinfrastructure (CICI)" program [2] that seeks secuirty research that benefits scientific computing infrastructure itself — often one of the easiest things for a student to get access to, because at least some scientific computing infrastructure is present at just about any research university. In any case, this greater degree of intermingling between researchers and companies building real systems can only help increase awareness about processes for making research more useful.

Finally, very fortunately, this magazine has researchers who span research, practitioner, and policy-making, and beyond, and it's my hope that this magazine's readership is and will continue to be on the forefront of producing and adopting useful computer security technologies. Indeed, I challenge this magazine's enlightened readership will try to apply the concepts they've read in this letter to their own

specific domains and problems. One thing we can't claim right now is that there is a shortage of security problems, and so with the right, user-centered approach, and forward-looking operational personnel, I have little doubt that great progress can be made.

References

[1] George H. Heilmeier. The Heilmeier Catechism. `https://www.darpa.mil/work-with-us/heilmeier-catechism`.

[2] National Science Foundation. Cybersecurity Innovation for Cyberinfrastructure (CICI). `https://nsf.gov/funding/pgm_summ.jsp?pims_id=505159`.

[3] Donald A. Norman. *User-Centered System Design: New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., 1986.

[4] Lavanya Ramakrishnan and Daniel Gunter. Ten Principles for Creating Usable Software for Science. In *Proceedings of the 2017 IEEE 13th International Conference on e-Science (e-Science)*, pages 210–218, Auckland, New Zealand, 2017.

[5] Sathya Chandran Sundaramurthy, John McHugh, Xinming Ou, Michael Wesch, Alexandru G. Bardas, and S. Raj Rajagopalan. Turning Contradictions into Innovations or: How We Learned to Stop Whining and Improve Security Operations. In *Proceedings of the Twelfth Symposium on Usable Privacy and Security (SOUPS)*, pages 237–251, Denver, CO, 2016. USENIX Association.