Sean Peisert
Associate Editor in Chief

Some Experiences in Developing Security Technology That Actually Get Used

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 because we just like the research process . . . or, perhaps, because we want to figure out what makes something work or 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, although 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? Just picking a practical problem and charging in has definitely not been easy, at least for me.

Solving Problems

One of the most important things I learned is that most of the practical problems I wanted to solve involved expertise beyond what I had and that they were often from outside my own domain of computer science. For example, in my work developing solutions for cybersecurity for the power grid, I realized that I could read all I wanted about supervisory control and data acquisition 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 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!

I went and found a power engineer to partner with. Even then, however, although finding someone with a power engineering background helped me 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 is 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 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. As I found out, more often than not, those two teams typically work, at best, in loose coordination with each other, but physically, they sit in different places and have very different sets of expertise, tools they use, and vocabularies to describe things. It is not uncommon for power experts and 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, one could argue, addresses the fourth central tenet of the immortal "Heilmeier catechism,"[1] namely, "Who cares?" Well, the people who need to use the technology definitely care!

Digital Object Identifier 10.1109/MSEC.2019.2899711
Date of publication: 2 April 2019

My experiences in the healthcare field in areas of cybersecurity parallel those of my experiences in the power field in many ways. For example, before 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 needs to be secured. For example, just considering access control alone, it is not merely the attending surgeon who 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 usability constraints, it is not sufficient to simply ask the people what they need; instead, a process must be developed to intuit the nature of the problem they face. When Henry Ford was asked what improvements to transportation people need, he is said to have replied, "If I had asked people what they wanted, they would have said faster horses." Steve Jobs, Apple's late cofounder and chief executive officer, 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, in the era of the Motorola 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. Even there, though, uncovering the real problem may still be confounding. Speaking from my own experience, the worst situation of all for the end user is when the researcher comes to the end user with a hammer, searching for a nail. Most often it takes the end user 10 seconds to realize that the researcher isn't really trying to solve the end user's problem; he or she is simply looking for a use case for a technique to be tested against and published in an article, with the researcher then never to be seen again. When end users realize that they are research subjects, or simply the means to a research end, they often check out of the conversation. In the future, a researcher's chances of changing the perception of the end users and redeeming himself/herself in the eyes of the end users can be very hard to do.

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. Although 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 that 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 levels) but presume that a voter is willing to believe mathematicians who tell them 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 they 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 users 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 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 testing 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 that they view the volume of bugs to fix as some kind of job security?" This was just one possible hypothesis for why the researcher was struggling to engage with the test team, but regardless of the answer, it gives an example of why I believe that deeply understanding the needs and motivations of the end user is vital.

The point I made previously about cross-disciplinary collaborations is 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 and saying something to the effect of, "Well, that's not a computer." And similarly, my colleague's electrical engineering graduate students, although 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?"

In cross-disciplinary partnerships, academics must remember that focusing on solving the problem is the real goal, not obtaining publication at 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 it may be possible to figure out creative ways to publish in both disciplines. At the end of the day, however, 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 for the curriculum vitae.

One final point 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 to remember that not every problem can be solved immediately, and that doesn't mean that the existing solution is necessarily bad. Furthermore, 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 who are not focused on seeing their work actually get used—which is fine and is often not even on the radar of people doing basic research—can certainly ignore what I've written here. Other researchers may well 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 to 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. It 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 to significantly help 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 Sundaramurthy et al.'s anthropological work,[5] which was a wonderful example of transformative security research applied to practice.

Students doing internships in private industry can obtain similar experience by working side by side with developers or operational personnel, and they 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 happened, it 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 by wildly waving my particular security hammer du jour.

Wearing my operational hat, I recall having a conversation with a researcher who was building a solution that required full packet capture and pointed out that all of the necessary equipment would be provided by the project or the project's sponsor, so 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 colocation space where the network hardware was stored were often full or didn't have available power supplies or that optical fiber can't simply 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, so it's both natural and beneficial that researchers 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 lead to even more future collaborative possibilities and better insights into what the "right" technology or combination of technologies to solve a problem are.

I don't intend to suggest that the roadblock between researchers and developers is entirely at the feet of researchers—I'd also love for operational personnel and policy makers to gain familiarity with research pertaining to their field, which might help them better understand the longer-term possibilities that research in a particular field can provide. I think 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 start-ups to large technology firms, and there is also a healthy flow of Ph.D. computer science students to industry jobs. Another reason for optimism is that funding agencies are also now encouraging such opportunities—consider the National Science Foundation's Cybersecurity Innovation for Cyberinfrastructure program,[2] which seeks security 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, *IEEE Security & Privacy* has researchers who span research, practice, policy making, and beyond, and it's my hope that this magazine's readers will continue to be at the forefront of producing and adopting useful computer security technologies. Indeed, I challenge this magazine's enlightened readership to apply the concepts they've read in this article 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. DARPA, "The Heilmeier catechism." [Online]. Available: https://www.darpa.mil/work-with-us/heilmeier-catechism
2. National Science Foundation. "Cybersecurity innovation for cyberinfrastructure (CICI)." [Online]. Available: https://nsf.gov/funding/pgm_summ.jsp?pims_id=505159
3. D. A. Norman, *User-Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: L. Erlbaum Associates Inc., 1986.
4. L. Ramakrishnan and D. Gunter. "Ten principles for creating usable software for science," in *Proc. 2017 IEEE 13th Int. Conf. e-Science*, pp. 210–218.
5. S. C. Sundaramurthy, J. McHugh, X. Ou, M. Wesch, A. G. Bardas, and S. Raj Rajagopalan, "Turning contradictions into innovations or: How we learned to stop whining and improve security operations," in *Proc. 12th Symp. Usable Privacy and Security (SOUPS)*, 2016, pp. 237–251.