CHAPTER

29

Beyond “Computer Says No”

At the start of this century, security technology was an archipelago of mutually suspicious islands – the cryptologists, the operating system protection people, the burglar alarm industry, right through to the chemists who did banknote inks. We all thought the world ended at our shore. By 2010, security engineering was an established and growing discipline; the islands were being joined up by bridges as practitioners realised we had to look beyond our comfort zones. The banknote ink chemist who didn’t want to understand digital watermarks, and the cryptologist who could only talk about confidentiality, were steadily marginalised.

Now, in 2020, everyone needs to have a systems perspective in order to design components that can be integrated usefully into real products and services. And as these are used by real people, and often at global scale, our field is embracing the humanities and social sciences too.

Security engineering is about ensuring that systems are predictably dependable in the face of all sorts of malice, from bombers to botnets. And as attacks shift from the hard technology to the people who use it, systems must also be resilient to error, mischance and even coercion. So a realistic understanding of people – staff, customers, users and bystanders – is essential; human, institutional and economic factors are as important as technical ones. The ways in which real systems provide dependability are becoming ever more diverse, and protection goals are not just closer to the application, they can be subtle and complex. Conflicts between goals are common: where one principal wants accountability and another wants deniability, it’s hard to please them both.

Starting in 2001, we began to realise that many persistent security failures are incentive failures at heart; if Alice guards a system while Bob pays the cost of failure, you can expect trouble. This led to the growth of security economics, which the first edition of this book helped to catalyse. The second edition in 2008 documented how failures were also increasingly about usability, and the decade after that saw a lot of research into security psychology.

So what next? By way of a conclusion to this book, I'd like to highlight three things.

First, complexity. Computer science has spent seventy years devising an impressive array of tools to manage technical complexity, but we're now coming up hard against social complexity. We can program cars to drive themselves fairly well on the freeway or in the desert, but we can't cope with cluttered city streets with all those unpredictable people. We can encrypt messages or strip people's names from databases but we can't stop social structure showing through. And bullying people has its limits; "computer says no" is a fast way to lose customers. It's not enough to study how a computer system can interact with a human; we need to figure out how it can work with many interacting humans.

Second, sustainability. As we put software in everything and connect everything online, we have to patch the software and maintain the servers. With durable goods like cars, pacemakers and electricity substations, we may have to maintain software for twenty or even forty years. We have no real idea how to do that, and if we don't crack it then our automation will be bad news for our planet's future. So-called 'smart' devices are often just things that have to be thrown away sooner, when "computer says no".

Third, politics. Security is not a scalar, but a relationship. It's not some kind of magic fairy dust you sprinkle on systems, but about how these systems exercise power. Who loses and who gains when "computer says no"? Does the social-network user get privacy, or does the advertiser get access? How is it used to turn money into political power? And if people want public goods such as a dependable Internet or a low rate of cybercrime, how can these be provided in a global world?

The stability of cybercrime over a decade in which the technology has changed completely suggests that it's not fundamentally about technology. The persistence of tech monopolies raises other questions about how tech and society can co-evolve, and about the nature of power. When Facebook becomes the arbiter of political speech, when Apple and Google can dictate policy on coronavirus contact tracing, and when Amazon, Microsoft and Google dictate policy on facial recognition (outside China), then I suspect that technology people should start reading up on political science, as well as on economics and psychology. The most intractable problems of the next ten years may be around governance.

Just as individuals can learn through experience, so our societies learn and adapt too. Democracy is the key mechanism for that. So a crucial way in which engineers can contribute is by taking part in the policy debate. The more we engage in the problems that technology poses around complexity, sustainability and the nature of power, the faster our societies will adapt to deal with them.