Preface to the Third Edition

The first edition of *Security Engineering* was published in 2001 and the second in 2008. Since then there have been several big changes.

The most obvious is that the smartphone has displaced the PC and laptop. Most of the world's population now walk around with a computer that's also a phone, a camera and a satnav; and the apps that run on these magic devices have displaced many of the things we were building ten years ago. Taxi rides are now charged by ride hailing apps rather than by taxi meters. Banking has largely gone online, with phones starting to displace credit cards. Energy saving is no longer about your meter talking to your heating system but about both talking to your phone. Social networking has taken over many people's lives, driving everything from advertising to politics.

A related but less visible change is the move to large centralised server farms. Sensitive data have moved from servers in schools, doctors' offices and law firms to cloud service providers. Many people no longer do their writing on word processing software on their laptop but on Google Docs or Office365 (I'm writing this on Overleaf). This has consequences. Security breaches can happen at a scale no-one would have imagined twenty years ago. Compromises of tens of millions of passwords, or credit cards, have become almost routine. And in 2013, we discovered that fifteen years' worth of UK hospital medical records had been sold to 1200 organisations worldwide without the consent of the patients (who were still identifable via their postcodes and dates of birth).

The biggest game-changer of the last decade was probably the Snowden revelations, also in 2013, when over 50,000 Top Secret documents about the NSA's signals intelligence activities were leaked to the press. The scale and intrusiveness of government surveillance surprised even cynical security engineers. This brings us to the third big change, which is a much better understanding of security threats. In addition to understanding the capabilities and priorities of western intelligence agencies, we have a reasonably good idea of what the Chinese, the Russians and even the Syrians get up to.

And where the money is, the crooks follow too. The last decade has also seen the emergence of a cyber-crime ecosystem, with malware writers providing the tools to subvert millions of machines, many of which are used as criminal infrastructure while others are subverted in various ways into defrauding their users. We have a team at Cambridge that studies this, and so do dozens of

other researcher groups worldwide. The rise of cybercrime is changing policing, and other state activity too: cryptocurrencies are not just making it easier to write ransomware, but undermining financial regulation. And then there are individual threats such as cyber-bullying, which usually fall below the threshold for criminal prosecution but which cause real distress, are made easier by social networks, and happen at such a scale as to matter.

So online harms now engage all sorts of people from banks and the military down to schoolteachers. It is ever more important to measure the costs of these harms, and the effectiveness of the measures we deploy to mitigate them.

Some of the changes would have really surprised someone who read my book ten years ago and then spent a decade in solitary confinement. For example, the multilevel security industry is moribund, despite being the beneficiary of billions of dollars of US government funding over forty years; the Pentagon's entire information security philosophy – of mandating architectures to stop information flowing downward from Top Secret to Secret to Confidential to Unclassified – has been abandoned as unworkable. While architecture still matters, the emphasis has shifted to ecosystems. Given that bugs are ubiquitous and exploits inevitable, we had better be good at detecting exploits, fixing bugs and recovering from attacks. The game is no longer trusted systems but coordinated disclosure, DevSecOps and resilience.

What might the future hold? A likely game-changer is that as we put software into safety-critical systems like cars and medical devices, and connect them to the Internet, safety and security engineering are converging. This is leading to real strains; while security engineers fix bugs quickly, safety engineers like to test systems rigorously against standards that change slowly if at all. A wicked problem is how we will patch durable goods. At present, you might get security patches for your phone for three years and your laptop for five; you're expected to buy a new one after that. But cars last for fifteen years on average and if we're suddenly asked to scrap them after five the environmental costs won't be acceptable. So tell me, if you're writing navigation software today for a car that will launch in 2022, what toolchain will you choose to ensure that you'll be able to keep on shipping security patches in 2032, 2042 and 2052?

Finally, there has been a sea change in the political environment. After decades in which political leaders considered technology policy to be for anoraks, and generally took the line of least resistance, the reports of Russian interference in the Brexit referendum and the Trump election really got their attention. The prospect of losing your job can concentrate the mind wonderfully. The close attention of lawmakers is changing the game, first with tighter rules (such as Europe's General Data Protection Regulation) and second as software and online connectivity find their way into products that are already regulated for safety, from cars and railway signals to children's toys.

The questions the security engineer has to ask today are just the same as a decade ago: what are we seeking to prevent, and will the proposed mechanisms actually work? However, the canvas on which we work is now much broader. Almost all human life is there.