Context Matters: designing security for contexts of use

Shamal Faily

At 10.24am on 3rd July 1988, Iran Air Flight 655 was shot down over the Straits of Hormuz by the American guided missile cruiser USS Vincennes. The direct cost of this event amounted to the death of all 290 passengers on board. The indirect costs are incalculable and still felt to this day, as the US and Iranian governments continue to blame each other for the incident. The accident report blamed "human error", but this has been contested. After all, how could such an advanced system, with the capability to track 200 missiles simultaneously, fail to recognise the difference between an F-14 and an Airbus A300?

The USS Vincennes was designed for conventional warfare in the deep ocean, an environment where the presence of civilian ships and aircraft was inconceivable. Consequently, hails sent from the Vincennes to Flight 655 were sent over military and civilian air-distress frequencies, rather than the civilian air-traffic frequencies commonly used. We will never know if the flight crew of Flight 655 received any of these hails, as its black box flight recorder was never found. We can, however, glean information about the Vincennes leading up to 10.24am. We know that the CIC (Combat Information Centre) was a scene of chaos shortly before the captain took the decision to open fire. The ship was engaged in skirmishes with Iranian Republican Guard gunboats when the unknown aircraft appeared on radar. Experts believe that despite the conflicting evidence, such as the civilian aircraft transponder code, and the availability of civilian air-traffic schedules, the crew was conditioned to expect an attack from an enemy aircraft. The Vincennes is more than a piece of technology, it is a socio-technical system: a system where both people and technology are equal partners. Rather than blaming the destruction of Flight 655 on the frailty of humans, we should instead accept that humans were part of a system, which was not designed for the context of operation it found itself in.

Socio-technical systems are big. They encompass a menagerie of technology, people, cultures, and are often designed to meet many different goals,

some of which conflict. In some cases, such systems do not even have a central point of control. Unfortunately, methods often employed for building such systems assume a single-organisation context, homogenous technology, and organisational policies which assume that best practice of the resulting system will be followed. The consequential impact of such processes is evident not only from the case of Flight 655, but also from reports on security problems 20 years later. Stories about security failures, such as the loss of personal data of 25 million people by HM Revenues and Customs, are common enough that the media no longer considers them news-worthy. Again, like the Vincennes, we find ourselves asking why we can put a spacecraft around a different planet, but remain unable to secure personal data, or avoid a deluge of spam mail.

One reason we continue to build poorly designed system is our propensity to look for technical solutions to what we think are technical problems. Technologists tell us that installing a "fire-wall" will stop hackers breaking into our computers, and the use of "digital signatures" will prevent electronic eavesdroppers from monitoring emails we send and receive. Following this solution driven approach has two problems. First, most people don't understand the technology they are required to use, nor do they find it particularly easy to use. Such technology is not designed from the perspective of the end-user, but from the designer building it. Such designers assume that security is the primary goal. In reality, the primary goal of a user is the purpose he or she uses a system for. As such, security should be designed as something which is ambient, rather than an artefact at the fore-front of one's mind. The consequence of not doing this becomes obvious when people find ways to circumvent security which gets in the way of their work. Second, security writers, such as Bruce Schneier, assert that the security of a system is only as good as its weakest link. Moreover, the weakest link is usually not the technology, but the person using it. If an attacker wanted to obtain some information from you, why go to the effort of trying to break the encryption of an email's digital signature, when he can simply use some social-engineering to persuade you to give him what he wants.

It is not enough to design systems assuming the presence of human agency, we must appeal to the norms and values of the human communities such systems serve. Usability researchers have carried out considerable work in this area, yielding a number of socio-technical design methodologies; these prescribe approaches for eliciting and managing human values during system design. Methods have also been designed for use in secure systems design, and provide support to the process of identifying concepts like assets, threats, vulnerabilities and risks, such that any resulting countermeasures can be situated for a system's context of use.

Despite their suitability for security design, socio-technical design approaches are rarely used in practice. The first barrier of use is that many system designers do not realise they exist. Most developers do not attend usability conferences, nor do they read the scholarly journals where the principles of such approaches are described. If we assume that these techniques are widely advertised, then designers face a second barrier: applying these approaches in practice. Many approaches assume that plenty of time is available to cull requirements for different stake-holders. In reality, the people with the biggest stake in a system are often the busiest, and have neither the time nor the interest in providing needed information. The approaches also take a myopic view of design. Design to one set of people may be the preparation of a requirements document, to others it may be the high-level architectural blue-print of a system, or a model illustrating a discrete software component. In most cases, the approaches focus on a particular aspect of 'design', and consider early or later stages of the design process as relatively trivial engineering tasks.

Security design also has intrinsic problems of its own. When we design a system, we usually have some purpose in mind. When we design security, we must appreciate aspects of the system which make it liable to attack, by considering notions such as preventing, deterring, detecting, and reacting. We can not simply decompose a system down into its component parts and analyse each component independently, we need to consider the system holistically, as any object of attack. We need to think about possible attackers and their motives, and understand enough about the context of operation to reason about how changes to it can make the system more or less secure.

We do not have any answers for how best to design security for socio-technical systems. There is, however, a growing number of researchers with an interest in tackling the problem of designing usable security. Recent work in this area has looked at the culture of security, and what sort of concepts strengthen or weaken an organisational culture's propensity to act in a secure manner. This work has led to a better understanding of the cultural context underpinning contexts of operation, and highlighted the particular importance of modelling the roles and responsibilities held by different users within a system. Other recent work includes investigating how well best-practice for modelling contexts of use can be integrated with best-practice from risk analysis and management. The work has not only generated prescriptive guidance a developer can follow, but also software to support use of the process.

We now see that technology alone will not make security more situated for its contexts of use, neither will design processes which promote human usability above all else. Security design needs to be inclusive: we need input

from many different communities such as theoretical computer science, social-science, cognitive psychology, social anthropology, and economics. Academic fora, such as the New Security Paradigms Workshop, exist to identify inter-disciplinary research with the potential to cause a paradigm shift in security design. Collaboration between the disciplines will not be easy. Early work on usable security was built on the the premise that 'users are not the enemy'. Some usability researchers go further, and specifically attribute bad usability to the short-sightedness of software developers. On the other hand, computer scientists and software engineers argue for formalism and engineering principles leading to the engineering of systems according to many different design priorities. Therefore, the first hurdles to be surmounted will be cultural. When researchers from different disciplines learn to work together, we will glean the insights necessary to engineer secure socio-technical systems. Until then, expect to see more examples of "human error".