Appropriation and Principled Security

Steve Dodier-Lazaro
UCL Computer Science, United Kingdom
s.dodier-lazaro@cs.ucl.ac.uk

Abstract. Secure systems have a reputation of being unusable and demanding on users, a situation attributed to a lack of usability and human factors expertise among security experts. We argue that the issue of unusable security might have deeper roots. Indeed, the design principles security relies on are out of touch with the reality of nowadays' computing practices. In particular, the security principles of least privilege and fail-safe defaults strip human users of their ability to reconfigure systems and leave them stranded when facing interaction breakdowns. Security principles therefore prevent the re-appropriation of systems they mediate both in unexpected practices and by unexpected users. We propose several leads to lessen the negative impact of those principles on secure systems.

Introduction

Computer security lays its foundations on Saltzer and Schroeder (1975)'s ten empirical security design principles. Those principles have been instrumental in enabling security practitioners to produce secure systems. Security design principles seek to constrain the space of possible designs and guide designers towards the least insecure design. However, traditional security mechanisms have been designed with little knowledge of human factors, and have a reputation of treating their users as 'enemies' (Adams and Sasse, 1999). Usable security is a research field that draws heavily from practical HCI. The role of usable security with regard to its parent discipline is to evaluate and improve the usability of security mechanisms.

We will focus our discussion on two principles[1]. The principle of *fail-safe de-*

[1] Owing to space constraints. However, the least-common mechanism could also be examined under the light of universal access in support of appropriation by unanticipated users.

faults (PFSD) "means that the default situation is lack of access, and the protection scheme identifies conditions under which access is permitted". Security design and policy authoring are error-prone activities, and preventing access to resources in scenarios which have not been thought through is an effective method of avoiding situations detrimental to the system's security: "In a large system some objects will be inadequately considered, so a default of lack of permission is safer". The principle of *least privilege* (PLP), similarly, states that "every program and every user of the system should operate using the least set of privileges necessary to complete the job". Whilst the previous principle applies when the security consequences of a specific configuration are unknown, the PLP is directly informed by a security requirements gathering process and encourages designers to deny access to resources mediated by the security mechanism as often as possible.

In ethnomethodology-inspired HCI, it is held that computer systems are *unable* to fully capture, model and react to user contexts (Dourish, 2004; Chalmers, 2004; Suchman, 2006). Dourish (2004) proposes that systems should expose their structure to users in ways intelligible to them, so that users can align them with the settings of their activity at hand. Reconfiguration can be seen as the process of making technology fit a practice, and is described by Suchman (2006) as one of the constituents of human agency with regard to machines.

We observe that systems cannot both be reconfigurable and implement the security principles as they are laid out. Not only is reconfiguration out of reach but the principles also, we argue, strip human actors of their agency by constraining their ability to control their interaction with systems. Since reconfiguration affords appropriation, it is the very core of security design practice that must be revisited by HCI researchers, rather than the mere interface of existing security artifacts.

In the remainder of this paper, we develop on the reconfiguration-security contradiction presented above, and we then propose courses of actions for security designers to lessen the negative effects of the Saltzer and Schroeder principles.

Why Security Principles Compete with Appropriation

Chalmers (2004) provides a pragmatic overview of the role of designers with regard to constraining the possible uses of designed artifacts. On the one hand, designers constrain the types of couplings afforded to users (and thus the types of creative associations made between the artifact's capabilities and the user's problem at hand). On the other hand, they must explicitly support the interpretation and appropriation of tools, so that transparent use can be achieved:

> "A system [...] necessarily involves underspecification of the situation of its use, and therefore openness to interpretation and variability of its normative effect. [...] People accommodate the characteristic affordances of a new tool, but they may also appropriate it to suit the practices and priorities of their own contexts and communities of use i.e. other, older tools and media, and other people."

In contrast, PFSD explicitly states that *all* unanticipated uses must be prevented, and PLP that only the resources necessary for performing the anticipated tasks

should be accessible. The ability to interpret and reconfigure the system is explicitly removed. Those two security principles require a specification of what activities are legitimate for each authorised user, and a model of the applicable threats. The PFSD constrains the use of the system to anticipated use scenarios, and the PLP constrains the set of permitted actions on the system based on the threats at hand.

The externalities of the PFSD were deemed acceptable on the basis that users will be able to detect and remedy denied accesses: "A design or implementation mistake in a mechanism that gives explicit permission tends to fail by refusing permission, a safe situation, since it will be quickly detected". This reasoning has come under question in the past decade. Users, say Adams and Sasse (1999), have a legitimate right not to be pestered and hindered by computer security mechanisms that failed to account for their needs. Besides, they are not always able to remedy denied accesses without detriment to their productivity (Bartsch and Sasse, 2013). In some systems, only the initial configuration errs on the side of safety, and users are free to reconfigure the system later on according to their needs. This interpretation of the PFSD is one that we encourage and wish to make more explicit.

Re-Agencing Users in Principled Security

We propose to give opportunities for human actors to sustain interactions that would otherwise be prevented by the Saltzer and Schroeder principles. The rationale for stripping actors of their agency in the first place is that users should not be trusted to make correct security decisions. Whilst we disagree with this assessment, a similar case can be made that requiring users to think security decisions through consumes too much of their time (Beautement et al., 2008) and is an unsustainable form of interaction. Let us use a radical metaphor: Suchman (2006) stated that agency in human-like machines is achieved via a "collaborative performance", that involves human actors who control the interaction arena of the machine:

> "An encounter with Stelarc's Head affords further evidence for the collective and contingent nature of sociomaterial agencies. [...] I am struck by the sense of collaborative performance involved, both within and beyond the gallery walls."

If we were to return agency to the objectified users of security, could we provide them with collaborating agents who can reconfigure the arena to their benefit?

Focus on User Practices before Technique

We security designers must keep in mind that the shortcomings discussed in this paper arise from a lack of consideration of usability *at the stages of requirements gathering and ideation*, and that usability research is not just a field of user-centered evaluations, but also of design principles and processes. Ongoing research from colleagues Becker et al. (2015) hints that even organisations that proclaim to develop usable security products are unaware of usability design processes.

We ought to rewrite security design principles to emphasize the goal of reconfigurability. We could rewrite the PLP to add an obligation for researchers to validate that the most common and desirable practices at the time of design can be sustainably performed within the proposed design. Robust methods to this end exist: participatory design, value-sensitive design, and in-the-wild observations (including critical incident reports, digital data collection, ethnographies) which can unveil unanticipated practices and better document actual user populations.

Empower Users to Mediate Security Policies

Systems accurately configured to support the activities ongoing at the time of deployment will not necessarily *remain* appropriate. Unexpected needs and accompanying practices will emerge. Users must thus possess efficient mechanisms to inform the system – or its administrators – of such evolutions. In a series of interviews with employees of a large company, Bartsch and Sasse (2013) described how ineffective security policy review mechanisms can hinder workers' productivity and force them to evade security mechanisms. Policy reviews are indeed best performed on a local scale. As illustrated by Suchman (1995), it takes understanding and experiencing the work settings of an individual to explain their – sometimes seemingly illogical – work practices. Systems where users are empowered to make security decisions can sometimes be beneficial: Mazurek et al. (2011) showed how users were best able to decide whether to grant access to a resource when the requesting party provided a situated justification for their request; Kirlappos et al. (2014) found that employees have some motivation to engage in security-related behaviour, even when they must give up on employers' inflexible security mechanisms.

In other words, we must explicitly design both the security systems and their sites of interaction to permit reconfiguration. We must think reconfiguration opportunities through, and understand and compensate for their security consequences. For instance, a system that grants users access to a resource in a semi-public context could allow them to delegate their access to a third party, endorsing the third party's actions (similarly to a guarantor scheme) to provide both flexible access and assurance of accountability to the resource owner and the delegating user. In general, the ability to amend or relax permission is beneficial to productivity and appropriability goals, and should be provided when possible.

Even if policy mediation is to be decentralised and if users are to ultimately set policy collectively, security experts have an important role to play in addressing the mental models of non-expert users. Experts can provide high-level overviews of what practices are beneficial or risky, and can provide arbitration. The key to scalable permission management is to spare the expensive expert resource and make full use of the knowledgeable local one.

Isolate and Make Dangerous Interactions Explicit

In the consumer world, fail-safe defaults and least privilege have been more radically applied in recent times. For instance, users of modern tablet and mobile

OSs are prevented from installing software from unofficial sources. Some email providers prevent users from sending attachments deemed potentially dangerous. These decisions are made to prevent real malware threats. Nevertheless, users may need to send files which appear to be malicious but are not, or may need to use in-house software on their personal devices that cannot be uploaded to an app store.

Partial technical solutions exist. For instance, untrusted applications could be installed in confined environments, and users be required to explicitly give permission for each resource applications attempt to access. Why is that an improvement over blunt denial? Currently, users prevented from using an application or document can only drop out from using their device's security model and switch to a competing device, or they must reformulate or abandon their original goal. They cannot resolve the interaction breakdown caused by the security mechanism, no matter whether they are taking an actual risk or the mechanism is mistaken. Instead, they could be offered explicit trade-offs: they can accept the drawbacks of the confinement mechanism (performance issues and the enforcement of present-at-hand interaction) rather that disengage from security or fail to fulfil their expectations.

Since security mechanisms should be designed to support the assumed legitimate activities of users, such tools should be needed rarely enough that users continue to rationalise their use and don't fall prey to habituation. Besides, present-at-hand use of technology is an integral part of users' learning process, and so designers could explicitly inform users about the cues they should be looking for, whilst interacting with suspicious artifacts, in order to determine their legitimacy. If habituation occurs and interaction with isolating tools becomes ready-to-hand, users must by then have been made able to not take unwarranted risks. Such situations cannot be achieved accidentally; rather, designers will be required to explicitly consider how users interact and co-evolve with systems, and whether the ways in which systems are appropriated ultimately lead to insecurity. Methods such as participatory design and in-the-wild deployment studies can provide such evidence.

Discussion

We observe that systems cannot both be reconfigurable and implement the security principles as they are laid out. Since reconfiguration affords appropriation, we argue that at least some of the usability issues of security mechanisms originates in security design principles. We call this the *reconfiguration-security contradiction*.

We suspect that the very core of security design practice must be revisited by HCI researchers, rather than only the artifacts it has already produced. Both modern investigation and design methods must be appropriated by security designers to tackle such challenges, since mingling with the interfaces of mechanisms that are likely to contain uncovered flaws in the interactions they afford may not suffice.

The reconfiguration-security contradiction is particularly noticeable when security mechanisms cause breakdowns to which users respond through disengagement (Dourish et al., 2004; Bartsch and Sasse, 2013). We argue that the principles at the core of security design encourage such responses by preventing reconfigura-

bility, and therefore other forms of breakdown solving. We propose designers to explicitly identify potential breakdowns and design for their solvability by affording human users a richer range of actions. For instance, users can be offerred the ability to reconfigure systems, or allowed to explicit perform dangerous interactions – in which case designers should provide all the guidance and assurance they can.

Security principles have taken away the agency of human actors by forbidding reconfiguration. It is no longer sufficient to state that users must not be treated as enemies by security designers. Going further, they must not be *treated as objects*. We must transit from a culture of experimentation on subjects to a culture of design with agents. Such a shift in attitude might help us find a path towards building security interactions that are meaningful, comforting and genuinely useful.

References

Adams, A. and M. A. Sasse (1999): 'Users are not the Enemy'. *Communications of the ACM*, vol. 42, no. 12, pp. 40–46.

Bartsch, S. and M. A. Sasse (2013): 'How Users Bypass Access Control and Why: The Impact of Authorization Problems on Individuals and the Organization'. In: *Proceedings of the 21st European Conference on Information Systems (ECIS 2013)*. Berkeley Electronic Press.

Beautement, A., M. A. Sasse, and M. Wonham (2008): 'The Compliance Budget: Managing Security Behaviour in Organisations'. In: *Proceedings of the 2008 Workshop on New Security Paradigms*. New York, NY, USA, pp. 47–58, ACM.

Becker, I., S. Parkin, and M. A. Sasse (2015): 'Applying Sentiment Analysis to Identify Different Conceptions of Security and Usability'. In: *Under Review*.

Chalmers, M. (2004): 'A Historical View of Context'. *Computer Supported Cooperative Work*, vol. 13, no. 3-4, pp. 223–247.

Dourish, P. (2004): 'What we talk about when we talk about context'. *Personal Ubiquitous Comput.*, vol. 8, no. 1, pp. 19–30.

Dourish, P., E. Grinter, J. Delgado de la Flor, and M. Joseph (2004): 'Security in the wild: user strategies for managing security as an everyday, practical problem'. *Personal Ubiquitous Computing*, vol. 8, no. 6, pp. 391–401.

Kirlappos, I., S. Parkin, and M. Sasse (2014): 'Learning from Shadow Security: Why understanding non-compliance provides the basis for effective security'. In: *Workshop on Usable Security*. San Diego, California, The Internet Society (ISOC).

Mazurek, M. L., P. F. Klemperer, R. Shay, H. Takabi, L. Bauer, and L. F. Cranor (2011): 'Exploring Reactive Access Control'. In: *Proceedings of the 2011 annual conference on Human factors in computing systems*. New York, NY, USA, pp. 2085–2094, ACM.

Saltzer, J. and M. Schroeder (1975): 'The Protection of Information in Computer Systems'. *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308.

Suchman, L. (1995): 'Making Work Visible'. *Communications of the ACM*, vol. 38, no. 9, pp. 56–ff.

Suchman, L. (2006): *Human-Machine Reconfigurations: Plans and Situated Actions*. New York, NY, USA: Cambridge University Press.