Shaping our mental model of security

Shaping our Mental Model of Security

Saša Radomirović

School of Mathematical and Computer Sciences
Heriot–Watt University

Abstract. The IT industry's need to distinguish new products with new looks, new experiences, and new user interface designs is bad for cybersecurity. It robs users of the chance to transfer previously acquired security-relevant knowledge to new products and leaves them with a poor mental model of security.
Starting from a comparison with physical safety, we explore and sketch a method to help users develop a useful mental model of security in cybersystems. A beneficial side-effect of our methodology is that it makes precise what security requirements the user expects the system to fulfill. This can be used to formally verify the system's compliance with the user's expectation.

1 Introduction

The safety of consumer products has tremendously improved over the course of the last century. In some industries, notoriously the automotive industry, safety improvements have been achieved in spite of a significant push-back by the industry and only after many hard-fought legal battles [8, 6]. Unfortunately, now that consumer products are increasingly being connected to the Internet we may be about to regress on safety. We have never really been secure in cyberspace, but we were physically safe from cyber attackers until the IT industry began to connect everything and the kitchen sink to the Internet.

In the IT industry security issues are still largely the customer's problem. Only the largest vendors provide automatic security patches for their products and only for a limited time. When the vendor ceases to provide security updates, the customer must buy new products or risk security breaches due to unpatched vulnerabilities. New products and services attempt to distinguish themselves with new looks, new user interface designs, and new functions. This leaves some users disoriented and thus vulnerable to attacks.

Even though user interface design is a well-researched area and user experience is directly related to a product's success, some people still struggle to comprehend computing technology, and fail to interact correctly with it, simply because there is always a learning curve to a new technology. Some of those that do comprehend technology now will eventually lose touch with its latest trends. When a new technology goes on to marginalise and eliminate previously established technologies, those unfamiliar with it will be forced to learn to use it. This problem is aggravated by the fact that different mobile apps, web apps,

applications, and all the Internet of Things devices making their way into people's homes use different design patterns, different terminology, and different interfaces. It further steepens the learning curve and increases user errors due to confusion and misunderstanding, which may be exploited by an attacker.

Therefore, in order to design systems that are better at keeping users and the Internet at large secure, we must ensure that independently designed systems represent security-critical interfaces in a unified manner. In this paper we explore the use of security signs to communicate security-relevant information and instructions. The purpose of this paper is not to discuss what the best graphical or auditory representation for security signs and signals is, but to discuss what types of instruction or information must be conveyed. We shall therefore refrain from suggesting any shape, color, symbol, or sound for any sign or signal.

2 Users' Mental Model of Security

A mental model is a cognitive representation of external reality. It is a functional, simplified representation of reality, a working model [7]. We form our mental models through trial and error. To improve our mental model we need interactive feedback. Wash and Rader [9] observe that information security provides very little direct feedback to users. This makes mental models for information security difficult as the positive or negative consequences of security-critical decisions may not manifest themselves in time to be associated with the decisions made. It follows that (design) changes are bad for people's mental model of security.

As long as our rapidly evolving hardware and software systems keep on changing the location, terminology or graphical representation of security-critical settings and notifications, we can expect that users' mental models of security-critical functionalities will be poor. To guide users through security-critical processes and decisions we must therefore either always make the costly assumption that the user's working model of the system is very poor, or support users in creating better mental models by keeping the presentation and functionality of security-critical elements of systems the same. Clearly, the second option is preferable, but it requires standardisation.

Standardised signs and signals are a crucial tool to reduce the risk of accidents in safety-critical systems. For example, we all rely on "green for go" and "red for stop" on the road. The standardisation of this choice of colors keeps those of us who can distinguish these colors safe even in places we haven't been to before. "Green for go" is not innate to us, but we are able to transfer this knowledge to new, previously unseen environments.

We can therefore expect that user security would benefit greatly if there was agreement on standard signs and signals for security and privacy options and notifications across applications and platforms. If future versions of applications and new technologies keep to such a standard, it would allow users familiar with previous technologies to transfer their accumulated knowledge to the new technology.

At present, we are in the unfortunate situation that the location and representation of security settings and notifications not only change between vendors, but even on a yearly basis from one major version of a software to the next. To give just one example, in an update of the iOS mobile phone operating system, the behaviour of the Bluetooth and Wi-Fi buttons in "Control Center" changed while their appearance (shape, color, and symbol) remained the same. In iOS 10 these buttons turn the respective services on and off. In iOS 11 these buttons temporarily disconnect some devices, but the respective service remains on[1] and the system continues to provide access to the service [2, 3]. The result of such an interface tweak is that the users' mental models further diverge from reality.

3 Shaping our Mental Model of Security

Our position is that human-computer interfaces should provide security-relevant information with standardised security signs. Ideally, the security signs alone should provide enough information for the user to understand the security implications of an action (or inaction) to the user's assets.

In the following we explore how signs and signals could be designed to support a user's security-critical decisions. We start with safety signs that are commonly found at workplaces and public areas in the European Union and are standardised by ISO 7010.

3.1 From Safety Signs to Security Signs

Safety signs help people to safely navigate a physical space. There are essentially four types of safety signs distingished by shape and color in ISO 7010. A red circle indicates a prohibition, a blue disk indicates an obligation (a mandatory action), a yellow triangle indicates a warning, and a green rectangle indicates a safe way (a safe condition)[2]. These shapes and colors are combined with pictograms to convey what the safety instruction or information relates to. For example, a cigarette in a red circle indicates that smoking is prohibited, a helmet on a blue disk indicates that head protection must be worn, a lightning bolt in a yellow triangle is a high voltage warning, and a phone in a green rectangle indicates the location of an emergency phone.

The obligation and prohibition signs give instructions about what to do or not to do in an environment. The warning and safe way signs provide information about dangers and safety features in the environment. If we are lost in a hospital, a "no unauthorised persons" and a radiation warning sign on a door should prevent us from trying to find our way out through that door. The prohibition sign instructs us not to enter, the warning sign informs us of the danger.

[1] A new symbol is introduced in iOS 11 for the state in which the Bluetooth or Wi-Fi service is off. Unless the respective service is off to begin with, this state cannot be reached from within Control Center in iOS 11.

[2] Signs indicating the location of fire equipment are depicted on red rectangles.

Our familiarity with safety signs could perhaps be leveraged to communicate security requirements and warnings with such signs to users. However, our focus in the following is on the instructions and information that can be conveyed with the four types of signs. Indeed, the four types of safety signs discussed are not sufficient to signal all security conditions. Safety signs are intended to minimise risk to a subject's safety in an environment that the subject has no or only limited authority over. For example, they must be set up by employers in workplaces as part of their duty of care to their employees. The safety signs are a communication from the employer to the employees.

In contrast, the responsibility to keep a user's assets secure in cybersystems is shared between the user and their system and it requires a two-way communication between the user and the system. The user inputs data into the system and has some control over the system. The system is supposed to store, processes, and protect the user's data as directed. Both user and system may signal an obligation or prohibition to each other. To avoid confusion, we must therefore distinguish between signs that represent an obligation/prohibition imposed on the user and signs that represent an obligation/prohibition imposed on the system. For example, if a system uses safety signs to communicate instructions to users, it would use a blue disk and red circle respectively to signal an obligation and prohibition imposed on the user. Consequently, obligations and prohibitions that the user wishes to impose on the system would have to be represented by shapes and colors that are clearly distinct from those used for safety signs.

Moreover, since the user has a choice whether to impose a restriction on the system, there must also be related signs for its opposite, i.e., for the "release from obligation" and the "permission" types of instructions. From a security requirements perspective, these instructions could be represented by a single type, i.e. a "no restriction" type, as they do not impose a restriction on the system. In practice, however, it might be clearer from a user's perspective to have distinct signs for the two types. We will refer to the no restriction type in the rest of the paper.

We argue below that these new types of instructions suffice to represent security requirements for standard data security properties. To this end, we must first define what security means.

3.2 Security Requirements

A security requirement is a set of acceptable system behaviours. If the system's behaviour is in the acceptable set, the system satisfies the security requirement. We say that the system is secure if it satisfies all of the user's security requirements. Otherwise the system is not secure.

Standard data security requirement types concern the control of read and write permissions to data and resources and impose availability and functionality requirements on services and resources. It is well-known that the former requirements are prohibition type requirements and the latter obligation type requirements.

With the security signs discussed in the preceding section, the user can define security requirements by setting obligations and prohibitions to be imposed on the system. It follows that the security signs can represent standard types of security requirements.

Once the obligations and prohibitions imposed on the system are set, the user *expects* the system to be secure, i.e., meet all imposed obligations and prohibitions. However, the system will frequently be unable to meet all user-imposed requirements, for example due to external factors, such as power loss, or conflicting requirements imposed by the user. The notion of system security defined above is impractical as it prevents the analysis of how the system and user can resolve some benign violations of security requirements. We therefore relax the conditions and define *user security* as the absence of bad surprises for the user. More precisely, we allow the system to break a user-defined security requirement if it previously informs the user about the need to change a requirement and offers options to do so. Thus the user's expectation of which security requirements hold is defined by the choices the user makes in the system's interactive dialogs and the information that the system communicates to the user. User security is satisfied if the system satisfies the user's expected security requirements.

For user security to be satisfiable in a non-trivial system, the security signs must be expressive enough for the system to communicate violations of security requirements to the user and provide options to resolve them. We discuss this next.

3.3 Communicating with security signs

Users may have control over a wide range of services, from an alarm clock to voice command recognition. A typical service runs in the background without a prominently audible or visible user interface. Enabling a service creates an obligation for the system to eventually or continuously perform a task. Similarly, a user can launch applications obliging the system to perform an immediate task. The user can define permissions and prohibitions by granting or revoking the permission for a service or application to access data or a device. For example a voice command recognition service may be prohibited to access information about the user's contacts while a messaging application may be permitted to access the microphone. The system communicates the obligations and prohibitions that the user can control and these are all represented as obligations and prohibitions imposed on the system.

Compliance and Violation. A user's security requirements (i.e., obligations and prohibitions for the system) may be impossible for the system to satisfy. In this case the system must alert the user with a warning sign and provide information about the violated requirement. To avoid confusion, a sign representing a violation of an obligation or prohibition, which is of an information type, must be distinguishable from the obligation and prohibition signs which denote an instruction. We therefore need a violation type whose signs are related to but clearly distinct from obligation, prohibition, and no restriction signs.

In order for the system to inform the user about the reason for a violation, the sytem may also need to refer to security requirements that are satisfied. We thus also need signs of a compliance type that indicate satisfied security requirements. Note that compliance and violation signs may refer to restrictions imposed on the system or the user.

The information alerting a user to a potential security violation is therefore composed from several signs: First, a warning sign to indicate that the user's security has been or is at risk of being violated. Second, the violation sign for the one or more security requirements that cannot be fulfilled. Third, the reason (if known) for the violation of the aforementioned security properties which are compliance or violation type signs.

If there are one or more options for the user to resolve the situation, the user must be presented with these options. This can be achieved with a safe way type sign followed by the available options which we will discuss in the next section.

We briefly give examples for various types of obligation violations that could occur.

1. Obligation violated due to a prohibition imposed on the system.
   The system may be unable to run a voice messaging application because it does not have permission to access the microphone.
   To alert the user, the system would signal a warning sign, a violation sign for the obligation imposed on the system to run the voice messaging application, and a compliance sign for the prohibition to access the microphone.
2. Obligation violated due to an obligation violation by the user.
   The system may be unable to run a voice messaging application because it does not have a microphone or may be unable to install security updates because it is operating on low battery power.
   In both cases there is a lack of an external resource which is impossible for the system to control. It is the user's obligation to provide the resource, i.e., ensure that there is a microphone, a sufficiently charged battery, or a power supply.
   To alert the user, the system would signal a warning sign, a violation sign for the obligation imposed on the system and a violation sign for the obligation imposed on the user to provide the missing resource.
3. Obligation violated due to an obligation imposed on the user.
   Similarly to the previous example, a portable loudspeaker may be unable to play music while it is being recharged. As recharging the device is an obligation imposed on the user, the system would signal a warning sign, a violation sign for the obligation imposed on the system and a compliance sign for the obligation imposed on the user to provide the missing resource.
4. Obligation violated due to another obligation imposed on the system.
   The system may be unable to perform a backup operation of a storage device while it is repairing the files system of the same storage device.
5. Obligation violated due to a prohibition imposed on the user.
   The system may be unable to open a file because the user is not authorised to access the file.

We have thus seen that the introduction of violation and compliance signs enables the system to communicate its state to the user. It remains to discuss the communication of options provided to a user to define security requirements or resolve a violation or conflicting set of requirements.

Options. To maintain or return to a secure state, i.e., satisfy the user's expected security properties, the system must allow the user to choose from all available options.

The security sign types defined thus far allow us to represent options for security requirements by signalling the present state with a compliance type sign and signalling the available options with obligation, prohibition, and no restriction signs. The user's choice can then be represented by changing the compliance type sign to the new state or by signalling a warning, as discussed above, if the user's choice leads to a security violation.

In some cases, the system must signal an option involving an obligation or prohibition to be imposed on the user. Such options are only available when the user is not fulfilling the necessary obligation or prohibition. Thus, such options are signalled with a violation type of the obligation or prohibition imposed on the user and the corresponding obligation or prohibition type. For example, if one of the options is for the user to plug in a currently unplugged device into an electricity supply, then the system would display a violation of the obligation imposed on the user to plug in the device and the obligation for the user to plug in the device.

4 Discussion

We have merely explored the feasibility of security signs with respect to the types of information that must be communicated between a user and a system. We have seen that a communication system could be based on 9 different types of signs: Warning signs and safe way signs to alert the user and provide information on how to resolve a problem. Obligation and prohibition signs imposed on the system and respectively on the user to communicate instructions and a type for no restriction imposed on the system. Finally, compliance and violation signs to indicate a state. Four of these types are analogous to standard types of safety signs which provide warnings, information, and instructions to people. The new types extend these to communicate instructions from a user to the system and for the system to communicate its current state to the user.

We have not discussed the number of symbols that this approach would require. There is clearly a trade-off between the memorability of a number of signs and symbols for users and the expressivity of the resulting language, and it is not certain that there is a good solution.

A non-negligible benefit for a system that implements standardised signs as sketched in this paper is that it makes precise what security requirements the system must satisfy. This makes it easier to formally test the system's conformance to the promised behaviour as signalled to the user, the user security property

defined above, as well as other security and usability properties, such as robustness to human error. Moreover, the type and number of signs appearing in user interface dialogues can be used to estimate the complexity of the information the system communicates to its user.

We have tacitly assumed that the system is trustworthy. This assumption is too strong and real-world systems will be using accidentally or adversarially misleading security signs. For example, phishing and overlay attacks [1, 10] are conceivable, but their impact on systems with security signs and their mitigation should be the same as for a system without security signs. A potentially interesting, more specific attack vector could be dark design patterns that are well-known in the advertising industry. These are user interface designs that guide the user towards an unfavourable outcome for the user. A simple dark pattern would be to use the tyranny of the default: Many users are weary of changing default settings or simply do not take the time to explore preference settings. A system could set unfavorable default security settings for the user and make them hard to find. Thus the deployment of security signs in itself would certainly neither guarantee that users' security nor their mental model of security improves.

5 Related Work

Mental models in cybersecurity have been investigated from a few different angles. We mention two that use a formal methods approach.

Combéfis and Pecheur [4] use labeled transition systems to describe the behaviour of a system and the user's mental model of it. They present an algorithm that computes from a system model the minimal mental model that a user must have in order to properly use the system. The minimal mental model is equivalent to the system model modulo a variant of weak bisimulation.

In his PhD dissertation, Houser [5] developed a formal model of user mental models and applied it to discover dangerous user-system interactions within the context of a cloud data storage system and analyse the threats faced by recipients of phishing emails.

6 Conclusion

We have sketched a method to help users develop a useful mental model of security in cybersystems. The basic idea is to always expose users to the same standard signs in the same security context. The security signs may be meaningless to a novice in the beginning, but their repeated use and the user's observation of the effects that follow gradually shape and improve the user's mental model. A wide-spread adoption and consistency across different platforms and software versions would ensure that users' mental models remain accurate across different systems and help reduce security risks due to confusion and misunderstanding.

References

1. Simone Aonzo, Alessio Merlo, Giulio Tavella, and Yanick Fratantonio. Phishing attacks on modern android. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 1788–1801. ACM, 2018.
2. Apple Inc. iPhone User Guide, 2017. `https://help.apple.com/iphone/10`, accessed: 2019-12-30.
3. Apple Inc. iPhone User Guide, 2018. `https://help.apple.com/iphone/11`, accessed: 2019-12-30.
4. Sébastien Combéfis and Charles Pecheur. A bisimulation-based approach to the analysis of human-computer interaction. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 101–110, 2009.
5. Adam Michael Houser. *Mental Models for Cybersecurity: A Formal Methods Approach*. PhD thesis, Department of Industrial and Systems Engineering, University at Buffalo, State University of New York, 2018.
6. Christopher Jensen. 50 Years Ago, 'Unsafe at Any Speed' Shook the Auto World. The New York Times, November 27 2015. Section B, page 3. Online: `https://www.nytimes.com/2015/11/27/automobiles/50-years-ago-unsafe-at-any-speed-shook-the-auto-world.html`, accessed: 2019-12-30.
7. Natalie A. Jones, Helen Ross, Timothy Lynam, Pascal Perez, and Anne Leitch. Mental models: An interdisciplinary synthesis of theory and methods. *Ecology And Society*, 16, 2011.
8. Jerry L. Mashaw and David L. Harfst. *The Struggle for Auto Safety*. Harvard University Press, 1990.
9. Rick Wash and Emilee J. Rader. Influencing mental models of security: a research agenda. In *2011 New Security Paradigms Workshop, NSPW '11, Marin County, CA, USA, September 12-15, 2011*, pages 57–66, 2011.
10. Yuxuan Yan, Zhenhua Li, Qi Alfred Chen, Christo Wilson, Tianyin Xu, Ennan Zhai, Yong Li, and Yunhao Liu. Understanding and detecting overlay-based android malware at market scales. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2019, Seoul, Republic of Korea, June 17-21, 2019*, pages 168–179. ACM, 2019.