# Eliciting Usable Security Requirements with Misusability Cases

Shamal Faily
*Department of Computer Science*
*University of Oxford*
*Oxford, UK*
*Email: shamal.faily@comlab.ox.ac.uk*

Ivan Fléchais
*Department of Computer Science*
*University of Oxford*
*Oxford, UK*
*Email: ivan.flechais@comlab.ox.ac.uk*

*Abstract*—**Although widely used for both security and usability concerns, scenarios used in security design may not necessarily inform the design of usability, and vice-versa. One way of using scenarios to bridge security and usability involves explicitly describing how design decisions can lead to users inadvertently exploiting vulnerabilities to carry out their production tasks. We present Mis-usability Cases: scenarios which describe how design decisions may lead to usability problems subsequently leading to system misuse. We describe the steps carried out to develop and apply misusability cases to elicit requirements and report preliminary results applying this technique in a recent case study.**

*Keywords*-**Scenarios, Misuse Cases, Personas, Goals, Obstacles**

## I. Introduction

Scenarios are widely used by both security and usability professionals, but for different reasons. To usability professionals, they describe how people use a system to carry out activities that achieve their personal or occupational goals. To security professionals, scenarios describe how a system might be misused towards an attacker's own ends. Although scenarios are flexible enough to be used in both contexts, an artifact from one context is not necessarily useful in another. A scenario describing how a student returns a borrowed book to a library may provide no more insight into the security of a library's loan management system than a scenario describing how a professional hacker might carry out a Denial of Service attack on the library's web-server provides insight into the usability of the same system.

Sindre and Opdahl [1] have proposed using scenarios to describe unwanted behaviour in a system. Such behaviour can be encapsulated in a *misuse case*: a sequence of actions, including variants, that a system or other entity can perform, interacting with misusers of the entity, and causing harm to some stakeholder/s if the sequence is allowed to complete. While a popular technique for threat modelling, it remains unclear how useful misuse cases are for understanding situations where poor usability causes users to inadvertently exploit vulnerabilities.

To discover the cause of inadvertent system abuse we need approaches that consider how specified systems are used unintentionally. An example of such thinking is Nathan's work on Value Scenarios [2] where scenarios are used to describe both the positive and negative systematic effects of technology without considering users as malevolent. Value scenarios are vignettes describing the systematic effects of a system to both direct and indirect users over extended periods of time. Not all software systems are as divisive or pervasive as those typically described by value scenarios. Nevertheless, it may be possible to stimulate similar narratives with more supplemental information about the system, its users, and its contexts of use. This information may not be available during the early stages of design where it is envisaged that value scenarios should be employed, but it might be available from the data collected during later stages.

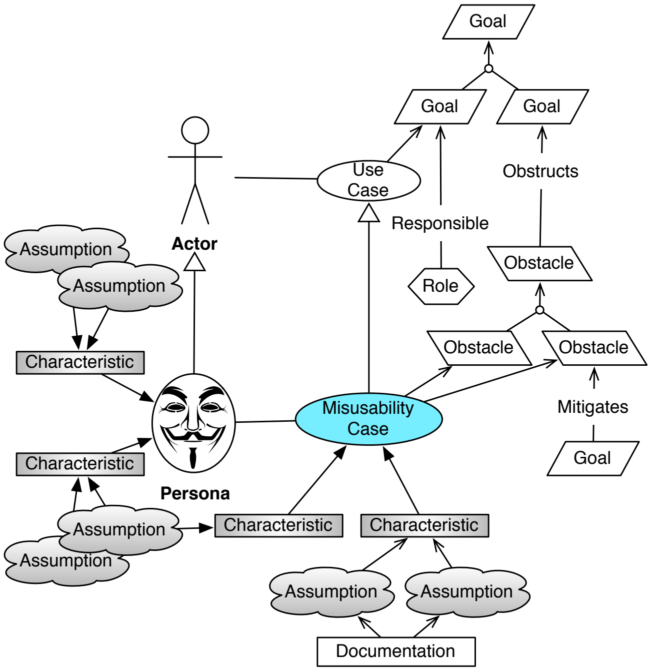

Figure 1. Misusability Case with other design concepts

## II. Our Approach

Misusability Cases are scenarios where a persona achieves a personal or work objective, but inadvertently exploits one or more vulnerabilities in order to do so. The aim of Misusability Cases is twofold. First, to identify cases of insecure misuse within the context of use where activities are carried out using a designed system. Second, to elicit the root causes of this misuse, together with the system requirements which mitigate them.

As figure 1 illustrates, Misusability Cases do not exist in isolation, nor are they used during the early stages of requirements analysis. We assume goals have been elicited corresponding to the requirements a system needs to satisfy. We also assume that use cases [3] have been elicited describing episodes of system behaviour carried out by actors, and one or more personas [4] have been developed to contextualise these actors. Misusability Cases are situated within the IRIS Meta-Model [5]. This meta-model illustrates how concepts from Requirements Engineering, Information Security, and Human-Computer Interaction can be integrated to support the elicitation and specification of secure system requirements.

The Misusability Case technique supports the elicitation of usable security requirements by following a four-step approach. In the first step, implicit assumptions that may give rise to misusability are identified from the design data. In the second step, building upon recent work on structuring the characteristics of personas using Toulmin's model of argumentation [6], characteristics of a scenario are developed where a persona inadvertently endangers a system while performing activities necessary to achieve his or her goals. The third step involves writing a misusability case supported by the characteristics developed in the previous step while, simultaneously, satisfying any related use cases. The final step involves using the KAOS goal-oriented method [7] to identify the obstacles directly contributing to the different aspects of misusability in the misusability case. Based on these obstacles, the higher-level obstacles these lower-level obstacles help satisfy are elicited. This step continues until system requirements are identified, or new requirements are elicited, which are obstructed by these obstacles. Although this step could be construed as an exercise in bottom-up analysis, fitting the misusability case and its contributing obstacles into the larger goal model necessitates both top-down and bottom-up analysis.

## III. Preliminary Results

We used the Misusability Case technique to elicit security requirements for a portal facilitating the sharing of medical study data. Goal models, architectural design documentation, and related usability design artifacts were used as data sources for misusability case elicitation and specification. We updated the CAIRIS Requirements Management tool [8] to support the elicitation and visualisation of argumentation model elements. Of the 21 obstacles and 6 key security requirements elicited during the case study, 15 obstacles and 4 requirements were elicited from misusability cases alone.

Our initial results indicate that, unlike many Security and Usability Engineering techniques which assume their application very early in the design process, misusability cases can be applied at a comparatively late stage. While deferring usability and security design techniques until late in the design process should not be condoned, many teams dedicate significant time and resources to understanding the complexity of a problem domain, leaving themselves little time for applying either Security or Usability Engineering techniques. Our approach demonstrates that such techniques can be effectively situated with Requirements Engineering practice at later, as well as earlier, stages of the design process.

We are currently using misusability cases to explore the impact of architectural design ambiguity and user expectations about security and privacy on the EU FP7 webinos project.

## Acknowledgment

This research was funded by the EPSRC CASE Studentship R07437/CN001, and by the EU FP7 webinos Project (FP7-ICT-2009-5 Objective 1.2) . We are also grateful to Qinetiq Ltd for their sponsorship of this work.

## References

[1] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, 2005.

[2] L. P. Nathan, P. V. Klasnja, and B. Friedman, "Value scenarios: a technique for envisioning systemic effects of new technologies," in *CHI '07: extended abstracts on Human factors in computing systems*. ACM, 2007, pp. 2585–2590.

[3] A. Cockburn, *Writing Effective Use Cases*. Addison-Wesley, 2001.

[4] J. Pruitt and T. Adlin, *The persona lifecycle: keeping people in mind throughout product design*. Elsevier, 2006.

[5] S. Faily and I. Fléchais, "A Meta-Model for Usable Secure Requirements Engineering," in *Proceedings of the 6th International Workshop on Software Engineering for Secure Systems*. IEEE Computer Society, 2010, pp. 126–135.

[6] ——, "The secret lives of assumptions: Developing and refining assumption personas for secure system design," in *Proceedings of the 3rd Conference on Human-Centered Software Engineering*, vol. LNCS 6409. Springer, 2010, pp. 111–118.

[7] A. van Lamsweerde, *Requirements Engineering: from system goals to UML models to software specifications*. John Wiley & Sons, 2009.

[8] S. Faily and I. Fléchais, "Towards tool-support for Usable Secure Requirements Engineering with CAIRIS," *International Journal of Secure Software Engineering*, vol. 1, no. 3, pp. 56–70, July-September 2010.