# User-centered information security policy development in a post-Stuxnet world.

FAILY, S. and FLÉCHAIS, I.

2011

# User-Centered Information Security Policy Development in a Post-Stuxnet World

Shamal Faily
*Department of Computer Science*
*University of Oxford*
*Oxford OX1 6UD, UK*
*shamal.faily@comlab.ox.ac.uk*

Ivan Fléchais
*Department of Computer Science*
*University of Oxford*
*Oxford OX1 3QD, UK*
*ivan.flechais@comlab.ox.ac.uk*

***Abstract*—**
**A balanced approach is needed for developing information security policies in Critical National Infrastructure (CNI) contexts. Requirements Engineering methods can facilitate such an approach, but these tend to focus on either security at the expense of usability, or vice-versa; it is also uncertain whether existing techniques are useful when the time available for applying them is limited. In this paper, we describe a case study where Usability and Requirements Engineering techniques were used to derive missing requirements for an information security policy for a UK water company following reports of the Stuxnet worm. We motivate and describe the approach taken while carrying out this case study, and conclude with three lessons informing future efforts to integrate Security, Usability, and Requirements Engineering techniques for secure system design.**

***Keywords*-personas; misuse cases; KAOS; CAIRIS ;**

## I. Introduction

Information security policies in Critical National Infrastructure (CNI) organisations need to be balanced. The growth in technologies such as distributed control systems and smart grids have meant that media reports about cyber-warfare and terrorism have heightened the security awareness of senior managers. However, the impact of such policies extend beyond the board-rooms and offices where they are drafted. Poorly written policies that constrain the ability of staff to carry out their day-to-day work might compromise operations, leading to the introduction of vulnerabilities to get around them. These problems can be compounded by unforeseen events causing organisations to re-think their current stance on information security. When under pressure, the perception that security design is time consuming may lead policy decisions to be driven by fear rather than rationality. Because few people are fired for making policies too secure, as long as usability and security continue to be treated as qualities to be traded off against each other, policies will err on the side of constraint over freedom of action.

Existing work argues that balanced security and usability can be achieved using Requirements Engineering best practice, but such work tends to emphasise security or usability, but not both. For example, while the SQUARE method [1] provides practical advice on selecting techniques for eliciting security requirements for different organisational contexts, it does not support the development of usability artifacts. Similarly, while the RESCUE method [2] prescribes techniques for human activity and context modelling, it implicitly assumes that secure and usable systems will naturally follow by applying the method. Unlike SQUARE, RESCUE fails to pay attention to the adversarial element intrinsic to security design.

A recent study demonstrated how Usability and Security Requirements Engineering techniques can be aligned in system design without considering Security and Usability as trade-off concerns [3]. However, because this study took place over a period of several months, it is difficult to determine how useful these techniques might be when working to a tight deadline. In such situations, limited time is available for collecting empirical data and running focus groups or workshops.

This paper reports the results of a case study where a user-centered approach was taken to elicit and analyse information security policy requirements following reports of the Stuxnet worm [4]. This policy covered operators working at water-treatment plants in a UK water company. The management imperative for responding to Stuxnet meant that policy decisions needed to be made where there was both a lack of time for data collection and restricted stakeholder availability. Nevertheless, we found that an initial, up-front investment in User-Centered Design activities paid dividends throughout the study from assisting in vulnerability and threat identification, through to spotting fallacious arguments during risk analysis discussions. In section II we motivate and present the approach taken to carry out the case study, and describe our results. In section III, we describe some of the lessons learned carrying out this study which, we believe, inform future approaches for secure system design.

## II. Approach

The aim of this study was to understand how successful Usability and Requirements Engineering techniques might be for eliciting and specifying organisational Information Security requirements. Because this evaluation would take place in a real-world context rather than a controlled environment, this case study was carried out as an *Action*

*Research* intervention [5]. Action Research is an iterative research approach involving the planning of an intervention, carrying it out, analysing the results of the intervention, and reflecting on the lessons learned; these lessons contribute to the re-design of the social action, and the planning of a new intervention. Although primarily used in social science and educational studies research, Action Research has also been used to validate security design methods, e.g. [6]. The objective of the intervention was to elicit and specify missing requirements for an information security policy, as indicated in section I. For reasons of confidentiality, this company will hereafter be known as ACME.

The Action Research methodology used in this paper is that proposed by Baskerville [7], who breaks an intervention into five distinct phases:

- Diagnosis: Identifying the influencing factors in the organisational context impacting the design of the intervention.
- Actions Planned: Devising the planned process to be adopted in order to meet the intervention's objectives.
- Actions Taken: Carrying out the planned steps taken as part of the intervention.
- Evaluating: Evaluating the outcome of the intervention.
- Specifying Learning: Stating the actions which need to feed forward to future interventions or research.

For reasons of brevity, we will describe the actions planned and taken in sections II-C, II-D, II-E, and II-F; the discussion in section III constitute the results of the *Evaluating* and *Specifying Learning* phases for this intervention.

### A. Influencing factors

In July 2010, early reports of how the Stuxnet worm had infected several industrial plants around Europe began to appear. These reports shook up senior management at ACME for several reasons. First, a long held assumption that the obscurity of their SCADA (Supervisory Control and Data Acquisition) systems made them immune to security was dispelled; the virus explicitly targeted the same type of SCADA software used by ACME. Second, by combining knowledge of zero day threats with a realistic means of spreading the virus, i.e. via USB sticks, plant control software no longer seemed as isolated as it once was. Finally, although the motivation of the attacker was, at the time, unknown, the technical sophistication of the virus suggested that the virus was professionally developed to cause harm. While ACME didn't believe they were the virus' target, they were acutely aware that the impact of being infected was largely unknown. They did, however, agree that an effective means of mitigating the likelihood of being threatened was to devise a specific information security policy for those staff working in plant operations.

Although many of ACME's sites were unstaffed, the planned policy would cover staffed clean water plants and sewage works serving large urban areas. These plants were staffed by operators responsible for the running of the plant, and its treatment operations. Plant operators were acutely aware of the safety implications of clean and waste water treatment. Waste water effluent which is not properly treated could have a significant impact on the ecosystem and the food chain. The clean water treatment processes are also critical enough that quality warnings are automatically forwarded to ACME chemists and quality assurance teams. Plant operators were also made aware of the security implications of deliberate attacks on the clean water infrastructure. Like other employees at ACME, information security communiques were regularly sent to all ACME staff, and police periodically visited clean water treatment plants due to the perceived risk of possible terrorist action. There was, however, a feeling held by the information security team that plant operators perceived the threats described in these communiques as irrelevant to their work.

The new security policy would need to cover both the existing infrastructure, and a new Enterprise SCADA system currently being rolled out to other parts of ACME. There were, however, two issues which would need to be considered when designing policy requirements for this system. First, access to stakeholders working in this project were limited. The project relied on external contractors, several of whom were paid a substantial amount of money for their expertise. Their insight would be required for this intervention, but their time needed to be carefully managed. Second, a number of technical requirements had been stipulated by the Enterprise SCADA system manufacturer. At the start of the intervention, it was unclear what impact these might have on the security policy, and ACME's ability to enforce it without compromising this new operating environment.

### B. Approach taken

Based on the influencing factors, we determined that the intervention needed to be completed in a timely manner; this would ensure that the initial analysis would be available to senior managers quickly. We also determined that stakeholders working at water treatment plants, from plant operators to managers, would need to be engaged in the process without underselling or overselling the importance of security and usability in policy decisions. Finally, design activities would need to be informed by the on-going design of the new Enterprise SCADA system, and access to resources working on the Enterprise SCADA project would need to be carefully managed.

To meet this criteria, we devised a user-centered process for eliciting policy requirements. This process was *user-centered* because of its early focus on the needs of the policy's users and tasks, and the grounding of these needs in empirical usage data. After agreeing the scope of the policy, a Fieldwork phase was undertaken; this involved holding in-situ interviews with users who would be affected by the policy at sites where the policy would be applied.

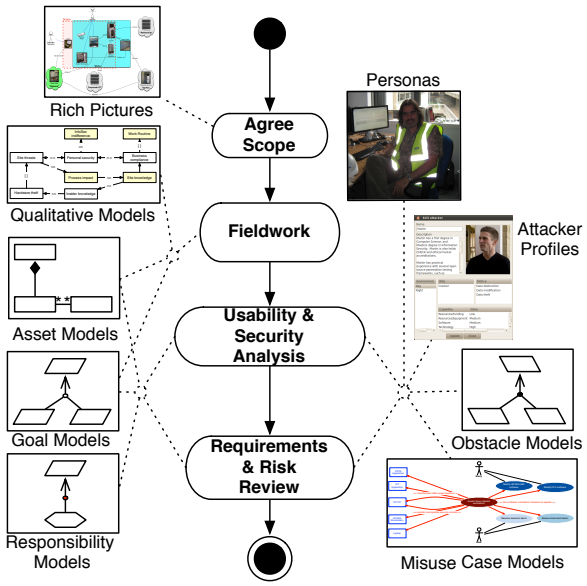


Figure 1. Policy Requirements elicitation process

This data was used to build a qualitative model of security perceptions held by plant operators. The results from this phase informed two further phases: Usability & Security Analysis, and Requirements & Risk Analysis.

Usability & Security Analysis entailed developing personas [8] and, using scenarios, describing the typical activities they carried out. In parallel with this activity, KAOS goal trees [9] were developed to model the policy requirements, and possible ways these requirements could be obstructed. These obstructions were modelled using KAOS obstacle trees. Where possible, obstacles were resolved at this stage using policy requirements. Risks were elicited on the basis of obstacles that could not be resolved without being first discussed by stakeholders.

The Requirements & Risk Review phase involved creating misuse cases [10] to describe the impact of the identified risks, and holding a focus group with key stakeholders to agree possible policy requirements for mitigating them.

As the UML diagram in figure 1 suggests, several different models were generated as part of this process. The artifacts elicited during the analysis and review phases were managed using the CAIRIS (Computer Aided Integration of Requirements and Information Security) Requirements Management tool [11]. CAIRIS builds upon the IRIS meta-model [12], which describes how the concepts underpinning these artifacts are linked. As a corollary, entering data about these artifacts into the tool automatically generates the different models according to the meta-model relationships.

Once the scope had been agreed, a little under two weeks were set aside for carrying out the Fieldwork phase and supporting qualitative data analysis activities. Usability & Security Analysis took place over the following 3 week period before initial policy requirements were available for the Requirements & Risk Analysis review.

Further information about the process and how the different artifacts were generated are described in subsequent sections.

### C. Agreeing Scope

Existing documentation about ACME information security policies was provided as an input to this process. On the basis of this input data, an initial rich picture diagram of the policy scope was developed. Due to time constraints, this was developed off-site and distributed to stakeholders via ACME's Information Security Manager. Although preparation for Fieldwork commenced during this stage, the scope of investigation was bounded only when the rich picture diagram was agreed with ACME. The feedback received from the ACME stakeholders involved word changes which seemed minor, but were semantically significant to ACME. For example, an association was drawn between one system in scope to another box named after the physical location of ACME's head office; ACME's telemetry group and their servers were located at this site. Although the association was valid, the box was renamed to *Bunker* to emphasise the data flow to the telemetry group rather than other groups located at the physical location; the name was commonly used to refer to the group because they were located in a bomb-proof building.

### D. Fieldwork

The objective of the Fieldwork stage was to develop a qualitative model of plant operations security; this would be used to derive one or more personas representing plant operators for later design activities. We visited 4 different water-treatment works (2 clean water and 2 waste water) to hold in-situ qualitative interviews with plant operators and related stakeholders. Although these interviews were largely open-ended, high level questions dealt with the nature of work undertaken, including what plant operators were responsible for, who they worked with, and how they obtained help if necessary. Plant operators were also asked about important work items and activities, and the problems they often faced. Following the interviews, qualitative data analysis was carried out on the interview transcripts and, from this, a qualitative model of plant operator security perceptions was derived

In addition to these fieldwork activities, goal modelling also commenced at this stage. The documents used to drive this activity included a draft security policy that ACME had prepared, an ACME information handling guidelines document, and ACME's organisational security policy. As the aim of the intervention was to elicit missing requirements from the first of these documents, this was the primary document used to elicit goals. For each policy recommendation in this document, a goal was defined. As they were elicited, a goal

tree was induced based on statements which relied on the satisfaction of other goals. Where supplemental documents were referenced, the referenced statements also formed the basis of goals.

In parallel with other activities, an asset model was progressively developed and, by the end of this stage, was mature enough to form the basis of analysing possible security issues. This asset model was based on the AEGIS Asset Model notation [6]. The qualitative data analysis carried out indicated that the two prevalent contexts of interest to plant operations staff would be activities taking place during daylight hours (Day) and the the hours of darkness (Night). With this in mind, assets and security values that stakeholders appeared to hold about them were modelled for each of these two contexts. Data about what constituted *Low*, *Medium*, and *High* value assets were based on ACME's own risk management documentation.

### *E. Usability and Security Analysis*

*1) Usability Analysis:* A plant operator persona (Rick) was derived from the qualitative model developed in section II-D. More details about the technique and how it was used to develop this persona can be found in [13].

Once the personas were ready, 3 scenarios were developed to describe how Rick would carry out his activities during the Day and Night contexts; These scenarios were modelled as tasks in CAIRIS, and textual narratives described how the task was carried out in each context. For example, the narrative associated with the *Resolve reservoir alarm* task during the Day context was as follows:

*Rick looks at the SCADA monitor nearest to him and notices that the levels of the reservoir nearby are unusually high. When the level gets too high, the entire works need to be shutdown. In this situation, Rick knows exactly what to do. After stopping the alarm, Rick logs into the ICT PC next to the SCADA workstation, and clicks on the Xtraview icon. After logging into Xtraview, he finds the location of a pumping station 10 miles upstream on the map and connects to it. After a few moments, he masters the main pump before switching it off. Rick then returns the pump to its normal slave setting before shutting down Xtraview. The alarm periodically starts and stops again but, after about an hour, the reservoir level normalises again.*

Although the above task was identical for both Day and Night contexts, there were a number of variations in other tasks. This was due to the necessity for on-call technicians to resolve problem that on-site staff could have fixed during working hours.

*2) Security Analysis:* At this stage, the goal tree was analysed to find obvious vulnerabilities requiring further analysis. Although no obstacles were forthcoming, a number of goals suggested policy requirements needing to be present in order for them to be satisfied. One such requirement was *Authorised STCS network point data shall be available to authorised plant operators on the ACME portal*. This requirement arose from a goal stating that information about authorised network points should be available to authorised plant staff; this was necessary to allow plant staff to identify network points which might be unauthorised.

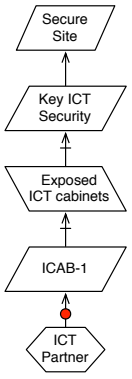


Figure 2. goal tree fragment associated with Exposed ICT Cabinets vulnerability

Although no obstacles were obvious from the goal tree, examining the empirical data collected during the Fieldwork stage identified several vulnerabilities. An example of how these were integrated into the goal tree is provided by figure 2. During the site-visits, cabinets containing network infrastructure were found in publicly accessible areas of certain plants. Based on this, the *Exposed ICT Cabinets* obstacle was introduced; this obstructed pre-existing goals for securing the physical infrastructure. This particular obstacle was mitigated with the requirement *Key ICT equipment shall be stored in a restricted access computer room*. In the figure, this requirement is abbreviated with the label ICAB-1 because it references the ICT Cabinets asset (abbreviated as ICAB).

When all possible vulnerabilities were mitigated, a threat analysis step was carried out to identify possible attackers and the threats they might carry out. From the empirical data, two classes of attacker were identified. The first related to thieves attempting to break into plants to steal scrap metal or other equipment. Several plant operators expressed concern about these attackers because the damage to monitoring equipment they cause is inevitably greater than the value of the items stolen. Plant operators were also worried about their own personal safety should they be required to confront them out-of-hours. A *Kit Theft* threat was defined to model the impact of this attack.

The second class of attacker arose from a general indifference that plant operators and engineers held about information security threats. Even after describing the recent reports of Stuxnet, participants interviewed were still unconvinced that "hackers" were as convincing a threat as the press and information security communiques would have them

believe. Consequently, to portray an attacker that would be believable, an profile was developed based on a penetration tester that could, potentially, be commissioned by ACME; this attacker was grounded in a number of open-source intelligence resources and texts on penetration testing. Based on this attacker, several threats were identified, such as war-dialling modems, *footprinting* to determine information about possible ACME network services, and enumeration of possible passwords using known defaults for software applications. Although several obstacles were elicited based on these threats, no mitigating requirements could be identified without further discussing the threats and their consequences with ACME stakeholders.

### *F. Requirements and Risks Review*

The final stage involved running a focus group with ACME stakeholders and presenting the misuse cases encapsulating the unmitigated risks. These stakeholders were operational managers responsible for plant security and a representative ICT manager. Because only a limited amount of time was available, the presentation of the analysis was centred around a discussion of risks of most interest to ACME: a virus-infected SCADA workstation, and a site-break in. The misuse case associated with each risk was presented, discussed and, based on the outcome, mitigating strategies were proposed. For each discussed misuse case, a misuse case model was developed; as indicated in section II-B, each model was generated automatically by CAIRIS.

For the first risk, a policy requirement was added to remove USB access to SCADA workstations. Responsibility for the second risk was provisionally passed to ACME's facilities management department.

After updating the CAIRIS model based on these discussions, a revised specification document was re-issued to ACME. Because of the limited time available during the focus group, a more detailed review of the analysis took place at ACME's head-office several weeks later. In this one-to-one session with ACME's Information Security Manager, the goal model and elicited policy requirements were validated, and the risk analysis results were reviewed. The purpose of this session was to ensure that all goals and requirements were assigned a responsible role and responses were elicited for each risk.

On completion of the study, 106 separate policy goal statements had been elicited. The vast majority of these were associated with the Day context; this reflects the many day-to-day concerns that participants had with regards to security policy coverage. Similarly, the threats most evident from the empirical data were based on attacks expected to take place during daylight hours.

## III. Lessons learned

We believe the outcome of the intervention was a success for two reasons. First, despite the challenging time constraints, the study was completed comparatively quickly without compromising the quality of the artifacts created. As section II-B reports, the study was largely completed in just over one month with ACME involvement limited to occasional email discussions, in-situ interview participation and a single focus group session to discuss key misuse cases. Second, all elicited policy requirements were accepted by ACME. Moreover, the design models created during the study were used to help with other security issues in ACME. For example, the Rick persona was subsequently used to inform design decisions about user account profiles. In the following sections, we describe three findings from this case study that, we believe, inform future efforts to harmonise Security, Usability, and Requirements Engineering techniques as part of secure system design.

### *A. Fieldwork is security sense-making*

Focusing on security design activities at the same time as Fieldwork activities heightened awareness of possible threats and vulnerabilities at an early stage. For example, on one site-visit, questioning the purpose of one particular PC led to the discovery that not only was it superfluous to plant operations, but the modem attached to it was vulnerable to war-dialling attacks. On another visit, a chance conversation about a car driving up to the plant's main gate on a CCTV screen led to the discovery that the plant had a second gate, and the access control system for this plant entrance was particularly weak. Based on these observations, we believe that fieldwork makes two important contributions to security design. First, de-familiarisation activities associated with in-situ interviews leads to identification of hitherto unseen affordances; these affordances are potentially exploitable by attackers. Second, opportunities for identifying and analysing vulnerabilities happen at any time and, quite often, such insights might have otherwise remained hidden.

### *B. Threats without up-front threat analysis*

Useful information about attackers and threats was collected without an up front threat elicitation exercise. This is because threat analysis could be informed by the sense-making activities associated with other analysis. There are two reasons why this is an improvement over security design methods relying solely on anecdotal information from stakeholders or security experts to derive threats, e.g. [6], [14]. First, threat elicitation is not exclusively contingent on participatory approaches, which rely on getting stakeholders together in a single location. Second, the task of eliciting attackers followed by threats is easier than trying to elicit attacks in their own right. While the empirical data can point to possible attackers, further research is often necessary to determine what threats these attackers can give rise to and, as a result, which assets might be threatened.

### C. Misuse Cases as cases

Misuse cases were useful for spotting more general fallacies made when arguing against the feasibility of a risk. In particular, we noticed a tendency by stakeholders to undermine the impact of the threat or the severity of the vulnerability by focusing solely on the threat's likelihood and the asset directly under threat. During discussion of the *Site break-in* misuse case, some participants highlighted the limited number of staffed sites, coupled with the relatively high frequency of PC theft, as a reason why incorporating policy requirements to mitigate this risk might be infeasible. However, when it was highlighted that the PCs themselves were less important than the monitoring they facilitated, and that the quantity of staffed and unstaffed sites had little bearing on the impact of the risk, it was agreed to transfer responsibility of the risk rather than ignore it. When discussing the risk during the follow-up meeting with ACME's Information Security Manager, it was highlighted that transferring the risk in its entirety was inappropriate. Consequently, the policy goals related to securing physical sites were reviewed to determine which were the responsibility of ACME's facilities management department, and which needed to be pro-actively managed by ACME's own security team.

## IV. Conclusions

In this paper, we have presented the results of a case study where Usability and Requirements Engineering techniques, supplemented with techniques from Secure Software Engineering, were used to elicit requirements for an information security policy for a CNI organisation. This paper has made three particular contributions towards improved harmonisation between Usability, Security, and Software Engineering.

First, we have motivated and presented the results of an Action Research intervention in a real-life context of contemporary interest; specifically, CNI in the immediate post-Stuxnet world.

Second, we have successfully evaluated the efficacy of integrating selected Usability, Security, and Requirements Engineering techniques. Specifically, we have demonstrated that rather than adopting a single process model, judiciously selecting and applying appropriate design techniques for the organisational context can be economical in terms of manpower and time.

Finally, we have illustrated how we can achieve *Security through Usability*. By focusing on the up-front development of Usability, rather than Security, Engineering artifacts, we can re-use the sense-making activities and empirical data to elicit hitherto unseen vulnerabilities. From this, we can also glean insights about possible system attackers and threats.

## V. Acknowledgement

The research described in this paper was funded by EPSRC CASE Studentship R07437/CN001. We are very grateful to Qinetiq Ltd for their sponsorship of this work.

## References

[1] N. R. Mead, "Identifying Security Requirements Using the Security Quality Requirements Engineering (SQUARE) Method," in *Integrating Security and Software Engineering*, H. Mouratidis and P. Giorgini, Eds. Idea Group, 2007, pp. 44–69.

[2] N. Maiden and S. Jones, "The RESCUE Requirements Engineering Process: An Integrated User-Centered Requirements Engineering Process. Version 4.1," City University, Tech. Rep., 2004.

[3] S. Faily and I. Fléchais, "Barry is not the weakest link: Eliciting Secure System Requirements with Personas," in *Proceedings of the 24th British HCI Group Annual Conference on People and Computers: Play is a Serious Business*, ser. BCS-HCI '10. British Computer Society, 2010, pp. 113–120.

[4] Control Engineering UK, "'Stuxnet' Trojan Targets Siemens WinCC," http://www.controlenguk.com/article.aspx?ArticleID=35267, 20 July 2010.

[5] K. Lewin, "Action research and minority problems," *Journal of Social Issues*, vol. 2, no. 4, pp. 34–46, 1946.

[6] I. Fléchais, C. Mascolo, and M. A. Sasse, "Integrating security and usability into the requirements and design process," *International Journal of Electronic Security and Digital Forensics*, vol. 1, no. 1, pp. 12–26, 2007.

[7] R. L. Baskerville, "Investigating information systems with action research," *Commun. AIS*, p. 4, 1999.

[8] A. Cooper, *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity (2nd Edition)*. Pearson Higher Education, 1999.

[9] A. v. Lamsweerde, *Requirements engineering: from system goals to UML models to software specifications*. Hoboken, NJ: John Wiley, 2009.

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

[11] 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.

[12] ——, "A Meta-Model for Usable Secure Requirements Engineering," in *Software Engineering for Secure Systems, 2010. SESS '10. ICSE Workshop on*. IEEE Computer Society Press, May 2010, pp. 126–135.

[13] S. Faily and I. Flechais, "Persona cases: a technique for grounding personas," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI '11. ACM, 2011, pp. 2267–2270.

[14] F. den Braber, I. Hogganvik, M. S. Lund, K. Stølen, and F. Vraalsen, "Model-based security analysis in seven steps - A guided tour to the CORAS method," *BT Technology Journal*, vol. 25, no. 1, pp. 101–117, 2007.