ArgueSecure: Out-of-the-Box Security Risk Assessment

Dan Ionita, Roeland Kegel and Roel Wieringa
University of Twente, Services, Cybersecurity and Safety group,
Drienerlolaan 5, 7522 NB, Enschede, The Netherlands
Email: d.ionita@utwente.nl, r.j.wieringa@utwente.nl

Andrei Baltuta
Studio Panacee SRL, Web Symphony team
Bucharest, Romania
Email: andreibaltuta@gmail.com

***Abstract*—Most established security risk assessment methodologies aim to produce ranked lists of risks. But ranking requires quantification of risks, which in turn relies on data which may not be available or estimations which might not be accurate.**

As an alternative, we have previously proposed argumentation-based risk assessment. In this paper, based on practitioner feedback, we introduce the latest iteration of this method accompanied by two dedicated tools: an online, collaborative web-portal and an offline version. We focus on the lessons learned in iteratively developing and evaluating these tools and the underlying framework.

This new framework – called ArgueSecure – focuses on graphically modelling the risk landscape as a collapsible tree. This tree structure intuitively encodes argument traces, therefore maintaining traceability of the results and providing insight into the decision process.

I. Introduction

Most risk assessment methodologies require security experts, quantitative data or both. For small companies or projects, this means risk assessment is either skipped, performed by a non-expert based on a checklist provided by a third party such as a standardization institution, or performed by an external consultant. Each of these approaches has advantages and disadvantages. Not doing an RA at all is obviously cheap and risky. Using checklists is fine for known risks, but misses new risks emerging from new technology. Hiring external security consultants allows the identification of risks but still requires the involvement of internal IT specialists and budget-responsibles to decide on mitigations. In this paper we propose a participative, qualitative, argument-based security risk assessment approach to supplement all these approaches. It can be used on its own, or in combination with checklists or external security consultants.

Our approach involves dedicated brainstorming sessions, which can provide an opportunity for stakeholders to collaboratively identify risks (including new and hybrid risks) [1], as well as discuss and agree on security requirements [2]. Furthermore, studies show that project teams – rather than in-house experts or external consultants – are the most common participants in risk analysis meetings and that qualitative approaches are generally preferred by such groups. [3], [4]

We aim to help structure these meetings by providing usable tools, capable of capturing and encoding the key arguments put forth during a qualitative risk assessment. These arguments serve a dual purpose. Firstly, they provide support for the results of the assessment, whether risks or countermeasures. Second, they promote reusability and can be used to construct a knowledge base of such risks and countermeasures.

To this end, we iteratively developed ArgueSecure: a light-weight, flexible, qualitative risk assessment and security requirements elicitation framework, consisting of a set of tools and an associated method. ArgueSecure builds upon previous research into qualitative, argumentation-based risk assessment, described in Section II and is supported by two dedicated tools: a Web server that can be deployed as an intranet or Internet portal, and an offline Java tool. Both tools are open source, and work out-of-the-box with minimal configuration: the Web server is available as a deployable VM while the Java tool is provided as a a single-file portable executable with import and export functionality.

The goal of the offline version is to provide bookkeeping for risk assessment or security requirements engineering sessions. The goal of the online version is to, in addition, allow stakeholders and experts to engage in a risk assessment without being in the same room and even without being available the same time, while maintaining full traceability between security requirements and risks. The development of an online version is based on the assumption that the stakeholders whose input is required for eliciting security requirements might not be available to participate in a dedicated session. In this paper, we describe the evolution of the ArgueSecure method and present our experience with developing and evaluating these tools.

This paper is structured as follows: Section II summarizes related previous work on qualitative risk assessment in general, and argumentation-based risk assessment in particular; Section III introduces the first version of the dedicated ArgueSecure software tool and draws conclusions with regard its applicability, utility and usability; Section IV describes how these conclusions led to a re-design of ArgueSecure as an on-line portal and how this portal was again evaluated on the same criteria; Finally, Section V summarizes the lessons learned throughout developing these tools and discusses implications for practice as well as potential for future work.

The authors would like to extend their gratitude to the Web Symphony development team, in particular Mr. Emanuel Sandu, for their continued support in the development of the ArgueSecure-online.

II. Background

Research into qualitative risk assessment is not new; qualitative risk assessment techniques have been extensively applied in the food and health industries to assess new products for which historical data was not available [5], [6]. More recently, as information systems have become ubiquitous, a diverse ecosystem of information security risk assessment methodologies have been developed. Although many of these methodologies are qualitative, they can take days or even weeks to prepare and apply or commonly require help from security experts [7]. Furthermore, most older established methods rely on checklists or formal models [8]: Checklists quickly explode in size or become obsolete while drawing attention onto procedural details without providing understanding towards the nature of the problem [9], while models can be time-consuming to construct for complex systems or may be unavailable for systems under development [10]. The advantages of "soft approaches", particularly those involving stakeholders, have been highlighted [11].

Notably, Haley et al. have proposed a security argumentation framework for eliciting and representing security requirements [12]. The approach is based on the argument structure originally proposed by Toulmin in 1958, which consists of a *claim* which (1) follows from a et of mutually-accepted *facts*, (2) is justified by a *warrant* which in turn is supported by some *backing*, and (3) potentially challenged by a *rebuttal* [13]. This security argumentation approach was shown to adequately capture the rationale behind security requirements (called the inner argument), as well as the relationship between requirements and vulnerabilities or risks (the outer argument) [14]. However, for realistic security requirements, the UML-like representation of a single argument can exceed available screen estate.

CAIRIS is a usable, industry-ready requirements management tool which supports eliciting security requirements based on a simplified, argumentation-supported risk analysis [15]. However, such an assessment still requires some preparation: defining personas, assets and tasks [16]. Furthermore, the tool is not designed for real-time use during brainstorming sessions.

We have previously proposed a light-weight argumentation-based approach to security requirements elicitation [17]. This approach used spreadsheets to maintain a semi-structured log of arguments for the existence of vulnerabilities and the mitigation potential of countermeasures expressed during the risk assessment session. It did not require any dedicated tools and could be used with virtually no training, during a live risk assessment session. More importantly, the tabular format made efficient use of screen estate, thereby being for projection. This spread-sheet based approach was found to be useful, and later adopted by an Estonian R&D company. However, the usability and scalability of using spreadsheets were brought into question. Therefore, we have since iteratively developed and tested dedicated tools, as well as streamlined the method and underlying conceptual model based on feedback from practitioners.

III. First iteration: ArgueSecure-offline

The first dedicated ArgueSecure tool (now ArgueSecure-offline) implemented the same argumentation-based method as our spread-sheet based tool [17] and was intended to be used during dedicated security requirements elicitation sessions. The application is built to be usable in real time and the GUI is designed to work on low-resolution screens, so that it can be easily projected. However, unlike the spreadsheets described in Section II, each risk assessment (i.e. each list of risks and mitigations) now follows, tree-like structure:

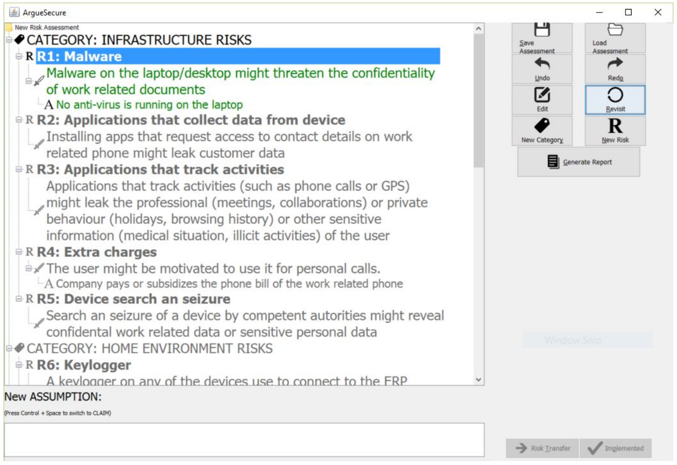
Cat : A category of risks
- **R1** : A risk
  - ⚔ Claim made by an attacker about the existence of an attack path.
    - **A** An assumption of the claim.
    - **A** Another assumption of the claim.
  - 🛡 Claim made by a defender, that partly or completely defeats the attacker's claim by pointing out that an attacker's assumption is probably, or certainly, false.
    - **A** An assumption of the defenders claim, e.g. about a mitigation that already exists or that will be implemented.
  - ⚔ Renewed claim of the attacker that bypasses the defenders argument.
    - **A** An assumption of this renewed claim.
- **R2** Another risk.

Cat : Another category of risks. [etc.]

This structure provides an visual representation of the identified risks and also shows the relationships between risks, attacks and mitigations. In line with previous work, each risk is treated as its own argument game with "attackers" and "defenders" taking turns until the risk is either accepted, reduced, eliminated or transferred [17]. Each turn consists of a single claim which - except for the first - rebuts a previous claim. Defender's claims can refer to ToA[1] components or architectural decisions that reduce or eliminate a risk, but can also refer to decisions, disclaimers, or policies that transfer the risk or potential loss to another party through a contract (e.g., a hold harmless clause) or to a professional risk bearer (e.g. to an insurance company or to a customer). Transfer claims are marked using an arrow instead of a shield.

The buttons and text are large enough to be visible from across the room when projected on a large screen (see Figure 1 for a screenshot). The tool is designed to be usable exclusively via the keyboard so as to support real-time book-keeping of the session. Save/load functionality allows assessments that span multiple sessions and even distributed assessments (by sending the file via e-mail, for instance). Together with various exporting and reporting features, it also supports reusability and dissemination of elicited security requirements.

[1] We use Target of Assessment (ToA) to refer to the software, system, or project which serves as the subject of a risk assessment



New Risk Assessment

CATEGORY: INFRASTRUCTURE RISKS

R R1: Malware

Malware on the laptop/desktop might threaten the confidentiality of work related documents

A No anti-virus is running on the laptop

R R2: Applications that collect data from device

Installing apps that request access to contact details on work related phone might leak customer data

R R3: Applications that track activities

Applications that track activities (such as phone calls or GPS) might leak the professional (meetings, collaborations) or private behaviour (holidays, browsing history) or other sensitive information (medical situation, illicit activities) of the user

R R4: Extra charges

The user might be motivated to use it for personal calls.

A Company pays or subsidizes the phone bill of the work related phone

R R5: Device search an seizure

Search an seizure of a device by competent autorities might reveal confidental work related data or sensitive personal data

CATEGORY: HOME ENVIRONMENT RISKS

R R6: Keylogger

A keylogger on any of the devices use to connect to the ERP

New ASSUMPTION:

(Press Control + Space to switch to CLAIM)

Fig. 1. Screen-shot of ArgueSecure-offline

A. Deployment and usage

The application is provided as a single, self-contained executable.

Conducting an ArgueSecure RA requires little preparation. Any number of stakeholders, domain experts and/or security experts can participate, but should be split up into two teams: attackers and defenders. The method assumes the participants posses pre-existing knowledge of the Target of Assessment. Ideally, but not mandatory, some sort of system model or diagram should be agreed upon by the participants. The preferred workflow is as follows:

1) Create new category and give it a name
2) Create a new risk under this category, and provide a brief name/description of it
3) Each risk starts with an attacker argument, describing an attack path or refining the risk. Each argument consists of a claim, supported by one or more assumptions.
4) Each attacker argument may be countered by a defender argument, describing a countermeasure mitigating the risk.
5) This back-and-forth rhetoric can continue until:
    a) The attacker team is unable or unwilling to counter the last defender argument. This means the risk has been eliminated.
    b) The defender team is unable or unwilling to counter the last attacker argument. This means the risk (or residual risk) has been accepted.
6) If other risks can be identified under this category, go back to step 2.
7) If a new category of risks can be identified, go back to step 1.

At any time during the assessment, defender arguments can be visually marked as ”implemented” if they describe existing risk countermeasures and/or ”transfer” if they describe a risk transfer.

B. Validation and lessons learned

We evaluated the usability and utility of the offline tool by a pilot study, a focus group and a case study.

1) Pilot study: The pilot study was carried out within the PISA[2] (Personal Information Security Assistant) project to obtain an initial overview of risks faced by employees working remotely or from home. After instantiating an assessment with several known risks, the tool, together with this draft assessment was sent via e-mail to various domain experts which were asked to complete the assessment as they see fit. This was because relevant stakeholders were unavailable at the same place at the same time (P0). Unfortunately, most e-mail servers block executable files and many computers do not have the Java Virtual Machine installed (P1). This has significantly hindered adoption to the point of falling back to a simple text editor. The participants which did contribute always added new risks to the assessment, and never elicited attacker claims rebutting previous defender claims (P2). Furthermore, assumptions were never substantiated (P3). Essentially, the tool was used as a running list of potential risks.

2) Focus group: The focus group consisted of security stakeholders of a major Dutch bank and was used to gather feedback with regard to the usability and utility of the ArgueSecure-offline tool. The goal was to collaboratively perform an ArgueSecure risk assessment of a new home banking authentication device. However, since planning a dedicated session with both security experts and responsible management was not possible (P0), the assessment was conducted in two phases: first, security experts created a list of risks and attacks; then, during a shorter meeting with bank stakeholders, decisions were made on which countermeasures to implement and which risks to accept. We observed that, similar to the pilot study above, renewed claims against elicited defences were rarely introduced (P2). While the tool was generally perceived as useful, participants also indicated that usability and scalability become issues as the depth of the tree increases (P4). However, unlike the pilot study, we were physically present during the meeting and therefore able to encourage participants to express their unstated assumptions.

3) Case study: The case study was aimed at identifying the limitations of the ArgueSecure approach when applied by two security practitioners to a fictional scenario involving ATM security. A facilitator was present to manipulate the tool as the two brainstormed about risks and countermeasures. We observed that, despite instructions by the facilitator, the division into attacker and defender teams was not respected, with both participants eliciting attacks as well as defences (P5). After a one hour session, the participants were also asked to fill out a questionnaire about their experience. The restrictive cardinality of the approach was highlighted as a main weakness. Namely, the inability to (1) map multiple attacks to the same risk (P6), (2) have a risk belong to zero or more categories (P7) and (3) state that an elicited defence mitigates several attacks or risks (P8).

[2] http://scs.ewi.utwente.nl/projects/pisa/

IV. Second iteration: ArgueSecure-online

Solis et al. [18], Cheng et al. [19] and Seyff et al. [20] claim that requirements engineering is becoming more and more a collaborative effort by distributed stakeholders. We combined this general insight in the future of RE with our experience with phenomena P0 and P1 by developing ArgueSecure-online: a distributed, web-based risk assessment and security requirements elicitation portal with real-time collaboration functionality. Its goal is to allow busy stakeholders to contribute to the security requirements elicitation process of a software and/or system in a flexible manner, (thereby avoiding phenomenon P0) and without having to download an executable (thereby avoiding phenomenon P1). The tool allows users to collaboratively or privately build and maintain structured lists of risks and mitigations for software and/or systems. The tool maintains the structure of the offline version, with some key differences:

- A defence claim can no longer be rebutted as this was rarely done (P2) and posed scalability issues (P4);
- Assumptions have been dropped as they were rarely used and commonly misused (P3);
- The separation between attackers and defenders has been dropped: any participant can elicit either risks and attacks or defences at any time (P5);
- Each risk can now consist of multiple alternative attacks (P6);
- Top-level categories have been dropped and replaced with node-level tags to further decrease the depth of the tree and allow filtering (P4) while permitting many-to-many mapping of risks and even individual attacks or defences to categories (P7);
- A single defence can now mitigate several attacks (P8).

A risk assessment in ArgueSecure-online also follows a tree-like structure. Similar to the offline version, the root node is the assessment itself, further decomposed into risks, then attacks and finally defences. The tree is only three instead of five levels deep due to categories becoming tags and assumptions being dropped:

Risk : a perceived weakness (such as a vulnerability) or threat (such as undesirable situation) of the system considered by the risk assessment. A risk is associated with a single risk assessment.

Attack : a specific attack path associated with a risk (such as a method of exploiting a vulnerability or producing undesirable effects). An attack is associated with one or more risks.

Defence : a security requirement (such as an architectural change or policy measure) that mitigates specific attack paths. Additionally, defences may refer to the transfer of the risk to a third-party and may be marked accordingly. A defence is associated with one or more attacks.

Each node, independent of its level, consists of a name, accompanied by a description and a set of optional notes.

Although a single defence may mitigate several risks thereby transforming the tree into a cyclic graph, for visualization purposes this graph is presented as a tree: leaf nodes which have two parents are simply duplicated. However, changes to any instance of the defence will propagate to all other instances.

A screen-shot showing a sample risk assessment is shown in Figure 2 and a demo version can be found online at ArgueSecure.ewi.utwente.nl.

A. Deployment and usage

The ArgueSecure application is deployable both as a VM and as stand-alone web-server. The source-code, VM images and configuration instructions are all freely available on GitHub[3].

Once deployed - locally, on an intranet or on the Internet - the application can be centrally managed via a built-in administrator account.

Regular users can log in individually and are able to create public or private assessments, as well as contribute to the public assessments of other users. Multiple users can contribute to the same assessment simultaneously. Changes are visible in real-time, both as updates to the graphical model as well as notifications. The tool also provides export functionality which prints the assessment as a bulleted list.

B. Validation and lessons learned

We evaluated the online version by means of an observational live study and a second focus group.

1) Live study: The live study was carried out entirely online, throughout the duration of REFSQ 2016[4] (a requirements engineering conference) [21]. Participants were provided with individual, anonymized access credentials which they could use to log into a private deployment of the ArgueSecure portal at any time. They were asked to imagine risks related to organizing and participating in a conference and fill in a questionnaire evaluating the tool. They were given no preceding instructions on how to use the tool.

Unfortunately, the questionnaire received only 6 responses. Most respondents claimed to have some experience with risk assessment. Participants, overall, found the interface suitable for brainstorming about risks, although some did point to more flexible alternatives such as Freemind[5] (offline mind-mapping software) and Trello[6] (online project management tool). While, on average, they rated the interface's understandability on first use with a 3 out of 5, after understanding the basic functionality, ease-of-use was scored with a a 4 out of 5.

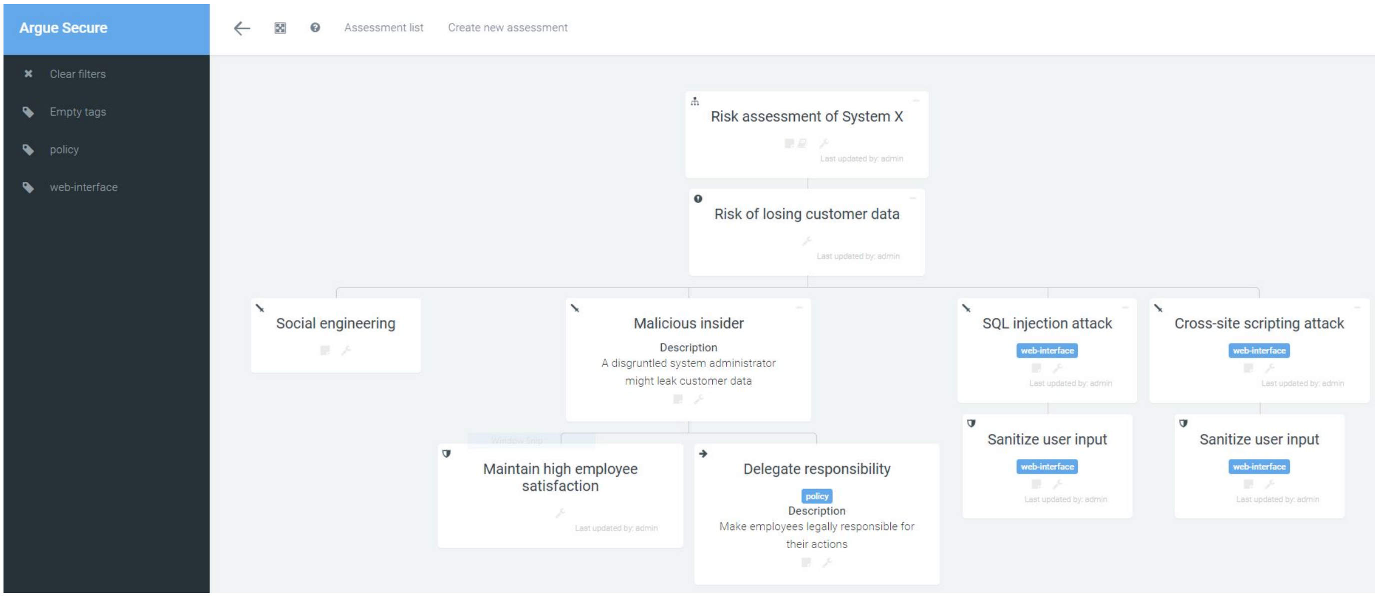
2) Focus group: During the one-hour focus group session, a total of eight information security researchers connected to the ArgueSecure portal using various devices such as laptops,

[3] https://github.com/hitandyrun/arguesecure-online/

[4] https://refsq.org/2016/conference-program/on-line-experiment/

[5] freemind.sourceforge.net

[6] https://trello.com



A disgruntled system administrator might leak customer data

Make employees legally responsible for their actions

Fig. 2. Screen-shot of ArgueSecure-online

tablets and smartphones. Each participant contributed to the assessment individually and simultaneously, without being given any instructions in advance. Despite several opportunities for improvement, all participants rated the interface as easy or very easy to understand and easy or very easy to use, with 4 out of 6 claiming it was very easy to perform desired tasks after only a few minutes. Furthermore, the tree representation was generally seen as suitable for brainstorming about risks. Finally, during the focus group, we have seen tags used to map threat agents to risks, or to categorize risks based on relevant factors for the particular application.

V. Conclusions and future work

Both the offline and online version of the tool, just as the earlier spreadsheet-based version, have been used successfully in real-world risk assessments, but the online version has the advantage of solving problems P0-P8 experienced with the offline version, and both on-line and offline versions have the advantage with respect to the spreadsheet-based tool of ease-of-use and of being more scalable in the number of risks.

Our experience with argumentation-based risk assessment does teach us some more general lessons: Formalized argument structures may be present in other domains, such as in the legal domain analyzed by Toulmin [13], but they are not used by any of the security or other experts with whom we did risk assessments. Assumptions were not stated, and claims once defeated were not revisited. Warrants and backings were present, but every expert found it a waste of time to document these in the RA as they were common knowledge. A modal qualifier indicating the strength of the support for the claim was not given; here the shared understanding was that no claim was supported with 100% certainty, and that the degree of certainty could not be quantified, nor was it worth the effort to estimate it. The decision to invest in a mitigation was made on subjective, unquantified assessments of the severity of a risk and the available budget for mitigations.

This means that the arguments stored in ArgueSecure consist of a claims supported by grounds. The grounds in turn are stated in terms of known vulnerabilities, the architecture of the ToA, and assumptions about attackers' capabilities. Our conclusion is that approaches to argument-based security that require elaborate argument structures, such as that of Toulmin are not usable in practice.

At the same time, the traceability between mitigations and the grounds for these mitigations was found important by all experts. However, while assumptions can help qualify and clarify a claim, most experts did not make these explicit. The tool cannot check for mal-formed or incomplete arguments, but a good facilitator can help in externalizing tacit knowledge without hampering the process.

The latest version of ArgueSecure is a result of several iterations of development and validation. Therefore, the argumentation-based risk assessment method that we started with [17] has evolved accordingly. While this paper is focused on presenting the lessons learned throughout this process as well as the evolution of the toolkit, it could be interesting to distill a formalized risk assessment method based on our findings, ideally after conducting a larger scale observational case study with the help of the ArgueSecure framework.

Finally, the approaches described in this paper have only been validated in isolation. It would be interesting to compare the updated approach with existing argumentation-based, tool-supported risk assessment methodologies, such as OpenRISA[7].

Acknowledgments

The research leading to these results has been funded from the European Union Seventh Framework Programme

[7] An Eclipse client application, available at http://sead1.open.ac.uk/risa.

(FP7/2007-2013) under grant agreement and ICT-318003 (TREsPASS,) and by NWO and KPN under contract 628.001.001. This publication reflects only the authors views and the Union, NWO and KPN are not liable for any use that may be made of the information contained herein.

REFERENCES

[1] E. Lumsdaine and M. Lumsdaine, "Creative problem solving," *IEEE Potentials*, vol. 13, no. 5, pp. 4–9, Dec 1994.

[2] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, 2005. [Online]. Available: http://dx.doi.org/10.1007/s00766-004-0194-4

[3] T. Raz and E. Michael, "Use and benefits of tools for project risk management," *International Journal of Project Management*, vol. 19, no. 1, pp. 9 – 17, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263786399000368

[4] T. Lyons and M. Skitmore, "Project risk management in the queensland engineering construction industry: a survey," *International Journal of Project Management*, vol. 22, no. 1, pp. 51 – 61, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S026378630300005X

[5] M. Wooldridge, "Qualitative risk assessment," in *Microbial Risk Analysis of Foods*. American Society of Microbiology, 2008, pp. 1–28. [Online]. Available: http://www.asmscience.org/content/book/10.1128/9781555815752.ch01

[6] B. L. Johnson, *Effective Risk Communication: The Role and Responsibility of Government and Nongovernment Organizations*. Boston, MA: Springer US, 1989, ch. Qualitative Risk Assessment, pp. 63–66. [Online]. Available: http://dx.doi.org/10.1007/978-1-4613-1569-8_8

[7] D. Ionita, *Current established risk assessment methodologies and tools*, July 2013. [Online]. Available: http://essay.utwente.nl/63830/

[8] R. Baskerville, "Information systems security design methods: Implications for information systems development," *ACM Comput. Surv.*, vol. 25, no. 4, pp. 375–414, Dec. 1993. [Online]. Available: http://doi.acm.org/10.1145/162124.162127

[9] G. Dhillon and J. Backhouse, "Current directions in is security research: towards socio-organizational perspectives," *Information Systems Journal*, vol. 11, no. 2, pp. 127–153, 2001. [Online]. Available: http://dx.doi.org/10.1046/j.1365-2575.2001.00099.x

[10] D. Ionita, R. J. Wieringa, J. H. Bullee, and A. Vasenev, "Tangible modelling to elicit domain knowledge: An experiment and focus group," in *34th International Conference on Conceptual Modeling, ER 2015, Stockholm, Sweden*, ser. Lecture Notes in Computer Science, P. Johannesson, M. Li Lee , S. W. Liddle , A. L. Opdahl, and López , Eds., vol. 9381. Berlin: Springer Verlag, October 2015, pp. 558–565.

[11] G. Dhillon and G. Torkzadeh, "Value-focused assessment of information system security in organizations," *Information Systems Journal*, vol. 16, no. 3, pp. 293–314, 2006. [Online]. Available: http://dx.doi.org/10.1111/j.1365-2575.2006.00219.x

[12] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," *IEEE Trans. Softw. Eng.*, vol. 34, no. 1, pp. 133–153, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1109/TSE.2007.70754

[13] S. E. Toulmin, *The Uses of Argument*. Cambridge University Press, 1958.

[14] Y. Yu, T. T. Tun, A. Tedeschi, V. N. L. Franqueira, and B. Nuseibeh, "Openargue: Supporting argumentation to evolve secure software systems," in *2011 IEEE 19th International Requirements Engineering Conference*, Aug 2011, pp. 351–352.

[15] S. Faily and I. Flchais, "Software for interactive secure systems design: Lessons learned developing and applying cairis," *International Journal of Secure Software Engineering*, vol. 1, no. 3, pp. 56–70, 2010.

[16] I. Fléchais and S. Faily, "Towards tool-support for usable secure requirements engineering with cairis," *Int. J. Secur. Softw. Eng.*, vol. 1, no. 3, pp. 56–70, Jul. 2010. [Online]. Available: http://dx.doi.org/10.4018/jsse.2010070104

[17] D. Ionita, J. W. Bullee, and R. J. Wieringa, "Argumentation-based security requirements elicitation: The next round," in *Evolving Security and Privacy Requirements Engineering (ESPRE), 2014 IEEE 1st Workshop on*. Springer, Aug 2014, pp. 7–12.

[18] C. Solis and N. Ali, "Distributed requirements elicitation using a spatial hypertext wiki," in *2010 5th IEEE International Conference on Global Software Engineering*, Aug 2010, pp. 237–246.

[19] B. H. C. Cheng and J. M. Atlee, "Research directions in requirements engineering," in *2007 Future of Software Engineering*, ser. FOSE '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 285–303. [Online]. Available: http://dx.doi.org/10.1109/FOSE.2007.17

[20] N. Seyff, P. Grunbacher, N. Maiden, and A. Tosar, "Requirements engineering tools go mobile," in *Proceedings of the 26th International Conference on Software Engineering*, ser. ICSE '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 713–714. [Online]. Available: http://dl.acm.org/citation.cfm?id=998675.999483

[21] D. Ionita and R. J. Wieringa, "Web-based collaborative security requirements elicitation," in *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, Research Method Track, and Poster Track co-located with the 22nd International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2016)*, vol. 1564. CEUR-WS, March 2016. [Online]. Available: http://ceur-ws.org/Vol-1564/paper32.pdf