Barry is not the weakest link: Eliciting Secure System Requirements with Personas

Shamal Faily
Ivan Fléchais
Oxford University Computing Laboratory
Wolfson Building Oxford OX1 3QD, UK
shamal.faily@comlab.ox.ac.uk
ivan.flechais@comlab.ox.ac.uk

Building secure and usable systems means specifying systems for the people using it and the tasks they carry out, rather than vice-versa. User-Centered design approaches encourage an early focus on users and their contexts of use, but these need to be integrated with approaches for engineering secure systems. This paper describes how personas can augment a process for eliciting and specifying requirements for secure and usable systems. Our results suggest that personas increase stakeholder empathy towards users represented by personas, and the empirical data used to build personas can also be used to obtain a better understanding of prospective attackers and their motivations.

1. INTRODUCTION

Over 35 years ago, Saltzer & Schroeder [24] penned the principle of Psychological Acceptability, stating that the human interface should be designed such that users routinely and automatically apply protection mechanisms correctly. Since then, seminal work in HCISec (Human Computer Interaction in Security) [2, 30] has reinforced this principle by illustrating how unusable controls introduce vulnerabilities if circumvented or used incorrectly. From a slightly different perspective, Schneier argues that the security of a system is only as good as its weakest link, and the weak links are people [25]. Elahi [13] argues that security is one design directive among many others, and system qualities such as security and usability need to be traded-off against functionality and other qualities. We believe, however, that simply considering people as weak links and treating the early design of a system as an exercise in trade-offs is injurious, and risks introducing unwarranted bias into a design. By thinking of security and usability as antagonistic, we ignore the possibility that creative thinking about people and their problems can lead to a solution where achieving functional, security, and usability goals can be synergistic. Therefore, we need to harness the early stages of design to specify systems around the people that use them and the activities they carry out, rather than vice-versa.

User-Centered approaches to design encourage an early focus on users and collection of empirical data about usage [17]. However, Cockton argues that invention precedes innovation, and User-Centered methods need to contend with technologists “getting there first” [9]. Because eliciting and specifying system requirements are inevitably time-boxed and bound by an agreed scope, if User-Centered design approaches are to be useful then stakeholders need to understand how they can contribute to obtaining these requirements.

Requirements Engineering approaches assume that a system is designed for users, but Cooper argues that the term user is an elastic term which can be stretched and misinterpreted based on the perspective of whoever is using it [10]. Moreover, not only do we need to understand the goals and perspectives of users, we also need to understand the tasks they intend to use the system for, and the contexts within which they work. Personas have been proposed as a technique for tackling these issues; these represent archetypal users, and embody their needs and goals [10, 11].

Unlike Requirements Engineering in general, Requirements Engineering for secure systems is also concerned with safeguarding assets within the system, and ensuring the system being specified appropriately mitigates identified risks. Consequently, requirements analysis also needs to contend with risk analysis activities, such as asset identification, and the identification of vulnerabilities, attackers, and threats. Accepted wisdom suggests that security, like usability, should be built into a system as early as possible, but we lack approaches which help take a synergistic approach to security and usability during requirements analysis.

This paper describes how personas can augment the process of eliciting and specifying requirements for a secure and usable system. By adopting this User-Centered design approach, not only are assumptions about prospective users and requirements discharged at an early stage, the resulting personas inform supplemental techniques used to analyse threats, vulnerabilities, and risks. In section 2 we describe related work in personas and secure system design, before describing the approach taken in section 3. In section 4, we discuss a number of findings from applying personas in the context of Requirements Engineering for security.

2. RELATED WORK

Although we are unaware of existing work using personas to supplement the Requirements Engineering process for secure systems, previous work has separately looked at using these to support Requirements Engineering (RE), and the process of developing secure software systems. In the usability literature, personas are treated as data-driven boundary objects; typically, personas are developed by eliciting data using techniques such as participant observation or interviews. However, the related literature either centres the design process around personas, or treats personas as an assumption-driven boundary object. In the case of the former, the success of the design process hinges on the success of developing the persona. In the case of the latter, the success of the design process hinges on the relevance of the data or assumptions used to derive personas.

From an RE standpoint, Castro et al. [8] have used personas and scenarios to supplement Requirements Engineering activities. Advocates of personas propose complementing them with scenarios, which describe system behaviour from a persona's standpoint[11, 22]. Scenarios describe expected persona behaviours, making them an effective way of validating assumptions underpinning both the persona and the context in general. Participants are also encouraged to think in terms of scenarios at an early stage; this reduces the possibility of participants treating personas as elastic users. Castro and his colleagues also align concepts from Cooper's methodology for developing personas [11] with those used for eliciting, analysing, specifying, and validating requirements. Although many of the techniques used for developing personas appear to align with techniques for eliciting and analysing requirements, Castro is less prescriptive on how personas contribute to other stages. Another contribution by the RE community is Aoyama's methodology, which combines personas, scenarios, and goals [4]. The methodology involves developing personas using conjoint analysis theory, supplemented with focus groups, to identify personas, identify scenarios and goals, evaluating scenarios against the primary personas perspective and his or her goals, and eliciting requirements from these scenarios. Aoyama's methodology facilitates the plug-in of goal models, but the approach is problematic for several reasons. First, Aoyama does not prescribe on how goal-modelling should be integrated into his methodology. Aoyama proposes specialising the concept of Goal with a Goal of Use, yet Aoyama does not describe how a goal is defined. In the KAOS Goal Oriented Requirements Engineering (GORE) methodology [19], a goal is a prescriptive statement of intent that a system much satisfy, but in the i* GORE methodology [31], a goal is defined as the intentional desire of an actor. Because the definition of Goal is ambiguous, by extension, so is the definition of a Goal of Use. Second, Aoyama indicates that requirements analysis is driven from the viewpoint of the persona. However, Pruitt and Adlin [22] argue that personas are best applied when they supplement other analysis rather than replace it. By driving the requirements analysis process from the perspective of a persona, rather than viewpoints more closely aligned to the problem domain, important requirements may be missed.

Although there appears to be no work purporting to use personas in the context of Requirements Engineering for secure systems, personas have been used to support the development of critical systems. This work prescribes persona development using pre-existing data, as a supplement to other design activities. For example, Van Der Linden [28] proposes developing personas at the same time as requirements are elicited, using the source data to pair human aspects such as age and technical familiarity with values such as teen and somewhat technical to develop customer archetypes, and deriving a narrative structure based on these. Moundalexis et al. [20] developed personas to profile different categories of users. These personas supplement rather than directly contribute to the development of scenarios and requirement; they were derived from existing artifacts, and validated with the users represented by the personas.

As well as using personas to better understand the contexts within which a user will interact with the system, personas have also been used to personify possible attackers. Negative personas [11] or anti-personas [22] have been proposed as a means for sketching people the system should not be defined for. Van Der Linden [28] describes how attacker personas can be developed from threat modelling data by identifying aspects of an attacker like motivation and identity, and pairing these with values such as defamation and script-kiddie. Similarly, Steele & Jia [27] have proposed developing personalised descriptions of attackers with descriptive attack scenarios; these scenarios are similar to the Requirements Engineering concept of Misuse Cases, which also describe how a generic attacker might carry out a threat [3]. Unfortunately, because these attacker personas are largely derived from assumptions, they are susceptible to the sort of problems motivating the original need for personas.

3. APPROACH

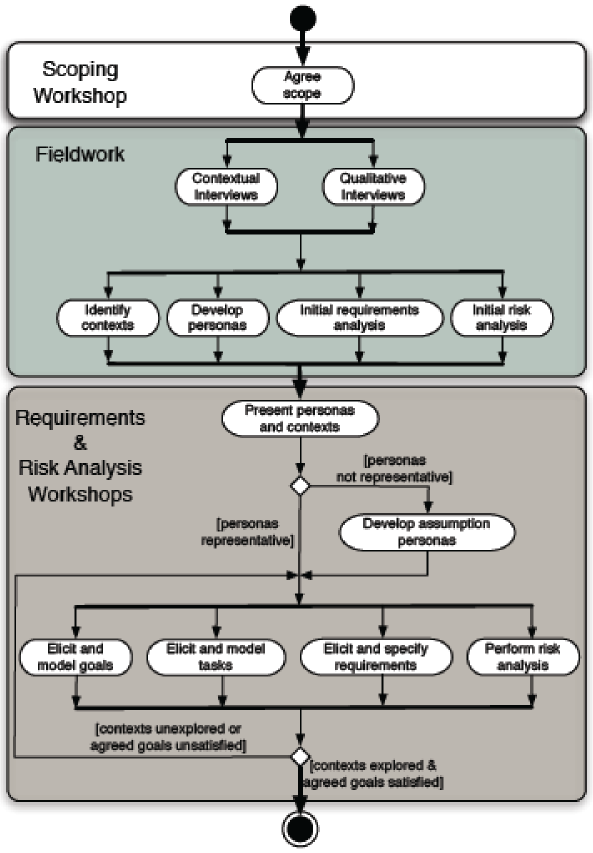
3.1 Process Overview

We have devised the IRIS (Integrating Requirements and Information Security) design process for developing and applying personas to support the elicitation and specification of requirements for secure and usable systems. This process is built upon the IRIS meta-model [14], a conceptual model describing the inter-relationships between concepts from Usability Engineering, Requirements Engineering, and Security.

As the UML activity diagram in figure 1 illustrates, the process was applied in three phases. The first phase involved a half-day workshop where the scope of the system to be specified was agreed. Based on the information collected in this workshop, high-level system goals were elicited, together with a rich-picture diagram bounding the scope of the system, and the key roles the system needs to be defined for. The second phase involved observing and interviewing indicative users; the data elicited was used to identify candidate contexts of use, develop personas, and carry out an initial requirements, threat and vulnerability analysis. The third phase was carried out in three one-day requirements and risk analysis workshops, where the personas were used to support requirements and risk analysis.

The design process was supported by the CAIRIS (Computer Aided Integration of Requirements and Information Security) software tool [1, 16], which is also based on the IRIS meta-model. The tool was used to store information about personas and their contexts of use, as well as data captured during the requirements and risk analysis workshops. This data was used to visually model the impact of security design decisions on personas and the tasks they carried out. Moreover, because CAIRIS supports the notion of an environment, information about how personas carried out tasks in different contexts of use could be elicited; this was useful for surfacing assumptions about personas, tasks, and environments that might otherwise have remained undiscussed.

Figure 1: Activity Diagram of the IRIS design process


Scoping Workshop
Agree scope
Fieldwork
Contextual Interviews
Qualitative Interviews
Identify contexts
Develop personas
Initial requirements analysis
Initial risk analysis
Requirements & Risk Analysis Workshops
Present personas and contexts
[personas not representative]
[personas representative]
Develop assumption personas
Elicit and model goals
Elicit and model tasks
Elicit and specify requirements
Perform risk analysis
[contexts unexplored or agreed goals unsatisfied]
[contexts explored & agreed goals satisfied]

Therefore, rather than personas being forgotten, they were frequently used in conjunction with other analysis. Based on the data provided, CAIRIS can generate a Volere [23] requirements specification document. After each workshop, a specification document was generated by CAIRIS and sent to all workshop participants for comment.

We validated this process by using it to specify a software repository for storing software used to control instrumentation for a UK water company. This software runs on many different devices and locations across wide geographic areas. As part of their responsibility for maintaining the water network, instrument technicians often make software modifications to telemetry outstations, PLCs (Programmable Logic Controllers), and SCADA (Supervisory Control and Data Acquisition) workstations. Without a central strategy for controlling such software, water treatment integrity may be compromised if software is lost, or incorrect software is accidentally, or deliberately, installed on critical instrumentation. However, because maintaining the water network can be physically and mentally demanding, any new technology needs to be situated for the contexts within which these technicians work.

The following sections describe how the personas were developed and applied in the project.

3.2 Persona Development

At the scoping workshop, it was agreed that instrument technicians were the primary users of the future system; the system needed to be situated for the tasks this role carried out. Three contextual interviews [18] were used to collect empirical data about Instrument Technicians and their work; these were supplemented with data from two on-site qualitative interviews and two telephone interviews with related stakeholders. An audio recording of each interview and contextual inquiry was taken and transcribed.

Using Grounded Theory [12] and the ATLAS. ti CAQDAS (Computer Aided Qualitative Data Analysis) tool [21], the transcripts were coded to induce a model of salient grounded concepts and their relationships. After reviewing this model and discussing these salient concepts with colleagues, the model was re-evaluated, and concepts were re-coded and clustered around an emerging set of themes. This emergent model is illustrated in the figure 2; the numbers after each concept relate to how many occurrences of this concept exist within the corpus of empirical data. This process was analogous to affinity diagramming [6], although ATLAS.ti was used to manage concepts and help identify underlying patterns, rather than sticky notes.

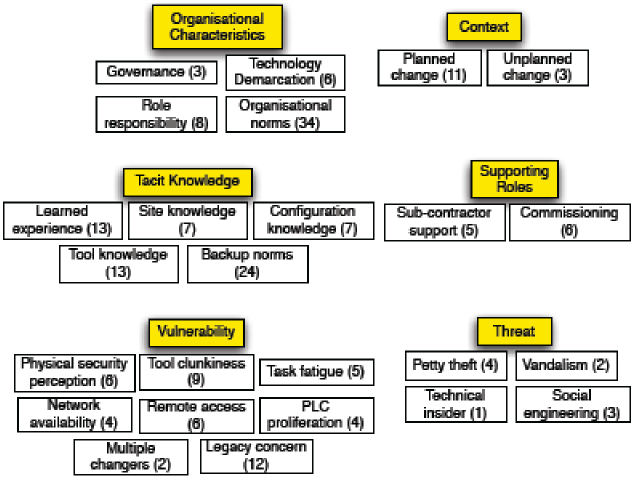


Figure 2: Affinity model of instrument technician concepts

From the various concepts and their grounding, it became apparent a single persona could not be derived which treated the most grounded, salient concepts. As a result, a narrative structure for two different personas were elicited; this structure was broken down by activities, attitudes, aptitudes, motivations, and skills; these sections were based on behavioural variable types recommended by [11]. The first persona (Barry) modelled an instrument technician who carries out software modifications as part of his day-to-day work maintaining instrumentation in the water network. The second persona (Alan) modelled a commissioning engineer who works off-site but is responsible for the initial installation of control software to plant equipment. The complete persona structure was entered into CAIRIS and, as figure 3 indicates, a representative photo was associated with each persona.

In addition to the personas, two contexts of use were identified from this data. The most grounded context of use encompassed work carried out as part of planned, scheduled changes. The other context of use encompassed unplanned work, invariably performed out-of-hours. Based on these contexts, a narrative describing each persona's thoughts and concerns in these contexts of use was also created. Information about these contexts of use was also entered into CAIRIS.

3.3 Working with the personas

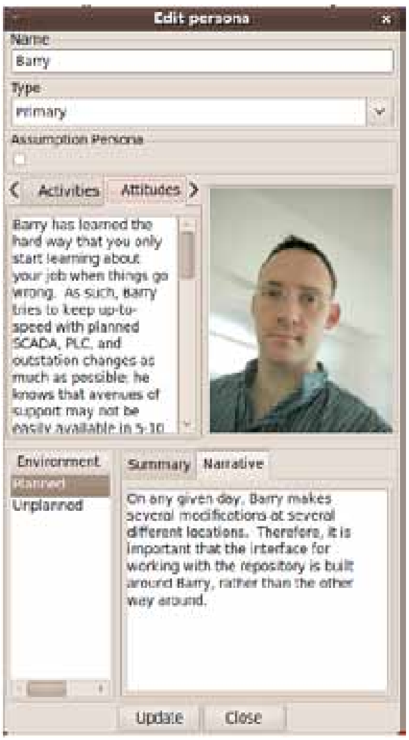
Participants in each workshop represented a cross-section of software repository stakeholders, including instrument technicians, software engineers, information security officers, and system administrators. At the beginning of each workshop, the personas were introduced in a short presentation where the salient characteristics were presented in a bullet point list; an example of the characteristics presented for Alan is provided in figure 4.

Although conscious of past work suggesting that personas should not be named by their developers because participants might not feel ownership for them [7], we also did not want participants to refer to personas as the user while trying to decide names. Therefore, we devised our own names for personas, making sure these did not correspond to any participants or known colleagues. After the personas were introduced by name, participants also referred to them by their pre-selected moniker. Occasionally, however, it was necessary to clarify some aspect of the persona's behaviour and the data underpinning them. In these cases, participants would lapse into referring to the role rather than the persona. When this occurred, rather than breaking the chain of discussion, the facilitator would instead make references to the name of the persona. Other participants would mirror this use of persona naming and start referring to the persona by name again, rather than the role.

3.4 Eliciting requirements with personas

Requirements were elicited during the workshops in a number of different ways. High-level system goals were progressively refined to requirements, which were operationalised by tasks. Participants specified several tasks, which described how personas would tackle the job of maintaining different classes of control software as part of their day-to-day work. After specifying each task, participants would categorise the usability of this task by categorising the amount of time it took a persona to complete the task, how often a persona would carry out the task, how demanding a persona found the task, and how closely the task matched a persona's own work and personal goals. This categorisation process is described in more detail by [15].

Even with the contextual support personas provided, participants occasionally found it difficult to determine how goals could be further refined. Personas were used to kick-start analysis using two simple techniques. The first technique involved simply asking participants questions like "what would Barry do?" This invariably stimulated further discussions about the sort of functionality which would need to exist to envisage possibilities for the persona. The question was also used when trying to gain a consensus. During one workshop, participants had conflicting views about how whether or not instrument technicians would update SCADA interface software using their laptop or a PC on-site at a plant. However, when specifically asked what Barry would do in the context of a software update in a planned situation, it was agreed that Barry would use an access PC at the plant, rather than a laptop he would normally carry around.



Name: Barry

Type: Primary

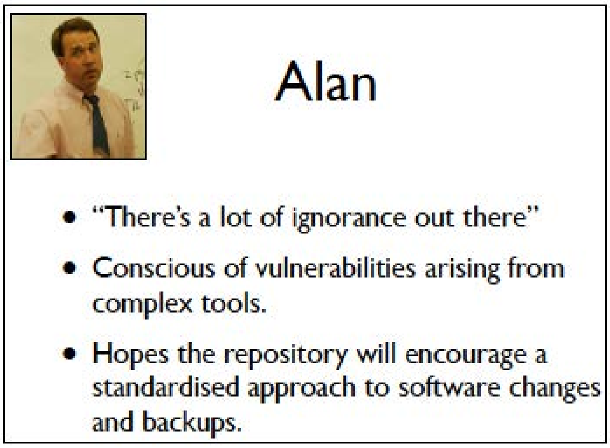
Barry has learned the hard way that you only start learning about your job when things go wrong. As such, Barry tries to keep up-to-speed with planned SCADA, PLC, and outstation changes as much as possible: he knows that avenues of support may not be easily available in 5-10

Planned
Unplanned

On any given day, Barry makes several modifications at several different locations. Therefore, it is important that the interface for working with the repository is built around Barry, rather than the other way around.

Figure 3: Persona details for Barry

Figure 4: Bullet points characteristics of Alan



Alan

- "There's a lot of ignorance out there"
- Conscious of vulnerabilities arising from complex tools.
- Hopes the repository will encourage a standardised approach to software changes and backups.

Further discussion and subsequent analysis led to a requirement being elicited stating the repository shall allow the latest version of SCADA interface software to be downloaded from an access PC. Much of the resulting discussion was also useful for recording the requirement's rationale, together with a fit criterion for testing the requirement.

The second technique involved taking a bottom-up approach to eliciting requirements. This involved participants describing the tasks carried out by personas, after which glossed over assumptions would be explored in more detail. Based on this discussion, assumptions would be explicated and modelled as goals or requirements needing to be satisfied. For example, requirements elicited to satisfy the goal of uploading the latest telemetry software changes to the repository were not readily apparent solely on the perspective of the goal. Therefore, the workshop participants walked through the task where Barry carries out a schedule task to a telemetry outstation. The task began when Barry arrived at the designed site and set about locating the outstation to be modified, and ended when he returned to his home depot to complete the requisite paperwork. After collectively authoring a narrative for the task, the participants walked through the task again to identify requirements which needed to be satisfied. Hitherto unknown requirements elicited for this task included explicitly displaying recent changes to outstation devices; this allows instrument technicians to be appraised of recent work on instrumentation which might impact the work about to be carried out.

As well as helping elicit new requirements, personas were also helpful in validating requirements elicited on the basis of risk analysis. To explore the impact of security on the system being specified, obstacles – conditions representing undesired behaviour and prevent an associated goal from being achieved – were elicited from certain goals [29]. These were progressively refined to identify the threats or vulnerabilities which give rise to these conditions.

Based on these threats and vulnerabilities, risks were identified and mitigation strategies for treating these were identified. These strategies were devised by treating risk responses as goals, and progressively refining these to additional requirements. Candidate countermeasures were derived from these requirements and the impact of introducing these countermeasures on the work carried out by personas was modelled. These requirements was validated by revising tasks carried out by personas, and determining the usability impact the design resulting from the requirements might have.

An example of such a validation occurred when a risk was identified where a malicious instrument technician plants a logic bomb in PLC software using shared repository credentials. It was decided to mitigate this risk by eliciting goals and requirements to detect the planting of malicious code. One of these goals stipulated that PLC software changes shall be peer-reviewed by a different instrument technician within 7 days of a software change. To ensure the reviewer would not be the same technician making the change, several requirements were elicited for a software component which could collect job information when a call is closed; this information included the identify of the instrument technician making the change. This component's design also facilitated the automatic generation of the change documentation a technician would normally need to complete. To understand the impact of these requirements, the task describing how Barry would update PLC software was modified to reflect the changes to his work based on this new software component.

4. DISCUSSION

As section 3.1 suggests, the approach was validated as part of an Action Research intervention [5]. Memos were kept during the collection of empirical data and persona development, and audio recording were taken of the requirements and risk analysis workshops. The following sections describe some of the findings drawn from this data.

4.1 Humanising design

Personas played an important role in humanising design decisions to workshop participants who, previously, had been unaware of the work carried out by the users the personas represented. Although the information security officers and IT support teams had visibility of their organisation's IT infrastructure, the issues relating to systems associated with the plant and water treatment instrumentation were hitherto largely unknown. The only contact IT staff had with instrument technicians arose from responding to requests for enhanced access privileges on laptops, which would allow the successful running of certain software applications. Similarly, because instrument technicians were largely responsible for supporting their own hardware and software, collaboration with IT staff was rare. Maintaining the software repository would, however, become the responsibility of IT support teams, so the personas played an important role in understanding the characteristics of a role they had not previously been exposed to.

When workshop participants the personas characterised were present, this empathy increased to the extent that the personas became a focal point of discussion above and beyond other analysis being carried out. When discussing goals an instrument technician would need to carry out in one workshop, the IT staff empathised with the challenges that Barry faced while working out-of-hours. While eliciting goals and tasks, these participants would periodically ask instrument technicians or participants with an experience in instrumentation how IT decisions related to the repository might affect Barry. Consequently, rather than treating Barry as a security antagonist, IT administrators were keen on understanding how their experiences of installing and testing software were similar or dissimilar to their own. In another workshop, the tasks carried out by Barry were used as a boundary object by a managerial participant to clarify subtle differences between the task and how the instrument technicians present would carry them out.

Because personas were designed as a communication tool, such discourse is inevitable. However, while validating information about personas and their characteristics is important, unfettered discussion risks distracting participants from the ultimate aim of these particular workshops: eliciting requirements for the system being designed. As such, we believe personas need to be carefully introduced and managed when combining them with more focused engineering design techniques.

4.2 Personas and Assumption

Personas Assumption Personas are persona sketches developed to articulate existing assumptions about the user population [22]. As figure 1 indicates, each requirements and risk analysis workshop began with a check that participants were happy with the personas being used. If discrepancies were found between the personas and the participants understanding of them, opportunities existed to modify the persona or created a new assumption persona to explicate these assumptions in more detail.

At the beginning of the 3rd workshop, potential issues were identified with the Alan persona. Although participants believed that Alan was a believable persona, they did not feel his activities were accurate. According to his specification, Alan not only handed over his design documentation to instrument technicians on the commissioning of instrumentation, he also provided informal support in the event of any problems. However, instrument technician participants in this workshop indicated that, more often than not, although information is handed over to a company representative or a process operator at a plant, Alan might not realise that Barry finds it difficult to obtain this documentation. Moreover, although Alan might be contracted to provide some support to Barry, this would only be for a limited time period following commission.

After more discussion about who would nominally support Barry, an assumption persona was specified to capture the characteristics of somebody in this role. This persona captured the characteristics of an engineer with specific expertise on the control systems Barry maintains, but is subcontracted to provide 3rd line support. The persona's name – Eric – was chosen by the participants. The first letter of the name corresponded to the real-life user the participants used to derive behavioural characteristics from, but the name was selected as one which would be believable. However, because of the origins of this assumption persona, participants tended to refer to the source user rather than the persona name.

Information about the assumption persona was collected by participants during the workshop and directly entered into CAIRIS. To ensure assumption personas were not confused with their empirically grounded counterparts, an Assumption Persona option was selected when entering data into the persona entry dialogue box. Setting this option had the affect of adding the <<assumption>> stereotype to the persona name in the visual CAIRIS models where personas are displayed, and the requirements specifications generated by the tool.

4.3 Personas and attackers

Unlike traditional approaches to threat modelling, attackers were not elicited purely by brainstorming in workshops. Before the start of the first workshop, attackers, vulnerabilities, and threats were elicited based on concerns identified in the empirical data. For example, during one of the contextual interviews, an instrument technician mentioned how a dial-in access PC was periodically used by colleagues to patch the SCADA or PLC software. This observation led to further discussion on possible vulnerabilities and threats which might be exposed by this activity. As part of the initial risk analysis, an attacker (Victor) was profiled based on Vitek Boden who exploited a similar vulnerability in Australia in 2001; this subsequently lead to the discharge of millions of litres of raw sewage into the local ecosystem [26]. This attacker was introduced, discussed, and used during the first requirements & risk analysis workshops. This example suggests that rather than deriving personas from secondary data, the empirical data used to derive personas should be analysed to identify data that might be useful for risk analysis.

Personas were also useful for exploring how threats might exploit them. One of the threats identified involved an inside attacker undermining Barry's work to elevate his own standing within the organisation. Initially, it was thought the attacker might have been Victor, but after further discussion about the threats this attacker might launch, it was felt this should represent a disgruntled insider, rather than a disgruntled contractor. This lead to the specification of an attacker (Bob) which reified the motivations and capabilities of this insider. This example reinforced the fact that attacks are rational from the perspective of an inside attacker, and by explicating the activities, aptitudes and motivations of personas, participants could better understand the motivations and characteristics of inside attackers as well. The example also illustrates that rather than changing information about an attacker to ensure it fits the risk analysis data, we should instead introduce a new attacker and explore the impact this has on risk analysis.

5. CONCLUSION

Personas are useful for supporting the design of systems which are situated for users and their context of use. To date, however, approaches for supplementing Requirements Engineering or secure system design have either been overly persona-centric, or overly reliant on assumptions and non- empirical data.

This paper has made a number of contributions towards both Requirements Engineering and User-Centered Design. First, we have detailed a process describing how personas can be developed and applied in the context of Requirements Engineering for secure systems, without overly prejudicing the design process for or against personas. Second, we have described how personas humanise design decisions where users represent previously marginalised user communities. Third, we have described how the persona development process can positively contribute to threat and risk analysis activities; this is a departure from existing work which instead uses the secure systems design process to devise attackers and personas.

Although this paper has shown that ad-hoc assumption persona specification, supplemented by tool-support, can be useful for exploring user assumptions, we have not explored the impact of developing more complete personas on the basis of assumption personas with respect to task, requirements, and risk analysis. Future work will examine how assumption personas can form the basis of more grounded personas, and how these can be evolved in line with other analysis.

6. ACKNOWLEDGEMENTS

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.

7. REFERENCES

[1] CAIRIS web site. http://www.comlab.ox.ac.uk/cairis.

[2] Adams, A., and Sasse, M. Users are not the enemy. Communications of the ACM 42 (1999), 41–46.

[3] Alexander, I. Negative scenarios and misuse cases. In Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle, I. F. Alexander and N. Maiden, Eds. John Wiley & Sons Ltd, 2004.

[4] Aoyama, M. Persona-scenario-goal methodology for user-centered requirements engineering. pp. 185 –194.

[5] Baskerville, R. L. Investigating information systems with action research. Commun. AIS (1999), 4.

[6] Beyer, H., and Holtzblatt, K. Contextual design: defining customer-centered systems. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[7] Blomquist, A., and Arvola, M. Personas in action: ethnography in an interaction design team. In NordiCHI '02: Proceedings of the second Nordic conference on Human-computer interaction (New York, NY, USA, 2002), ACM, pp. 197–200.

[8] Castro, J., Acua, S., and Juristo, N. Integrating the personas technique into the requirements analysis activity. In Computer Science, 2008. ENC '08. Mexican International Conference on (Oct. 2008), pp. 104–112.

[9] Cockton, G. Revisiting usability's three key principles. In CHI '08: CHI '08 extended abstracts on Human factors in computing systems (New York, NY, USA, 2008), ACM, pp. 2473–2484.

[10] Cooper, A. 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.

[11] Cooper, A., Reimann, R., and Cronin, D. About Face 3: The Essentials of Interaction Design.Wiley, 2007.

[12] Corbin, J. M., and Strauss, A. L. Basics of qualitative research : techniques and procedures for developing grounded theory, 3rd ed. Sage Publications, Inc., 2008.

[13] Elahi, G., and Yu, E. Agoalorientedapproachfor modeling and analyzing security trade-offs,vol.4801 LNCS. 2007.

[14] Faily, S., and Fl´echais, I. A Meta-Model for Usable Secure Requirements Engineering. In Software Engineering for Secure Systems, 2010. SESS '10. ICSE Workshop on (May 2010), IEEE Computer Society Press, pp. 126–135.

[15] Faily, S., and Fl´echais, I. Analysing and Visualising Security and Usability in IRIS. In Availability, Reliability and Security, 2010. ARES 10. Fifth International Conference on (2010).

[16] Faily, S., and Fl´echais, I. Towards tool-support for Usable Secure Requirements Engineering with CAIRIS. In International Journal of Secure Software Engineering (2010), IGI Global. To Appear.

[17] Gould, J. D., and Lewis, C. Designing for usability: key principles and what designers think. Communications of the ACM 28,3 (1985), 300–311.

[18] Holtzblatt, K., and Jones, S. Contextual inquiry: a participatory technique for systems design. In Participatory Design: Principles and Practice, D. Schuler and A. Namioka, Eds. Lawrence Erlbaum Associates, 1993, pp. 177–210.

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

[20] Moundalexis, M., Deery, J., and Roberts, K. Integrating human-computer interaction artifacts into system development, vol. 5619 LNCS. Springer, 2009.

[21] Muhr, T. User's Manual for ATLAS.ti 5.0. ATLAS.ti Scientific Software Development GmbH, Berlin, 2004.

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

[23] Robertson, J., and Robertson, S. Volere Requirements Specification Template: Edition 14 January 2009. http://www.volere.co.uk/template.htm,2009.

[24] Saltzer, J., and Schroeder, M. The protection of information in computer systems. Proceedings of the IEEE 63, 9 (Sept. 1975), 1278–1308.

[25] Schneier, B. Secrets & Lies : Digital Security in a Networked World.Wiley,2000.

[26] Slay, J., and Miller, M. Lessons learned from the maroochy water breach. In Critical Infrastructure Protection. IFIP WG 11.10 series in critical infrastructure protection, E. Goetz and S. Shenoi, Eds., vol. 253. Springet, 2007, pp. 73–82.

[27] Steele, A., and Jie, X. Adversary Centered Design: Threat Modeling Using Anti-Scenarios, Anti-Use Cases and Anti-Personas. In Proceedings of the 2008 International Conference on Information & Knowledge Engineering, IKE 2008 (2008), H. R. Arabnia and R. R. Hashemi, Eds., CSREA Press, pp. 367–370.

[28] Van der Linden, M. A. Testing code security. Auerbach Pub, Boca Raton, FL, 2007.

[29] van Lamsweerde, A., and Letier, E. Handling obstacles in goal-oriented requirements engineering. Software Engineering, IEEE Transactions on 26,10 (2000), 978–1005.

[30] Whitten, A., and Tygar, J. D. Why Johnny can't encrypt: a usability evaluation of PGP 5.0. In SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium (Berkeley, CA, USA, 1999), USENIX Association, pp. 14–14.

[31] Yu, E. Modeling Strategic Relationships for Process Reengineering. PhD thesis, University of Toronto, 1995.