Security through usability: a user-centered approach for balanced security policy requirements.

FAILY, S. and FLÉCHAIS, I.

2010

Security through Usability: a user-centered approach for balanced security policy requirements

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

I. Motivation

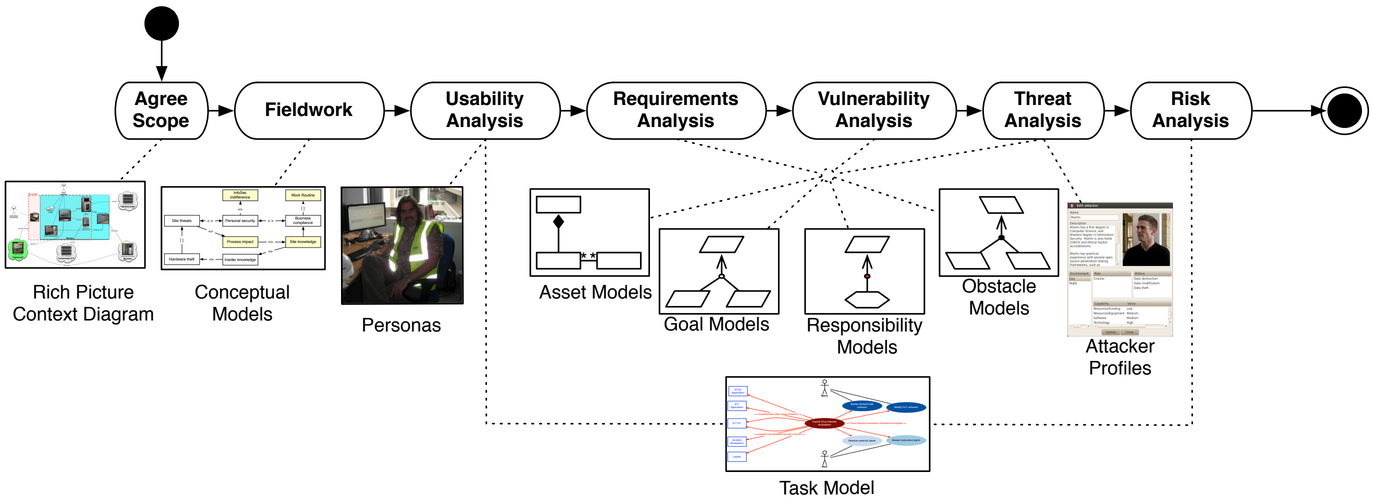
Security policy authors face a dilemma. On one hand, policies need to respond to a constantly evolving, well reported threat landscape, the consequences of which have heightened the security awareness of senior managers. On the other hand, the impact of policies extend beyond constraints on desktop computers and laptops; an overly constrained policy may compromise operations or stifle the freedom needed for staff to innovate. Because few people are fired for making a policy too secure, as long as usability continues to be treated as a trade-off quality together with functionality then policies will err on the side of constraint over freedom of action. Existing work [9] argues that balanced security can be achieved using Requirements Engineering best practice. Such approaches, however, treat usability as another class of quality requirement, and prescribed techniques fail to elicit or analyse empirical data with the same richness as those used by usability professionals. There is, therefore, a need to incorporates techniques from HCI into the task of specifying security, but without compromising Requirements Engineering practice. Recent work demonstrated how user-centered design and security requirements engineering techniques can be aligned [5], [6]; this approach was validated using a general system design project, where ample time was available to collect empirical data and run participatory requirements and risk workshops. The question remains whether such an approach scales for eliciting policy requirements where time is an imperative rather than a luxury.

II. Approach

We have devised a user-centered process for eliciting security policy requirements. Like the process in [6], it begins by agreeing the system scope and key roles with project stakeholders. This stage is followed by fieldwork and usability analysis. During the fieldwork stage, empirical data is collected about how users carry out their day-to-day work. This data is analysed and modelled using a qualitative data analysis methodology [3] to discover important user characteristics. This data is fed into the design of personas representing archetypical users [2], and task scenarios describing the activities they carry out.

Rather than undertaking lengthy participatory requirements and risk analysis workshops, the subsequent analysis is carried out exclusively by analysts in a number of stages. First, assets of value are modelled in UML based asset models. Second, the KAOS goal oriented requirements engineering methodology [4] is used to construct a goal tree of policy requirements. Leaf goals in these trees are assigned to specific roles. Goals may be obstructed using obstacles, which represent conditions

Fig. 1. Policy development process overview



preventing a goal from being achieved [8]. Although analysing goals may suggest possible vulnerabilities and threats, these are primarily defined from the empirical data used to define personas and tasks; previous work [6] found that fieldwork data can also be used to glean information about possible threats and vulnerabilities. Associated with each threat are attacker profiles; these are created using Open Source Intelligence, or published literature on related attacks. Where possible, policy requirements for mitigating these vulnerabilities and threats are elicited at this stage. The next stage involves specifying risks, where a single unmitigated threat exploits a single unmitigated vulnerability. Associated with each risk is a Misuse Case describing a scenario where the attacker carries out the activities necessary to realise the risk. Data from all stages are entered into a dedicated software tool [1], which structures the data, automatically visualises different models, and generates a DocBook policy requirements specification. This tool is described in more detail in [7].

The final stage involves a participatory workshop, where Misuse Cases are discussed by project stakeholders. Using Misuse Cases as boundary objects, this workshop aims to collectively elicit the remaining policy requirements mitigating these risks. Each Misuse Case is illustrated in a Task Model; this model contains the related attacker (or attackers), endangered assets, and personas and tasks related to these assets. Both the tasks and Misuse Cases are represented as coloured ellipses based on a quantitative usability and risk impact score generated by the software tool. When it has been agreed that a Misuse Case has been mitigated, the next Misuse Case is discussed.

III. Preliminary results and Future Work

We applied this approach to define the policy requirements for SCADA (Supervisory Control and Data Acquisition), Telemetry, and Control System software for plant operation staff at a UK water company. After agreeing the system scope and roles, we visited 4 different water treatment plants and, after subsequent usability analysis, one persona and 4 task scenarios were defined. During the initial requirements analysis, 102 policy goals were elicited, the leaf goals of which were assigned to 8 different roles. At an early stage, it was observed that IT support staff were responsible for a disproportionate number of policy goals, as opposed to plant operation and security staff. Based on the empirical data, 8 vulnerabilities, 8 threats, and 4 possible attackers were identified; 3 of these vulnerabilities were mitigated at an early stage. Eight risks were identified based on specific combinations of unmitigated threats and vulnerabilities. A participatory workshop was held to review the Misuse Cases associated with these risks; of these, two Misuse Cases were discussed in detail during the workshop. Based on the discussions, several additional policy requirements were elicited, while others were de-scoped from the study.

Preliminary results suggest that structuring risk analysis discussion around Misuse Case narratives helped ground the discussion about risk impact in the real world. One of the Misuse Cases explored the impact of a virus infecting SCADA workstations. Discussing the impact of policy decisions on the plant operator persona's work simplified policy requirements; in this specific case, the system attack surface was reduced by removing system functionality unused by the persona. Our results also indicate that this process can be completed comparatively quickly. The fieldwork and usability analysis stage was carried out in a little over a week, while the complete study took less than a month. Our results also re-affirm the usefulness of user-centered design techniques for providing supplemental data for security analysis. These techniques do, however, rely on the ability to carry out ethnographic research and qualitative data analysis, the results from which must be interpreted before personas and tasks can be built. We are currently exploring how argumentation structures can be used to build design rationale links between qualitative data and personas. This work potentially enables usability specialists to develop personas based on specific characteristics, which can be tracked back to its originating data.

IV. 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.

References

[1] CAIRIS web site. http://www.comlab.ox.ac.uk/cairis (September 2010)

[2] 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)

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

[4] Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming 20(1-2), 3 – 50 (1993), http://www.sciencedirect.com/science/article/B6V17-45FSTB2-10/2/605ddb5671131b8a28eb4d38d29cab5e

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

[6] Faily, S., Fléchais, I.: Barry is not the weakest link: Eliciting Secure System Requirements with Personas. In: BCS HCI '10: Proceedings of the 2010 British Computer Society Conference on Human-Computer Interaction (2010)

[7] Faily, S., Fléchais, I.: Towards tool-support for Usable Secure Requirements Engineering with CAIRIS. International Journal of Secure Software Engineering 1(3), 56–70 (July-September 2010)

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

[9] Mead, N.R., Hough, E.D., Stehney II, Theodore R.: Security Quality Requirements Engineering (SQUARE) Methodology. Tech. Rep. CMU/SEI-2005-TR-009, Carnegie Mellon Software Engineering Institute (2005)

Security through Usability: a user-centered approach for balanced security policy requirements

Shamal Faily and Ivan Fléchais
Computing Laboratory, University of Oxford
Email: {shamal.faily, ivan.flechais}@comlab.ox.ac.uk

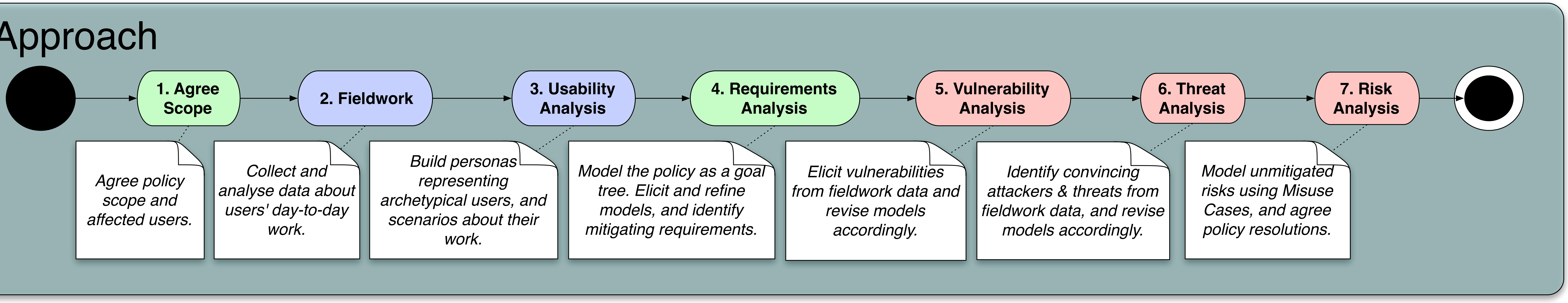
The Problem

- Information Security policies need to respond to evolving threats without over-specifying security.
- There is a noticeable lack of support for writing security policies which balance security and usability.

The Solution

- Make policy development user-centric by applying User-Centered Design [1,2].
- Augment User-Centered Design with complementary techniques & tools from Information Security and Requirements Engineering [3,4,5,6].

Our Approach

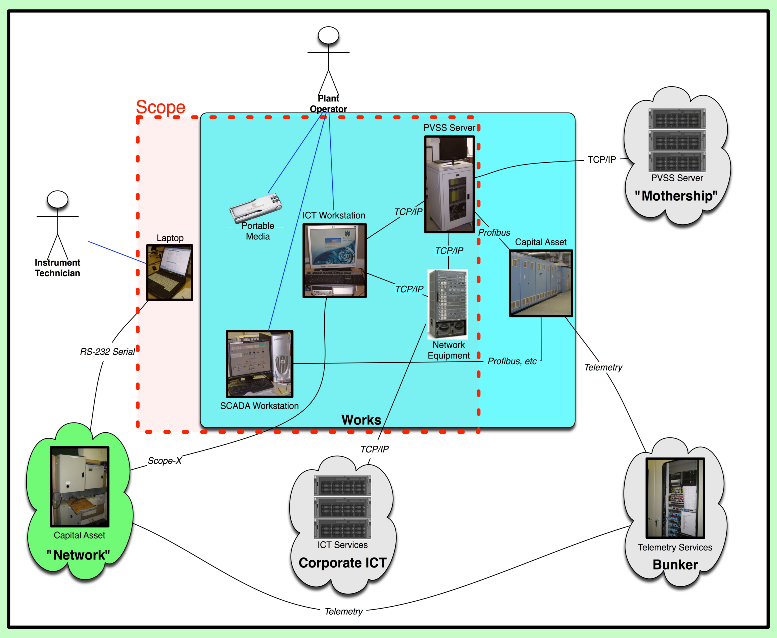


1. Agree Scope: *Agree policy scope and affected users.*
2. Fieldwork: *Collect and analyse data about users' day-to-day work.*
3. Usability Analysis: *Build personas representing archetypical users, and scenarios about their work.*
4. Requirements Analysis: *Model the policy as a goal tree. Elicit and refine models, and identify mitigating requirements.*
5. Vulnerability Analysis: *Elicit vulnerabilities from fieldwork data and revise models accordingly.*
6. Threat Analysis: *Identify convincing attackers & threats from fieldwork data, and revise models accordingly.*
7. Risk Analysis: *Model unmitigated risks using Misuse Cases, and agree policy resolutions.*

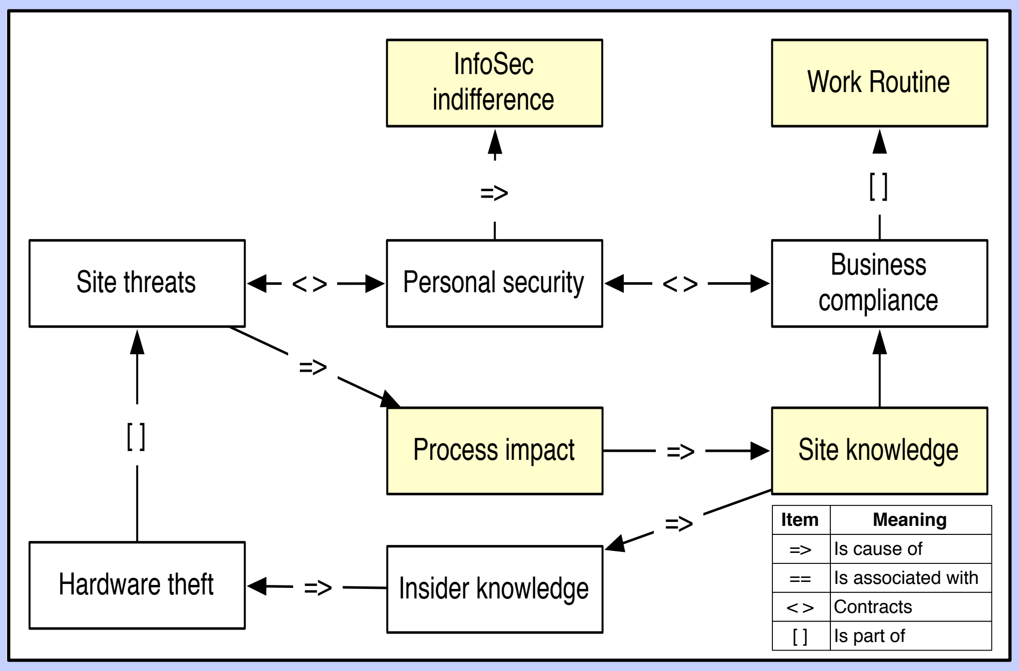
Preliminary Results

✓ Eliciting policy requirements for SCADA and Control Systems used by plant operations staff at a UK water company.

1. The policy scope was agreed & modelled using a Rich Picture Context Diagram.

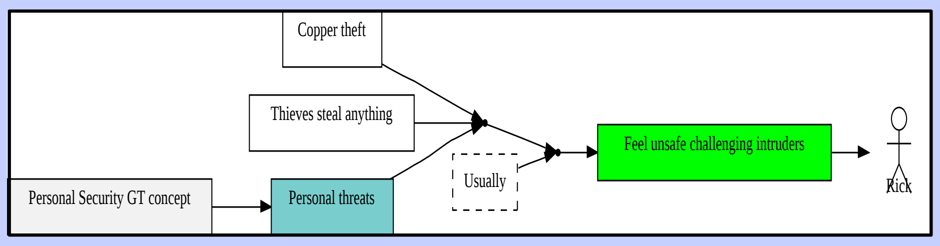


2. We visited 4 different water treatment plants, interviewing plant operators, and other staff. A conceptual model of plant security was developed from a qualitative data analysis of the collected data.



Item	Meaning
=>	Is cause of
==	Is associated with
<>	Contracts
[ ]	Is part of

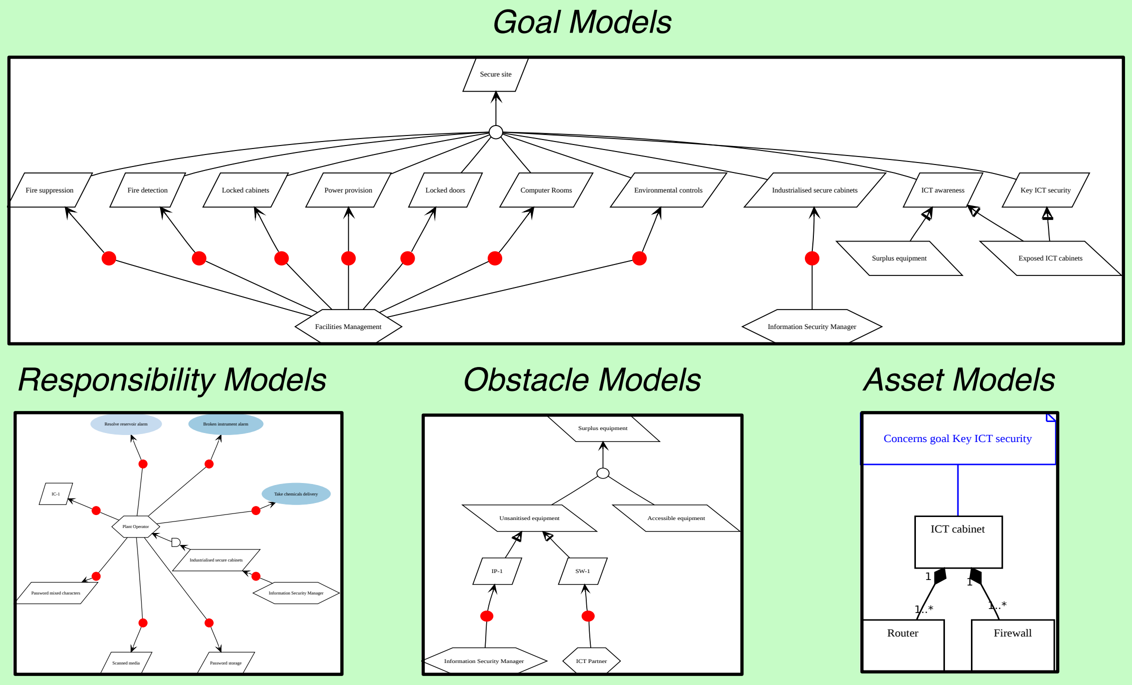
3. Using the results of the qualitative data analysis, a plant operator persona (Rick), and several task scenarios were elicited.


Copper theft
Thieves steal anything
Personal Security GT concept
Personal threats
Usually
Feel unsafe challenging intruders
Rick

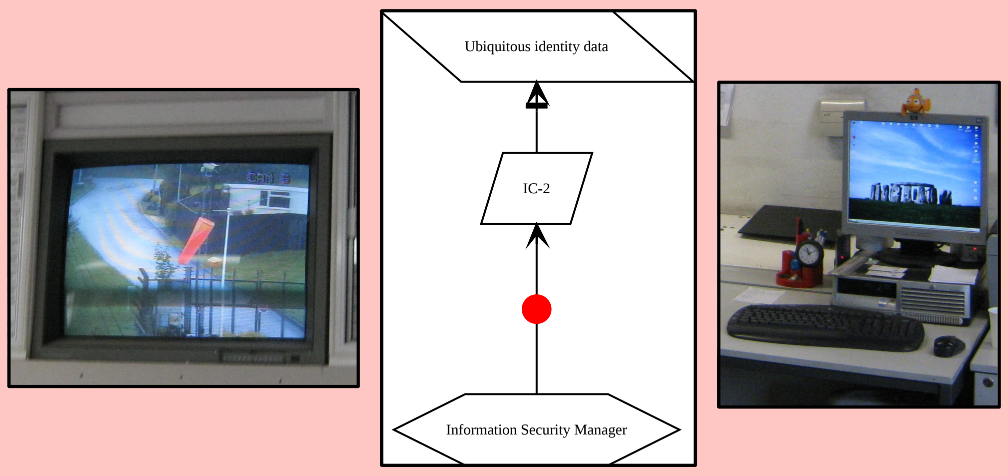
Although information security doesn't phase Rick too much, personal security does. Potentially facing off a scrap metal thief is a big worry for Rick.
"The police don't respond to intruder alarms at a nearby pumping station any more due to false alarms", says Rick.
"Because of this, we've been told not to go out to these places on our own.
We have a lone-worker system when people call us when we get to a particular station, but what happens if we get problems on the way?"

Motivation details about Rick, a plant operator persona.

4. Based on the collected data & documentation, 102 policy goals, 8 roles, and 18 assets. Based on obstructing policy goals alone, several vulnerabilities and threats were identified and mitigated.

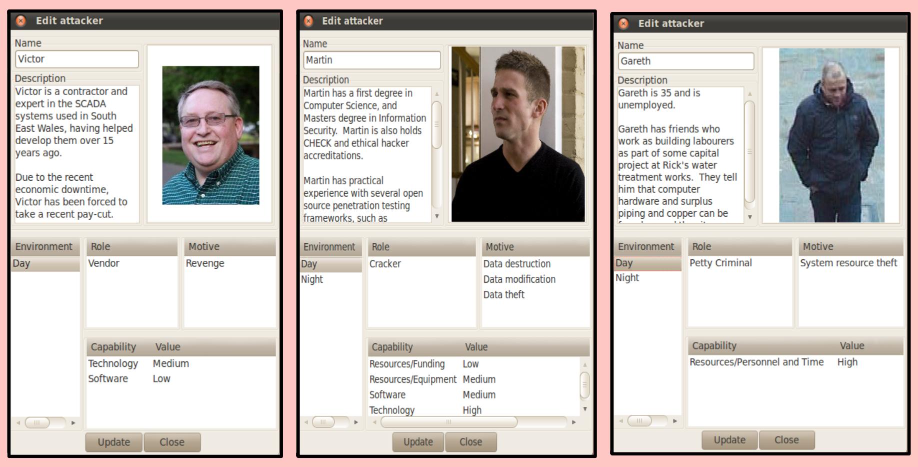


5. Based on the usability analysis data, 8 vulnerabilities were identified, 3 of which were mitigated at this stage.



Data contributing to ubiquitous identity vulnerability, and mitigating requirement.

6. Collected and open-source data helped identify 4 convincing attackers, and 8 possible threats.



Name: Victor
Description: Victor is a contractor and expert in the SCADA systems used in South East Wales, having helped develop them over 15 years ago.

Due to the recent economic downtime, Victor has been forced to take a recent pay-cut.

Environment	Role	Motive
Day	Vendor	Revenge

Capability	Value
Technology	Medium
Software	Low

Name: Martin
Description: Martin has a first degree in Computer Science, and Masters degree in Information Security. Martin is also holds CHECK and ethical hacker accreditations.

Martin has practical experience with several open source penetration testing frameworks, such as

Environment	Role	Motive
Day	Cracker	Data destruction Data modification Data theft
Night		

Capability	Value
Resources/Funding	Low
Resources/Equipment	Medium
Software	Medium
Technology	High

Name: Gareth
Description: Gareth is 35 and is unemployed.

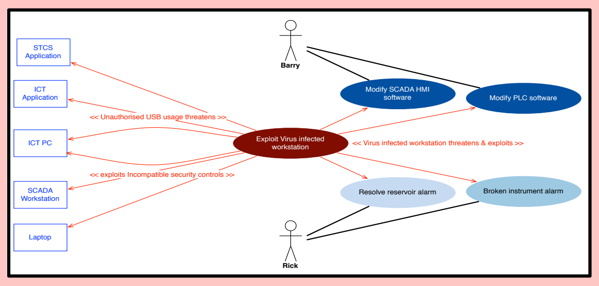
Gareth has friends who work as building labourers as part of some capital project at Rick's water treatment works. They tell him that computer hardware and surplus piping and copper can be

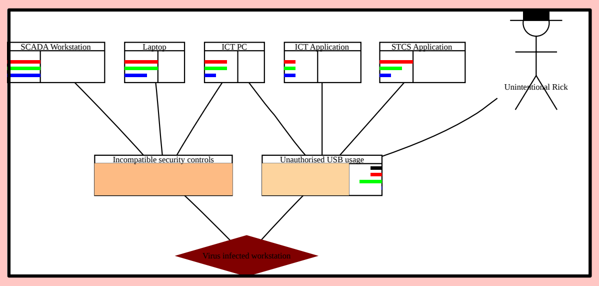
Environment	Role	Motive
Day	Petty Criminal	System resource theft
Night		

Capability	Value
Resources/Personnel and Time	High

Inside attacker (left), penetration tester (centre), and petty thief attacker profiles

7. Finally, the most topical risks were modelled as Misuse Cases, analysed, and mitigated in participatory design workshops.





Task (left) and Risk Analysis (right) model of virus-infected workstation risk

References

[1] Faily, S., and Fléchais, I. Barry is not the weakest link: Eliciting Secure System Requirements with Personas. *Proceedings of the 24th British HCI Group Annual Conference on People and Computers*, 2010, 113–120

[2] Faily, S., and Fléchais, I. The secret lives of assumptions: Developing and refining assumption personas for secure system design. *Proceedings of the 3rd Conference on Human-Centered Software Engineering* (2010),111–118

[3] Faily, S., and Fléchais, I. Towards tool-support for Usable Secure Requirements Engineering with CAIRIS. *International Journal of Secure Software Engineering* 1 (3), 2010, 56–70

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

[5] van Lamsweerde, A., Letier E., Handling Obstacles in Goal-Oriented Requirements Engineering. *IEEE Transactions on Software Engineering 26 (10)*, 2000, 978-1005

[6] Sindre, G., Opdahl L., Eliciting Security Requirements with Misuse Cases. *Requirements Engineering 10 (1)*, 2005, 34-44

Acknowledgements

This research was funded by the EPSRC CASE Studentship R07437/CN001.

We are also grateful to Qinetiq Ltd for their sponsorship of this work.