Security Requirements Engineering: When Anti-requirements Hit the Fan

Robert Crook* Darrel Ince Luncheng Lin Bashar Nuseibeh

Security Requirements Group
Department of Computing, The Open University
Walton Hall, Milton Keynes, MK7 6AA, UK

Email: Robert.Crook@t-online.de, {D.C.Ince, L.C.Lin, B.A.Nuseibeh}@open.ac.uk

Abstract

Everyone agrees that security is a problem, ranging from Microsoft to the banks that have been recent victims of rogue traders. What is paradoxical is that there does not seem to be a wholehearted commitment by both academics and industry to treat this topic systematically at the top level of requirements engineering. Our vision is of a future in which we inform the security requirements engineering process by organisational theory. This would act as the bridge between the well-ordered world of the software project informed by conventional requirements and the unexpected world of anti-requirements associated with the malicious user. We frame a vision for the requirements engineering community that would involve the community solving six difficult problems.

1. Information Security

The British Standards Institution [BSI199] defines information security as the protection of information assets from a wide range of threats. Information can be stored on many different forms of medium but, increasingly, the storage of information on computers has become the most critical resource for most businesses. Information in whatever form needs to be appropriately protected. This means maintaining confidentiality, integrity, and availability

Confidentiality is concerned with maintaining privacy and secrecy, allowing read access to only those users who have been authorised. Integrity is about ensuring the accuracy and completeness of information. Maintaining integrity involves allowing only authorised users to change or create data and applying controls to ensure the correctness of the data. Availability is concerned with ensuring that access to information systems is maintained when required.

2. Threats and anti–requirements

An *anti-requirement* is a requirement of a malicious user that subverts an existing requirement. The key here is that they are generated by the *malicious* user – a typical requirements engineering process will often generate conflicting or inconsistent requirements that have nothing to do with security. These can be generated from either a front-end threat analysis or from a post-hoc reaction to an operational attack. Building on existing work in goal-oriented requirements engineering, some researchers have started to tackle this problem [Chun93, JF01, Yu00]. Also, abuse cases [MF99] and misuse cases [SO01] have demonstrated how one can make explicit, and counteract, threatening scenarios. This literature has generated a challenging research agenda and highlighted our view that existing requirements engineering methods can go some way towards alleviating security problems. However, we would contend that a number of significant difficulties remain. Our vision here is of these difficulties being overcome. In order to start the process of demolishing these difficulties, we have elucidated a number of images that form part of our vision. These are described in the last part of our paper. A significant problem we face is that the *security policy* often never appears, explicitly or even implicitly.

3. Security policy and why it is often ignored

The literature does not lack examples of security policies [BL73, CW87, McClean94, HS97, SFK00]. Security researchers have developed a continuum of approaches ranging from early multi-models, such as Bell La Padula [BL73], to later work exemplified by role-based access control models such as [SFK00]. However, these derive from the solution world rather than from the problem

* Also an independent consultant

world and do not encourage a systematic exploration of security policy issues. Current industrial practice ranges from, at best, a process of adding requirements at a late stage in the development of system, to, at worst, coding security controls during implementation.

Why does this happen? There is the obvious reason that project teams—particularly those developing distributed systems—are under huge pressure to deliver functionality. However, we contend that something subtler is working away: that although researchers are beginning to address security policy goals, for example [AE01], conventional requirements modelling is inadequate to represent the organisational procedures that underpin a security policy.

A good example of this occurred when a bank in Hastings, England, did not audit address changes. A clerk changed her address and issued an ATM card and a PIN, changed the address back and withdrew a large amount of money [And01]. This is an example where the organisational procedures were inadequate for future threats. Our long-term aim is to integrate organisational security issues into the requirements engineering process in a systematic way.

4. Where is the organisation?

Organisations can take many forms, ranging from very simple structures such as those found in small start-ups, to the complex divisionalised structures found in multi-national corporations. Although no two organisations are exactly the same, neither are they truly unique. There are common aspects that enable us to categorise their structure and generalise.

An organisational structure is designed by top-level management and defines the lines of authority and division of work. These are the two principle characteristics that determine the responsibilities of each member of the organisation. The starting point for defining security requirements is the identification of actions and operations that human users carry out. For each of these actions, it is usual to define the agent or actor who carries it out. This can be used as the basis for defining a role. The crux of any security policy is the restriction of access to information assets. *Role-based access control* provides a flexible way of achieving this and different types of roles and role hierarchies have been proposed [ML99]. Unfortunately, roles often seem to be conjured out of thin air. In the future, we need to be able to define the notion of 'role', understandable in a security context, in order to address the organisation-based problems that we alluded to above. We remain to be convinced that current requirements engineering methods do this.

5. Breaking out of compartments

We firmly believe that it will be necessary to conduct research that cuts across the traditional boundaries that usually circumscribe academic subject areas. It will be necessary to draw together important contributions from disciplines of computer security, requirements engineering, organisational behaviour, and software applications engineering, to get to grips with the complexity that security poses if it is to be tackled in a systematic way.

6. Where we go from here: the vision thing

Our vision for the future of security requirements engineering encompasses the following images:

- Thomas *et al.* [TS94] describe a taxonomy of security policies that is a good illustration of how security policies can be tackled at different levels of abstraction. These levels are separated into: (i) organisational requirements, (ii) computer policy models, (iii) access control models, and (iv) implementation models. At the highest level, the main concern is the organisational structure, responsibilities and procedures. The next level concerns how the organisation interfaces to the computer system designated by the computer policy model and implemented by an access control model representing controls internal to the system. And finally, how this maps on to an implementation model. *Our vision is that the requirements engineering community will have devised a framework into which these are all integrated.*
- We have already stressed the importance of organisation. *Our vision is of a requirements engineering community that will draw upon the wealth of research carried out by organisational behaviourists over the last thirty years*[1].
- Security is a multi-faceted topic concerned with multi-level security, discretionary access control, separation of duties, delegation, roles, groups, generalisation hierarchies, and supervisory hierarchies. *Our vision is that we will be able to model these concepts and integrate them into the requirements engineering process*.
- There is a close relationship between non-technical and technical security policies, and an important distinction between organisational procedures and technical security mechanisms. *Our vision encompasses the requirements engineering community identifying the*

[1] Some would even say that this work extends further into the past, for example the work of the early sociologist Weber; and, in particular, his landmark research on offices and bureaucracies.

criteria for separating them and providing advances based on a full appreciation of the synergistic relationship between them.

- Not all our challenges are top-down. There is a need for an important bottom-up view of security requirements engineering. *Our vision includes the development of a framework to examine systematically requirements and anti-requirements building on preliminary work, such as that by van Lamsweerde combining goal and obstacle analysis* [VL00].
- There are a number of policy description languages, for example Ponder [DDLS00], that have been used for low-level specifications of distributed system management. Moffett [Moff99] has posited that these might be used for high-level policy specification within requirements engineering. *We would hope that, in the future, the requirements engineering and security policy communities join together to investigate the scope and applicability of these languages and examine how organisational requirements can be captured by them.*

Acknowledgements

We would like to acknowledge the advice and help of our colleagues in the security requirements group at The Open University. Particular thanks are due to Michael Jackson for feedback during earlier discussions, and to Jonathan Moffett for feedback on an earlier version of the paper.

References

[AE01] A. I. Anton and J. B. Earp, "Strategies for Developing Policies and Requirements for Secure Electronic Commerce Systems", *Recent Advances in Secure and Private E-Commerce*. Kluwer, 2001.

[AEPA01] A. I. Anton, J. B., Earp, C. Potts and T. A. Alspaugh, "The Role of Policy and Stakeholder Privacy Values in Requirements Engineering", Proc. *Symposium on Requirements Engineering for Information Security*, 2001.

[Alex02] I. Alexander, "Misuse Cases", *Proceedings of International Requirements Engineering Conference (RE'02)*, Germany, 2002.

[And01] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley, 2001

[BL73] D. Bell and L. J. La Padula, "Secure Computer Systems: A Mathematical Model", *MITRE Technical Report 2547*,Volume II. 1973.

[BSI199] BS799-1:1999 *Information Security Management - Part 1: Code of Practice for Information Security*. British Standards Institution, London. 1999.

[CW87]D. D. Clark and D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies", *Proc. of the 1987 IEEE Symposium on Security and Privacy*, 1987.

[Chun93] L. Chung, "Dealing with Security Requirements During the Development of Information Systems", In *Proc. CAiSE '93, 5th Int. Conf. Advanced Information Systems Engineering*, Colette Rolland, Francois Bodart, Corine Cauvet (Eds.). Springer, 1993.

[DDLS00] N. Damianou, N. Dulay, E. Lupu and M. Sloman, "Ponder, A Language for specifying Management and Security Policies for Distributed Systems*", Imperial College Research Report DoC2001*, Jan 2001.

[HS97] T.F. Himdi and R.S. Sandhu, "Lattice-based Models for Controlled Sharing of Confidential Information in the Saudi Hajj system", *Proceedings 13th IEEE Annual Computer Security Applications Conference*. 1997.

[JF01] Pierre Jean-Fontaine Goal Oriented Elaboration of Security Requirements. *Project Dissertation*. Universite Catholique de Louvain, Belgium, 2001.

[McClean94] J. McLean, "Security Models", *Encyclopaedia of Software Engineering*. Wiley, 1994.

[MF99] J. McDermott and C. Fox, "Using Abuse Case Models for Security Requirements Analysis", *Proceedings 15th IEEE Annual Computer Security Applications Conference*. 1999.

[ML99] J. D. Moffett, E. C. Lupu The Uses of Role Hierarchies in Access Control. *Proceedings of the 4th ACM workshop on Role-based Access*, 1999.

[Moff99] J. D. Moffett, "Requirements and Policies", *Proc. Policy Workshop* HP-Laboratories Bristol, UK, 1999.

[SCFY96] R. Sandhu, E. Coyne, H. Feinstaein and C. Youmann, "Role Based Access Control Models" *IEEE Computer*, 29(2). 1996.

[SO01] G. Sindre, and A. L. Opdahl, „Templates for Misuse Case Description", *Proceedings of the 7th International Workshop on Requirements Engineering, Foundation for Software Quality (REFSQ'2001)*, Switzerland, 4-5 June 2001.

[TS94] R. K. Thomas and R. S. Sandhu, "Conceptual Foundations for a model of Task-based Authorizations", *Proc. IEEE Computer Security Foundations Workshop VII*, 1994.

[VL00] A. van Lamsweerde and E. Letier, "Handling Obstacles in Goal-oriented Requirements Engineering", *IEEE Transactions on Software Engineering*, 26(10). 2000.

[Yu00] E. Yu and L. Liu, "Modelling Trust in the i* Strategic Actors Framework", *Proc. of the 3rd Workshop on Deception, Fraud and Trust in Agent Societies*, Barcelona, 3-4 June 2000.