Patterns of Information Security Postures for Socio-Technical Systems and Systems-of-Systems

Tim Storer, Karen Renaud
School of Computing Science
University of Glasgow
Glasgow, Scotland, G12 8QQ

William Bradley Glisson
Digital Forensics Laboratory
University of Glasgow
Glasgow, Scotland, G12 8QQ

***Abstract*—This paper describes a proposal to develop patterns of security postures for computer based *socio-technical* systems and systems-of-systems. Such systems typically span many organisational boundaries, integrating multiple computer systems, infrastructures and organisational processes. The paper describes the motivation for the proposed work, and our approach to the development, specification, integration and validation of security patterns for socio-technical and system-of-system scale systems.**

:

I. Introduction

The concept of a software design pattern, a tailorable solution to a recurring design problem, was first popularised by Gamma et al. (1994). These patterns addressed small scale, object oriented, software design problems, typically involving the arrangement of a small number of software classes. Design patterns have also been applied at other scales in software and systems engineering, including software architectures (Shaw and Garlan (1996) is one of many texts on the topic) and the analysis of patterns of work (Martin and Sommerville, 2004).

There has been considerable interest in the application of patterns to a range of information security domains. Security patterns have been proposed in contexts such as distributed access control (Erber et al., 2007), operating systems (Fernandez et al., 2006), and web application development (Kienzle and Elder, 2001). There have also been surveys of the state of the art, Yoshioka et al. (2008) for example, and a book on development with security patterns (Schumacher et al., 2005).

This existing research has addressed the application of a patterns approach to information security design problems at the software system scale. However, as Siponen and Heikka (2008) also noted, there is currently a lack of established engineering practice, tools or methods that address information security design problems at the socio-technical system and system-of-system scale. Such systems are based around information technology (IT) applications and infrastructures, and can be characterised as:

- very large scale, consisting of multiple heterogeneous organisations, work flows, actors, software applications, middlewares and hardware platforms; and
- very long lived, with evolutionary development occurring across organisational boundaries as the system context changes, users change, individual components are updated and new requirements are established.

The challenges of addressing general engineering problems for systems at these scales is well reported in the literature (Sommerville et al., 2011; Boardman and Sauser, 2006). Diverse examples of such systems include inter-agency civil emergency response (Sommerville et al., 2009a), logistics management (Baskerville and Siponen, 2002), election and voting management systems (Lock et al., 2008), space based infrastructures (Boin and Fischbacher-Smith, 2011) and electronic trading systems (Sommerville et al., 2011). The adoption of virtualised, remote and distributed cloud computing infrastructures will exacerbate the complexity of engineering at this scale.

It is recognised that such systems are vulnerable to information security threats that are both evolving and increasing in scale and complexity. Future 'cyber' attacks will be directed at multiple systems, infrastructures, organisations and nation states simultaneously (Research Councils UK, 2011; UK Office of Cyber Security and UK Cyber Security Operations Centre, 2009). Threats will combine both external activity and malicious 'insider' behaviour. In the future, systems that enable and support collaborative information security across organisational boundaries will be essential. To respond to this challenge, there is a need to develop engineering methods that address the design of security into systems at the socio-technical and system-of-systems scale.

To begin to address this challenge, this paper argues for the development of:

- methods for the identification, documentation and validation of *design patterns of information security postures for socio-technical systems and systems of systems*;
- an initial repository of validated design patterns which can be applied to recurring security problems at the socio-technical and system-of-systems scale;
- a method for selecting and tailoring a design pattern to a specific system context; and
- methods for evaluating the suitability of a proposed socio-technical security design for a specific context through modelling and simulation.

The work proposed will adapt the well established pattern approach to system design to the socio-technical and system-of-system scale. The pattern repository will provide a source of guidance for engineers seeking to address large scale

information security design problems. The application method will allow design patterns to be selected and tailored to specific socio-technical contexts. This approach will allow existing craft and experience to be formalised in engineering methods and tools.

This paper describes the initial results of this work, and is structured as follows. Section II illustrates our approach to documenting information security design patterns for socio-technical systems. The patterns approach is illustrated using a detailed example of a security incident response team in an organisation. Section III briefly enumerates other patterns that we have identified, illustrating the breadth of applicability of the approach. Finally, Section IV summaries the work and points to future directions. In particular, we describe future work on the the validation of socio-technical system designs through the use of multi-agent simulation.

II. Description of Method

Following the patterns approach, a Socio-Technical Information Security Design Pattern will include:

- **The context of an socio-technical security problem, explaining the circumstances in which the pattern can be applied.** The context section of the pattern describes the configuration of organisational vulnerabilities, known threats and attacker capabilities that can be addressed or mitigated through the application of the pattern. This part of the pattern is equivalent to a *threat model* that is typically described before a security design is proposed to address it.
- **A template for a socio-technical solution combining information systems, human actors and processes.** Specifically, the template solution will describe the roles and responsibilities of relevant actors within the organisation, and the communication channels and information requirements for the actors to discharge their responsibilities.

  There are several notations for modelling socio-technical systems, notably: goal based notations, such as TROPOS (Giorgini et al., 2005); and the SysUML (Hause, 2006). There is also the STAMP method for identifying systemic hazards in systems of systems (Leveson, 2004). However, we have found the notion of *responsibility* useful in analysing and describing socio-technical systems in terms of the actors in an organisation, the responsibilities they hold and discharge and the information they require and produce (Sommerville et al., 2009b).

  Lock and Sommerville (2010) demonstrated the feasibility of using an extension of the responsibility notation to capture key aspects of systems-of-systems. Consequently, we anticipate that the graphical notation (Sommerville et al., 2009a) we have previously developed for modelling the socio-technical and system-of-system contexts will be appropriate for describing socio-technical security pattern templates, in a similar way to the specification of software design patterns in UML.
- **The trade-offs, including any disadvantages, that must be considered when applying the pattern.** This part of the pattern will be developed through analysis of previous case studies and work with industrial partners. We will identify both the benefits of applying the pattern for information security at the socio-technical level, as well as any risks that are introduced.
- **Any related socio-technical and/or software security patterns.** Many design pattern schemes list relations within the same family of patterns, so that, for example, patterns which complement each other can be identified. In this section, we will identify related socio-technical information security patterns that complement one another. In addition, we will identify *software* security patterns, already documented in the existing literature, that complement a socio-technical pattern. This approach will support an integrated socio-technical approach to system security, allowing the consequences of security design decisions at different levels of abstraction and scale to be evaluated.

We briefly sketch an example socio-technical security pattern to illustrate our proposed approach. Figure 1 illustrates the pattern for a *Security Incident Response Team* (SIRT).

The context describes the threat model in which the pattern is applicable: when an organisation perceives some information security breaches as inevitable due to a large, distributed, heterogeneous infrastructure or rapidly changing insider capabilities, which make a secure system difficult to achieve. In addition, the pattern is applicable when it may be cheaper to respond to and mitigate attacks, rather than attempting to prevent them occurring at all.

The solution template responsibility model contains five actor roles, denoted using the UML stick figure stereotype. The **Incident coordinator**, responsible for collecting incident reports and initiating investigations. **Employee** represents a general member of the organisation. The **Automated monitor** is a software component that can be configured to report suspicious system activity, such as inappropriate file accesses, failed login attempts or firewall probes. The **Incident manager**, responsible for directing the focus of a specific investigation. The **Incident investigator**, responsible for gathering and analysing evidence under the direction of the investigation manager.

Note that the model describes actor roles, rather than specific actors. Consequently, the pattern is applicable to both large organisations, where the role of incident coordinator, manager and investigator may be distinct, and smaller organisations, where all these roles may be undertaken by a single IT worker.

In addition, actor roles may be undertaken by humans, organisations or technical components. In the responsibility modelling notation, any object can be modelled as an actor if the modeller perceives the actors to have *intent* and be able to hold a responsibility. This flexibility makes responsibility modelling useful for capturing different scales of socio-technical system.

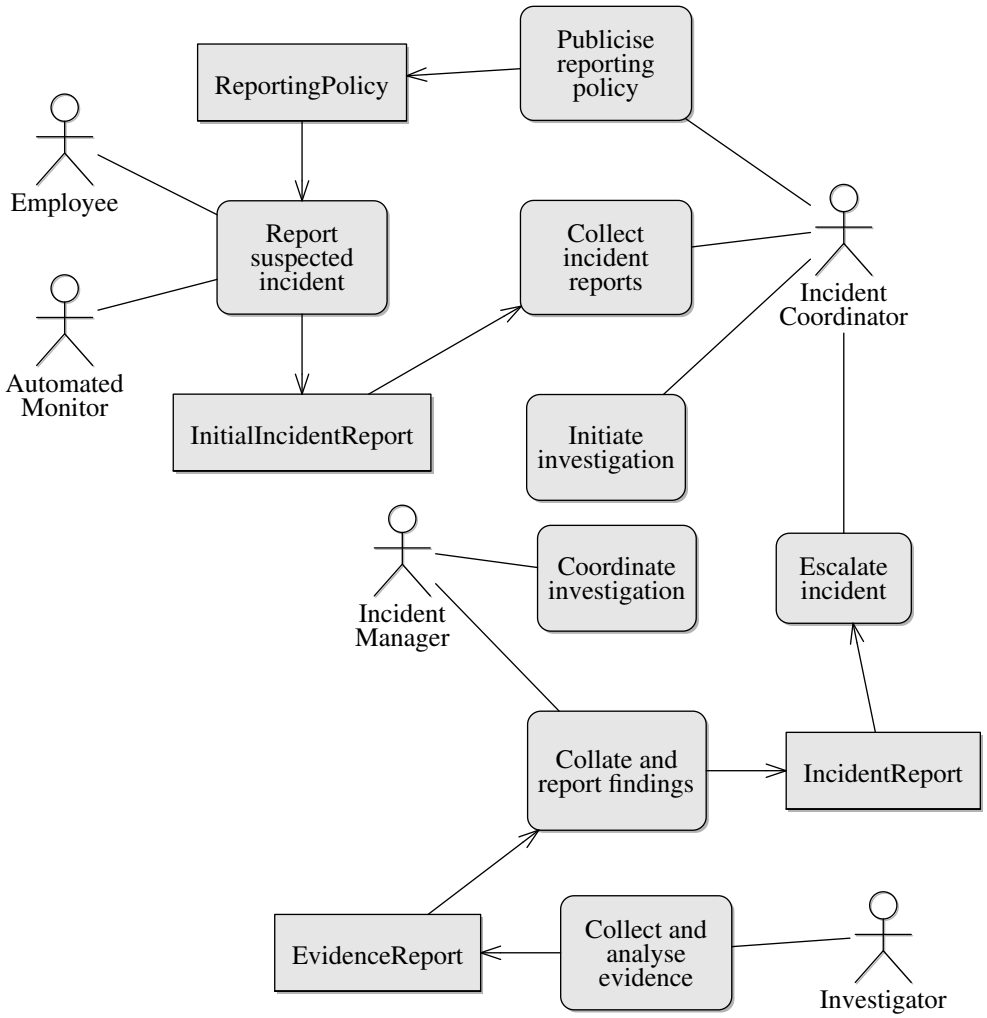
Title: Security Incident Response Team (SIRT)	
Problem Context	An organisation maintains a diverse collection of interconnected systems and infrastructures, used by many different actors with rapidly changing privileges. Attackers may be either external, or internal, exploiting one or more vulnerabilities in different parts of the organisation's systems. It is not considered feasible to prevent every security incident, but the extent of damage caused by attacks on the systems is often dependent on the time taken to respond to the attack.
Solution Template	
Advantages and disadvantages	The pattern provides the organisation with a means of identifying and responding rapidly to evolving security incidents. There is a risk that the system will generate too many false positives (particularly in the aftermath of a breach) swamping the organisation's ability to respond effectively.
Related patterns	Security policy management team; Secure logging framework;Incident reporting information system

Fig. 1. An example socio-technical security pattern.

The model illustrates the responsibilities held by each of the roles, and the information that is required or produced by them. Responsibilities are denoted in lozenges, while information resources are represented as UML classes (it is possible to specify class attributes if desired). An agent holding a responsibility is denoted using a UML association dependency. The UML transition dependency is used to denote the production or requirement of information to discharge a responsibility.

The Incident Coordinator is responsible for publicising the incident reporting procedure and the responsibility of employees to report incidents. The model shows that an employee is responsible for reporting suspicious events for further investigation. The Incident Coordinator will undertake an initial assessment and initiate a further investigation if deemed necessary. Part of this responsibility is to appoint a manager for the incident, responsible for coordinating investigation efforts. The incident manager will direct the evidence gathering efforts of investigators and collate them into a full incident report. Finally, the Incident Coordinator is responsible for deciding whether to escalate an incident or not.

The pattern explains both the advantages of disadvantages of the SIRT model. In particular, it is noted that the system may generate too many false positive reports, leading to resources being consumed investigating non-incidents. This may be a particular problem where poorly trained employees are required to report incidents, or where automated monitoring software is not correctly configured. There may also be variability in incident reports, particularly in the aftermath of a reported breach.

As can be seem from the figure, socio-technical security patterns modelled on responsibilities can also be used to denote the information that is required and produced through the discharge of a responsibility. This documentation can then be used to identify the requirements for the supporting technical and administrative of an organisation. These requirements can sometimes be expressed as related design patterns (both technical and socio-technical).

In the pattern described in Figure 1, for example:

- The Employee and the Automated Monitor needs a means of communicating an incident to the Incident Coordinator in a standardised way. A standard *layered* information system for collecting reports should support this requirement by providing a software interface to the automated monitor and a user interface to the employee.
- The incident manager and investigators need information concerning the state of parts of the technical infrastructure in the lead up to an incident. The socio-technical security pattern thus drives the need for a secure logging framework for the technical infrastructure.
- There needs to be a means of ensuring that employees of the organisation are familiar with their responsibilities to report incidents and the procedure for doing so. Consequently, the security policy management team pattern is recorded as a related pattern. This pattern describes how the organisation manages and communicates security policies to its employees.

The example illustrates how socio-technical security patterns can provide an over-arching and integrating framework for software level security patterns.

III. Other Patterns

This section of the paper provides a brief catalogue of further (elaborated) socio-technical security design patterns that we have identified at the time of writing.

- Security policy management team, including a security policy coordinator for the organisation.
- Inter–organisational response team, providing a means of communicating security incidents between organisations within a trusted environment.
- Inter-organisational ad-hoc secure channel, inspired by the security breaches (and remedies) encountered during data transmission between the United Kingdom's Revenue and Custom and National Audit Office (Poynter, 2008). The pattern provides a means for secure communication when relationships between agencies are ad-hoc and intermittent.
- Credential and authorisation manager, provides a single point of contact within an organisation for registering new users, removing privileges of departing employees and managing authorisations as roles change.
- Software patch management team, providing for a disciplined approach to evolving software configurations in a large-scale heterogeneous Information Infrastructure.
- Off-site fail-over, prevents the organisation from failing should the main IT infrastructure site be damaged due to man-made or natural disaster.
- Intra–organisational firewall, allows an organisation to partition information internally to prevent conflicts of interest arising.
- Secure decommissioning process for data storage devices, ensuring they are disposed of without compromising organisational data security.

Space considerations prevent a detailed description of these patterns. However, the examples listed (and others) hints at the breadth of the applicability of the approach.

IV. Summary and Future Work

The global information rich society is increasingly dependent on large scale computer based systems and infrastructures. These systems are increasingly threatened by threats capable of mounting attacks of increasing sophistication and scale. We believe that the development of patterns of socio-technical security postures addresses these security threats at an equivalent scale. This paper has outlined our envisioned approach, including the use of responsibility modelling as a core abstraction.

A key challenge at this scale is the validation of designs prior to their deployment. Mistakes in system design can be much more expensive to correct once implementation has begun. In the current state of the art, validation of the design of large scale, socio-technical systems is notoriously difficult. A trial and error approach, until an acceptable design is achieved,

is often employed. This can often result in system failures during initial deployments as problems are gradually rectified. Alternatively, systems may fail catastrophically and projects abandoned if socio-technical failures cause rapid declines in confidence in a system.

We propose to use the development of executable simulations of large scale systems as a means of validating socio-technical security patterns. We are investigating the use of multi-agent technologies that incorporate the responsibility modelling technique for capturing socio-technical structures. A repository of patterns will provide a re-usable collection of designs that can be applied to a multi-agent simulation of a socio-technical system, prior to deployment. Our vision is that the validity of applying a given socio-technical security pattern to a design problem can be tested first through simulation before a decision to commit to a substantial re-organisation is made. This paper has described the first steps in this direction.

REFERENCES

E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns : elements of reusable object-oriented software*, 1st ed., ser. Professional Computing Series. Addison Wesley, October 1994.

M. Shaw and D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall, 1996.

D. Martin and I. Sommerville, "Patterns of cooperative interaction: Linking ethnomethodology and design," *ACM Transactions on Computer-Human Interaction*, vol. 11, no. 1, pp. 59–89, 2004.

R. Erber, C. Schläger, and G. Pernul, "Patterns for authentication and authorisation infrastructures," in *1st International Workshop on Secure Systems Methodologies using Patterns (SPattern'07)*, Regensburg, Germany, 2007.

E. B. Fernandez, T. Sorgente, and M. M. Larrondo-Petrie, "Even more patterns for secure operating systems," in *Pattern Languages of Programs (PLoP)*, Portland, Oregon, 2006.

D. M. Kienzle and M. C. Elder, "Final technical report: Security patterns for web application development," DARPA, Tech. Rep., 2001.

N. Yoshioka, H. Washizaki, and K. Maruyama, "A survey of security patterns," *Progress in informatics*, vol. 5, pp. 35–47, 2008.

M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns: Integrating Security and Systems Engineering*, ser. Software Design Patterns. The Atrium, Southern Gate, Chicester, West Sussex, PO19 8SQ, United Kingdom: John Wiley & Sons, December 2005.

M. Siponen and J. Heikka, "Do secure information system design methods provide adequate modeling support?" *Information and Software Technology*, vol. 50, no. 9–10, pp. 1035–1053, 2008.

I. Sommerville, D. Cliff, R. Calinescu, J. Keen, T. Kelly, M. Kwiatkowska, J. McDermid, and R. Paige, "Large-scale complex systems engineering," 2011, submitted to Communications of the ACM.

J. Boardman and B. Sauser, "System of systems - the meaning of of," in *Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering*. Los Angeles, California, USA: IEEE, April 2006, pp. 118–123.

I. Sommerville, R. Lock, T. Storer, and J. Dobson, "Deriving information requirements from responsibility models," in *Advanced Information Systems Engineering, 21st International Conference, CAiSE 2009*, ser. Lecture Notes in Computer Science, P. V. Eck, J. Gordijn, and R. Wieringa, Eds., vol. 5565. Amsterdam, Netherlands: Springer Verlag, June 2009, pp. 515–529. [Online]. Available: http://www.dcs.gla.ac.uk/~tws/research/papers/sommerville08deriving-r1447.pdf

R. Baskerville and M. Siponen, "An information meta-policy for emergent organisations," *Logistics Information Management*, vol. 15, no. 5, pp. 337–346, 2002.

R. Lock, T. Storer, N. Harvey, C. Hughes, and I. Sommerville, "Observations of the Scottish elections 2007," *Transforming Government: People, Process and Policy*, vol. 2, no. 2, pp. 104–118, 2008.

A. Boin and D. Fischbacher-Smith, "The importance of failure theories in assessing crisis management: The columbia space shuttle disaster revisited," *Policy and Society*, vol. 30, no. 2, pp. 77–87, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1449403511000105

Research Councils UK, "An RCUK green paper for cybersecurity research," 2011.

UK Office of Cyber Security and UK Cyber Security Operations Centre, "Cyber security strategy of the united kingdom safety, security and resilience in cyber space," June 2009.

P. Giorgini, F. Massacci, and N. Zannone, "Security and trust requirements engineering," in *Foundations of Security Analysis and Design III, FOSAD 2004/2005 Tutorial Lectures*, ser. Lecture Notes in Computer Science, A. Aldini, R. Gorrieri, and F. Martinelli, Eds. Springer Verlag, 2005, vol. 3655, pp. 237–272.

M. Hause, "Cross-cutting concerns and ergonomic profiling using UML/SysML," in *INCOSE International Symposium on Systems Engineering*, 2006.

N. G. Leveson, "A systems-theoretic approach to safety in software-intensive systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 66–86, January 2004.

I. Sommerville, T. Storer, and R. Lock, "Responsibility modelling for civil emergency planning," *Risk Management*, vol. 11, no. 3-4, pp. 179–207, 2009.

R. Lock and I. Sommerville, "Modelling and analysis of socio-technical system of systems," in *10th International Conference on Engineering Complex Computer Systems*. Oxford, UK.: IEEE Computer Society, March 2010, pp. –.

K. Poynter, "Review of information security at hm revenue and customs final report," HM Stationary Office, St Clements House 2–16 Colegate, Norwich, NR3 1BQ, June 2008.