To boldly go where invention isn't secure: applying security entrepreneurship to secure systems design.

FAILY, S. and FLÉCHAIS, I.

2010

To Boldly Go Where Invention Isn't Secure: Applying Security Entrepreneurship to Secure Systems Design

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

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

ABSTRACT

When designing secure systems, we are inundated with an eclectic mix of security and non-security requirements; this makes predicting a successful outcome from the universe of possible security design decisions a difficult problem. We propose augmenting the process of security design with the paradigm of Security Entrepreneurship: the application of innovation models and principles to organise, create, and manage security design elements to bring about improved system security. We propose three initial Security Entrepreneurship techniques as examples of this paradigm, describe how their underlying models align with secure systems design, and help predict the social and technical impact of possible design decisions. We also pose a number of thought experiments, and suggest possible research agendas for Security Entrepreneurship.

Keywords

Security, Innovation, Entrepreneurship

General Terms

Design

Categories and Subject Descriptors

K.6.0 [**General**]: Economics

1. INTRODUCTION

Designing a secure system is hard. Irrespective of the methodology or process one uses to understand a problem domain and elicit a set of requirements, sooner or later certain requirements will arise which may not have obvious solutions. Eliciting solutions to such problems requires a mixture of creativity and knowledge, and the results can be difficult to predict. For example, ISO 27002 provides a useful catalogue for possible security controls which deal with different problems [4], but what appears to be a reasonable security control in one context, may fall foul of a variety of problems in another.

The exercise of designing a secure system shares many of the characteristics of solving a *wicked problem* [48]. A wicked problem can be characterised in security by the lack of clarity in what it means to secure the system, the lack of an immediate test proving that a system is secure, and the large number of possible controls which are potential solutions to a specified security problem. Creative thinking helps identify new ideas which bridge the problem/solution chasm. However, despite the claims of security vendors purporting to sell "innovative security solutions", successful innovation does not automatically follow from a creative idea.

It might seem inventive ideas are not useful for deriving successful, innovative security designs from requirements, but we know this to be otherwise. After all, we need only look around us to find examples of people who find practical solutions to wicked problems all the time. One class of exemplars are technology entrepreneurs, successful examples of which include the founders of Microsoft and Google. In some sense, force of character has an important role to play in this success, but so does an ability to innovate. Successes and failures in technology innovation have been the subject of a growing body of research which attempts to synthesise the reasons why some inventions succeeded in the face of overwhelming odds, while others have failed in cases where both good ideas and the capability to commercialise them were present. Many of these insights have been incorporated into models of innovation, which are now mainstays in MBA courses at many leading business schools. Consequently, a new generation of technology entrepreneurs now use these models on a day-to-day basis to identify opportunities with the potential to be "the next big thing".

Many of the ideas used to develop technology innovations have also been used by Social Entrepreneurs—entrepreneurs who have used the same principles to develop innovations that address social and environmental issues, many of which affect the developing world. Given the "wickedness" of problems faced in these contexts, we argue that many security design problems can also be tackled by taking an entrepreneurial mind-set. Such an mind-set is emancipatory; by adopting it, we treat security as an opportunity rather than a constraint, and the user community as a social-network, rather than an unpredictable, extended attack surface. Like technology and social entrepreneurs, entrepreneurial designers break down barriers and leverage resources at all levels to design creative and workable security solutions. We do not purport that following this approach automatically

solves design problems, but they do provide alternative perspectives from which to ask informed questions and identify design criteria that may not have been otherwise considered.

This paper presents the paradigm of Security Entrepreneurship: the application of innovation models and principles to organise, create, and manage security design elements to bring about improved system security. In section 2, we introduce some tenets from the innovation literature, before introducing three sample Security Entrepreneurship techniques in section 4 and illustrating these with a working example. Finally, in section 5, we discuss the consequences of this paradigm, and propose research directions for the mainstream introduction of Security Entrepreneurship for security design.

2. INNOVATION AND SYSTEM BUILDING

2.1 Creativity and design

The idea of informing the design process with ideas from the creativity literature is not new. In particular, the Requirements Engineering community has used these ideas to "invent" requirements, which may have otherwise remained hidden. Karlsen et al. [38] describe how a creativity support tool [36] was used to generate ideas by seeding the tool with key words and phrases from a security scenario; the tool generates associations between these seeds and related words and images. This resulting imagery stimulates discussion, which can lead to inventive requirements.

Some Requirements Engineering researchers go a stage further, arguing that the discipline itself can be a direct driver for innovation. Robertson asserts that analysts should invent requirements rather than expecting customers to ask for them; by understanding customer values, abstracting ideas from known good ideas, and thinking laterally, we can come up with innovations as powerful as Post-It notes, eBay, and peer-to-peer networking [49].

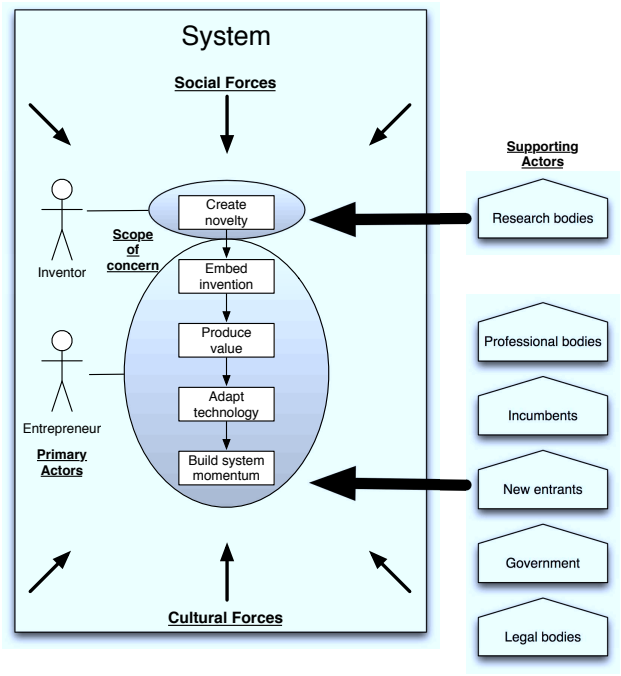
2.2 Entrepreneurs as system builders

Robertson also suggests that innovation can be achieved by abstracting away the elements of a problem, and tracing the origin of the need giving rise to the problem. Understanding this need is the tinder which, when the spark of invention is added, fires up an innovation. Unfortunately, scholarly research in innovation theory indicates that *envisioning* an invention is not enough to *precipitate* an innovation.

The traditional view of innovation, e.g. [28] prescribes a linear process, where an innovation is systematically derived from an invention via a process of applied research. This process is akin to uncovering nature, formulating creative abstractions based on these observations, and devising technology realising these abstractions. Seminal work by Hughes [35], who carried out a comparative study on the electrification of different European countries in the late 19th century, suggests other factors play a part in innovation as well.

Hughes identified several phases associated with innovative system building; only one of these is the creation of the invention itself. The other phases are concerned with (i) embedding the invention into its system, (ii) producing value from the emergent technology, (iii) adapting the technology based on different external forces, such as the market and other industries, and (iv) maintaining system momentum.

Figure 1: The Innovation Ecosystem



The system building process, illustrated in figure 1, is also influenced upon by internal social and cultural forces.

This model suggests the inventor's scope is comparatively small in the big picture. Much of the job of system building falls on the shoulders of entrepreneurs.

The role of the entrepreneur was first described by Richard Cantillon in the late 17th century. Cantillon described entrepreneurs as catalysts to production and exchange, and the *highly visible hand* – as opposed to Adam Smith's invocation of the *invisible hand* – that ensures that markets work [43]. Entrepreneurs have been characterised as forces creating paradoxical "gales of creative destruction" [51]; although their work is creative, unleashing this creativity can be both generative and disruptive to a system. Martin & Osberg [41] describe some of the characteristics of an entrepreneur:

- They are inspired to alter an unpleasant equilibrium.
- They need the ability to think creatively, and develop new solutions which may break from existing thinking.
- They must be prepared to take direct action to develop a new innovation.
- They need the ability to influence others.

Like Robertson's notion of the inventive requirements engineer, the entrepreneur will envision the possibilities needed to realise an innovation. However, the entrepreneur will also marshal the necessary technological, market, and organisational capabilities to make the innovation happen.

2.3 Models of Innovation

To better understand how innovations develop, and what characteristics are necessary for innovations to survive in

constantly changing organisational and market contexts, many different models of innovation have been developed. Each model has attempted to reason about innovation from a different perspective.

The work of Abernathy & Clark [7] is concerned with how knowledge underpins innovation. Their model is founded on the argument that technological capabilities may be retained while market capabilities are destroyed, or vice-versa. This model explains why, in some markets, an existing piece of technology from an established firm with complementary assets and market knowledge may continue to thrive in the face of a technically superior product from a rival start-up. Abernathy & Clark also introduce the notions of *incremental* and *radical* innovation. Incremental innovations are those which are non-competency destroying, whereas radical innovations can destroy existing competencies by making them obsolete, but, at the same time, open an innovation to new markets and opportunities.

While the incremental–radical dichotomy is useful for reasoning about innovation in technological and market terms, it fails to explain why modest incremental changes can, in some cases, have disastrous consequences. For example, despite a dominant position in the photocopier market in the 1970s, Xerox lost its market-leadership in the face of newer rivals who introduced products which were, to them, only incremental improvements. After analysing this and similar cases, Henderson & Clark [32] proposed that the knowledge needed to develop an innovation is based on components, and the architectures these components form part of. Consequently, it may be more fruitful to consider innovations in terms of whether innovation necessitates internal changes to components and component knowledge, or modification and re-configuration of architectures, and the — quite often — related tacit knowledge. From a design perspective, reasoning about innovations in terms of *component* and *architectural* knowledge and capabilities is an attractive proposition; such concepts tie in with definitions of components and architectures in an engineering sense.

2.4 Social Entrepreneurship analogies

In section 1, we suggest that because Social Entrepreneurship can solve "wicked" problems, a case exists for an entrepreneurial approach to security design as well. To see why, examining the similarities between classic entrepreneurship and Social Entrepreneurship is useful. Martin & Osberg [41] observed that entrepreneurs in both cases are motivated by market opportunity rather than direct financial gain, and profit is essential for sustaining the venture. In the case of social enterprises, these profits are usually re-invested in the business.

By considering three analogies between Social and Security Entrepreneurship, we can appreciate why Security Entrepreneurship has the potential to make a positive impact to the design of secure systems.

First, the problems addressed by both Social and Security Entrepreneurship have a social context. In the case of Social Entrepreneurship, innovations tackle problems in disadvantaged social contexts. In Security Entrepreneurship, situating innovations improves the design of a secure system, which safeguards physical or logical assets. The ultimate consequences of damage or loss of these assets are social, rather than technical.

Second, the value propositions nurtured by Social Entrepreneurs are designed to empower under-served or neglected populations. While the populations served by Security Entrepreneurs are not usually as impoverished as those served by Social Entrepreneurs, the need to empower the population is the same. Properly situating security controls allows people within the related environment to securely use their software, and thereby achieve their primary, and invariably non-security, goals.

Third, the success of a social innovation can be marked when traditional organisations attempt to enter the market place they had hitherto ignored [57]. This is both a blessing and a curse in a security context. The arrival of a new entrant with complementary assets might lead to a reconfiguration of people, processes, and technology; this may or may not have a disruptive impact on the overall system. On the other hand, the new entrant may be an attacker, and the innovation an asset which needs safeguarding as well as developing. Therefore, it would be useful to consider how innovation models can help to reason about how to deal with the attack; this issue is discussed in more detail in section 5.4.

Notwithstanding analogies with other forms of entrepreneurship, we need to consider how entrepreneurial techniques can be aligned with concepts associated with security and its design before we can ascertain the usefulness of the idea of Security Entrepreneurship.

3. SECURITY ENTREPRENEURSHIP AND THE SECURITY ENTREPRENEUR

In section 1, we defined Security Entrepreneurship as the application of innovation models and principles to organise, create, and manage security design elements to bring about improved system security. Based on this, the role of the Security Entrepreneur is to:

- identify opportunities for system insecurity,
- explore solutions for dealing with this insecurity, and
- remove the insecurity by re-configuring the system using the resources available.

We do not propose a new design method, but rather a new design paradigm where innovation models and techniques are used to supplement security design methods; these techniques identify, situate, and evolve solutions to insecurity. The novelty of our paradigm is that we treat the solution to information insecurity in much the same way that we might nurture a technology or social enterprise. The set of exploitable models is large, and precedents have already been set for using them for purposes other than predicting technology innovation. The Bass diffusion model [14] has been used to help predict the obsolescence of technology [30], and Disruptive Innovation Theory [19] has been generalised and re-purposed to better understand innovation in military doctrine [42]. Moreover, not only is the set of models large, it is also growing. In particular, the Social Entrepreneurship community has identified new innovation models based on successful social innovation in the developing world. For example, Leadbeater's theory of Structured Self-Organisation [40] is inspired by successful city-wide social-entrepreneurship initiatives in Curitiba, Brazil. Waste recycling and city planning may sound a world away from the information security problems we face, yet the success of these

initiatives relied on values such as collaborative engagement and a pragmatic (rather than perfectionist) working philosophy; we do not believe these values are incompatible with the design of secure systems.

Our model of a system is based on figure 1. We assume that the system is the amalgam of a socio-technical system, which represents a system of technology used within a system of activity, and the social system which surrounds it; this system needs to be safeguarded, as well as designed. The primary and supporting actors remain the same, but the role of the inventor is represented by a stakeholder who owns the security requirements during the design process. The innovations an entrepreneur wishes to situate may be novel design elements, or known security controls.

At first blush, analogies can be drawn between the Security Entrepreneur's role and that of an architect or design authority for a system. Indeed, if we map *insecurity* to risks, and *opportunities* to security requirements, the resulting approach might look like one of the many existing approaches to risk analysis [23, 20] or threat modelling [50, 54]. To understand the novelty of the Security Entrepreneurship paradigm, we need to differentiate it from the existing paradigms for designing secure systems. The easiest way to do this is to compare and contrast the role of a *software architect* with that of an *entrepreneur*.

Like entrepreneurs, architects make design decisions based on the downstream effect they will have on the systems they build [15]. An empirical study on the role of system architects [31] identified a number of characteristics shared with entrepreneurs:

- Architects leverage social networks; they position themselves on a project in such a way that they are approachable to developers in the event of problems, and develop their networks by nurturing ties with other groups inside and outside an organisation.
- Architects span boundaries within an organisation, to garner support or obtain commitments from complementary groups.
- Managing change and negotiation are core elements of an architect's work.

There are, however, several differences between these two roles.

First, the scope for an entrepreneur is wider than that of an architect. The architect has independence within the scope of his project. The entrepreneur has independence within the scope of the entire system. Moreover, unlike the architect, his remit extends to people and processes, as well as just software and hardware.

Second, where an architect is *system-centered*, an entrepreneur is *opportunity-centered*. Architects may work in teams and, where there is a chief architect, he may act as a mediator between the different sub-system architects and project managers; the resulting architecture is a pragmatic realisation of a system's goals, and an appeal to conceptual integrity. In contrast, an entrepreneur is a lone, empowered agent-of-change. If an entrepreneur is interested in the conceptual integrity of his design, it is because he wants this to be centered around his innovation strategy, rather than a system's goals.

Third, an architect needs to be mindful of ensuring his architecture is delivered in a timely manner, usually as soon as possible. Time is often an imperative for entrepreneurs as well, but for different reasons. Unlike software architects, entrepreneurs may be in competition with other entrepreneurs. As such, they may choose to make the strategic choice of not bringing an innovation to market when it's ready, but to instead allow a competitor to take an early lead. In this case, the entrepreneur's strategy is to let the competitor resolve uncertainties about technology or the market, before leap-frogging him with an improved product or better complementary assets.

Finally, the operating environment, which is analogous to the market in section 2.2 has a different part to play for both roles. For both roles, failing to situate a design to the environment can lead to system failure. However, although the environment is non-mutable to the architect, it is shapeable to the entrepreneur. In the same way that nascent markets form around dominant designs, the elements of an operating environment may shape itself around a system.

4. SECURITY ENTREPRENEURSHIP TECHNIQUES

In this section, we describe how creativity and innovation models and techniques can be used to foster Security Entrepreneurship. We present three sample techniques, which can be used to support secure systems design. Our approach assumes that the process of specifying requirements and proposing design elements to implement these is intertwined and self-reinforcing, as suggested by Nuseibeh [45]. The techniques presented explore the impact of security requirements (section 4.1), or determine what requirements need to be put in place to re-configure the system (sections 4.2 and 4.3).

To illustrate how these techniques might work in practice, we describe the design of security controls for an imaginary project developing a data-grid for clinical research. The nodes in the grid are servers located in hospitals around the UK. The data grid stores anonymised and partially anonymised clinical data, and is accessible anywhere by authorised end-users. The grid infrastructure has been developed and maintained by a single group of grid developers; this group is based at a university, and consists of system administrators and web service developers. End-users of the grid interact with the data grid in two ways: they can use an interactive shell to upload and download data and workflow instructions, or they can use customised applications developed and maintained by a separate application developed, located at a different university. Grid nodes are maintained by local IT support teams at the different hospitals. Although this scenario is imaginary, it is reflective of data grids used for clinical research in the UK, e.g. [18, 25].

In our example, a system architecture for the data grid has been provisionally set-out, and a number of security requirements have been elicited. These are currently being evaluated to determine candidate security controls which meet this architecture.

4.1 Security Chindōgu

Sometimes ideas which fail to realise their potential in one context, can be successfully repurposed in another. To appreciate an object's repurposing potential, it is useful to first identify its *affordances*: the qualities associated with the object and the user, which suggest action possibilities

Figure 2: Baby Mop (top), and Forget-Me-Not Digital Certificate Chindōgu (bottom). Top Image taken from [39]

for using the object [26]. By thinking of an object in terms of affordances offered, we can reason about how people might realistically use an object in different contexts of use.

A major challenge in security design is building usable security controls; the consequences of not doing so have been well-reported [9, 58]. Unfortunately, many of the affordances associated with security control invite perceptual uncertainty, not least because their physical representation is often far removed from its specified purpose. For example, the ASCII representation of a X.509 Digital Certificate suggests different action possibilities to a Certificate Authority than they do for a Clinical Researcher with no knowledge of Public Key Cryptography.

Bell et al. [16] propose using the literary device of defamiliarization to obtain different perspectives on possible affordances; these perspectives are obtained by viewing artifacts in strange and unfamiliar ways. By viewing what would normally be considered mundane objects through the eyes of someone with no familiarity of an object or its contexts of use, we identify opportunities or challenges associated with an object's use that might otherwise have remained hidden. To understand the possible impact of security requirements and possible design criteria for controls which refine them, we precede the defamiliarization exercise with a creativity technique, and follow it with an analysis of affordances. These affordances suggest possible design criteria, more refined requirements, or further opportunities for insecurity.

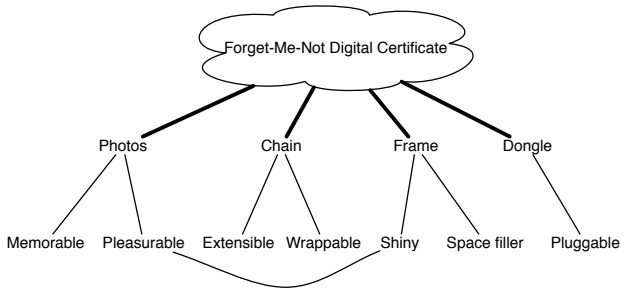
The creativity technique involves using the art of Chindōgu to prototype a possible security control which satisfies the requirement or requirements. This entails inventing an ingenious gadget, which may seem like an ideal solution to a problem, but introduces so many new problems that it effectively has no utility [6]. An example of such a Chindōgu is the Baby Mop in figure 2 (top). The Baby Mop aids busy parents with their household chores by allowing the baby to mop the floor while he or she crawls upon it. For an artifact to be Chindōgu, it must meet the following criteria:

- A Chindōgu cannot be for real use.
- A Chindōgu must exist.
- Inherent in every Chindōgu is the spirit of anarchy.
- Chindōgu are tools for everyday life.
- Chindōgu are not for sale.
- Humour must not be the sole reason for creating a Chindōgu.
- Chindōgu is not propaganda.
- Chindōgu are never taboo.
- Chindōgu cannot be patented.
- Chindōgu are without prejudice.

In the case of the baby mop, what seems like an interesting idea inevitably raises many, irreconcilable implementation issues; these range from child exploitation, through to health problems if the Baby Mop is exposed to harmful liquids, such as bleach.

Our technique involves taking an individual security requirement, or a pair of related security requirements, and

Figure 3: Forget-Me-Not Digital Certificate Chindōgu affordances



developing a Chindōgu implementing it. Once the Chindōgu has been built, its affordances are examined objectively, and possible design ideas are elicited. Chindōgu are usually physical rather than software-based because the affordances of a physical artifact are easier to perceive. However, prototyping a software Chindōgu, and examining its interface, architecture, and code may also suggest affordances, albeit in a more subtle manner.

Our rationale for developing a control as a Chindōgu rather than a more useful prototype is the challenging nature of the Chindōgu tenets; building an artifact which looks useful but is deliberately designed to be useless is unorthodox to most engineers, and demands creative thinking. Breaking from conventional orthodoxy is useful for viewing the artifact from an unfamiliar standpoint.

The sharing of digital certificates is endemic in our data grid example, thereby making access to the data grid by unauthorised users a real threat. In our example, the following security requirement has been stipulated: *a digital certificate shall only be used by the user to whom it is issued.* Satisfying the requirement may mitigate the threat, but its wording offers little in the way of guidance for refining the requirement.

The Forget-Me-Not Digital Certificate in figure 2 (bottom) is one solution which satisfies the requirement. The solution involves storing a digital certificate on a dongle, and permanently attaching this to a tasteful photograph of a loved one, or a particularly memorable picture. As users would want to keep an icon of such beauty with them always, they would no more share something so emotionally valuable than they would their wallet or car keys. The idea for using the picture frame could have arisen naturally from brain-storming, however our idea was developed with the aid of the combinFormation creativity support tool [36]. After seeding combinFormation with the keyword *certificate*, one of the images generated was a picture of a couple huddled over a picture frame in a country setting.

The proof-of-concept for the Chindōgu was sketched on a white-board, but was fabricated using raw materials commonly found in a Computer Science department. While scavenging for building material, pictures of loved ones were a fairly frequent sight, as were USB sticks, which could represent USB dongles. Because the USB sticks and picture frames we found had small loops on the back, we could build a simple *chain* using paper-clips, which were also available in abundance. A digital certificate added to the USB stick was a public key generated by one of the authors using PGP [56].

Although the Chindōgu meets the explicit security requirement, the number of practical problems associated with its implementation are large enough to render the device useless. We examined the affordances of this Chindōgu objectively, without any preconceived biases, to understand what behaviours this artifact invites. The act of fabricating the Chindōgu aided this analysis because the affordances were perceived directly, rather than mentally by thought experiment using a whiteboard drawing. A chart of these affordances is presented in figure 3. Some of the questions raised by analysing these affordances were as follows:

- What happens if we decide to attach more dongles, extend the chain by adding more links, or simply remove the dongle and replace it with another? This suggests a possible design criteria such that any physical or logical association between the digital certificate and another physical or software components cannot be re-purposed.

- What happens if the imagery we have selected no longer gives us the pleasure it once did? Reflecting on this question implies that any security designs should be built around values likely to be consistent across the population of its users?

- What happens if the imagery in our Forget-Me-Not is so effective that users are overwhelmed with a sense of altruism, and feel the urge to share the pleasure of this object with other users? By using an attractive picture frame, we have imbued our Chindōgu with reflective design values. As a consequence, the artifact introduces an unplanned tension between selfishness and altruism. Although ambiguity can be a useful resource for fostering reflective design in an artifact [24], the lesson to draw from this observance is that security controls meeting this particular requirement should appeal to the visceral and behavioural design principles [44], rather than aesthetics.

The Security Chindōgu helps fill the space between security problems and possible solutions. Using security requirements as seeds, participatory design workshops for risk analysis can be supplemented with *Security Chindōgu Design Sessions* to find possible ideas for mitigating risks. For the security entrepreneur, the technique also leads to further opportunities. In particular, the technique can be used to bridge Open Innovation – a paradigm where both ideas and paths to market are generated from both from within and outside an organisation – with security design. Security requirements could potentially be anonymised, and the building of Security Chindōgu crowd-sourced using one of the many available Open Innovation services, e.g. [2].

4.2 Innovation Value-added Chain

Hughes [35] found that system growth relies on correcting *reverse salients*; these are imbalances which occur when some parts of a system develop faster than others. Correcting these reverse salients may require incremental or radical innovation on the system components concerned.

Figure 4: Innovation Value-added Chain for stakeholder groups

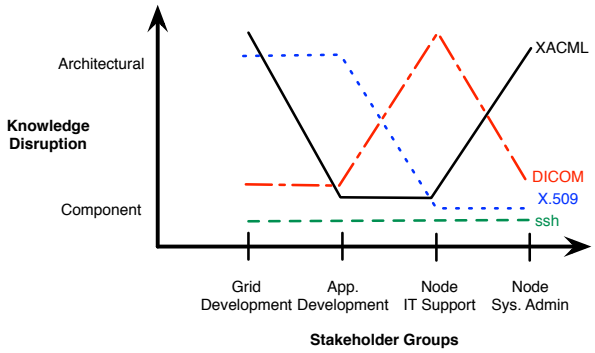
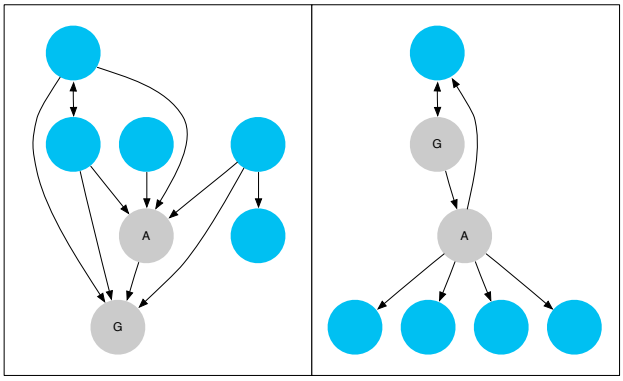


Figure 5: Existing security data social-network (left), and a social-network optimised for X.509 and XACML (right)



The Innovation Value-Added Chain examines the implications of an organisation's innovation on its suppliers, customers, and complementary innovators [10]. Innovations which may require only an incremental addition to one firm's processes and knowledge, may be disruptive and require a rethink in many areas of another. This model is visualised as a simple graph where the y-axis refers to the type of innovation, which may be incremental or radical, and the x-axis refers to the different stakeholder groups in the chain.

In our aligned version of this model, we make one minor modification. The terms *Incremental* and *Radical* can be confusing to the lay reader; the terms might suggest the innovation's perception to different groups is incremental or disruptive. Therefore, to reflect our modelling of an innovation's impact on knowledge and processes, we instead use the terms *Component* and *Architectural* as described in section 2.3.

In our example, we consider the different stakeholder groups in the value-chain for several security controls; these are commonly found in data grids used for storing sensitive medical data:

- XACML policies [27]: These model the resources that users have access to.
- X.509 Digital Certificates [3]: These are issued to authorised users to enable access to grid services.
- Secure Shell (ssh) [12]: An interactive command shell; this implements a secure channel for uploading and downloading data and workflow instructions to the grid.
- DICOM [5]: These standards and protocols are used for handling, storing, and transmitting medical information.

As well as the grid and application development teams, we also include the IT support teams at the node locations, and the system administrators responsible for maintaining the node servers.

This model illustrates how the successful integration of a security control will be contingent on the ability of relevant stakeholder groups to situate it within their own organisational contexts.

As figure 4 suggests, some stakeholder groups will integrate some controls faster than others. For example, application developers may only have to adapt to DICOM APIs to ensure any data their applications work with is anonymised. The software changes necessary to achieve this may be slight, and the developers already familiar with DICOM. On the other hand, IT support teams at local sites may need to develop fundamental changes to the way PC software and hardware is resourced and maintained to ensure these standards are met.

If the success of a project hinges on the successful integration of all of its security controls, the Innovation Value-added chain is useful for examining how alternative controls may help or hinder overall adoption.

4.3 Social Network Analysis

Social Network Theory is grounded in the idea that systems encapsulate thick webs of social relations and interactions. Recent work in this area [17] reports on the existence of different types of dyadic links (similarities, social relations, interactions, and flows) and, given the same nodes, social structures can behave very differently based on their patterns of relationship.

Existing work has examined how network topologies have interesting properties; these can be used to reason about attacker behaviour [11]. Of particular interest to Security Entrepreneurs, who are interested in optimising social networks to create the greatest positive impact for a security design decision, is the Strength of Weak Ties theory [29]. The theory suggests that novel information is more likely to flow between weak, as opposed to strong, ties; this is because weak ties can be easily disconnected from social networks. Nodes with weak ties can be reconfigured in different ways, based on the Security Entrepreneur's innovation strategy.

Using our example, we examine the social network associated with the data-grid and consider ways of obtaining the most usable impact for our XACML and X.509 controls. In figure 5 (left) we modelled a network structure based on the security information flow between different actors in the data-grid, where the arrow head represents the direction of information flow. This structure was derived by analysing interview transcripts from a qualitative study study on data-

grid security [22], and culling graph data based on known associations between different interviewees. The interviewees were either end-users (blue nodes), application developers (grey node labelled as *A*), or grid developers (grey node labelled as *G*). These associations were modelled declaratively, and visually rendered using GraphViz [1].

To optimise this social network, we review the requirements related to the use of XACML and X.509 controls, which must be satisfied:

- Access control decisions shall be implemented by the grid infrastructure team, based on requests by designated users, who represent line managers for different research groups.
- Designated users shall be the same users who authorise requests for X.509 digital certificates.
- X.509 digital certificates shall be sent to individual users once they have been created.

Although the collaborative ties between users is reasonably strong, the requirements suggest the levels of acquaintance between users and grid & application developers are comparatively weak. There are also degrees of centrality associated with the application and grid developer nodes; these have the potential to slow down, or even distort information. Moreover, according to the requirements, the application developer node may, in reality, only need to be loosely connected. This gives the Security Entrepreneur an opportunity to reconfigure the social network, and situate security tasks to make optimal use of this characteristic.

Figure 5 (right) represents one reconfiguration of the social-network; this fulfils the security requirements while simultaneously reducing the number of requisite ties. From a usability point of view, it is desirable to remove redundant associations, as this frees user nodes to form more relevant [non-security] ties. There are, of course, several possible configurations based on the social network the entrepreneur wishes to build. In this particular configuration, we want to be parsimonious with information flows.

5. TOWARDS SECURITY ENTREPRENEURSHIP

The techniques we propose have a number of benefits. First, because they are conceptually simple and impact different groups, they make good boundary objects; these are artefacts flexible enough to be used within a conceptual boundary, but robust enough to retain a common identity across boundaries [52].

Second, these underlying models and techniques have been empirically validated. Chindogōu's anarchistic nature of cherishing failure is embodied by a number of product design consultancies, such as IDEO, who consider failure as *enlightened trial-and-error*. For this reason, IDEO offices maintain collections of failed product designs; these are used to identify whether a product which failed in one environment may be successful in another [55]. Although the models of innovation presented were conceived to reason about past innovations, they have now developed into conceptual tools used by many practising Technology and Social Entrepreneurs; as section 2.4 suggests, these problems are not dissimilar to those faced by Security Entrepreneurs.

These techniques do, however, raise a number of questions about how Security Entrepreneurship can be applied in practice. Security Entrepreneurship espouses a worldview where system building is an opportunity to catalyse an innovation, where the *innovation* is one or more design elements. But this worldview is just one of several perspectives which may be taken when designing a system. For Security Entrepreneurship to be convincing, it needs to be seen by the community as complementary and free-spirited, rather than simply disruptive and anarchistic. Therefore, we need to consider what some of this paradigm's unintentional consequences might be, and ask how security entrepreneurship can be both applied in conventional secure systems design, and progressed as a security research topic.

5.1 Security Entrepreneurship considered harmful?

Gibson states that affordances of the environment are what they offer the environment for good or for ill [26]. Given the opportunities afforded by Security Entrepreneurship, are Security Entrepreneurs a positive influence in all possible contexts? This paper has described how innovation theories can be used to better inform the design of security. However, whether the ends of such entrepreneurship results in *better* security is open to debate. An unscrupulous consultant may use innovation models and techniques, which use fear to scare stakeholders into making sub-optimal security design decisions that benefit him financially. Similarly, an attacker may use the same techniques to organise different design elements to bring about *decreased* security for a targeted system.

Schumpeter's proposition that entrepreneurs act as agents of disruptive change suggests that while innovation adds value, this value comes at a cost. This has been described as the *innovation design dilemma*; structured processes generate few ideas, while more unstructured processes generate more diversity, but at the cost of conflict that might hamper the implementation of innovation [34]. Therefore, we need to understand the developmental contexts where Security Entrepreneurship may and may not be useful.

5.2 How should Security Entrepreneurship be situated with other design activities

We also need to understand how the Security Entrepreneur can work with, rather than be in conflict with, other design stakeholders like architects. An unintended consequence of innovation disruption is human and technical elements of a system may try to fight the disruption in some way, rather than allowing the innovation's environment to stabilise around a dominant design. To do this, we need to consider how Security Entrepreneurship and conventional secure software engineering approaches complement each other. Considering this raises a number of questions, which include:

- How does design data flow to and from Security Entrepreneurship techniques? The data which contributes to the techniques presented are by-products of security design; security requirements are the seed for Chindōgu, and the understanding gleaned developing them informs the thinking underpinning the other techniques. If we successfully elicit requirements from a Security Chindōgu, how do we preserve traceability links from the initial requirements to the subsequent requirements derived from it, via the Chindōgu?

- How do we manage the design data generated by Security Entrepreneurship techniques, and how do we ensure they are not marginalised or abused? The Security Entrepreneur may need to revisit a current strategy if the data forming the basis of an Innovation Value-added Chain becomes stale, or a social network evolves. Therefore, these models need to be managed in the same way as design documents or risk management artifacts. When building Security Chindōgu, we also need to be mindful that these are throw-away, rather than evolutionary, prototypes. Consequently, a change in mind-set may be required to encourage people to retain and re-purpose Chindōgu in different contexts, without explicitly developing these into products in their own right.

- Who is the Security Entrepreneur? Security Entrepreneurs are central nodes in the social network of actors contributing to the implementation of the secure system. However, as the data grid example suggests, the multi-organisational nature of some systems means it might not be obvious who should best fulfil this role. In the classic Information System literature, this role might have been the preserve of the *business analyst* but, given the cross-cutting nature of security, perhaps this role may, in future, be better fulfilled by information security officers? Irrespective of who, the responsibilities of a system builder have remained largely unchanged since first identified by Hughes [35]. Consequently, training in the transferrable skills necessary for system builders should be considered a pre-requisite for the successful adoption of Security Entrepreneurship.

5.3 How is Security Entrepreurship validated?

The practical, hands-on nature of security entrepreneurship means we cannot realistically evaluate it without applying it in the real world; this involves some form of interventionist research methodology, such as Action Research. Action Research aims to contribute to both the practical concerns of people in an immediate problematic situation, and to the goals of social science by joint collaboration within a mutually acceptable ethical framework [46]. In other words, the output of an intervention contributes both to practice and research at the same time. An action research intervention involves identifying a research question, developing an action plan, implementing the plan, gathering and analysing data, and reflecting on the findings of the investigation [13].

Action Research is a popular research method for evaluating case studies in entrepreneurship; it has even been argued that techniques used as part of an intervention can strengthen an entrepreneurial process [47]. The Empirical Software Engineering community is also taking a growing interest in adopting interventionist methods to evaluate tools and processes, yet despite pre-existing precedents [37, 53], we continue to rely heavily on student projects and well-publicised exemplars to validate security design approaches. To validate approaches to Security Entrepreneurship, we need to re-engage with our industry partners; industry has real security problems which need solving, and rich socio-technical systems within which we can exercise our approaches.

5.4 Can Security Economics help?

The models presented here are static models of innovation; these are only concerned with the capabilities of different groups, and the knowledge underpinning them. Several innovation models are more dynamic, and consider the radical and incremental phases of an innovation following its initial adoption. For example, the Utterback-Abernathy model [8] is concerned with how an innovation evolves from a *fluid phase* of technical and market uncertainty, through to the *transition stage* where the innovation becomes a dominant design, and then onto the *specific phase* when the product focus moves from design competition through to product performance and cost.

If we are to transpose dynamic models of information to reason about the long-term evolution of a security control, we need to consider what data contributes to these models. Such models might be useful if we wish to explore adversarial relationships associated with introducing attackers as new entrants, which may wish to shape the market to exploit certain assets.

Identifying this data, and exploring how these dynamic models might transpose to security may be a fruitful area of research for the Security Economics community. Any future research agenda does, however, need to be conscious of the paradoxical relationship between economics and entrepreneurship. This paradox arises because, although the market- and price-based economics community has historically been indifferent towards the interpretive and comparatively anarchistic nature of entrepreneurship, entrepreneurs are an essential element of market economies.

Examining what these two communities currently have in common might be a step towards collaboration between Security Economics and Security Entrepreneurship research. For example, an alternative perspective on Herley's findings on users' rationale rejection of security advice [33] might be obtained by analysing how much these controls contribute to destroying the competency knowledge of different actors associated with these controls.

6. CONCLUSION

Creativity on its own is not enough to implement innovative security controls. We also need to predict the impact a control might have from different perspectives of a problem.

Our proposal for Security Entrepreneurship makes three contributions towards a more innovative approach to secure systems design. First, we have described how theories from the Technology and Social Entrepreneurship literature can be re-purposed to develop security innovations with only a modicum of changes. Second, we have demonstrated how three practical techniques provide hitherto unseen insights on a working secure system design. Finally, we have analysed some of the consequences of adopting these and other models of innovation, and propose research questions to address them.

7. 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.

We are also grateful to Marc Ventresca and Victor Seidel at the Oxford Centre for Entrepreneurship and Innovation for their comments and insights both before and during

the preparation of this paper, members of the OUCL Security Reading Group, our paper shepherd: Deb Frincke, the anonymous reviewers, and the attendees of the NSPW 2010 workshop for their valuable feedback on this paper.

8. REFERENCES

[1] Graphviz web site. `http://www.graphviz.org`.

[2] Open Innovators web site. `http://www.openinnovators.net`.

[3] *X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks*. International Telecommunication Union, 2005.

[4] *ISO/IEC 27002: Information Technology – Security Techniques – Code of Practice for Information Security Management*. ISO/IEC, 2007.

[5] *Digital Imaging and Communications in Medicine (DICOM): Part 1: Introduction and Overview: PS 3.1-2009*. National Electronic Manufacturers Association, 2009.

[6] Chindōgu Wikipedia entry. http://en.wikipedia.org/wiki/Chindōgu, April 2010.

[7] Abernathy, W., and Clark, K. B. Innovation: Mapping the winds of creative destruction. *Research Policy 14*, 1 (1985), 3–22.

[8] Abernathy, W. J., and Utterback, J. M. Patterns of innovation in technology. *Technology Review 80*, 7 (1978), 40–47.

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

[10] Afuah, A. *Innovation management: strategies, implementation and profits*, 2nd ed. Oxford University Press, New York, 2003, ch. 2.

[11] Anderson, R., and Moore, T. The economics of information security, 2006. Science.

[12] Barrett, D. J., Silverman, R. E., and Byrnes, R. G. *SSH, the secure shell: the definitive guide*, 2nd ed. O'Reilly, Sebastopol, CA, 2005.

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

[14] Bass, F. A new product growth model for consumer durables. *Management Science 15*, 5 (1969), 215–227.

[15] Bass, L., Clements, P., and Kazman, R. *Software architecture in practice*, 2nd ed. Addison-Wesley, Boston, 2003.

[16] Bell, G., Blythe, M., and Sengers, P. Making by making strange: Defamiliarization and the design of domestic technologies. *ACM Trans. Comput.-Hum. Interact. 12*, 2 (2005), 149–173.

[17] Borgatti, S., Mehra, A., Brass, D., and Labianca, G. Network Analysis in the Social Sciences. *Science 323*, 5916 (2009), 892–895.

[18] Brady, M., Gavaghan, D., Simpson, A., Parada, M. M., and Highnam, R. eDiamond: a Grid-enabled federated database of annotated mammograms. In *Grid Computing - Making the Global Infrastructure a Reality*, F. Berman, A. Hey, and G. Fox, Eds. John Wiley & Sons, 2003.

[19] Christensen, C. M. *The innovator's dilemma: when new technologies cause great firms to fail*. Harvard Business School Press, Boston, Mass., 1997.

[20] den Braber, F., Hogganvik, I., Lund, M. S., Stølen, K., and Vraalsen, F. Model-based security analysis in seven steps - A guided tour to the CORAS method. *BT Technology Journal 25*, 1 (2007), 101–117.

[21] Faily, S., and Fléchais, 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.

[22] Faily, S., and Fléchais, I. A Model of Security Culture for e-Science. In *Proceedings of the South African Information Security Multi-Conference (SAISMC 2010)* (2010), N. Clarke, S. Furnell, and R. von Solms, Eds., University of Plymouth, pp. 154–164.

[23] Fléchais, I., Sasse, M. A., and Hailes, S. M. V. Bringing security home: a process for developing secure and usable systems. In *NSPW '03: Proceedings of the 2003 workshop on New security paradigms* (New York, NY, USA, 2003), ACM, pp. 49–57.

[24] Gaver, W. W., Beaver, J., and Benford, S. Ambiguity as a resource for design. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2003), ACM, pp. 233–240.

[25] Geddes et al. Neurogrid: using grid technology to advance neuroscience. *Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium on* (June 2005), 570–572.

[26] Gibson, J. J. *The ecological approach to visual perception*. Houghton Mifflin, Boston, 1979.

[27] Godik, S., and Moses, T. EXtensible Access Control Markup Language (XACML) version 1.1, committee specification, August 2003. *http://www.oasis-open.org* (5 May 2005).

[28] Godin, B. The linear model of innovation: The historical construction of an analytical framework. *Science Technology and Human Values 31*, 6 (2006), 639–667.

[29] Granovetter, M. The strength of weak ties: A network theory revisited. 201–233.

[30] Gravier, M. J., and Swartz, S. M. The dark side of innovation: Exploring obsolescence and supply chain evolution for sustainment-dominated systems. *Journal of High Technology Management Research 20*, 2 (2009), 87–102.

[31] Grinter, R. E. Systems architecture: product designing and social engineering. *SIGSOFT Softw. Eng. Notes 24*, 2 (1999), 11–18.

[32] Henderson, R. M., and Clark, K. B. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly 35*, 1 (1990), 9.

[33] Herley, C. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *NSPW '09: Proceedings of the 2009 workshop on New security paradigms workshop* (New York, NY, USA, 2009), ACM, pp. 133–144.

[34] Hobek, J. The innovation design dilemma: some notes on its relevance and solution. In *Innovation: a cross-disciplinary perspective*, K. Grønhaug and G. Kaufmann, Eds. Norwegian University Press, 1988.

[35] Hughes, T. P. *Networks of power: electrification in Western society, 1880-1930*. Johns Hopkins University Press, Baltimore, 1983.

[36] Interface Ecology Lab. combinFormation Home Page. http://ecologylab.net/combinFormation/ .

[37] James, H. Managing information systems security: a soft approach. *Information Systems Conference of New Zealand, 1996. Proceedings* (30-31 Oct 1996), 10–20.

[38] Karlsen, I. K., Maiden, N., and Kerne, A. Inventing requirements with creativity support tools. In *REFSQ '09: Proceedings of the 15th International Working Conference on Requirements Engineering: Foundation for Software Quality* (Berlin, Heidelberg, 2009), Springer-Verlag, pp. 162–174.

[39] Kawakami, K. *The Big Bento Box of Unuseless Japanese Inventions (101 Unuseless Japanese Inventions and 99 More Unuseless Japanese Inventions)*. W. W. Norton & Company, 2005.

[40] Leadbeater, C. The Socially Entrepreneurial City. In *Social Entrepreneurship: New Models of Sustainable Social Change*. Oxford University Press, 2006, pp. 233–246.

[41] Martin, R. L., and Osberg, S. Social entrepreneurship: The case for definition. *Stanford Social Innovation Review 5*, 2 (2007), 29–39.

[42] Mukunda, G. We cannot go on: Disruptive innovation and the first world war royal navy. *Security Studies 19*, 1 (2010), 124–159.

[43] Murphy, A. E. *Richard Cantillon, entrepreneur and economist*. Clarendon Press, Oxford, 1986.

[44] Norman, D. A. *Emotional design: why we love (or hate) everyday things*. Basic Books, New York, 2004.

[45] Nuseibeh, B. Weaving together requirements and architectures. *Computer 34*, 3 (2001), 115–117.

[46] Rapoport, R. N. Three dilemmas in action research. *Human Relations 23*, 6 (1970), 499–513.

[47] Rasmussen, L. B., and Nielsen, T. Entrepreneurial capabilities: Is entrepreneurship action research in disguise? *AI and Society 18*, 2 (2004), 100–112.

[48] Rittel, H. W. J., and Webber, M. M. Dilemmas in a general theory of planning. *Policy Sciences 4*, 2 (1973), 155–169.

[49] Robertson, J. Eureka! why analysts should invent requirements. *Software, IEEE 19*, 4 (jul/aug 2002), 20 – 22.

[50] Schneier, B. Modeling security threats. *Dr. Dobb's Journal* (1999).

[51] Schumpeter, J. A. *Capitalism, Socialism, and Democracy*. Allen & Urwin, 1944.

[52] Star, L., and Griesemer, J. R. Institutional ecology, "translations" and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. *Social Studies of Science 19*, 3 (1989), 387–420.

[53] Straub, D. W., and Welke, R. J. Coping with systems risk: Security planning models for management decision making. *MIS Quarterly: Management Information Systems 22*, 4 (1998), 441–464.

[54] Swiderski, F., and Snyder, W. *Threat modeling*. Microsoft Press, Redmond, Wash., 2004.

[55] Thomke, S., and Nimgade, A. IDEO Product Development (HBS-9-600-143). *Harvard Business School Case Study* (2007).

[56] Various. Pretty good privacy wikipedia entry. *http://en.wikipedia.org/wiki/Pretty_Good_Privacy* (2010).

[57] Weinberg, A. S., Pellow, D. N., and Schaiberg, A. *Urban Recycling and the Search for Sustainable Community Development*. Princeton University Press, 2000.

[58] 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. 169–184.