Against Mindset

Arne Padmos
hi@arnepadmos.com

ABSTRACT

The security field has adopted the social construct of the security mindset: the idea that there exists a single attitude that allows individuals to think like an attacker. However, there is little evidence that the security mindset is an appropriate construct. We suggest an alternative approach, consisting of multiple security-relevant attitudes, which are linked to security roles within the systems development life-cycle. To illustrate the usefulness of our approach, we show how the framework can be used to help shape curricula.

CCS CONCEPTS

Social and professional topics → Computing education

KEYWORDS

Security mindset, attitudes, curriculum design, alignment.

ACM Reference Format:
Arne Padmos. 2018. Against Mindset. In *New Security Paradigms Workshop (NSPW '18), August 28–31, 2018, Windsor, UK.* ACM Press, New York, USA.

OVERVIEW

This paper contributes:

- A novel framework for exploring the role that attitudes play in supporting security within system development life-cycles.
- An alternative approach to the singular security mindset, based on attitudes from other professions that serve as metaphors.
- A new taxonomy of roles, in which diverse security-relevant archetypes, attitudes, responsibilities, and tasks are aligned.

Instead of using a single archetype to represent security professionals, we have chosen to move from one to many. Based on an analysis of the system development life-cycle, we distilled several archetypes with a role to play in ensuring the security of systems. The set of attributes is derived from common ways of grouping actors, with a preference for binary attributes to split groups along broad lines within development processes.

1. Security analyst: takes an attacker perspective, looks at lower-level implementations, checks whether things are done right.
2. Security engineer: takes a defender perspective, develops one or more lower-level implementations, tries to do things right.
3. Security forecaster: tries to simulate offensive action, looks at higher-level designs, sees whether the right thing was picked.
4. Security architect: tries to simulate defensive action, prototypes higher-level designs, tries to pick the appropriate thing to do.
5. Security manager: has an overarching view, seeks alignment with the business, tries to balance relationships within teams.

This paper is in the public domain. Copying or redistribution is allowed, provided that the article citation is given and that the author is clearly identified as the source.
NSPW '18, August 28–31, 2018, Windsor, UK
ACM ISBN 978-1-4503-6597-0/18/08
https://doi.org/10.1145/3285002.3285004

Using other, older professions as metaphors, relevant attitudes for the archetypes were identified. These indicate that there is more to security than the security mindset and the hacker ethic.

1. Security analyst: critical and independent, in line with the world view of auditors, military intelligence analysts, and scientists.
2. Security engineer: humble and restrained, something that many traditional engineering professions learnt to value over time.
3. Security forecaster: playful and inquisitive, which is valued in the approaches popular amongst planners and policy makers.
4. Security architect: empathic and reflective, advocated for within the architectural field and within many other design sciences.
5. Security manager: transparent and proactive, which supports accountability, standards of good governance, and innovation.

In order for the archetypes and attitudes to become meaningful, they need to be integrated into roles in which attitudes support, and are supported by, appropriate skill sets. These roles can be linked to responsibilities and tasks within the development life-cycle.

1. The *security analyst* delivers audit reports by setting rules and focus areas, exploring different views, hypothesising and finding flaws, and creating demos and tooling.
2. The *security engineer* delivers module builds by picking and following standards, carrying out and codifying tests, integrating quality gates, and tracking and fixing vulnerabilities.
3. The *security forecaster* delivers threat models by modelling information flows, enumerating possible threats, exploring likely attack paths, and writing relevant scenarios.
4. The *security architect* delivers design drafts by eliciting end-user requirements, questioning basic assumptions, drawing sketches and diagrams, and drafting detailed specifications.
5. The *security manager* delivers process guides by prioritising and aligning goals, assessing and weighing risk, identifying and assigning tasks, and monitoring and reporting progress.

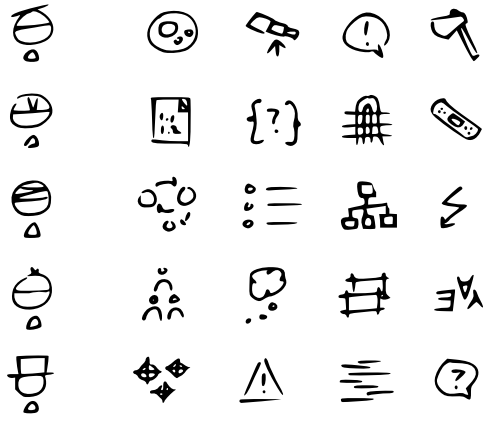

Figure 1: Graphical representation of the roles and tasks.

Note that an individual can hold multiple roles. Depending on the context, some roles may be complementary, while others are in potential conflict with one another. Besides teamwork questions, the taxonomy also raises questions concerning curricula and culture. Specifically, filtering and teaching on the basis of attitudes is harder than teaching and filtering on the basis of knowledge and skills, and the dynamics shaping culture formation in the security field are hard to control, as is spillover into system development.

1 INTRODUCTION

The concept of the security mindset has been discussed within the fields of computer security education [18] and security awareness training [36], and within information security more generally. It has been illustrated in the form of a story on the nefarious use of a paper form for shipping live ants [120], as well as more broadly in the balancing of "wolves" and "sheep" for a healthy society [121]. Attempts to teach such a mindset have included forcing students to "cheat" [24] and having students create everyday threat models [76]. However, in most cases, the concept of the security mindset remains vague and is not operationalised. When things are operationalised it is not a mindset that is taught, but a recipe or skill. The recent CSEC2017 curricular guidelines on cyber security [72] talk about the importance of aspects that are related to the idea of the security mindset, but its learning objectives focus on information reproduction. Note that there is an alternative definition of the security mindset as continuously asking yourself "How do I know my system is secure?", but we approach the concept from the conventional interpretation of thinking like an attacker.

This paper looks at the current state of the broader educational debate, with an emphasis on the question of attitude. It relates the questions that arise out of this debate to approaches to teaching security. Through the lens of curricular design theory, the idea of the security mindset is critiqued, and alternative conceptions are presented. The resulting views are relevant for instructors looking to integrate security into their curriculum, as well as trainers who are weighing different approaches to training developers, especially from the perspective of how to integrate "culture" into a development process. Furthermore, ideas relating to the security mindset touch the core of security and help us question what security is about. It provides a perspective on problems such as the rock-star culture within security, the emphasis of attack over defence efforts (beyond the simple view of "defenders need to protect everything, attackers need to find just one hole"), and the widespread failure of integrating the human factor.

The framework that is presented for analysing the concept of the security mindset and for operationalising the approach of multiple attitudes is not claimed to be the only possible approach. In line with the infinite number of ways a curriculum can be built, so there are many ways of analysing the secure development life-cycle in terms of archetypes, and subsequently translating these to relevant knowledge, skills, and attitudes. What is claimed is that the framework contributes a valuable way of aligning attitudes with the secure development life-cycle, as well as with the required underlying theory, skills, and methods (which often seems to be forgotten or neglected in debates to instil a nebulous "security mindset" or "security awareness").

2 BACKGROUND

In recent years, educational theory has increasingly emphasised the importance of studying more than just the concept of learning. In addition to getting students to learn various things, they should be "formed" [11]. Besides qualification, both personification and socialisation are put forward as important goals of education. This builds forth on the previous note that learning should not be focussed on mere memorisation, but should include work on competences that take the form of intertwined knowledge, skills, and attitudes [142]. Just as knowledge, skills, and attitudes should support each other, so there should likely be a positive relationship between the goals and efforts related to socialisation, personification, and qualification. Related concerns and criticisms include trends towards a greater emphasis on the cognitive domain, widespread teaching to the test (and resulting impoverishment of the curriculum), greater use of standardised testing (with parallel greater focus on those elements of the curriculum that are easy to test), reduced intrinsic motivation, and deprofessionalisation of the teaching profession [38]. Some other indicators include such practices as the apparent widespread use of Bloom's (revised) taxonomies, where the affective and psychomotoric domains seem to have been all but forgotten, with the term "Bloom taxonomy" referring singly to the cognitive domain. Also, while the idea of competence and competence-based education has gained ground over the past decades, the result appears to have mostly been additional emphasis on marketable skills, with key competence profiles missing any discussion of the relevant attitudes students should possess for them to be called "competent". Both the NICE [102] and e-CF [20] competence frameworks lack a detailed discussion of attitudes.

Given this situation, it is not surprising that the aspect of skills, and especially the aspect of attitude might not be widely integrated within curricula and training programmes. Looking at leading security certificates, there is still a widespread focus on knowledge (e.g. CISSP, CISA), with evaluation of skills only slowly becoming a part of the wider educational landscape within the security certification community (e.g. OSCP). In light of what appears to be slow extension of the area of concern to include skills, it seems unlikely that attitude will get much attention. Given the importance of strong attitudes in driving long-term behavioural outcomes [8], we claim this is not a positive state of affairs. A related question that those responsible for education and training of, for example, developers in secure coding could ask themselves is "to what extent have my lessons and efforts paid off in improved security posture 0.5, 1, 2, 5, 10 years after my intervention?". The same goes for security awareness efforts. As will be discussed further in the rest of this paper, the answer might not involve looking for the holy "security mindset", nor the training of purely hands-on step-by-step methods. Instead, we propose that knowledge, skills, and attitudes should be integrated and aligned into a coherent whole, relevant attitudes should be identified as a function of context (including, among other factors, looking at the place that a role occupies within the development life-cycle), and security should not be seen as a special way of thinking (instead, attitudes from, for example, architects and designers are very much seen to be both applicable and sufficient when talking about required attitudes). The latter point presupposes at the same time recognising the importance

of context while also recognising the generality of many issues. This can be viewed from the perspective of security as a quality attribute, implying both a dependence on a specific product, while sharing concerns across other quality criteria from similar development stages that multiple products go through. Note that canonical bodies of knowledge are held to still be of value, especially from the perspective of the "T-shaped professional", but they may need to be contextualised for optimal alignment, and they should complement practical experience.

3 FRAMEWORK

In the previous sections we suggested that educational efforts may have a lopsided focus on knowledge, and we noted the missing operationalisation of the concept of the security mindset. Where discussion of attitude does take place, the question is often approached using the cultural construct of a singular security mindset (which assumes that the thinking of security experts should be similar to the way that attackers think). This paper breaks with tradition, and illustrates why security seems to call on multiple attitudes in different contexts. We sketch an approach to security education based on archetypal roles with integrated knowledge, skills, and attitudes, derived from places within the systems development life-cycle. We also discuss why we think the attitudes required at different stages of the development life-cycle are so different from one another that it is unlikely that we can ever speak of something like a unitary atomic "security mindset". In later sections, we apply the framework to sketch rough outlines of possible curricula to illustrate the potential value of taking a multidimensional view of attitude in the context of security.

The design of our framework is based around five archetypes that we take to be representative of the different roles that security specialists can take within the development life-cycle. For each archetype there is a coherent set of knowledge, skills, and attitudes that appear to be of importance. In the next section we will discuss the attitudes that we think are important for the different archetypes, after which we look at the respective skills, along with several concepts that may be relevant for a future body of knowledge. We illustrate how each archetype is taken to have several "natural" attitudes that seem to be conductive to proper performance within the given job function. We also discuss some of the relevant problems related to assessment, especially those that have to do with assessing attitude. Throughout the analyses, where possible, relevant attitudes are inspired by common attitudes within other more mature fields of study.

To more directly impact the lives of people, we define the goal of security education to be the delivery of professionals with a (more) direct influence on the security of digital products that are produced. As such, a focus on the systems development life-cycle seems appropriate. To do so, we develop archetypes that will likely have a positive influence within a given moment of a product's life-cycle. For the development of these archetypes we have used an approach that is common in scenario planning: defining two axes that appear to be major drivers and using these to derive four quadrants. Note that we include an additional "wildcard", which is an approach that is sometimes used to capture elements outside the quadrants. The two axes that we have identified as important are analysis versus synthesis (or break versus build) and design versus implementation (or high-level versus low-level, abstract versus concrete). Roles are thus determined on the basis of a split between "getting the right design" (i.e. high-level design) and "getting the design right" (i.e. low-level implementation), as well as a split between identifying underlying facets and bringing constituent parts together.

The four quadrants that result from the analysis-synthesis and abstract-concrete axes are architecture, forecasting, engineering, and analysis. The wildcard is taken to be the manager that guides the development life-cycle. These five categories are explored in more detail in later sections of the paper. Several issues that have been identified in this approach are the possibility that concepts and skills may be company-specific, that roles may overlap in a given organisations, that non-standard attitudes may be needed when dealing with unique problems, and that external factors, such as legally mandated standards, may be necessary in addition to attitudes and intrinsic motivation. For now, let us note that role separation is recommended in security on the basis that people should not check their own work, that generic starting roles exist in the security industry with people recommended to start as application programmers or penetration testers [58], and that general models necessitate abstractions that may not include all edge cases (similar to how idealised systems development life-cycle models regularise a messy reality [99]).

Alternative frameworks exist that try to capture the nature of security. One such framework is the Attack Navigator [110, 114] which describes the activity of security professionals as planning attack routes on maps of a system. In contrast to our framework, the Attack Navigator takes an attacker-focussed perspective. The main disadvantage of this seems to be that education along these lines may turn into a self-fulfilling prophecy as a focus on existing attacks may preclude explorations of aspects such as underlying architectural causes, novel attacks, and social factors. On the other hand, attacks are concrete and measurable, and practices focussed around the security mindset may provide direction and clarity of purpose. Similar to our approach, the Attack Navigator framework makes use of metaphor, although it is built around the metaphor of navigation to both motivate and explain the methodology, while our approach uses the metaphor of well-known professions.

The archetype framework answers the call by the Dutch college capstone project assessment framework [7] for personas that illustrate degree programme profiles, as these could enable greater understanding, buy-in, alignment, and transparency between students, assessors, and external entities. Our approach contrasts with other competence models which generally only provide an unorganised list of roles, or roles split into arbitrary categories. Compared to competence profiles such as e-CF [20] and curricular frameworks such as CSEC2017 [72], our framework has an underlying structure linked to the development life-cycle (which is something that appears to either be absent or not made explicit in other frameworks) and uses metaphors of common professions from other fields (such as scientists, engineers, planners, architects, and managers) as a means for easing communication and common understanding.

4 ATTITUDES

The attitudes of the archetypes are derived from fields that show an apparent natural affinity and similarity to the tasks, approaches, methods, and perspectives of those of the archetypes. While these fields are obviously not one-to-one comparable, we claim they offer sufficient similarities for meaningful comparisons. Note that archetypes are discussed in reverse-chronological order with respect to their place in the development life-cycle.

4.1 Security analyst's attitudes

Starting with the archetype of the security analyst (or researcher), the required attitude is arguably the one that is closest to the idea(l) of the "hacker ethic" [91]. A security analyst needs to be *critical* and *independent*. These attitudes are nothing new: Auditors throughout the centuries have had to be constantly aware of the need to remain independent both in appearance and in fact. Military intelligence analysts have long known how easy it is for blind spots to form, as a result of looking at situations from their own world view and other biases [53], and as a result of hierarchical chain of command filtering bad news [147]. These insights have led to the nurturing of a culture where approaches such as devil's advocate and others [134] should be ingrained. An awareness of fundamental human biases, suspicion of authority, and healthy scepticism is also something that forms the basis of a scientific world view [112]. A long history and many examples of dogma clouding scientific insight [40] should have made most scientists suspicious of inherited wisdoms, however self-evident those wisdoms may appear at first glance. This also includes being critical of leading paradigms [79]. These fields indicate that security as a field does not have a unique claim to the importance of a mindset that questions authority. Instead, independence and being critical seem to be attitudes that are important for all fields that deal with questions regarding what the world is really like, in contrast to what it could or should be like.

4.2 Security engineer's attitudes

In contrast to the analyst, the engineer has to construct something. They have to build something on the basis of a blueprint, which by its very nature specifies only part of what will be built. The blueprint has to be translated into a physical entity in such a way that its design objectives are achieved. As a result of limited knowledge about the world and (unavoidable) mistakes, it seems important that the engineer approaches the task with an attitude that is both *humble* and *restrained*. Mistakes will surely be made, and are all the more likely with novel constructions, complex procedures, new materials, etcetera. This is a classic engineering mentality that should also be present in civil engineers, aerospace engineers, transportation engineers, and the like. This is also in line with the precautionary principle [111] advocated by ecologists, climatologists, and ethicists. One could argue that security is different, in that the security engineer needs to think about issues such as supply chain attacks, hardware implants, insider threats, and more. But this appears to be a difference in scale, not in the kind of attitude required. One of the most lethal civil engineering accidents in the US was the result of a contractor proposing a different attachment configuration for a series of walkways than those specified in the original design [95]. The Tacoma Narrows Bridge collapse was the result of constructing a bridge using a novel approach, adding elements to the bridge whose interaction had not been tested previously [4].

The situation with respect to security technologies only being gotten right in the fourth or fifth iteration [6] is not much different from the slow evolution of the safety and functionality of flight (see accidents such as the Hindenburg disaster [136] and various other crashes). Unintended consequences such as acid rain and smog illustrate that security isn't special when it comes to how hard or impossible it can be to predict adverse effects. Other (sub)disciplines, such as safety engineering [89] and resilience engineering [54], also entail a focus on a humble and restrained attitude. Many high-risk high-impact industries include explicit attention to the fostering of a safety culture that does not accept cowboy behaviour [48]. Enculturation takes the form of different approaches such as always holding stair railings, the acceptance of the utility of checklists (and the importance of continuously testing them) [44], as well as instilling a mentality of KISS (keep it simple, stupid) even for the engineer themself.

4.3 Security forecaster's attitudes

The forecaster (also known as modeller, simulator, threat modeller, and architectural risk analyst) works in the space of ideas. Their objects of study are not physical, in the sense that they do not yet exist as independent physical entities. In order to "evaluate" various architectural constructions before they are built, the forecaster tries to predict possible issues with their design. This does not belong to the traditional field of study of the sciences, as it is not about studying what is, but about studying what could be [126]. As the object of study has not been built yet, the knowledge gained takes the form of a simulation with its respective outputs. This is similar to the approach taken in design thinking where prototypes help to answer questions about a design [34]. Within the field of information security, such prototypes may take the form of information-flow models, and the simulation can be a "game" whereby players simulate both attackers and defenders in a make-believe synthetic world. Within the field of games studies this is referred to as the magic circle [74].

In line with this is the idea that the key identifying feature of games is that participants agree to artificial constraints (i.e. rules) and that the game can only be declared to be won by those who play according to those rules [130]. Similarly, simulations taking the shape of security games also involve a set of rules meant to ensure that the hypothetical world is reflective of the information flows and constraints embedded within the design of the architectural structure. A player in such a game could claim the action of transporting themself into a locked space, but, without clear links to the capabilities of their character and the properties of the game world, such an action would be viewed as breaking the rules. Of course, the challenge in such a game is finding a way to abuse the rules of the game in such a way that the objectives of the attacker are achieved. The mentality that seems to be required for such an exercise is a *playful* and *inquisitive* attitude, i.e. a willingness to enter a make-believe world that does not yet exist in reality, as well as the drive to explore such a setting. These techniques are not unique to the security context: in many other situations that involve uncertainty around future situations, the approach of gaming is used to simulate thought worlds. For example, within warfare planning

tabletop exercises are common (e.g. see [86]), within (public) policy planning they are used as a way of evaluating potential behaviours (e.g. see [73]), and within behavioural economics they are used to construct simplified models within which human action can be studied (e.g. see [9]). These games or models provide a (battle)space in which specific elements of architectural designs can be studied.

4.4 Security architect's attitudes

The security architect creates abstract designs that seek to meet some need, that seek to achieve some security target. Although such designs are made at a high level, they will eventually need to be embedded, where they will interact with actual users (in contrast to idealised entities within, for example, an entity-relationship diagram). Over time, product design has increasingly emphasised the importance of thinking about the end-user early in the design process [105]. In this light, an *empathic* attitude towards users appears important as it may enable designers to view the problems to be addressed from the perspective of other people. This could drive both the easier spotting of likely problems in existing implementations, as well as a greater understanding of the motivations of the entities in a novel architecture, which may eventually end up as a new implementation.

Within the field of security, and especially in traditional hacker communities, users have often been looked at from a position of superiority, as illustrated by terms such as "luser" and "PEBKAC" (problem exists between keyboard and chair). The field of usable security [25], which has grown in recognition in recent years, shows the vanity and serious security consequences of not integrating a concern for the user within the development life-cycle of a security system. With the traditional security mindset concept being closer to the idea of the hacker ethic, the postulated importance of a distinctive empathic attitude for real-world security further indicates that the idea of a single security mindset is not the whole story.

Besides empathy, within architecture (and more broadly within the design sciences) the idea of the reflective practitioner has been put forward [122]. In contrast to pure academic knowledge, the practitioner is said to build knowledge in action through action on the world and reflection on the effects that their actions have. This reflection may be especially important for architects, given the overarching impact their designs generally have, the overall difficulty of changing underlying designs, as well as the big separation between the blueprint and the final operational system. By looking at the relationship between the actions taken during the design and the final outcome, the architect can revisit and tune the design process. Parallel to this, reflection on preconceptions is an important aspect during the design process. As discussed in detail in Alexander's seminal work, *Notes on the Synthesis of Form* [1], the architect's language, the fundamental building blocks, and the tools used during the conception of (architectural) designs shape both the interpretation of requirements as well as the solution space that is explored. Designers are thus advised to go beyond "simple" analyses based on words, disentangling the design problem into competing forces, and using diagrams to explore the problem and solution space. This is done by creating small diagrams of tightly coupled competing forces, which can later be aggregated into full solutions in a bottom-up fashion.

Again, similar to the issue of the absence of empathy from the hacker ethic, a *reflective* attitude also seems to be missing. Besides the above notes on the importance of reflection for both learning and solution exploration, reflection may also play an important role when it comes to the integration of ethics within the development cycle (which is related to the concept of empathy). As noted by Lessig in *Code v2* [87], the design of a system can act as the law of the land and encode unwritten constraints and laws into computer systems that are different from those that are part of the traditional body of law. As such, in line with Lessig, we think that architects need to reflect on various difficult questions, such as the concept of intellectual property and its (excessive) enforcement, the impact of business models on privacy, and end-user control of technologies that are becoming increasingly complex. Also, ethics are embedded into architecture, and as a result security architects should start early with reflecting on such concerns and integrating mitigations into the design of systems (as later adjustments will be much harder). Architects should continue to reflect on these concerns throughout the lifetime of a system.

4.5 Security manager's attitudes

The security manager or lead is (stereotypically) concerned with questions of money, but this can be seen as a proxy for the continuity of an organisation, which relates to controls that support stock-owner protection, as well as activities whose objective is continued cash flow in the face of changing external forces [113]. The indirect management model of most publicly-traded companies (and of various organisations that are not publicly traded) requires some kind of control structure to ensure that management and employees are looking after the interests of the owners, and not after their own interests. These controls often include independent audits, liability of management for company actions, and requirements around statements of truthfulness and completeness in (yearly) financial statements (e.g. see the US Sarbanes-Oxley Act [135] and similar legislation in other countries). In line with these measures, a *transparent* attitude seems to align well with good governance. Besides the apparent value provided to stock owners, this also appears to provide value to other stakeholders. Employees are in a dependent position, and transparency (around policies, direction, and more) could help build trust.

Given the nature of security breaches, it is generally considered helpful to engage in data sharing of detected attacks, including the modus operandi of attackers. Buy-in from the top for such efforts is vital for such efforts to be successful [148]. Legislation concerning privacy is also pushing for greater transparency in the area of data processing (e.g. see the EU GDPR [37]). Note that, although ideals of public disclosure of vulnerabilities and free distribution of information are values that have become common in the security community, they follow principles that are widespread in communities such as journalism. Of course, the concept of free speech also wasn't invented by the security community.

Next to transparency and the instigation of appropriate controls, managers are responsible for setting goals and long-term vision. One issue faced by managers is the "innovator's dilemma" [21]. It is similar to Romein's "law of the handicapping head start" [117], whereby the ingrained organisational culture, production processes,

and workforce that made initial success possible work against future success that requires a different approach. As a part of the risk management and business continuity at the forefront of management's planning, such issues should receive attention, as they can impact the sustained profitability of the company. From this perspective, one part of the task of management is to instigate change where needed. We think this requires a *proactive* attitude from the leader in order to increase the odds of achieving sustained change. The importance of being transparent and proactive is further illustrated by the suggested reading list of Stefan Lueders [94], CERN's security officer: *The Art of War* [46], *Animal Farm* [107], *1984* [108], and *Who Moved My Cheese?* [71]. From an organisational perspective, he notes that it is very important to be aware of issues around getting people to adopt new techniques and processes, as well as the problems inherent in security systems, especially where they automatically enforce security policies in an autocratic manner.

Summarising the above discussion on attitudes deemed desirable for specific steps of the systems development life-cycle, we see that these attitudes go beyond a security mindset focussed on breaking things. Additionally, the attitudes are specific to their respective roles and the places of these roles within the development life-cycle. To some extent, this should not come as a surprise: many of the attitudes are not "natural neighbours", e.g. empathy vis-a-vis being critical or finding fault in things. We put forward these results as the primary argument for abandoning the concept of a unitary security mindset, to be replaced by the idea of multiple attitudes that are each conductive to helping achieve one aspect of the development of a secure product. Note that this is not the whole story, as educational designs are most impactful when they focus on all three of attitudes, skills, and knowledge. Next, we describe, at a high level, a collection of task-supporting skills that we believe are aligned well to the archetype's attitudes described in the preceding paragraphs, as well as concepts that may be relevant to the shaping of one part of a broader body of knowledge.

5 SKILLS

As with the derivation of attitudes, this analysis is built around the roles and responsibilities that we associate with the location of the archetype within the development process. This collection of knowledge and skills is far from complete and does not claim to be canonical (although this is a goal for future work). Instead, the objective is to illustrate how the teaching of attitudes could be supported by the careful choice of other elements in the curriculum, as well as providing hints of ways in which certain attitudes might influence how students approach (collections of) tasks and task sequences. The knowledge and skills covered in this section are specific to the given roles and they do not preclude curricular coverage of "broader" (i.e. less in-depth) knowledge and skills. As noted previously, generic frameworks such as the CISSP CBK [47] and CSEC2017 [72] have a function, especially when conceptual understanding and application across contexts are emphasised. How they can be integrated into a curriculum is outside the scope of this paper, but some approaches are front-loaded theory, just-in-time delivery (which can be hard to get right!), and something in between. Also, note that in a complete curriculum both a broad knowledge base and in-depth practical experience may play a part.

5.1 Security analyst's skills

For security analysts, there are some methodologies that may provide an important basis for guiding their role in the development process. Examples of generic approaches are FHM (flaw hypothesis method) [92, 141], IDART red teaming [145], FPVA (first principles vulnerability assessment) [81], and ITAF (information technology assurance framework) [62]. Fundamental principles such as attack vectors, scoping, and relevant representations are covered, as well as how they can be applied. The process of security analysis can be split up into the following task elements that broadly match with the above methodologies: scoping the problem and determining which areas/elements of the system are to be studied, determining what views (i.e. perspectives) to take throughout the analysis, spotting security problems and identifying underlying causes of these vulnerabilities, and, finally, automating the analysis process, where possible, to ensure that more aspects of the system can be tested, to allow regression testing to be performed, and to distil analyst know-how into conceptual and procedural tools.

The approaches advocated by the methodologies could help to provide the analyst with a critical and independent attitude. Vice versa, such an attitude may support analysts by making them less susceptible to forces that seek to undermine the process of uncovering vulnerabilities and determining the general security state of the system. Specifically, explicit scoping could help force the identification of security assumptions, which is vital for a scientific approach to security evaluation [52]. Thinking of the appropriate views to take may drive a discussion about different measuring instruments, including a discussion of the representativeness and validity of those carrying out the assessment, i.e whether they can adequately emulate enemies that are deemed relevant.

Naturally, the ways of looking influence what is found. Once a potential vulnerability has been found, a critical look at the potential impact is required, including how much is actually known about the impact in practice. For findings to be actionable, it is also important for analysts to engage in root-cause analysis (e.g. 5 whys [64], fault-tree analysis [49], why-because analysis [82]). Given the nature of such an exercise, it seems important to keep a critical attitude throughout (e.g. see threats such as inadequate models and simplification of causes [55]). During the process of tool development, an independent attitude appears important to prevent excessive reliance on tools and to ensure that tasks can still be completed manually. A critical attitude may also be important, as tool usage raises questions about their reliability, whether they continue to measure what they should be measuring, and whether the positives of introducing the tool outweigh the negatives.

The examples above of different (sub)tasks of security analysis indicate that these tasks themselves might support a critical and independent attitude, and vice versa.

5.2 Security engineer's skills

As with building codes, security engineers may require norms for the construction of their systems. We believe all systems engineering projects should start with the identification of relevant security norms (including national, international, and de-facto standards). There are collections of common norms such as the ISO 27000 series of standards and NIST's SP 800, SP 1800, and FIPS series. There are

also much more specific norms. For example, relevant norms for secure embedded coding include the MIRSA C development guidelines [98], the CERT C coding standard [127], and JPL's ten rules [57]. The humble attitude of a good engineer could help prevent not-invented-here syndrome and the vanity that they can single-handedly develop better systems. The history of security is littered with cryptographic systems that were custom-built, and that were quickly broken once they had been reverse engineered.

Additionally, testing is required. Specifically, for software and hardware there is a need for unit, integration, and system-level testing. Also taken to be important for the release process is the implementation of quality gates, where formal sign-off is given on the basis of predefined quality standards. Once a product is out in the field, it is close to inevitable that security problems will be found. As such, engineers of a system might have to recognise their own fallibility and implement mechanisms and processes for responding to reported vulnerabilities (they can be guided by standards such as ISO 29147 [68] and NCSC's responsible disclosure guidelines [100]).

All of the above steps appear to align with a frame of mind that actively acknowledges the fallibility of human beings. Thus, it seems highly desirable that a security engineer has a humble and restrained attitude as this may provide the understanding and motivation to comply with these processes. In the other direction, properly implemented vulnerability disclosure processes, quality gates, (automated) testing, and construction norms could help catch errors and indicate to engineers that they are not perfect.

5.3 Security forecaster's skills

Within the role of security forecaster, there are a handful of threat modelling frameworks that are widely used within the (software) security community. These include Microsoft's threat modelling method [123, 124, 132] (and the very similar architectural risk analysis [97]), Trike [84, 119], and PASTA [133]. Each of these methodologies is built around information flows (and to a certain extent control flows) as salient aspects of a system that deserve special attention. The make-believe world that might be constructed to put a system architecture through its paces is built around how data (and other elements) flow through the system. The goals of the participants could be to try to attack or defend the flow of information on the basis of one or more abstract descriptions that represent the world (e.g. data-flow diagrams, access-control matrices, etcetera). On the basis of these descriptions, the players might make a "move" by identifying potential vulnerabilities (through techniques such as STRIDE [75], HAZOP [128, 129], and attack patterns [29]). These could be linked together to indicate the dependencies of different vulnerabilities. The possible "outcome" of the game may be a collection of attack stories (i.e. scenarios) that are derived from attack trees showing vulnerability dependencies. "Scoring" could take place by looking at the apparent likelihood and impact of stories of how attackers might navigate a path through the imagined system. When scenarios are deemed to be too far-fetched or unlikely, it might be necessary to build out part of the abstract prototype into a physical form where an attack can be demonstrated in practice, and/or (historical) data or experiences of similar games/simulations may turn out handy. It is important to note that these simulations cover one specific aspect of the system, and that as such they only provide answers with respect to that aspect (and even then only very preliminary answers).

All of these steps and the general process of security simulation seem to require a level of wilful suspense of disbelief in order to be fully present within the specially constructed game world. As noted previously, this appears to require a playful attitude. Of note is that the structured approach of both modelling the system architecture in terms of information flows, as well as the clear rules on how to proceed through the generation of possible vulnerabilities, the creation of attack graphs, and the distillation of attack scenarios with the greatest impact and likelihood, all provide a guiding framework that could support the players in creating a world for themselves to play in. Also, as the game is not bounded in a traditional sense, and does not have any fixed outcome, it seems important for players to possess an inquisitive attitude. While the simulation methodologies do not provide any "levels", they do provide players with a map (systems flows, invariants, goals) and compass (threat-elicitation techniques) that may support them in both situational awareness and direction finding or treasure hunting. As with the other archetypes, we see indications that the attitudes related to the archetype are both supported by and support their (sub)tasks.

5.4 Security architect's skills

Security architects need to recognise that a problem exists out in the world and they need to iteratively come to clear specifications regarding how the gap can be closed. This process starts with a requirements analysis. For understanding the problem, it appears to be important that architects are able to empathise with the users of a system. As such, an empathic attitude seems helpful. Additionally, there is a variety of methods, common in the field of human-centred design [60], that support and enable architects and designers to better understand their users (which should in turn support an empathic attitude). Such methods include the creation of personas that are representative of user groups, the shadowing of users in their natural habitat, several forms of role-play, tools to emulate handicaps, and more. These can be supported by rapid prototyping, especially in contexts where the requirements are unclear due to factors such as the novelty of the project.

Having collected the requirements (which should generally be in the form of negative requirements, as positive quality requirements are hard to test), architects get to work exploring solutions to the problem [1]. In doing so, we think they should adopt a reflective attitude, especially with respect to the influence of preconceptions on how they approach this exploration, and how this limits their effective search space (e.g. see the dynamics of classification systems [14]). By being aware of this effect, an architect can focus their attention on the nexus of competing requirements, and might address groups of forces that are most fiercely in conflict. Solutions could be explored through sketches embodying the structure of the underlying forces. Promising approaches may then be combined into an overall solution direction for the problem. This approach to handling complexity is similar to that identified by Herbert Simon: hierarchies based on a division of system elements on the basis of minimal "crosstalk" between parts [125].

Once the solution directions are clear, the architect should keep in mind the importance of preserving the conceptual integrity of the

system design [16]. One approach is to avoid design-by-committee. Another approach is to iteratively refine the initial idea into a high-level informal description, and afterwards refine this into a formal specification or blueprint of the system, all the while going back and forth to check whether the original design intent has been preserved. One common approach for informally describing systems is through block-level diagrams [61] and pseudocode. More formal approaches to such descriptions are based on a mathematical foundation, such as temporal logic [56, 83], first-order logic [70], or typed higher-order logic [10, 103]. For the approach of refinement, a reflective attitude seems helpful as it appears to support an awareness of the design process itself. For the tasks of creating formal and informal specifications, an empathic attitude may help recognise the need for such specifications, as well as to help understand the issues that users of such specifications might run into. The process itself might help to instil these attitudes, given that iterative refinement can force the architect to go back and look at the previous step or layer of abstraction, which may encourage at least some form of reflection. Additionally, being forced to formally describe the solution approach may help the architect understand how and why informal specifications might be ambiguous, which could lead to (hints of) an empathic attitude.

5.5 Security manager's skills

There are numerous management and control frameworks that may help security managers in structuring their approach. Examples include COSO-IC [115], ISO 27001 [66], NIST's cyber-security framework [104], and NIST SP 800-160 [118]. The standards are generally based around the concept of the goals to be achieved, in the form of the selection of controls and the writing of policies. They also commonly include an approach based on risk management, which is specified in more detail in standards such as COSO-ERM [116] and ISO 27005 [65]. Another fundamental part of management tends to be the planning of tasks: what needs to be done by whom and by when, based on methods of optimising throughput (or other variables). Consequently, tracking progress is important, as is status reporting (including yearly and monthly reports, daily stand-ups, kanban, etcetera), and closing of projects.

For setting goals, standards such as COBIT [63] can help to align security goals to the objectives of the organisation, and control baselines such as the CIS controls [22] and ISO 27002 [67] can help to select security controls in line with security objectives. Risk management methodologies such as OCTAVE [19] and FAIR [41] can support professionals and students in acquiring and adapting a process for the identification, quantification/qualification, and treatment of risk. Classical tools such as Gantt, PERT, and critical path charts, as well as work breakdown structures, can help split up projects into tasks and help to sequence those tasks appropriately. Other common practices may be relevant, such as status reporting and incident, complaint, and ticket management, which form the pillar of the ISO 9001 quality management standard [69] and are a key element of the ISO 27001 information security management systems standard [66].

The compliance landscape and the compliance burden require organisations to be in control, and to be visibly so. A transparent attitude from management may help to achieve this. Additionally, transparency and clarity in policies, procedures, processes, and mechanisms may facilitate the identification of problems and inefficiencies, and it could further enable employees to be kept informed of the demands made of them. The attitude of management being proactive appears helpful both for ensuring challenges resulting from a changing environment can be faced, as well as helping managers to instil the need for security policies and associated behaviours (which employees often view as a burden).

Besides their own attitudes and their project and portfolio management tasks, managers may also be held responsible for nurturing a culture where those with respective (security) roles are encouraged to adopt appropriate attitudes and methods, as well as more directly inspiring appropriate attitudes and methodologies in their juniors. As a result, management roles should probably be taught only after those of the "subservient" roles have been completed. We will talk more about this and other practical concerns in the next section. While it is not the only role that raises questions around how attitudes and methodologies can be taught, it does raise important challenges such as the importance of an authentic training and assessment context, and how to disentangle the teaching of security management principles form the development process itself. We will cover various practical issues in the next section.

This section has illustrated how the steps that are part of various methods and methodologies may help to construct the roles and responsibilities that are part of the systems development life-cycle. We have provided indicators of how the practical steps in the development process could be related to the attitudes identified in the previous section. This hints at the need for taking an integrated approach to the teaching of secure development. In the following section we will look at several important practical questions related to the teaching and assessment of attitudes within a curriculum.

6 IMPLEMENTATION

There are several important questions when it comes to operationalising security education with an attitudinal component. The ones we discuss in greater depth in this section include the question of how attitudes can and should be assessed, how to sequence the various attitudes (and their related roles and tasks) within a curriculum, and whether there are any teaching techniques that contribute to the nurturing of attitudes. The emphasis within this section will be on formal security curricula, although part of the discussion will be relevant to problems surrounding security awareness training. We do not present a complete curriculum, and many questions are still open. These and more are covered in the next sections which will go into open questions and promising avenues for further research.

Note that research with respect to security education tends to take the form of specific teaching activities (e.g. [31]), platforms for approaches based on CTF-like challenges (e.g. [35]), and high-level discussions on what content should be taught (e.g. [138]). As such, the discussion in this section refers to research done in the broader context of didactics, learning sciences, pedagogy, instructional design, and curriculum theory. While claims have been made that security is special, and requires a different teaching approach, here we take the position of building on general education research into questions involving assessment theory, curriculum development, and learning mechanisms and processes.

6.1 Assessment

Assessment of attitudes is different from the assessment of knowledge and skills. While knowledge can be assessed by asking for reproduction of the relevant knowledge within a given constrained time window, and skills can be assessed by asking for the creation of an artefact for which the skills are a prerequisite, attitude cannot be measured directly. Instead, the common approach settled on over the previous decades is to measure attitude by providing the subject with an object or situation and asking them to rate their favourability on a Lickert scale [96]. The answers are mapped to a continuum, and the result is a direction (positive or negative) and an intensity on this continuum. This approach to attitude measurement has several important implications for assessment. As the results depend on self-report, the bias resulting from tying grades to the assessment and the dynamics relating to social expectations imply that formative assessment may be preferable over summative assessment of attitude. Additionally, anonymity in assessment of attitudes also seems preferable. It is not a surprise that surveys related to employee satisfaction are generally both anonymous and do not carry any punishment or reward for (not) filling them out, or for answering in a particular manner.

Although not directly related to the assessment of attitudes, the assessment of both knowledge and skills may indirectly influence attitudes, and assessment should thus be shaped appropriately. Hands-on skills might be best assessed by looking at one or more concrete deliverables, preferably ones that naturally allow assessment of both the end-product as well as the process leading up to it. A deliverable that allows feedback to be given and integrated throughout the process could facilitate an emphasis on the learning process over teaching to the test. Note that evaluation of professional products is inherently a subjective endeavour. Trying to fully objectify the assessment criteria may lead to an impoverishment of the curriculum and a push towards low-level learning objectives. Instead, having grading indicators in the form of rubrics or checklists and having multiple people check the work (e.g. the student themself, peers, and multiple instructors) could provide both a level of transparency as well as intersubjective grading supported by triangulation of results. Additionally, all those involved are commonly recommended to continue to calibrate their respective expectations and to make those expectations clear to each other, e.g. in the form of calibration sessions between assessors, and through grading rubrics and frequent (peer) feedback.

Note that, in a context of theory-heavy education, there is often both more experience with the assessment of knowledge and this knowledge tends to be easier to assess due to the naturally lower-level learning objectives associated with it. Common psychometric techniques can be used to study the validity and reliability of assessments, provided that the general checks on the alignment between learning objectives and assessment criteria have been performed, as well as checks between assessment criteria and the exam coverage, answer model, etcetera. One warning that is appropriate in light of a curriculum in which attitudes take a central position is that multiple choice answers or similar stimulus-response-type questions associated with a behaviouralist view of education should most likely be avoided. Instead, open-format questions testing higher-level cognitive processes through more elaborate responses are probably preferable, also because of the signals they send. The choice of assessment method and its communication to students can influence the learning approach and culture that develops among students [45], and assessment methods should therefore be carefully chosen. Low-level assessment can have a negative impact on student motivation. As such, it seems inappropriate for a curriculum that is not built solely around knowledge.

6.2 Sequencing

There are many ways of sequencing a curriculum and many theories that can support the choice of how to order learning objectives, assessment, and learning activities. Two historic classic theories that we will use here to illustrate one promising approach to ordering are concreteness fading [17, 42] and scaffolding [146]. A related more recent theory is cognitive load theory [28], which is conceptually linked to the idea of flow [27] as well as to the idea of the zone of proximal development [140]. Concreteness fading states that beginners should start with concrete examples and exercises, and should slowly and gradually move to more abstract concepts and contexts. Scaffolding implies the presence of social support at the start of a learning activity, which is gradually removed as the learner becomes more capable. Cognitive load theory seeks to explain why these and related mechanisms are effective. Comparing the cognitive processes of experts with those of novices, the idea put forward is that beginners need various kinds of support and a carefully designed learning environment because—among other things—they make greater use of their short-term memory than experts, who use long-term memory more extensively. As a result of short-term memory limitations, beginners are more likely than experts to experience cognitive overload in the same situation. Flow is a theory that seeks to explain learning from the perspective of motivation. When plotting the difficulty of an activity against the capability of a student there is a "flow channel" in which there is optimum motivation. The postulated mechanism for increasing capability and maintaining motivation is to ensure that at no point in the learning process are activities too easy, and ensuring that at every point an activity is performed that is slightly harder than is easily doable with the current abilities of students. The difference should present an interesting challenge to students, and allow them to grow their capabilities, at which point the tasks can be made slight harder. The zone of proximal development is the area to which the student can "move" only if they are provided with additional assistance. These theories can serve as lenses through which to view the sequencing problem. As these theories are primarily based on the perspective of knowledge and skills, we will first look at the problem from the perspective of the skills relevant to the different archetypes and their tasks. We will later illustrate that the resulting curricular sequence also appears to make sense when looked at from the perspective of attitudes.

Taking a life-cycle approach to security (as espoused by NIST SP 800-160 [104], Microsoft's SDL [93], and others), it seems logical to sequence the curriculum chronologically according to the steps in the development life-cycle. However, given that the earlier stages in the process seem much more abstract, less tangible, more dependent on context, more open, and less amenable to real-world scaffolded examples, this does not appear to be a promising approach in light

of the theories just discussed. Although a theoretical overview of the security development life-cycle might make sense at the start to sketch the context, we claim that a practical chronological approach is not the way to go. Instead, the coverage could be ordered from concrete to abstract in reverse-chronological order, i.e. analysis, engineering, simulation, architecture, management. This provides the advantage of starting out with concrete objects of study that are widely available, thus making initial scaffolding potentially easier. It also seems like a more natural extension of front-loaded theory that focusses on concrete examples of how systems may be attacked and defended. Furthermore, we think that archetypes related to analysis come before those related to synthesis: learning ways of seeing before building the objects of analysis is a natural progression. Aside from this ordering of archetypes, there is also the question of how to order the teaching of the tasks those archetypes are expected to do. Although within one archetype the cycle is shorter than the full security development life-cycle, similar arguments can be put forward for starting with the last step in the task sequence, as it tends to be the most concrete. Do note that starting with an overall demo in chronological order, as well as overarching theoretical coverage for each archetype, might help put things in context. For both the ordering of the archetypes, and the tasks of those archetypes, the approach of starting at the end allows repetition of the final steps by having students study each additional step on its own and then going through the other tasks in the development process until a final deliverable is produced. This enables repeated practice and reinforcement of what has been learnt.

Besides the perspective of didactics, it is interesting to take a slight detour and look at the sequence just sketched from the perspective of maturity models of secure development. Models such as CMM [59] view maturity as the extent to which processes are in place and are being followed, defined, and optimised. Another way of looking at the maturity of implementations of security within the systems development life-cycle is by observing how security may be introduced into an existing development process. Often the first thing that is done is not some abstract "security by design", but a security audit mandated by legislation (i.e. shoved down a team's throat) or requested by the head of security to illustrate the scale of the problem and to ask for budget from upper management. Similar arguments can be made as to why engineering will become part of the life-cycle before a focus on the underlying architecture is introduced. The parallels between the curricular sequence and the sequence in which security may be seen to be introduced into a development life-cycle seem to support didactic approaches such as service learning. It also provides a general sanity check on the appropriateness and relevance of the setup of such a curriculum.

As noted previously, the theories for curriculum sequencing that we have describe here are not focussed on the teaching of attitudes. Frameworks for explaining learning tend to focus on the cognitive element instead of the affective element. This difference in focus is also apparent in the relative popularity of Bloom et al.'s cognitive taxonomy [13] over Krathwohl et al.'s affective taxonomy [78]. The revised taxonomy for the cognitive domain is as follows: remember, understand, apply, analyse, evaluate, create [5, 77]. This matches with the setup of front-loaded theory and covering analysis before synthesis, as sketched in the previous paragraphs. The revised taxonomy also includes knowledge dimensions for each level, with meta-cognition as the highest level. The affective taxonomy is very different, and it has not been revised. Krathwohl et al.'s affective taxonomy is structured as follows: receive phenomena, respond to phenomena, valuing, organisation, internalise values [78].

While the usefulness of such general taxonomies to specific disciplines can be and has been questioned (especially as related to helping teachers structure their education [32]), they nonetheless provide conceptual tools to analyse a curriculum (of course there are many other such tools). Bloom et al.'s cognitive taxonomy indicates dependencies between knowledge and skills, while Krathwohl et al.'s affective taxonomy indicates how attitude change is not a sudden, but a gradual shift from being a passive recipient to being an active entity through a process of enculturation. Looking at the concept of attitudes through the lens of internalisation of affect does not directly point us toward an ordering of when to teach which attitudes, but there are several other indicators that provide some direction. Broadly speaking, the traditional "hacker ethic" relates to independent and critical attitudes, and it can be argued that these come most naturally through the process of enculturation within a traditional security community as well as in academic environments. As such, similar to the parallels between what is naturally introduced first in the software development life-cycle (a security audit), emphasising the critical and independent attitudes at the start of the curriculum would fit with this. The other attitudes can be argued to be in the order from most to least affinity with the classic stereotype of many "techies". However, we argue for this ordering of attitudes on the basis of what we see as their clear alignment to the archetypes and the logical ordering of when their tasks and related skills would naturally be taught. In other words, while the perspective of enculturation is not a strong indicator for taking a specific approach, it does provide some support for the ordering taken. The relevance of maintaining alignment between attitudes and skills in the teaching of roles derived from the archetypes is of possibly greater importance.

Note that if the approach chosen involves "working backwards" by teaching the last task first (based on using worked examples and context from the other preceding tasks, while gradually increasing scope), this raises the question of whether such an approach matches with the perspective of attitude formation as a gradual process. There are several indicators that this is indeed the case. Firstly, the setup of "working backwards" and having the preceding set of tasks demonstrated can allow students to perceive the teacher at work (who is expected to both possess the relevant attitudes as well as to actively profess them) before responding to phenomena. Having the process repeated may allow students to gradually become encultured. Secondly, it seems that the tasks that most call for the attitudes of relevance to the archetype are those that are at the beginning more so than those at the end. For example, identification of requirements seems to call for more of an empathic attitude than the other architectural tasks; selection and conformance to norms seems to require more of a humble attitude than the engineering tasks of fixing systems or writing tests. By covering these earlier tasks at the end of the instruction sequence, students should have both been exposed to more worked example and demonstrations, and should have had the ability to apply the attitudes in practice in the other steps. In summary, like the ordering of archetypes, for the ordering of tasks there seem to be a couple of arguments for the

ordering presented when looked at from the perspective of teaching attitudes. The argumentation is not as strong as the arguments on the basis of skills, but this is to be expected when we assume that there is greater focus on, and experience with, knowledge-based and skills-based education.

6.3 Teaching

There are many teaching formats for both knowledge and skills, including general approaches such as lectures, group work, lab work, videos, exercises, debates, poster sessions, paper writing, student-led presentations, fieldwork, observation, and more. There are also formats that are described in much more detail, with some thoroughly formalised. For example, the "Training Within Industry" process [30, 33] used in the U.S. training efforts in industry during World War II, and later exported to Japan for the rebuilding of the country (becoming one of the pillars of what is now fashionably called "lean"), used heavily scripted lessons to ensure uniform standards and to make it possible to teach the teachers of the teachers. However, most education today appears to consist of teachers selecting their own teaching methods and building a personal repertoire of such methods. Cookbooks of "teaching recipes" (i.e. formats) with a plethora of approaches are common. However, these seem to often lack inclusion of guidelines regarding when their use is appropriate. Furthermore, as with curricula, they appear to be focussed around the requirements of knowledge over skills, and especially on the requirements of knowledge over attitude. This leads to the question of identifying appropriate hands-on teaching techniques appropriate for teaching the security skills discussed in this paper, as well as the question of whether these techniques sufficiently support the development of attitudes.

Before looking at teaching techniques related to attitudes, we will first look at techniques from the perspective of skill acquisition. We will emphasise general prescriptive frameworks that have seen widespread use. The discussion is not about specific teaching techniques, but instead it is about underlying frameworks to structure many different kinds of teaching activities. One such overarching framework for lesson organisation is Gagne's "nine steps of learning" [43], which is based on an information-processing view of human learning, and which prescribes that lessons should consists of the following nine stages: get attention, describe objectives, stimulate recall of prior learning, present content, provide learning guidance, elicit performance, give feedback, assess performance, enhance retention and transfer. Allen's 1919 model of instructional organisation [2] predates Gagne's model by around half a century, and consists of the following steps: preparation, presentation, application, and testing (or inspection). It was developed in the context of vocational training and focusses on skill acquisition. An adapted version was used in the Training Within Industries project discussed earlier, with the following steps: prepare the worker, present the operation, try out performance, follow up. Both TWI's and Gagne's model include initial motivation of the student, activation of prior learning, demonstration and explicit instruction by the teachers (including a clear description of expected performance), practice coupled to feedback, assessment of performance, and ways of encouraging transfer to practice. This sequence is even compatible with more recent constructivist approaches to learning [15], provided that attention is paid to conceiving of instruction as the active creation of mental models by the student instead of seeing teaching as the direct transfer of knowledge (which may be facilitated by the checking of prior learning in addition to the activation of prior learning).

To achieve each of the individual steps sketched in the previous paragraph, there are many different teaching techniques that might be relevant. Techniques that appear promising for encouraging student motivation at the start of a lesson or unit include coverage of recent security vulnerabilities and other news [39], an illustration of how their behaviour is insecure [80] (for example by using examples of student code), demonstrations of contrasting approaches whereby the approach to be taught is clearly superior, and examples of how a naive "self-evident" approach does not work (TWI used the underwriter's knot to show that teaching by merely telling does not work [30]). For activation of prior learning, short quizzes, exercises, as well as in-class oral assessment may be useful. The questions used can be directed at the uncovering of common conceptual misunderstandings. Additionally, a bridge between the activation of prior learning, clarifying learning objectives, and explicit instruction is the use of key questions [144] that form the conceptual foundation of a field of study, as well as the use of Socratic dialogue [109], whereby a chain of questions is constructed that seeks to lead learners to a goal in a step-wise manner.

For explicit instruction, teaching techniques such as lecturing have gotten a bad reputation, but they can be effective vehicles of knowledge transfer, provided that visuals have an appropriate structure [85] (they should not encode information as a linear sequence), that dual-coding principles are followed [23] (transmit information both verbally and visually, and don't have the exact same information on both channels), and that sufficient interaction is present [131], e.g. through questions to random students to check for understanding. Another important part of explicit instruction, especially when it comes to the teaching of skills, appears to be the inclusion of demonstrations. A common approach in, for example, teaching medicine, is first showing without verbalisation, followed by showing with verbalisation, then asking the student for a verbal walk-through directing the process, and finally having them practice the skill while they verbalise their actions before they are carried out [3].

The idea of practice coupled to feedback is in line with the view of formative assessment being a type of interaction between student and teacher, student and student, and internal metacognitive monitoring [12]. In the context of teacher-student interaction, feedback should be provided on the quality of the product, but also on the process taken, as well as on the underlying motivation of the student [51]. On the topic of peer feedback, "80% of the feedback learners receive in the classroom is from other learners, and 80% of that feedback is wrong" [106]. As such, students should be supported in their ability to give feedback to one another, for example through structured rubrics, key points to emphasise, and standard work templates (with all of these also being valuable for supporting teacher-student interaction). For metacognitive monitoring, students should be provided with questions that elicit reflection before, during, and after task completion [50]. Later on, once the student is deemed ready, their performance can be summatively assessed. Assessment techniques that seem appropriate have been discussed

in a previous subsection. In short, the assessment could consist of realistic tasks for testing skills, deep questions for testing knowledge, and anonymous formative assessment for testing attitude. To encourage transfer to other situations, such as the workplace, tools like metaphors and reference cards may be appropriate. Additionally, students could be visited in their workspace, observed at practice, and asked to repeat specific processes and procedures.

The teaching approaches discussed focus on the teaching of skills, but they may also support the teaching of attitudes. Properly integrated, techniques for teaching skills appear to also be applicable to the teaching of attitudes. Generally speaking, the teaching of attitudes might be performed by having the instructor model desirable behaviour. A culture that is conductive to the desired attitudes may also be helpful: appropriate models could be provided both by the instructors, as well as by fellow students. Specific methods may be used to complement the approaches described for the teaching of skills. One such method is "role modelling" [101], whereby students learn from the observation of role models. For role modelling to be effective, it is important that the implicit is made explicit, and that improvements to institutional culture are made [26]. Note that role modelling can happen both in the context of the performance and teaching of tasks within a formal security curriculum, as well as outside such a framework. Besides supporting the introduction of attitude formation within the halls of academia, our framework may also support the identification of, and connection to, other spheres and communities that play a role in shaping attitudes.

7 DISCUSSION

The archetype framework, selected attitudes, taxonomy of roles, and questions around implementation discussed throughout the paper have several possible implications for security practice. Among these, the debates concerning approaches to teamwork, curricula, and culture are especially important.

7.1 Teamwork

The concepts presented in this paper may be used to analyse the performance and composition of development teams. Roles within existing teams can be analysed for their match with the taxonomy of roles given in this paper in order to spot any gaps. The same goes for setting up new teams, where the taxonomy can function as a secondary check. When any gaps are found, the attitudes and skill sets can be used to validate the appropriateness of potential team members. They may also be used in the evaluation of existing employees. Note that not all teams may need all roles, as certain roles could be outsourced.

Other team dynamics that could be explained on the basis of the archetype framework presented here include possible conflicts of interest that are likely to appear as a result of divergent attitudes in different archetypes. One possible reasons for such conflicts is the presence of mutual misunderstanding due to different world views.

7.2 Curricula

As a design science, curriculum design is in the uneasy situation that many knowledge claims are contextual. Choices regarding what and how to teach the next batch of security professionals are very specific to the lecturer concerned. Although we may not want to leave the choice of materials and methods solely to expert judgement and politics, alternatives are hard to come by as we are working with systems that do not yet exist. Complementing the value judgements of individuals and committees, clear identification of desired effects, measurement of their achievement, and feedback into the design process may have a role to play.

Also, before we can look at creating a design vision on security curricula, it is important to address the assessment question. Specifically, both filtering and teaching on the basis of attitudes is harder than teaching and filtering on the basis of knowledge and skills. As such, rigorous assessment of attitude is an important open question for security education. A parallel concern relates to the implications for security careers and career progress, where the possibility of selecting on the basis of attitudes requires substantiated choices.

7.3 Culture

It may be necessary to not only look at individual attitudes but also at organisational attitudes. Regulation is often seen as one way influencing the behaviour of organisations, and it is generally accepted that senior executives have extensive influence on the culture within an organisation. However, the influence of regulation on organisational culture is less clear. There is a nebulous relationship between attitudes and regulation, and between intrinsic motivation and extrinsic reward and punishment. As Leveson notes [88], the steam engines on boats stopped exploding as a result of federal laws. However, as behaviour within the financial sector indicates, while laws may be a necessary condition, they may not be a sufficient condition for appropriate activity.

Another interesting target is looking at cultural leaders or influencers and the role that they might play in the adoption of the identified attitudes. Note that activities looking to increase the adoption of these attitudes could also have positive knock-on effects within the broader culture in the software development field. When working on this, it is important to emphasise moving from a security culture of penetrate and patch to one that has greater maturity. One possible area to address first is the disconnect between the approaches and views advocated by usable security researchers and the hubris, rock-star culture, and "0day" glorification seen at some security conferences. Either way, careful attention should be paid to the question of who creates (or drives) the adoption of particular attitudes and who maintains them.

8 FUTURE WORK

There are several open questions to focus on in future work:

1. Given the framework that is described in this paper for the development life-cycle, what are the attitudes and skills necessary for different roles within operations? Such work may find inspiration in, among others, crew resource management [143] practised within the aviation sector. A quick scan of common security-related operational activities—installation, configuration, support, monitoring, recovery—indicates that relevant attitudes are possibly divergent. Additionally, having investigated these attitudes, it may also be interesting to take a look at how the attitudes identified as relevant for development and operations relate to DevOps teams.

2. What mental models best support the attitudes and skills identified in this paper? The 4C/ID instructional design model [137] describes the key role played by mental models in enabling transfer of complex skills to novel contexts. In the field of security, there has been descriptive work into the mental models of novices (e.g. [139]), and, in the field of safety, the accident model STAMP [90] is based on mismatched control models in subsystems. Further work would be valuable to identify (causal) relationships between mental models and the introduction and detection of flaws and bugs throughout the development lifecycle. The identification, extraction, and teaching of reference models would be a follow-up challenge. Both mental models directly supporting the application of skills in a known context, as well as mental models that play a more significant role in enabling far transfer, are relevant objects of study.
3. To what extent can the framework be further validated through additional interviews with developers and managers from the industry? Orthogonally, can instruments be built and validated to measure the different attitudes and skills discussed in this paper? Once the framework has received further validation and once the key concepts have been further operationalised, ways of integrating aligned attitudes and skills into curricula can be explored in greater depth, and the real-world security impact of graduated students can be studied to compare the approach to one that places more emphasis on knowledge. Parallel to this, relationships with personality traits could be explored.

9 CONCLUSION

We have illustrated why basing security education on the concept of a singular security mindset may be unwise. As an alternative, we have proposed archetypes with attitudes linked to their place within the systems development life-cycle. We have also explored ways to align the teaching of such attitudes with the teaching of relevant skills. Prior to implementation, several issues need to be addressed, including how to assess attitudes and skills, and how to sequence and teach the curriculum. Note that, whereas we have sketched one possible approach, and have illustrated why it appears promising, much research remains to be done on the concept of multiple security-relevant attitudes and its implications.

10 ACKNOWLEDGEMENTS

We would like to sincerely thank the anonymous reviewers for providing valuable suggestions that helped focus the paper. Their reviews also proved helpful in identifying fruitful questions to debate at NSPW. Furthermore, we would like to thank L. Jean Camp for shepherding the pre-proceedings, and Filipo Sharevski for shepherding the proceedings. Our thanks also go out to all participants of NSPW 2018 for the inspiring debates. Lastly, we would like to thank Diederik de Vries for many interesting discussions.

11 REFERENCES

[1] C. Alexander. 1964. *Notes on the synthesis of form*. Harvard University Press, Cambridge, USA.
[2] C.R. Allen. 1919. *The instructor, the man, and the job*. J. B. Lippincott Company, Philadelphia, USA.
[3] L. Allery. 2009. How to teach practical skills. *Education for Primary Care*. 20, 1 (2009), 58–60.
[4] O.H. Ammann, T. von Karman, and G.B. Woodruff. 1941. *The failure of the Tacoma Narrows Bridge*. Federal Works Agency, Washington, USA.
[5] L.W. Anderson, D.R. Krathwohl, P.W. Airasian, K.A. Cruikshank, R.E. Mayer, P.R. Pintrich, J. Raths, and M.C. Wittrock. 2001. *A taxonomy for learning, teaching, and assessing*. Longman, New York, USA.
[6] R.J. Anderson. 2010. *Security engineering*, 2nd edition. John Wiley & Sons, New York, USA.
[7] D. Andriessen, D. Sluijsmans, M. Snel, and A. Jacobs. 2017. *Protocol verbeteren en verantwoorden van afstuderen in het HBO 2.0*. Vereniging Hogescholen, The Hague, The Netherlands.
[8] J.N. Bassili. 2008. Attitude strength. In *Attitudes and attitude change*. W.D. Crano and R. Prislin, eds. Psychology Press, New York, USA, 237–260.
[9] J. Berg, J. Dickhaut, and K. McCabe. 1995. Trust, reciprocity, and social history. *Games and Economic Behavior*. 10, 1 (1995), 122–142.
[10] Y. Bertot and P. Casteran. 2004. *Interactive theorem proving and program development*. Springer, Berlin, Germany.
[11] G.J.J. Biesta. 2013. *The beautiful risk of education*. Paradigm Publishers, Boulder, USA.
[12] P. Black and D. Wiliam. 2009. Developing the theory of formative assessment. *Educational Assessment, Evaluation, and Accountability*. 21, 1 (2009), 5–31.
[13] B.S. Bloom, M.D. Engelhart, E.J. Furst, W.H. Hill, and D.R. Krathwohl. 1956. *Taxonomy of educational objectives*. David McKay Company, New York, USA.
[14] G.C. Bowker and S.L. Star. 2000. *Sorting things out*. MIT Press, Cambridge, USA.
[15] J.D. Bransford, A.L. Brown, R.R. Cocking, M.S. Donovan, and J.W. Pellegrino. 2000. *How people learn*, 2nd edition. National Academies Press, Washington, USA.
[16] F.P. Brooks. 1995. *The mythical man-month*, 2nd edition. Addison-Wesley, Reading, USA.
[17] J.S. Bruner. 1966. *Toward a theory of instruction*. Harvard University Press, Cambridge, USA.
[18] J. Cappos and R. Weiss. 2014. Teaching the security mindset with reference monitors. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (SIGCSE '14). ACM Press, New York, USA, 523–528.
[19] R.A. Caralli, J.F. Stevens, L.R. Young, and W.R. Wilson. 2007. *Introducing OCTAVE Allegro*. Carnegie Mellon University, Pittsburgh, USA.
[20] CEN. 2014. *European e-competence framework 3.0*. CWA 16234-1-2014. European Committee for Standardization, Brussels, Belgium.
[21] C.M. Christensen. 1997. *The innovator's dilemma*. Harvard Business School Press, Boston, USA.
[22] CIS. 2016. *The CIS critical security controls for effective cyber defense (version 6.1)*. Center for Internet Security, East Greenbush, USA.
[23] J.M. Clark and A. Paivio. 1991. Dual coding theory and education. *Educational Psychology Review*. 3, 3 (1991), 149–210.
[24] G. Conti and J. Caroland. 2011. Embracing the Kobayashi Maru. *IEEE Security & Privacy*. 9, 4 (2011), 48–51.

[25] L.F. Cranor and S. Garfinkel. 2005. *Security and usability*. O'Reilly Media, Sebastopol, USA.
[26] S.R. Cruess, R.L. Cruess, and Y. Steinert. 2008. Role modelling. *British Medical Journal*. 336, 7646 (2008), 718–721.
[27] M. Csikszentmihalyi. 1990. *Flow*. Harper & Row, New York, USA.
[28] T. de Jong. 2010. Cognitive load theory, educational research, and instructional design. *Instructional Science*. 38, 2 (2010), 105–134.
[29] D. Dhillon. 2011. Developer-driven threat modeling. *IEEE Security & Privacy*. 9, 4 (2011), 41–47.
[30] W. Dietz and B.W. Bevens. 1970. *Learn by doing*. Walter Dietz, Summit, USA.
[31] T. Dimkov, W. Pieters, and P. Hartel. 2011. Training students to steal. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (SIGCSE '11). ACM Press, New York, USA, 21–26.
[32] J. Dolin and R. Evans. 2017. *Transforming assessment*. Springer, Berlin, Germany.
[33] C.R. Dooley. 1946. Training within industry in the United States. *International Labour Review*. 54, 3 (1946), 160–178.
[34] d.school. 2013. *Bootcamp bootleg*. Stanford University, Stanford, USA.
[35] W. Du. 2011. Hands-on lab exercises for computer security education. *IEEE Security & Privacy*. 9, 5 (2011), 70–73.
[36] W. Dutton. 2017. Fostering a cyber security mindset. *Internet Policy Review*. 6, 1 (2017), 110–123.
[37] European Union. 2016. General data protection regulation. *Official Journal of the European Union*. L 119, 4.5.2016 (2016), 1–88.
[38] J. Evers and R. Kneyber. 2015. *Flip the system*. Routledge, Abingdon, UK.
[39] N. Ferguson, B. Schneier, and T. Kohno. 2011. *Cryptography engineering*. John Wiley & Sons, New York, USA.
[40] R.P. Feynman. 1974. Cargo cult science. *Engineering and Science*. 37, 7 (1974), 10–13.
[41] J. Freund and J. Jones. 2014. *Measuring and managing information risk*. Butterworth-Heinemann, Oxford, UK.
[42] E.R. Fyfe, N.M. McNeil, J.Y. Son, and R.L. Goldstone. 2014. Concreteness fading in mathematics and science instruction. *Educational Psychology Review*. 26, 1 (2014), 9–25.
[43] R.M. Gagne and L.J. Briggs. 1974. *Principles of instructional design*. Holt, Rinehart & Winston, New York, USA.
[44] A. Gawande. 2009. *The checklist manifesto*. Metropolitan Books, New York, USA.
[45] G. Gibbs. 2010. *Using assessment to support student learning*. Leeds Met Press, Leeds, UK.
[46] L. Giles. 1910. *The art of war (a translation of a work by Sun Tzu)*. Luzac & Company, London, UK.
[47] A. Gordon. 2015. *Official (ISC)2 guide to the CISSP CBK*, 4th edition. CRC Press, Boca Raton, USA.
[48] F.W. Guldenmund. 2000. The nature of safety culture. *Safety Science*. 34, 1 (2000), 215–257.
[49] D.F. Haasl. 1965. Advanced concepts in fault tree analysis. In *Proceedings of the System Safety Symposium*. Boeing Company, Seattle, USA.
[50] D.F. Halpern. 1998. Teaching critical thinking for transfer across domains. *American Psychologist*. 53, 4 (1998), 449–455.
[51] J. Hattie and H. Timperley. 2007. The power of feedback. *Review of Educational Research*. 77, 1 (2007), 81–112.
[52] C. Herley and P.C. van Oorschot. 2018. Science of security. *IEEE Security & Privacy*. 16, 1 (2018), 12–22.
[53] R.J. Heuer. 1999. *Psychology of intelligence analysis*. Central Intelligence Agency, Langley, USA.
[54] E. Hollnagel, D.D. Woods, and N. Leveson. 2006. *Resilience engineering*. Ashgate, Aldershot, UK.
[55] C.M. Holloway and C.W. Johnson. 2006. Why system safety professionals should read accident reports. In *Proceedings of the 1st IET International Conference on System Safety* (SSCS '06). Institution of Engineering and Technology, Stevenage, UK, 325–331.
[56] G.J. Holzmann. 2003. *The SPIN model checker*. Addison-Wesley, Boston, USA.
[57] G.J. Holzmann. 2006. The power of 10. *Computer*. 39, 6 (2006), 95–99.
[58] Homeland Security Advisory Council. 2012. *CyberSkills Task Force report*. United States Department of Homeland Security, Washington, USA.
[59] W.S. Humphrey. 1988. Characterizing the software process. *IEEE Software*. 5, 2 (1988), 73–79.
[60] IDEO.org. 2003. *The field guide to human-centered design*. IDEO, Palo Alto, USA.
[61] INCOSE. 2014. *Systems engineering handbook*, 4th edition. International Council on Systems Engineering, San Diego, USA.
[62] ISACA. 2008. *Information technology assurance framework*, 3rd edition. ISACA, Rolling Meadows, USA.
[63] ISACA. 2012. *Control objectives for information and related technology*, 5th edition. ISACA, Rolling Meadows, USA.
[64] K. Ishikawa. 1976. Cause-and-effect diagram. In *Guide to quality control*. Asian Productivity Organization, Tokyo, Japan, 18–28.
[65] ISO. 2011. *Information security risk management*. ISO 27005-2011. International Organization for Standardization, Geneva, Switzerland.
[66] ISO. 2013. *Information security management systems*. ISO 27001-2013. International Organization for Standardization, Geneva, Switzerland.
[67] ISO. 2013. *Code of practice for information security controls*. ISO 27002-2013. International Organization for Standardization, Geneva, Switzerland.
[68] ISO. 2014. *Vulnerability disclosure*. ISO 29147-2014. International Organization for Standardization, Geneva, Switzerland.
[69] ISO. 2015. *Quality management systems*. ISO 9001-2015. International Organization for Standardization, Geneva, Switzerland.
[70] D. Jackson. 2012. *Software abstractions*, 2nd edition. MIT Press, Cambridge, USA.
[71] S. Johnson. 1998. *Who moved my cheese?* Putnam, New York, USA.
[72] Joint Task Force on Cybersecurity Education. 2017. *Curriculum guidelines for post-secondary degree programs in cybersecurity (CSEC2017)*. ACM Press, New York, USA.
[73] L. Kimbell. 2015. *Applying design approaches to policy making*. University of Brighton, Brighton, UK.
[74] J.H.G. Klabbers. 2009. *The magic circle*, 3rd edition. Sense Publishers, Rotterdam, The Netherlands.
[75] L. Kohnfelder and P. Garg. 1999. *The threats to our products*. Microsoft Corporation, Redmond, USA.

[76] T. Kohno. 2009. *What to contribute (Winter 2009 CSE 484 / CSE M 584)*. Retrieved from https://cubist.cs.washington.edu/Security/2009/01/04/what-to-contribute-winter-2009-cse-484-cse-m-584/.
[77] D.R. Krathwohl. 2002. A revision of Bloom's taxonomy. *Theory Into Practice*. 41, 4 (2002), 212–218.
[78] D.R. Krathwohl, B.S. Bloom, and B.B. Masia. 1964. *Taxonomy of educational objectives*. David McKay Company, New York, USA.
[79] T.S. Kuhn. 1962. *The structure of scientific revolutions*. University of Chicago Press, Chicago, USA.
[80] P. Kumaraguru, Y. Rhee, A. Acquisti, L.F. Cranor, J. Hong, and E. Nunge. 2007. Protecting people from phishing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '07). ACM Press, New York, USA, 905–914.
[81] J.A. Kupsch, B.P. Miller, E. Heymann, and E. Cesar. 2010. First principles vulnerability assessment. In *Proceedings of the 2010 ACM Cloud Computing Security Workshop* (CCSW '10). ACM Press, New York, USA, 87–92.
[82] P. Ladkin and K. Loer. 1998. *Why-because analysis*. RVS-Bk-98-01. Bielefeld University, Bielefeld, Germany.
[83] L. Lamport. 2002. *Specifying systems*. Addison-Wesley, Boston, USA.
[84] B. Larcom. 2012. *Trike version 1.5.06 spreadsheet (help)*. Retrieved from https://sourceforge.net/projects/trike/files/trike/1.5.06/.
[85] J.H. Larkin and H.A. Simon. 1987. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*. 11, 1 (1987), 65–100.
[86] T. Lenoir and H. Lowood. 2005. Theaters of war. In *Collection, laboratory, theater*. H. Schramm, L. Schwarte, and J. Lazardzig, eds. Walter de Gruyter, Berlin, Germany, 427–456.
[87] L. Lessig. 2006. *Code v2*. Basic Books, New York, USA.
[88] N. Leveson. 1994. High-pressure steam engines and computer software. *Computer*. 27, 10 (1994), 65–73.
[89] N. Leveson. 1995. *Safeware*. Addison-Wesley, Reading, USA.
[90] N. Leveson. 2011. *Engineering a safer world*. MIT Press, Cambridge, USA.
[91] S. Levy. 1984. *Hackers*. Anchor Press, Garden City, USA.
[92] R.R. Linde. 1975. Operating system penetration. In *Proceedings of the 1975 National Computer Conference and Exposition* (AFIPS '75). ACM Press, New York, USA, 361–368.
[93] S. Lipner. 2004. The trustworthy computing security development lifecycle. In *Proceedings of the 20th Annual Computer Security Applications Conference* (ACSAC '04). IEEE Computer Society, Washington, USA, 2–13.
[94] S. Lueders. 2016. *CERN computer security (presentation given at Rotterdam UAS on 2018-09-16)*. CERN, Geneva, Switzerland.
[95] R.D. Marshall, E.O. Pfrang, E.V. Leyendecker, K.A. Woodward, R.P. Reed, M.B. Kasen, and T.R. Shives. 1982. *Investigation of the Kansas City Hyatt Regency walkways collapse*. NBS BSS 143. National Bureau of Standards, Washington, USA.
[96] D.B. McCoach, R.K. Gable, and J.P. Madura. 2013. *Instrument development in the affective domain*. Springer, Berlin, Germany.
[97] G. McGraw. 2006. *Software security*. Addison-Wesley, Boston, USA.
[98] MISRA. 2013. *Guidelines for the use of the C language in critical systems*, 3rd edition. MIRA Limited, Nuneaton, UK.
[99] B. Moggridge. 2007. People and prototypes. In *Designing interactions*. MIT Press, Cambridge, USA, 641–735.
[100] NCSC. 2013. *Policy for arriving at a practice for responsible disclosure*. Ministry of Security and Justice, The Hague, NL.
[101] T.P. Nelms, J.M. Jones, and D.P. Gray. 1993. Role modeling. *Journal of Nursing Education*. 32, 1 (1993), 18–23.
[102] W. Newhouse, S. Keith, B. Scribner, and G. Witte. 2017. *NICE cybersecurity workforce framework*. NIST SP 800-181. National Institute of Standards and Technology, Gaithersburg, USA.
[103] T. Nipkow, L.C. Paulson, and M. Wenzel. 2002. *Isabelle/HOL*. Springer, Berlin, Germany.
[104] NIST. 2018. *Framework for improving critical infrastructure cybersecurity (version 1.1)*. National Institute of Standards and Technology, Gaithersburg, USA.
[105] D. Norman. 2013. *The design of everyday things*, 2nd edition. Basic Books, New York, USA.
[106] G. Nuthall. 2007. *The hidden lives of learners*. NZCER Press, Wellington, New Zealand.
[107] G. Orwell. 1945. *Animal farm*. Secker & Warburg, London, UK.
[108] G. Orwell. 1949. *Nineteen eighty-four*. Secker & Warburg, London, UK.
[109] R. Paul and L. Elder. 2016. *The art of Socratic questioning*. Foundation for Critical Thinking, Tomales, USA.
[110] W. Pieters, J. Barendse, M. Ford, C.P.R. Heath, C.W. Probst, and R. Verbij. 2016. The navigation metaphor in security economics. *IEEE Security & Privacy*. 14, 3 (2016), 14–21.
[111] W. Pieters and A. van Cleeff. 2009. The precautionary principle in a world of digital dependencies. *Computer*. 42, 6 (2009), 50–56.
[112] K. Popper. 1959. *The logic of scientific discovery*. Hutchinson & Company, London, UK.
[113] M.E. Porter. 1979. How competitive forces shape strategy. *Harvard Business Review*. 59, 2 (1979), 137–145.
[114] C.W. Probst, J. Willemson, and W. Pieters. 2015. The attack navigator. In *Proceedings of the 2nd International Workshop on Graphical Models for Security* (GraMSec '15). Springer, Berlin, Germany, 1–17.
[115] PwC. 2013. *Internal control integrated framework*. Committee of Sponsoring Organizations of the Treadway Commission, New York, USA.
[116] PwC. 2017. *Enterprise risk management integrated framework*. Committee of Sponsoring Organizations of the Treadway Commission, New York, USA.
[117] J. Romein. 1937. De dialektiek van de vooruitgang. In *Het onvoltooid verleden*. Querido, Amsterdam, The Netherlands, 9–64.
[118] R. Ross, M. McEvilley, and J.C. Oren. 2018. *Systems security engineering*. NIST SP 800-160v1. National Institute of Standards and Technology, Gaithersburg, USA.
[119] P. Saitta, B. Larcom, and M. Eddington. 2005. *Trike version 1.0 methodology document*. Retrieved from http://www.octotrike.org/papers/Trike_v1_Methodology_Document-draft.pdf.
[120] B. Schneier. 2008. *The security mindset*. Retrieved from https://www.schneier.com/blog/archives/2008/03/the_security_mi_1.html.
[121] B. Schneier. 2012. *Liars and outliers*. John Wiley & Sons, New York, USA.
[122] D.A. Schoen. 1983. *The reflective practitioner*. Basic Books, New York, USA.
[123] A. Shostack. 2008. Experiences threat modeling at Microsoft. In *Proceedings of the Workshop on Modeling Security* (MODSEC '08). Sun SITE Central Europe, Aachen, Germany, 41–51.

[124] A. Shostack. 2014. *Threat modeling*. John Wiley & Sons, New York, USA.
[125] H.A. Simon. 1962. The architecture of complexity. *Proceedings of the American Philosophical Society*. 106, 6 (1962), 467–482.
[126] H.A. Simon. 1996. *The sciences of the artificial*, 3rd edition. MIT Press, Cambridge, USA.
[127] Software Engineering Institute. 2016. *SEI CERT C coding standard*, 2016 edition. Carnegie Mellon University, Pittsburgh, USA.
[128] T. Srivatanakul. 2005. *Security analysis with deviational techniques*. University of York, York, UK.
[129] T. Srivatanakul, J.A. Clark, and F. Polack. 2004. Effective security requirements analysis. In *Proceedings of the 7th International Conference on Information Security* (ISC '04). Springer, Berlin, Germany, 416–427.
[130] B. Suits. 1978. *The grasshopper*. University of Toronto Press, Toronto, Canada.
[131] M. Svinicki and W.J. McKeachie. 2014. Facilitating discussion. In *McKeachie's teaching tips*. Wadsworth, Belmont, USA, 38–57.
[132] F. Swiderski and W. Snyder. 2004. *Threat modeling*. Microsoft Press, Redmond, USA.
[133] T. UcedaVelez and M.M. Morana. 2015. *Risk centric threat modeling*. John Wiley & Sons, New York, USA.
[134] UFMCS. 2016. *The applied critical thinking handbook (version 8.1)*. University of Foreign Military and Cultural Studies, Fort Leavenworth, USA.
[135] United States Congress. 2002. Sarbanes-Oxley act of 2002. *United States Statutes at Large*. 116, 1 (2002), 745–810.
[136] United States Department of Commerce. 1937. Report of airship Hindenburg accident investigation. *Air Commerce Bulletin*. 9, 2 (1937), 21–36.
[137] J.J.G. van Merrienboer, R.E. Clark, and M.B.M. de Croock. 2002. Blueprints for complex learning. *Educational Technology Research and Development*. 50, 2 (2002), 39–64.
[138] R.B. Vaughn, D.A. Dampier, and M.B. Warkentin. 2004. Building an information security education program. In *Proceedings of the 1st Annual Conference on Information Security Curriculum Development* (InfoSecCD '04). ACM Press, New York, USA, 41–45.
[139] R. Wash. 2010. Folk models of home computer security. In *Proceedings of the 6th Symposium on Usable Privacy and Security* (SOUPS '10). ACM Press, New York, USA, 130–145.
[140] R. Wass and C. Golding. 2014. Sharpening a tool for teaching. *Teaching in Higher Education*. 19, 6 (2014), 671–684.
[141] C. Weissman. 1995. Penetration testing. In *Information security*. M.D. Abrams, S. Jajodia, and H.J. Podell, eds. IEEE Computer Society, Washington, USA, 269–296.
[142] W. Westera. 2001. Competences in education. *Journal of Curriculum Studies*. 33, 1 (2001), 75–88.
[143] E.L. Wiener, B.G. Kanki, and R.L. Helmreich. 2010. *Crew resource management*. Academic Press, Cambridge, USA.
[144] G.P. Wiggins and J. McTighe. 2005. *Understanding by design*, 2nd edition. ASCD, Alexandria, USA.
[145] B.J. Wood and R.A. Duggan. 2000. Red teaming of advanced information assurance concepts. In *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition* (DISCEX '00). IEEE Computer Society, Washington, USA, 112–118.
[146] D. Wood, J.S. Bruner, and G. Ross. 1976. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*. 17, 2 (1976), 89–100.
[147] M. Zenko. 2016. *Red team*. Basic Books, New York, USA.
[148] C. Zimmerman. 2014. *Ten strategies of a world-class cybersecurity operations center*. MITRE Corporation, Bedford, USA.