Information & Computer Security

An experimental evaluation of bow-tie analysis for security

Per Håkon Meland, Karin Bernsmed, Christian Frøystad, Jingyue Li, Guttorm Sindre,

Article information:

To cite this document:
Per Håkon Meland, Karin Bernsmed, Christian Frøystad, Jingyue Li, Guttorm Sindre, (2019) "An experimental evaluation of bow-tie analysis for security", Information & Computer Security, https://doi.org/10.1108/ICS-11-2018-0132
Permanent link to this document:
https://doi.org/10.1108/ICS-11-2018-0132

Downloaded on: 20 June 2019, At: 05:54 (PT)

References: this document contains references to 64 other documents.

The current issue and full text archive of this journal is available on Emerald Insight at:
www.emeraldinsight.com/2056-4961.htm

An experimental evaluation of bow-tie analysis for security

Per Håkon Meland
Department of Digital, SINTEF for Industriell og Teknisk Forskning, Trondheim, Norway and Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

Karin Bernsmed and Christian Frøystad
Department of Digital, SINTEF for Industriell og Teknisk Forskning, Trondheim, Norway, and

Jingyue Li and Guttorm Sindre
Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

Received 23 November 2018
Revised 14 February 2019
Accepted 27 February 2019

Abstract

Purpose – Within critical-infrastructure industries, bow-tie analysis is an established way of eliciting requirements for safety and reliability concerns. Because of the ever-increasing digitalisation and coupling between the cyber and physical world, security has become an additional concern in these industries. The purpose of this paper is to evaluate how well bow-tie analysis performs in the context of security, and the study's hypothesis is that the bow-tie notation has a suitable expressiveness for security and safety.

Design/methodology/approach – This study uses a formal, controlled quasi-experiment on two sample populations – security experts and security graduate students – working on the same case. As a basis for comparison, the authors used a similar experiment with misuse case analysis, a well-known technique for graphical security modelling.

Findings – The results show that the collective group of graduate students, inexperienced in security modelling, perform similarly as security experts in a well-defined scope and familiar target system/situation. The students showed great creativity, covering most of the same threats and consequences as the experts identified and discovering additional ones. One notable difference was that these naïve professionals tend to focus on preventive barriers, leading to requirements for risk mitigation or avoidance, while experienced professionals seem to balance this more with reactive barriers and requirements for incident management.

Originality/value – Our results are useful in areas where we need to evaluate safety and security concerns together, especially for domains that have experience in health, safety and environmental hazards, but now need to expand this with cybersecurity as well.

Keywords Security, Threats, Bow-tie analysis, Misuse case, Controlled experiment

Paper type Research paper

The research leading to these results has partially been performed by the Cyber Security in Merchant Shipping (CySiMS) project, which received funding from the Research Council of Norway under Grant No. 256508. The authors would like to thank all the participants in the experiment and the group of NTNU students developing the bow-tie modelling tool for supporting their work.

Information & Computer Security
Emerald Publishing Limited
2056-4961
DOI 10.1108/ICS-11-2018-0132

1. Introduction

There is an increasingly tight coupling between the cyber and physical world, which leads to new forms of risks that have not been considered adequately, such that the cyber element adversely affects the physical environment (Banerjee *et al.*, 2012). This is typically seen in industries that up until now have been running on isolated platforms and networks but through rapid digital transformations find themselves exposed to hostile cyber attacks from new categories of adversaries, as well as unintentional disclosure of sensitive data. For instance, a *Shodan* search conducted by Trend Micro in 2017 found more than 83,000 industry robots exposed on the internet, whereas more than 5,000 of these had no authentication whatsoever (Maggi *et al.*, 2017). These robots were operating in sectors such as automotive, aerospace, defence, food, and beverages. Similarly, the increased connectivity and lack of security awareness in the shipping industry are making stakeholders worried that this will become the *next playground for hackers* (WMN, 2014). A common trait of these industries is that there are already well-established practices for managing safety concerns. If these practices can be extended to encompass security, we might have an easier path than introducing a set of security analysis techniques that are unfamiliar to them and must be used in parallel.

Security models provide a useful basis for security analysis and requirements elicitation, e.g. supporting comparative evaluations of threats and intended security properties (Bau and Mitchell, 2011). Security modelling comes in many different forms and flavours (Bernsmed *et al.*, 2017), and there is not necessarily one single best or correct approach (Shostack, 2008). In many practical situations, this is a choice depending on factors such as available resources, focus area, domain, level of abstraction and personal preferences, but there is currently little empirical knowledge that can guide us when making these trade-offs. Just as with many other tasks within software engineering, there are many techniques and methods that are used because conventional wisdom suggests that they are the best approaches. As a remedy to this, experiments can investigate the situations to validate whether such claims are true (Pfleeger, 1994). According to Tichy (1998), *experimentation can accelerate progress by quickly eliminating fruitless approaches, erroneous assumptions, and fads. It also helps orient engineering and theory into promising directions.*

Our research objective has been to gain empirical knowledge on the use of bow-tie analysis applied for cybersecurity. Bow-tie analysis has a long tradition from the safety and reliability domain, where identified preventive and reactive barriers are used as sources for eliciting requirements. We wanted to evaluate how well the same analysis technique performs in the context of security, and complements to existing security modelling techniques, such as misuse case diagrams (Sindre and Opdahl, 2001). The research hypothesis central to this work is that *the bow-tie notation has a suitable expressiveness for security as well as safety*, and we have performed controlled experiments with both experienced and aspiring security professionals to get a wider range of people who are representative for the techniques. There already exists evidence that bow-tie analysis performs well for safety considerations, but if the hypothesis is falsified, then applying bow-tie analysis in assessments where we need to consider both safety and security in combination would make no sense.

This paper is an extension of previous work in Meland *et al.* (2018) and is structured as follows. We briefly show related work and explain the history and notation of bow-ties and misuse cases in Section 2, as well as how they can be compared to each other. In Section 3, we explain our research method and the details of the experiments at hand. This is followed by a summary of results in Section 4. These results are then interpreted and discussed as a part of Section 5, and the paper is concluded in Section 6.

2. Background

2.1 Models covering safety and security

There are many examples in the literature of models that allow combinations of safety and security considerations. For instance, Johnson (2011) shows how to build cybersecurity assurance cases for Global Navigation Satellite Systems (GNSS) using Boolean Driven Markov Processes (BDMP), extending conventional fault trees. Winther *et al.* (2001) include security as part of HAZOP studies, which is a systematic analysis on how deviations from the design specifications in a system can arise and whether these deviations can result in hazards. Raspotnig *et al.* (2012) make use of UML-based models within a combined safety and security assessment process to elicit requirements. Kumar and Stoelinga (2017) combine fault and attack trees so that both safety and security can be considered in combination. Fishbone diagrams are similar to bow-ties and are mentioned in Nolan's book (2014) on safety and security reviews for the process industries, but examples here only focus on safety incidents. FMVEA (failure mode, vulnerabilities and effect analysis) (Schmittner *et al.*, 2014) is a safety and security co-analysis method extended from FMEA (failure mode and effect analysis). Like FMEA, FMVEA proposes to use the STRIDE model (Shostack, 2008) to identify threat modes first, and then analyse the effect of each threat model. Further examples of methods, models, tools and techniques in the intersection of safety and security can be found in the surveys by Zalewski *et al.* (2012), Piètre-Cambacédès and Bouissou (2013), Chockalingam *et al.* (2016), as well as Kriaa *et al.* (2015).

2.2 The bow-tie history and notations

Bow-tie analysis has since the 1970s been used by organisations worldwide for risk-management purposes, but primarily to demonstrate control over health, safety and environmental (HSE) hazards (Lewis and Smith, 2010). For instance, Khakzad *et al.* (2013) show this application in safety risk analysis in offshore drilling, Trbojevic and Carr (2000), as well as Mokhtari *et al.* (2011), do the same for safety assessment in international maritime ports, and Lu *et al.* (2015) apply bow-ties in the context of leakage from natural gas pipelines.

In our modern cybersecurity world, we have to consider the intertwined relationship between safety and security during risk assessment, and make sure that requirements can be traced back to a *source*, such as an intended barrier. As already described by Bernsmed *et al.* (2017), there have been several efforts at adopting the bow-tie notation for cybersecurity within areas such as engineering environments and maritime operations. This is because these areas are already familiar with the notation from safety assessments, and therefore it is assumed to be easier obtaining community buy-in by evaluating cybersecurity threats in the same way as accident scenarios. Abdo *et al.* (2018) have also proposed a combined bowtie/attack tree methodology to consider the effect of cyber security on safety risk scenarios. However, we are not aware of any empirical evidence from the literature proving that bow-ties are suitable to cover security concepts in addition to safety.

A central part of bow-tie analysis is the creation of graphical bow-tie diagrams. A bow-tie diagram is something that resembles a fault-tree on the left-hand side with an event-tree on the right (Lewis and Smith, 2010). Figure 1 gives an overview of the modelling elements that have been included in our experiment, based on Bernsmed *et al.* (2017). First, the *Hazard* element represents the risky environment in which one or several *Unwanted events* (aka *top event*) can occur but which is also necessary to perform business. Note that we only model one top event per diagram. A *threat* is anything that can potentially cause an unwanted event (ISO/IEC, 2011), and there can be several types of such threats in a single diagram. To prevent or eliminate threats, we can add *barriers* (aka *controls*) that interfere between threats and the top event. An *Escalation factor* is a specific type of threat that targets a barrier,

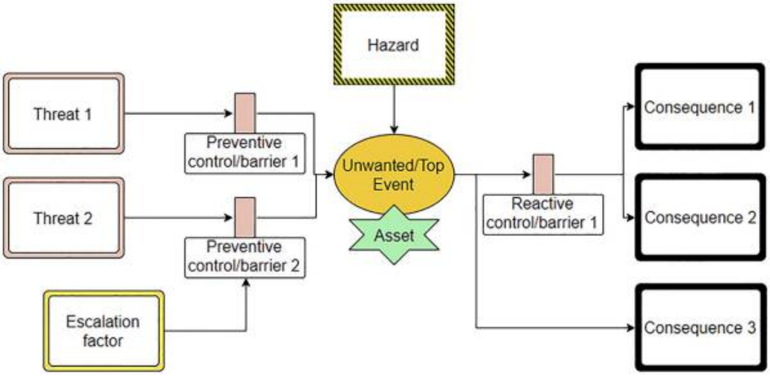


Figure 1.
The basic elements of the bow-tie notation with security extension

opening up for the original threat. A top event can result in one or several *consequences*. As with threats, we can add *controls/barriers* that can reduce the probability or eliminate the consequences, but these are now of a reactive nature because the top event has already occurred. Finally, and specifically added for security, an *asset* is anything tangible or intangible with value and should be protected. We allow one or more assets to be modelled per diagram.

2.3 The misuse case history and notation

Misuse case modelling is a well-known technique for graphical security modelling, and can be summarized as an extension to regular UML use cases (Jacobson, 1993), also covering misuse and used to elicit security requirements (Sindre and Opdahl, 2001). Misuse cases have already been proven useful in different industrial cases when considering security (Matulevicius *et al.*, 2008). They have also been used in controlled experiments to identify safety hazards (Stålhane and Sindre, 2008). Misuse cases are therefore a good basis for comparison with bow-tie diagrams, though one might say that they have an opposite historical path (coming from the security domain and subsequently applied to safety).

The misuse case notation can be summarized as shown in Figure 2. Here, we have included the extensions from Røstad (2008) that also cover vulnerabilities and insiders.

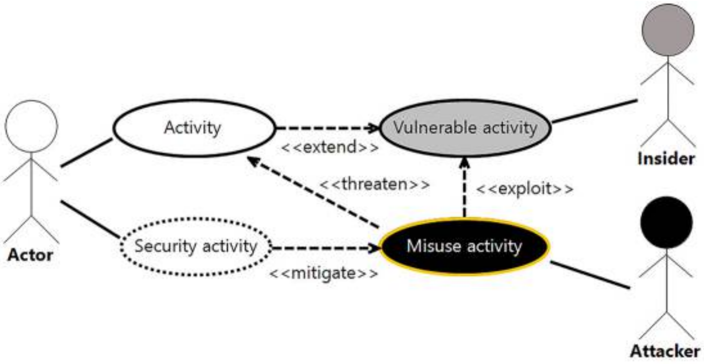


Figure 2.
Overview of the misuse case notation

Activities represent the normal behaviour of the system. Normal *actors* instantiate the activities and represent anyone (could be human users but also other systems) interacting with the system as intended, i.e. they do not harm the system either intentionally or unintentionally. *Misuse activities* represent threats towards the system, typically something an attacker would like to perform or achieve. The *attacker actors* have malicious intents towards the system. *Vulnerable activities* are part of the normal behaviour of the system, but represent functionalities that make the system exploitable. *Insider actors* are trusted users of the system that could intentionally or unintentionally cause harm to the system. *Security activities* show what can be done to mitigate misuse activities or vulnerable activities.

Though misuse case models and bow-tie diagrams share some of the same traits, it can be difficult to directly replace one with the other in an analysis. In Table I we compare them together to show similarities and differences. We would argue that misuse case and bow-tie diagrams are more complementary than competing types of security models, something we have exploited in our bow-tie experiment.

2.4 Related security requirements techniques

Both bow-tie and misuse case diagrams mainly focus on identifying threats, while a key aspect of requirements engineering would then be to specify requirements concerning the necessary level of security in mitigating these threats. An approach closely related to misuse cases would be security use cases (Firesmith, 2003), which go somewhat further in the direction of requirements rather than threats. Other UML-related approaches that offer more detailed specification of requirements are SecureUML (Lodderstedt *et al.*, 2002) and UMLsec (Houmb *et al.*, 2010; Jürjens, 2002) that offer security extensions to several other UML diagrams (e.g. class, activity, sequence diagrams) and not just use case diagrams. Another related approach is the extension to state-transition diagrams proposed by El-Attar *et al.* (2015).

Bow-tie analysis has been less used in security but could be seen as related to the concept of risk, which is the central focus of the modelling language proposed by Mayer *et al.* (2007). Other well-known approaches to security requirements modelling include goal-oriented

Misuse case models	Bow-tie diagrams
[Both] Defined by a simple to understand graphical notation with an open-ended method, allowing for a lot of creativity by the modeller	
Originate from computer security and requirements engineering, based on UML use case diagrams	Originate from the safety and reliability domain, related to fault analysis
Developed to identify malicious actions (misuse) for a given system	Developed to investigate accident scenarios and define barriers
The misuse activity element represents an unwanted event (something that threatens regular activities)	The top event element represents an unwanted event
Broad scope. Suitable for describing many different misuse activities in a single model	Narrow scope. Focus on a single unwanted top event per diagram
Show actors (attackers, misusers, threat agents) related to misuse activities	Do not represent actors, but in which risky environment (hazard) the top event can occur
Mitigations are modelled as security activities	Mitigations are modelled as barriers, which are clearly defined as either preventive or reactive
Can depict vulnerable activities that a can be exploited	Represent various threats/causes that can lead to the top event
Consequences are not part of the model	Explicitly depict possible consequences following the top event

Table I. A comparison of misuse case models and bow-tie diagrams

approaches such as KAOS (Van Lamsweerde, 2004), Secure i* (Elahi *et al.*, 2010) and Secure Tropos (Mouratidis and Giorgini, 2007). There are also many other security requirements techniques, beyond what can be covered in this section. Good overviews of techniques that existed by 2010 are provided by Fabian *et al.* (2010) and (Mellado *et al.*, 2010) and a more recent mapping study by Souag *et al.* (2016) focusing specifically on reuse of knowledge in security requirements engineering.

3. Experiment method

To plan our bow-tie experiment, we adopted and applied the guidelines by Kitchenham *et al.* (2002), originally designed for empirical studies in software engineering. The form of the study is a *controlled experiment*, which is a scientific method for identifying cause – effect relationships (Sjøberg *et al.*, 2005), and a means to *acquire general knowledge about which technology (process, method, technique, language or tool) is useful for whom to conduct which tasks in which environments*. The intervention we introduce is the use of the bow-tie notation for security analysis on two sample population that are both working on the same case.

As there are no random assignments, this should be classified as a *quasi-experiment*, and as a formal experiment because we have a high level of control over the variables that can affect the truth of the hypothesis (Pfleeger, 1994).

One of the sample populations consists of students, and therefore it has been important to make sure that they perceive a value from participation (Carver *et al.*, 2004). By carefully scoping the case of the experiment and having an approach that is new to the student sample and professionals in general, we expect to get relevant results with external validity (Salman *et al.*, 2015).

To have a better basis for experiment evaluation, we present the result from a similar experiment with misuse case modelling, though applied to a different case. This allows us to see whether the phenomena related to the dual populations are generalisable or local to bow-ties. Both cases in focus and experiment setups are described in the sections below.

3.1 Case A: digital exams

For the bow-tie analysis, we chose a security modelling assignment related to use of digital exams, something that is rapidly growing in popularity at universities and other educational institutions. Here, exams are created, solved and graded using online systems. This is meant to be more efficient than traditional exams done on paper, however, relies on technology and opens up to new types of threats that need to be identified and dealt with. For instance, a survey by Chen and He (2013) shows that there is a great diversity of security risks for online exams, nevertheless, security is not considered as a top priority among learning providers and practitioners. Additionally, there is evidence that both digital and “analogue” exams suffer because of new technical ways of cheating. According to *The Guardian* (Marsh, 2017), there has been a 42 per cent rise in cheating cases between 2012 and 2016, involving gadgets such as mini cameras and micro earbuds. London (2017) gives an overview of further inventive and not-so-inventive ways that have been used for cheating on online exams. In some developing countries, such as Algeria, Ethiopia, Syria and Iraq, internet access in the whole country is shut down during the exam period to prevent cheating (Bradbury, 2018). All in all, a case related to digital exams provides an interesting and relevant arena for looking at security issues and possible solutions.

In our case, there are many students participating in the exam in the same confined room and within the same time frame. This is a bit different from other types of digital exams, which can be done from home and at any given time. Furthermore, the students are allowed to use their own personal computers with internet access through WiFi, but are not allowed to use supporting materials, such as curriculum books and notes. A specific Web browser must be installed on their computers, known as the *Safe Exam Browser* (SEB)[1], which regulates access to websites, search engines, other applications and system calls, also referred to as *browser lockdown*. Vegendla *et al.* (2016) report on a case study doing penetration testing on the SEB, identifying some vulnerabilities that could be used for cheating. However, it must be noted that this cheating is less likely today, as the software has since been improved.

3.2 Case B: Web shop for digital goods

For the misuse case modelling, we selected a system description that most people can relate to through personal experience. Web shops are virtual marketplaces that are accessed online and used to browse for interesting items and complete purchases. Web shops are suitable for security analysis because there have been plenty of examples of real-life compromises. For instance, a 2018 report by a security firm show that almost 90 per cent of the people logging into some popular retailers' e-commerce sites were hackers using stolen data (Green, 2018). A lot of the Web shops also use the same code base; hence, they share a lot of the same vulnerabilities. If the store owners are lazy updating their software, they quickly become easy prey to attackers looking for known vulnerabilities. In 2016, a Dutch developer reported almost 6,000 Web shops with proven vulnerabilities and that were exploited to steal the credit card details of customers (BBC, 2016). The OWASP Juice Shop Project (Kimminich, 2018) is an example of an intentionally insecure Web shop that is being used to train software developers.

In our case, we limited the Web shop inventory to be digital goods, which are non-tangible items, such as music files, wallpapers, games and other types of software that are directly downloaded from the Web shop. The main assets are the digital goods and customer information such as personal data, order history and credit card information. There could be a number of different threat agents/attackers with different motivation, such as cyberthieves or business competitors.

3.3 Bow-tie experiment setup

This experiment engaged two types of populations: a small sample of security experts and larger sample of computer science MSc graduate students. The characteristics of these groups can be described as follows: the students participated in the experiments as a part of a classroom exercise in a course on secure software engineering and were motivated to learn security modelling to apply such techniques for their exercises and final exam. Before the experiment, the students had taken several lectures, including security concepts and principles, OWASP top 10, crypto introduction, multilevel security and multilateral security. The students had limited knowledge of security modelling beforehand and no experience at all from bow-tie modelling. Moreover, the students had significant practical experience related to digital exams, as they had already been exposed to this on several occasions. It is unknown how experienced and reflected they were related to cheating.

The security experts had a great deal of prior knowledge and practical experience in various types of security modelling techniques, and in particular bow-tie for specific domains. In contrast to the students, the experts had limited practical experience of participation in digital exams, though one of them was skilled with setting up exams using the online system. The experts were motivated by the research itself, and the desire to create a good reference model that the student results could be compared to.

As an introduction, the students were given a lecture on threat modelling, including the misuse case and bow-tie notations. As we know from prior experiences, one of the challenges of bow-tie diagrams, is setting the scope of the unwanted event. Therefore, the students were presented with a misuse case model that we hoped would better define the scope and the relationship between the events. This model is shown in Figure 3 and depicts a number of actors and typical activities related to digital exams, as well as misuse case activities and associated threat agents. For example, the actor *professor* will need to *log in* to the system and *create exam assignments* prior to the examination day. An external *attacker* actor would possibly want to *steal assignments* and maybe sell this online to students who want to cheat. After the examination day, an additional *external examiner* is involved in the process of *grading exams*. The attacker could at this point try to *change the results* of the exam. During the examination day itself, the main legitimate actor is the *student* who needs to *setup* his/her computer, which also involves sub-activities such as *connecting to the network* and *installing the correct SEB software*. To *do the exam*, the student must authenticate by *logging in*, *enter the exam pin* for this particular exam, *solve the assignments* and finally *submit the exam*. On the right side of the diagram, we have depicted a *bad student* insider actor who inherits all the activities from the legitimate student actor. The bad student has a misuse activity mostly relevant prior to the examination day, which is to *buy the assignments in advance*, and two others that threaten the regular activities during the exam. The first one, *disrupt exam*, is basically a way of sabotaging the examination for everyone, possibly motivated by a wish of cancelling/delaying the exam. The second one is *cheat during exam*, which a student would do to illegitimately improve his/her grade. The *proctor* is a type of examination guard that *supervises the exam* and is there to mitigate cheating attempts and disruptions.

The next step of the introduction was to show how a misuse activity can be detailed as a bow-tie top event. This was demonstrated with *disrupt exam* as shown in Figure 4. In this model, there are a number of threats that can lead to a disruption, such as *tampering with the fuse box* to cause power outage, *jamming the wireless network* or performing some other action to *make the online server unavailable*. The assets that needs to be protected are the *network*, the *SEB software* and the *physical premises* themselves. We added some example preventive controls/barriers, such as *locking the fuse box cabinet* and having a system *mirror site* on hot standby. In terms of disruption consequences, computers can stop working and the bad student can be expelled. The only reactive control/barrier shown here is *switching to paper* to complete the exam.

Having introduced the notation, defined the scope and given examples, the populations were now ready to work on their own diagrams. We predefined *digital exam* as the risky environment, *cheat during exam* as the top event and the asset *answers* as a starting point.

Both populations worked on this same case, with access to external information such as SEB documentation and articles about online exams and cheating. The students worked in teams, typically two-three persons per model, spending about 30 min on their task, and were observed by two of the authors of this paper. The experts worked independently of each other for about one hour. Both populations used an online modelling tool[2] to create their models. The tool itself has an intuitive drag-and-drop interface for the basic bow-tie elements, and runs within any Web browser. A screenshot of this tool is shown in Figure 5.

The students were informed that all participation was anonymous and voluntarily, and that we wanted to make use of the result to evaluate the bow-tie notation for security.

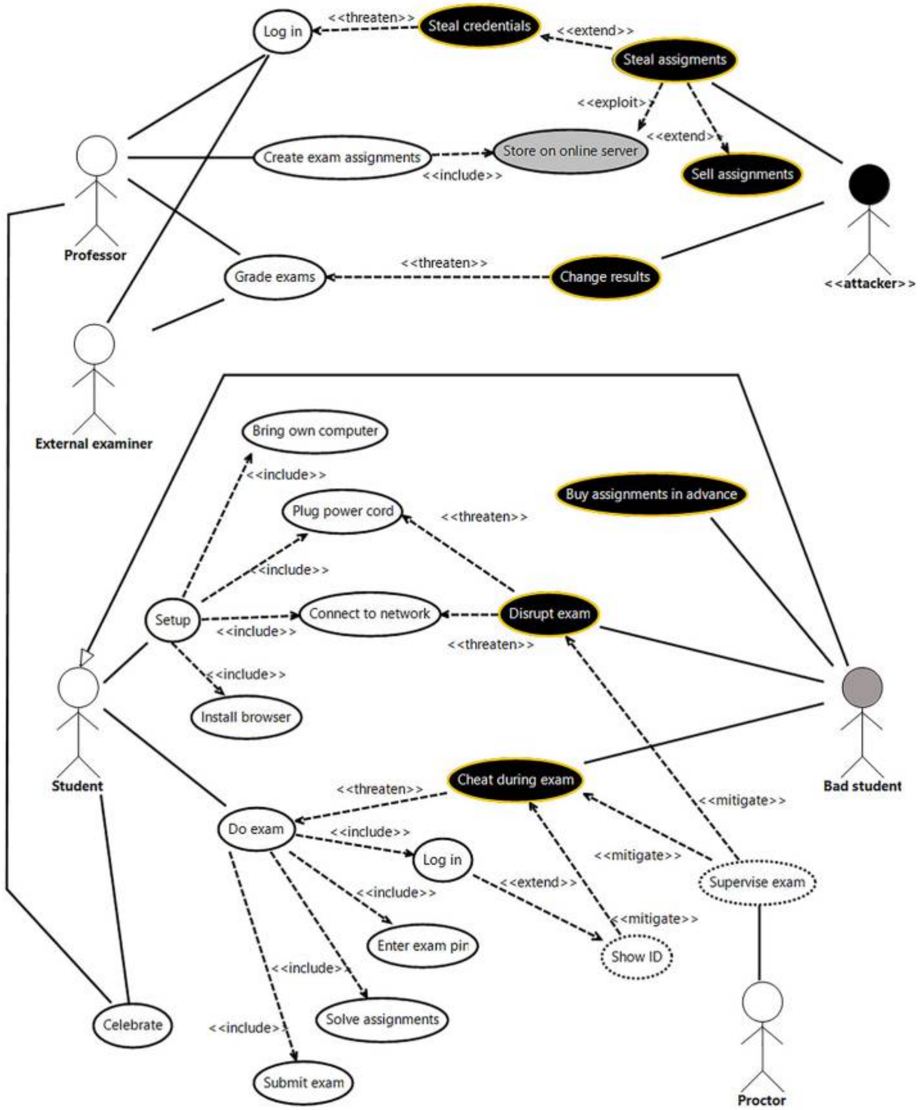


Figure 3.
Defining the scope with a misuse case diagram

3.4 Misuse case experiment setup

The setup of this experiment was almost identical to the bow-tie experiment, with a few notable exceptions:

- The resulting models had already been created by students taking the same course during three previous years. Hence, it was not the same population of students but a larger set of students with the same characteristics. The experts were the exact same individuals as in the bow-tie experiment.

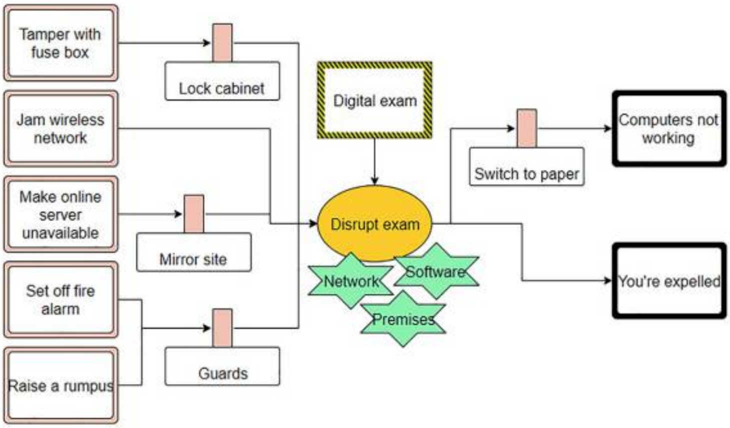
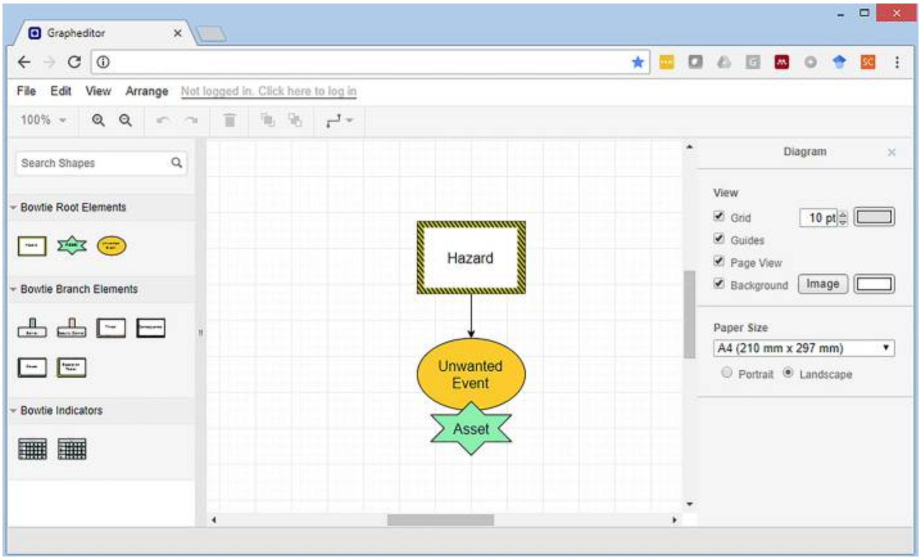


Figure 4.
Example model showed as a preparation



☑ Grid 10 pt
☑ Guides
☑ Page View
☑ Background
Paper Size
A4 (210 mm x 297 mm)
○ Portrait ● Landscape

Figure 5.
The online tool used for making the bow-tie diagrams

- The students had some familiarity with UML use case modelling, but no significant experience with misuse case modelling beforehand.
- Both populations used pen and paper to create their models and not an online modelling tool.

The students had been given the same kind of introduction to security modelling, including a walkthrough of misuse case analysis with a few simple example models. When the assignment started, the students were handed out a paper sheet containing a use case template for them to work on in pair-wise groups. This template is shown in Figure 6, and depicts two normal actors, a *customer* and the *Web shop* service. These are associated to a set of

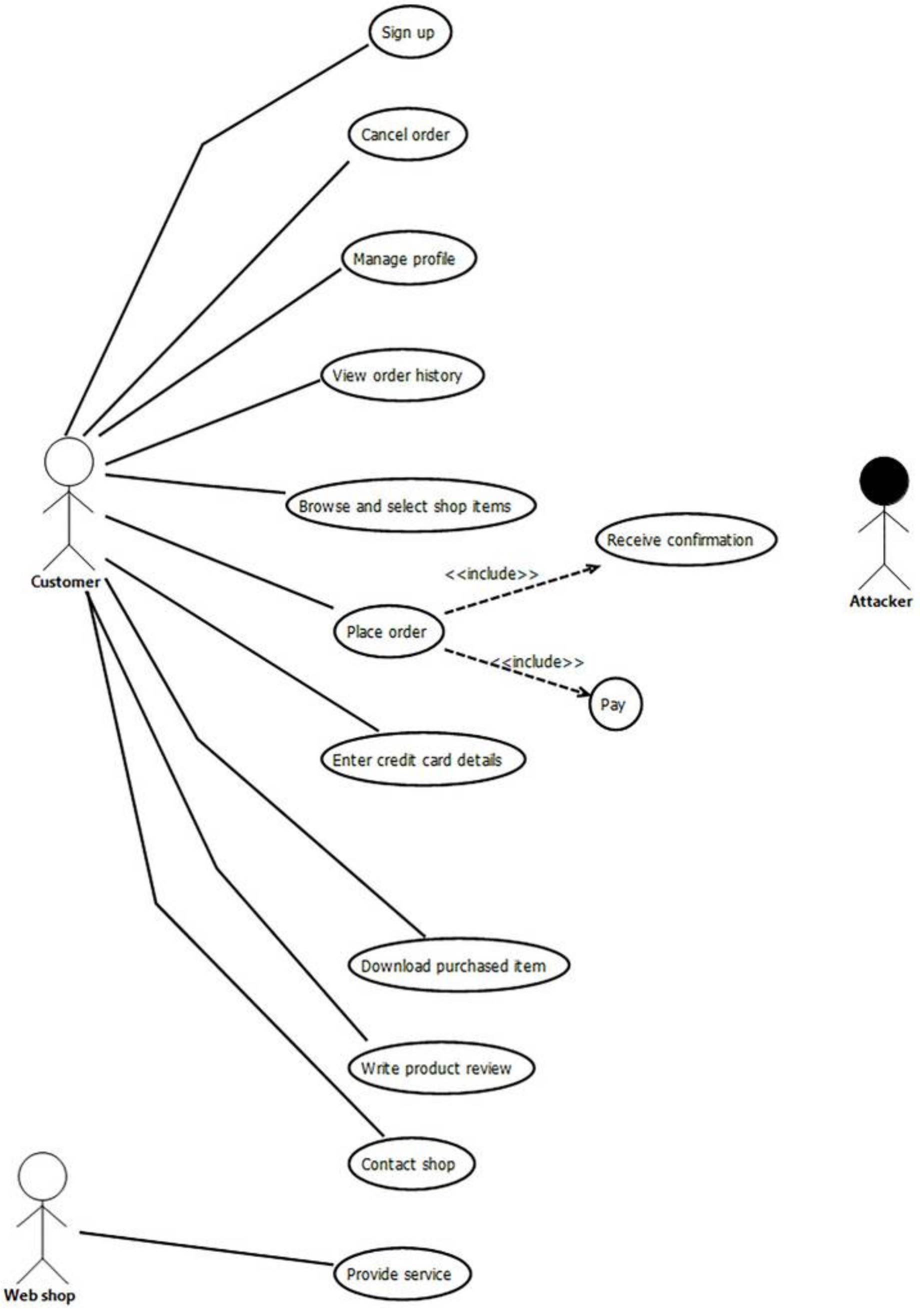


Figure 6.
The Web shop use case template

predefined use case activities defining the functional scope of the assignment. An *attacker* actor was also included in the template, but with no associated misuse activities. The use case activities indicate what kind of assets that are involved, such as personal data, order history, shop items, credit card information and product reviews. Both populations were instructed to add misuse case model elements to the template and spent about 30 min on their models. Just as with the bow-tie experiment, all participation and hand-in was voluntarily and anonymous.

4. Results of the bow-tie analysis

4.1 Diagrams made by students

A total of 40 students were present in the experiment session, which resulted in 13 different diagrams. Observations from the classroom indicated that approximately 30 students contributed to these diagrams. This estimate is based on the average size of the groups and that we also know that not all diagrams were submitted (this was voluntarily). The diagrams were then analysed, and we created a small taxonomy of threats, controls/barriers and consequences to be able to compare them. Based on this, we developed a combined bow-tie diagram, shown in Figure 7, which also indicates the frequency of the threat and consequence elements found in the diagrams made by the students. As can be seen from the figure, the top threats were:

- *Analogue cheat sheet*, the most popular threat, appeared in 6 out of the 13 models that we collected (6/13). This is probably the most "traditional" way of cheating, and it involves the use of some concealed written material, e.g. paper notes hidden inside the wrapper of a candy bar or somewhere on the body of the student.
- *Access external information* (4/13) encompasses using the computer to search and access information on the internet.
- *Another person takes exam* (4/13) is related to impersonation and not something that is unique to digital exams.
- *Digital chat with others* (3/13) is when the student computer is used to communicate with others in the same room or on the outside.
- *Hack browser* (3/13) is done by somehow modifying the source code or exploiting an existing vulnerability in the SEB software to disable the lockdown functionality.
- *Run browser in virtual machine* (3/13) was represented as a threat in two of the models, and as an escalation factor in a third. In the combined model, we represent it as an escalation factor because this is basically a way of circumventing a preventive barrier by letting the SEB software lockdown the virtual machine instead of the computer itself.
- *Digital communication with others* (3/13) covers all kinds of gadgets besides the student computer that are used for communication with others. This typically includes Bluetooth devices and other radio equipment.
- *Spy on other screens* (3/13), also denoted as "shoulder surfing", is simply ways of looking at other people's answers without them noticing it. Peeking at the answers of others is a relevant cheating threat for paper exams too, but may be accentuated for digital exams because screens are nearly vertical, while paper lying horizontally on a desk is harder to read from a distance.

Some additional threats can be found in Figure 7, but these were only present in one or two of the diagrams. Additionally, we discarded three threats that were out of scope for this top event, namely, *Retrieve exam answers beforehand*, *Disrupt exam* and *Blackmail professor*.

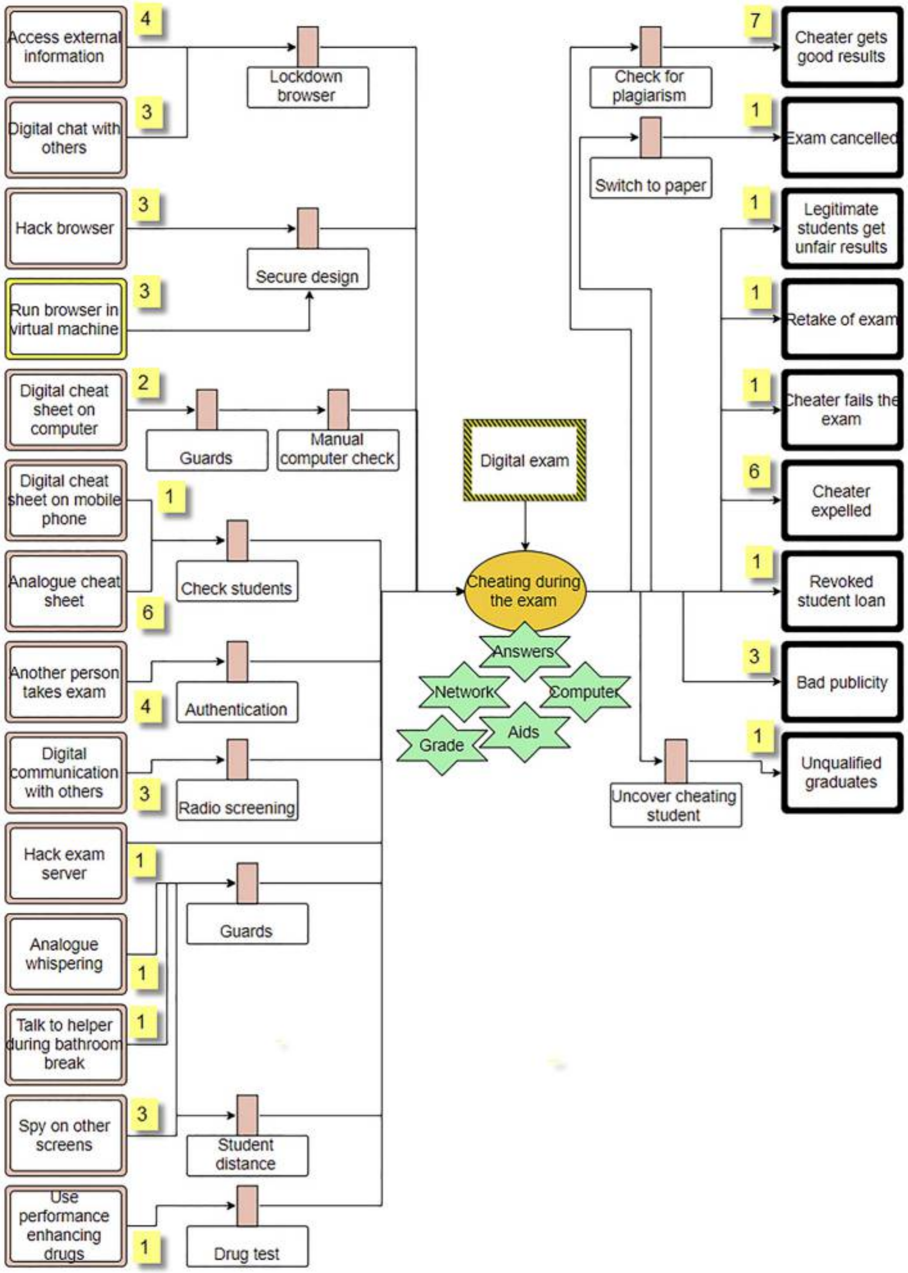


Figure 7.
A combination of the bow-tie diagrams made by the students

On the consequence side of the diagram, *Cheater gets good results* (7/13) was most prevalent, followed by *Cheater expelled* (6/13) and *Bad publicity* (for the university). Interestingly, these are consequences for both successful cheating as well as consequences for the cheater if he/she gets caught.

The combined diagram does not show the frequency of barriers/controls because a lot of them overlap over more than one threat/consequence. We also noticed that some of the diagrams (4/13) contained additional assets, so we added these to the combined diagram as well (Figure 7).

4.2 Diagrams made by security experts

There were three security experts participating in this experiment, resulting in three independent bow-tie diagrams. These were analysed in the same manner as the student diagrams and aligned using the same taxonomy. The resulting combined diagram from the experts is shown in Figure 8. There were only four threats that had an overlap between the expert diagrams; *Access external information*, *Another person takes exam*, *Hack browser* and *Phone outsiders*. The three first were all present among the top threats from the student diagrams as well, while the latter was not. We discarded one threat from the diagram,

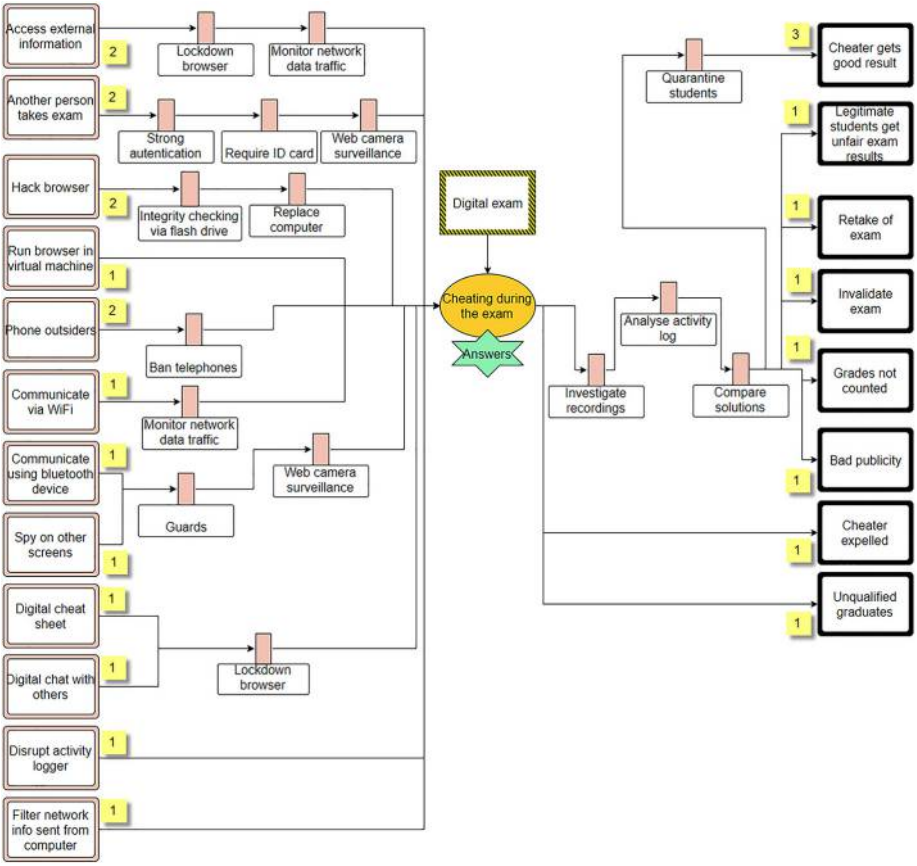


Figure 8.
A combination of the bow-tie diagrams made by the experts

Introduce vulnerability in SEB OSS project, as this is something that must be done prior to the exam and hence out of scope for this top event. The expert and student diagrams shared their top consequence, namely, *Cheater gets good result*. Besides from that one, there was little overlap between consequences among the experts. Note that there are several threats and consequences that are without any barriers. It turned out that one of the experts forgot about adding these, and therefore spend more time on finding threats and consequences compared to the others.

Table II shows a numerical comparison of the diagrams created by the two populations. The last row shows how many distinct elements that are common between the combined diagrams from each population. As the level of detail vary, it was not possible to always create direct mappings. Therefore, *Communicate* via *WiFi* and *Communication using Bluetooth device* in the expert diagram is mapped to the single threat *Digital communication with others* in the student diagram. Likewise, the preventive barrier *Strong authentication* in the expert diagram is mapped towards the less strict *Authentication* in the student diagram. The complete set of original diagrams is openly available from Meland (2018a).

5. Results of the misuse case experiment

5.1 Models made by students

Because we collected misuse case models made by students from three previous years, the total number of models was increased to 31 in total. The classroom setup had been the same, with two and two students sitting together, so approximately 62 students contributed to these models. During the analysis, we grouped together similar model elements and created a taxonomy of misuse case activities, vulnerable activities and security activities. As misuse case models by nature have a broad scope, we also got a very broad set of misuse case activities (48 distinct), where the top 20 were:

1. *Trojan/corrupt code in digital goods* (22/31) targets other customer of the Web shop.
2. *Disrupt Web shop service* (13/31) for instance using DDOS/DOS attacks.
3. *Phishing* (13/31) is an attack technique used to obtain sensitive information.

Table II. A numerical summary of bow-tie model elements

Measurement	Experts	Students
Number of participants	3	~30
Number of models	3	13
Total number of threats	18	49
Number of distinct threats	12	14
Average number of threats per model	6	3.8
Total number of consequences	10	27
Number of distinct consequences	8	9
Average number of consequences per model	3.3	2.1
Total number of preventive barriers	16	41
Number of distinct preventive barriers	10	9
Average number of preventive barriers per model	5.3	3.2
Total number of reactive barriers	6	6
Number of distinct reactive barriers	4	3
Average number of reactive barriers per model	2	0.5
Common (threats/consequences)/(preventive/reactive) barriers	7/5/3/0	

(4) *SQL/Code injection* (12/31) is an attack technique typically used wherever there is some kind of user input, e.g. Web forms.

(5) *Write false review* (10/31) can manipulate the user ratings of the digital goods, potentially influencing what the customers buy.

(6) *Pharming* (10/31) is a general concept where the goal is to steal and collect other customer's payment data.

(7) *Spoof payment* (9/31) tricks customers into paying at a false payment service.

(8) *Information theft* (9/31) is a high-level concept where the attacker obtains information that should have been protected.

(9) *Spoof Web shop* (9/31) tricks the customers into using an imitated, fake version of the Web shop.

(10) *Steal account/password* (8/31) enables an attacker to impersonate a legitimate user.

(11) *Malicious input in review form* (7/31) encompasses active code or offensive content inserted into the review form functionality of the Web shop.

(12) *False/fake signup* (7/31) is when non-existing or impersonated users are registered to the Web shop.

(13) *Drive-by download* (7/31) is when the information belonging to the digital goods have been replace by or contains malicious code.

(14) *Man-in-the-middle* (7/31) is a general concept where requests are intercepted and manipulated before they reach their destination.

(15) *Eavesdropping/sniffing* (7/31) means to tap into some communication. This can be considered a sub-type of information theft.

(16) *Social engineering* (6/31) is when humans are exploited rather than technical systems. It is mostly associated with the *Contact shop* activity.

(17) *Manipulate payment* (6/31) threatens the integrity of the payment transaction, for instance the target account or the sum of the payment.

(18) *Send fake order confirmation* (6/31) targets the customers, typically used for delusion or click-bait.

(19) *Modify/delete other customer's profile* (6/31) is a broad category, e.g. involving changing personal information and password, as well as replacing the profile picture with explicit photos.

(20) *Change contact info* (6/31) of the Web shop, subsequently tricking the customers into unknowingly contacting fraudsters.

In addition to these, we would highlight *Pay with stolen credit card* (5/31), *Steal/copy digital goods* (4/31) and *Trolling* (2/31) as particular relevant for the Web shop case. Ten misuse activities only appeared once in the models but were still relevant.

There were noticeable differences in the level of abstraction used to describe the misuse activities. We could have created a smaller taxonomy with more generalized activities, but this would mean loss of some of the more specific information. For instance, *Information theft* is a very broad concept that some of the students used. *Steal account/password* or *Pharming* are more specific and connected to the use case activities at hand, while *SQL/Code injection* is an attack technique that can be used to accomplish information theft (among other things).

The number of distinct vulnerable activities was only seven, making an average of only 0.3 per model. This number is a bit skewed, as 24 of the models had no vulnerabilities at all. The vulnerable activities appearing in more than one model were:

- Retrieve personal information (2/31);
- store private data (2/31); and
- input forms (2/31).

There were 19 distinct security activities, but only 5 that appeared in more than one of the models. These were:

(1) *Network encryption* (8/31) protects the integrity of the communication through SSL/TLS mechanisms.

(2) *Checksum for digital goods* (4/31) is used as a mean to verify the integrity or authenticity of the downloaded items from the Web shop.

(3) *Email verification on profile alterations* (4/31) is used to inform the customer about possible misuse of the account.

(4) *Sanitize input* (4/31) is used to check user input and mitigate injection attacks.

(5) *CAPCHA* (2/31) can be used to mitigate bots creating false user accounts or inserting malicious content/spam into the review forms.

Without going too much into details, it is safe to say that that the students mostly focused on misuse activities, and not so much on identifying vulnerable or security activities. Figure 9 shows a combined model with the top 20 misuse case activities and vulnerable/security activities that appear in more than one model. Some relationships have been omitted for readability purposes.

5.2 Models made by security experts

The same three security experts as in the bow-tie experiment participated and created their models individually. The experts identified 18 distinct misuse case activities, whereby 17 of these were also covered by the students. The last one was *Enumerate usernames* (1/3), where the goal is to harvest existing usernames by misusing the *Sign up* activity. These usernames can have a value for an attacker because they give away information about the customer base, and they can also be used in brute force passwords attacks, sending out phishing emails or locking out other users. Figure 10 shows a combined model based on the results from the experts. Some of the relationships have been omitted to increase readability.

The average number of misuse case activities per model was 9, which is only slightly higher compared to the student models. As can be seen in the Table III summary, it is for the vulnerable and security activities there are significant differences between expert and student models. In fact, the expert had an average of distinct vulnerable activities more than 22 times compared to the students, and more than 7 times for distinct security activities. The complete set of original models is openly available from Meland (2018b).

6. Discussion

6.1 Interpretation of bow-tie experiment results

It was interesting to see how well the students were able to grasp the concepts of bow-tie modelling and apply it to the digital exam case after just a relatively short introduction. There are a few notable differences when comparing results from students with experts, such that the average numbers of threats, preventive barriers and consequences per model

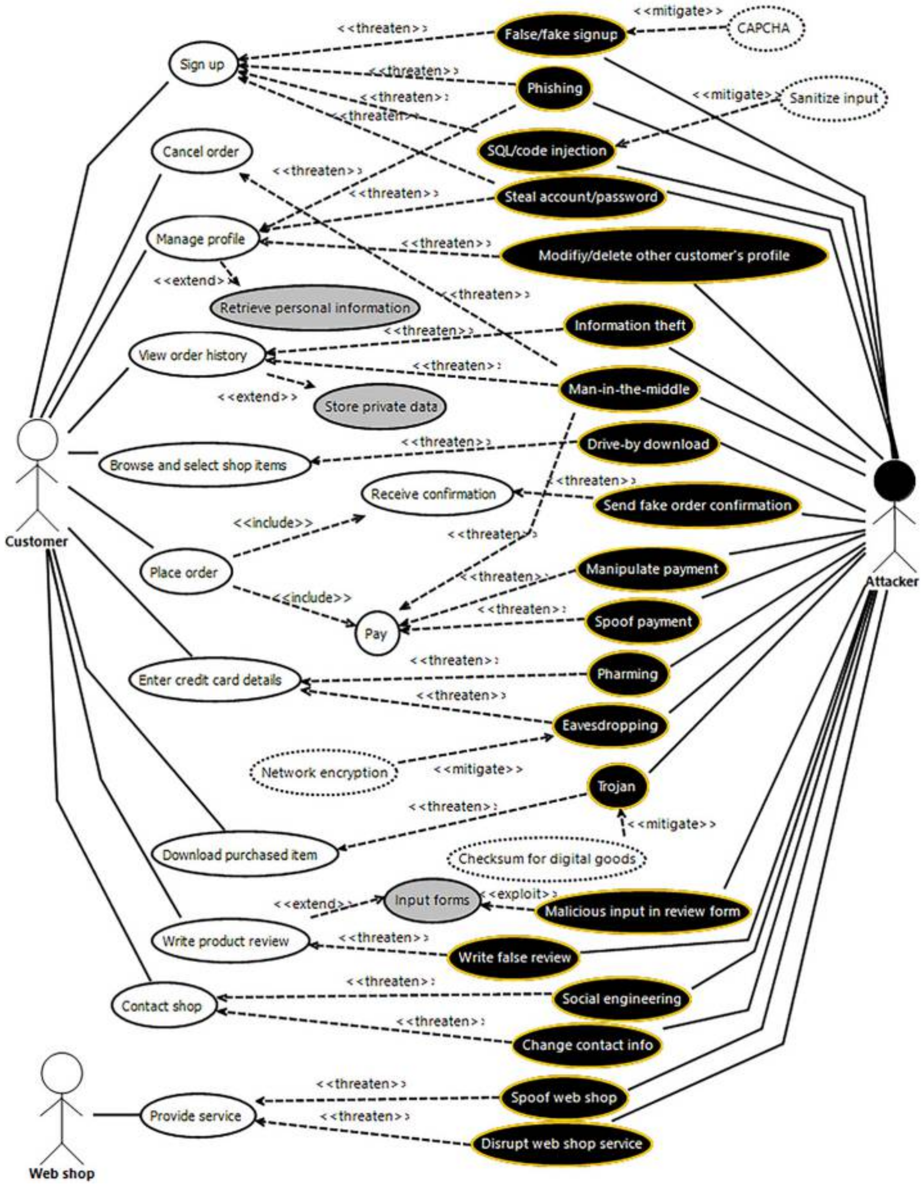


Figure 9.
A combination of the misuse case models made by the students

are all about 60 per cent higher for the experts. This is to be expected, as the experts had a deeper security knowledge and did also have some additional time for developing their models. The number of reactive barriers was clearly higher for the experts, but this is in line with a general observation that the students tended to focus on the left side of the diagram. In fact, 3 of the 13 models from the students had no elements on the right side whatsoever.

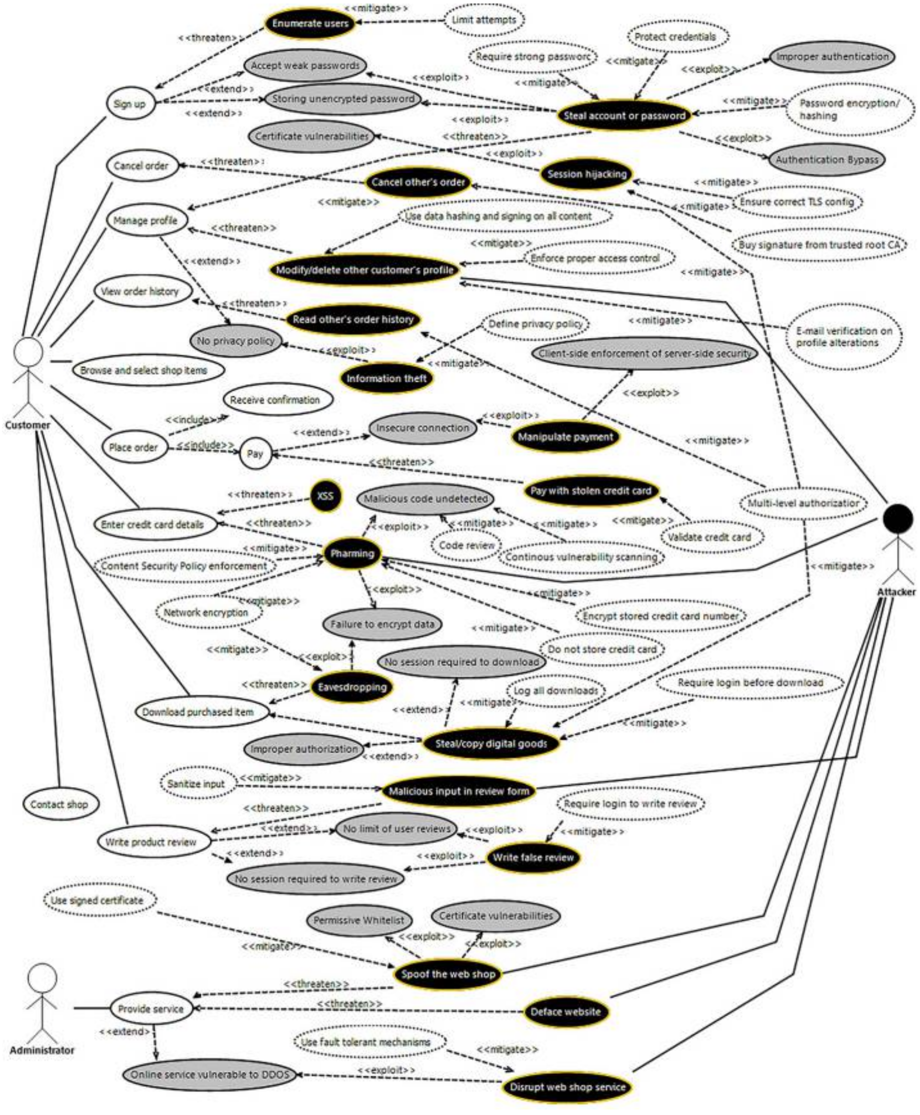


Figure 10.
A combination of the misuse case models made by the experts

Another significant difference was that two of the experts modelled two or three barriers for most of their threats, while this was not observed in any of the student models where all threats had just a single control/barrier. This can be interpreted in two ways; the students did not fully understand that the tool supported adding more than one barrier per threat, or the students did not think that it is necessary to implement more than one barrier per threat in a real system. The last experts did, as mentioned above, not model any barriers, and this skews the average barrier per threat significantly. Identifying a wide range of barriers is

Table III.
A numerical summary of misuse case model elements

Measurement	Experts	Students
Number of participants	3	~62
Number of models	3	31
Total number of misuse activities	27	251
Number of distinct misuse activities	18	48
Average number of misuse activities per model	9	8.1
Total number of vulnerable activities	20	10
Number of distinct vulnerable activities	16	7
Average number of vulnerable activities per model	6.7	0.3
Total number of security activities	26	37
Number of distinct security activities	24	19
Average number of security activities per model	8.7	1.2
Common (misuse/vulnerable/security) activities	17/1/5	

considered one of the primary advantages of bow-tie modelling, and we have made a note to encourage this a bit more in later work.

When we consider the students as a collaborative group, the numbers of the distinct threats, consequences and both types of barriers are almost identical to what the experts produced. When we look beyond these numbers and compare the type of elements in the taxonomy, there is a clear tendency for the experts to focus on technical threats and threats that are specific for digital exams, while the students have included more of the traditional ways of cheating. We believe that both these inputs can be important, and advocate for a combination of security experts and end-users (in our case, the students) when developing these kinds of security models, and consequently defining requirement based on barriers.

Our general impression is that the students showed great creativity, covering most of the same threats and consequences as the experts identified, and discovering additional ones as well. The bow-tie notation did not seem like an obstacle for expressing this, which confirms our hypothesis that the bow-tie notation has a suitable expressiveness for security as well as safety issues. The students also identified additional elements on the consequence side that the experts had not thought of, even though it seems like the students spent most of their time on the threat side. The students seemed just as good as the experts at staying inside the scope of the top event, something we believe can be attributed to the misuse case presentation in the introduction of the experiment.

6.2 Comparing results from bow-tie and misuse case experiments

Having performed similar experiments with different modelling techniques allowed us to see if some phenomena can be generalized or if they are specific for the technique at hand.

Just as with the bow-tie threats, the combined mass of students was able to identify a broader set of misuse case activities. This might not be a surprise given that the number of students was much greater, especially in the misuse case experiment. However, this observation was not consistent when it comes to the other model elements, such as vulnerable and security activities. This phenomenon of imbalance in the models made by the student population was present in both experiments, but that does not necessarily make the models poor compared to the ones from the experts. In both case A and B, the students had domain knowledge and practical experience as service end users, and were able to imagine lots of “bad stuff”, i.e. malicious or deviant behaviour. The security experts had (and should have) a better repertoire of common pitfalls and security solutions from years developing real-life systems, and this became visible in both cases as well.

Looking at the average number of model elements that were added to each model, this number was exactly the same for the student populations in both experiments (9.6). The experts had an average of 16.6 for the bow-tie diagrams and 33.7 for the misuse case models. This is a bit of a surprise because they had more time than the students on the bow-tie experiment, and the same time as the students on the misuse case experiment. This phenomenon can be explained by the fact that the students were almost equally inexperienced to both modelling techniques, while the experts were more experienced to misuse cases than bow-ties. Hence, for inexperienced users, there is no reason to believe that misuse case models outperform bow-tie diagrams in a security context if we consider content generation in isolation.

6.3 Limitations and threat to validity

There are several factors to consider regarding the validity of these experiments. Convenience sampling is a threat to a lot of experiments that involve a population consisting of students, as this can come at the cost of low external validity, but we argue that our samples already had taken an interest in security and represent an aspiring group of people that are likely to work with security engineering in their professional careers. According to a survey on controlled software engineering experiments by Falessi *et al.* (2018), there are pros and cons with both the use of professionals and students, and it is impossible to state that one is always better than the other. Studies by Salman *et al.* (2015), Svahnberg *et al.* (2008) and Höst *et al.* (2005) show that there is little difference in performance between these groups, especially for graduate students (Runeson, 2003).

Though the participation was voluntarily and anonymously, the students seemed motivated and we did not see any submitted diagrams with frivolous content. Furthermore, it was in their own interest to get some relevant experience in security modelling for their course exercises and final exam.

The time that the students had available for the analysis was very limited. In real life, a thorough bow-tie analysis would include defining a series of top events within the same risky environment, and there would be several iterations on each model to improve their coverage and quality. We have tried to address this by letting the students collaborate directly, and by spending time in the introduction on defining a narrow scope for a single top event. Alternatively, we could also have given different top events to different groups and thus have a wider analysis, but that would impose limitations to the comparison afterwards.

Another limiting factor is that we did not perform any systematic user evaluations. Our evidence is thus solely based on the resulting diagrams, aided by observations and comments received during the experiments. For future work, this can be done in several ways, e.g. with standardised usability surveys or adopt from the *Information Systems* (IS) field Moody's *Method Evaluation Model* (2003) that combines measurable constructs such as effectiveness, perceived usefulness and ease of use, intention to use and actual usage (Moody, 2003). Another approach could also be to engage participants in interacting focus groups where they more freely discuss their opinions.

As reported in a previous work (Bernsmed *et al.*, 2017), there are more informal evaluations of situations that combine both safety and security within the same bow-tie diagrams. Though this would have been desirable to try out in this experiment as well, we chose to focus on security issues as we could not find a suitable case where the student would have enough domain knowledge to consider safety, in addition to security.

A final note is related to the different setups of the two experiments. For instance, the modelling cases were not the same, and the bow-ties were made with an online tool, while the misuse cases made with pen and paper. Such elements make direct comparisons more questionable, but as we pointed out in section 2.3, the modelling techniques have different natures, and we do not intent to prove that one is better than the other. We have rather tried to verify that bow-ties used in a security context do not suffer from significant penalties compared to an established security modelling technique.

6.4 Further research directions

Both misuse case models and bow-tie diagrams are high-level modelling techniques, and in their basic forms they are not concerned about attack sequences, relationships between threats, or attributes such as costs and likelihood. Attack (-defence) trees (Schneier, 1999), (Kordy *et al.*, 2014) can for instance be used to further drill down the details of how the unwanted event/attacker goal can be realised, but there is a need to obtain more practical knowledge about what level of granularity and level of detail to represent with various security modelling techniques, and when we should switch between them.

In both experiments, the students and experts did not attempt to transform the bow-tie barriers or misuse security activities into well-defined security requirements. In addition, prioritisation would be the next step of this process, but that would require quantification of risk and mitigation costs. Both these steps are natural continuations that we would like to follow up.

The bow-tie modelling tool itself was not something we set out to evaluate as a part of this study, but observations and comments suggest that the built-in support for creating and connecting the right elements together was helpful indeed. With the misuse case models, which were drawn by hand, the semantics were less strict and the content more difficult to interpret afterwards.

In our bow-tie experiment, the collaborating students were sitting closely together using the same computer, but it would be interesting to see how well a Web-based tool can facilitate online collaboration. Our tool has already built-in functionality for sharing diagrams between users, as well as getting a quick start by importing templates made by others. During the analysis, it also occurred to us that an online voting mechanism could help create consensus about which threats, consequences and associated barriers should be prioritised.

7. Conclusion

Our research hypothesis has been that the bow-tie notation has a suitable expressiveness for security as well as safety, and our results go a long way in verifying this. One of the main strengths of bow-tie analysis is the identification of preventive and reactive barriers, which can be used as traceable sources for the following requirements elicitation process. Naïve professionals might tend to focus on preventive barriers, leading to requirements for risk mitigation or avoidance, while experienced professionals seem to balance this more with reactive barriers and requirements for incident management.

Our results are useful in areas where we need to evaluate safety and security concerns together, especially for domains that have experience in HSE hazards, but now needs to expand this with cybersecurity as well. Of course, there should be further studies on a wider range of situations before this can be generalized across domains. The experiment results also advocate for a combination of people involved when creating security models. Our observations show that the security experts were better at finding technical threats and alternative barriers, while the combined mass of students found a wider range of threats (i.e. ways of cheating) and consequences that would affect individuals such as themselves.

Notes

1. This is an open source tool available and further documented at https://www.safeexambrowser.org/
2. Freely available at https://github.com/KDPRO-SINTEF/BowtieTool

References

Abdo, H., Kaouk, M., Flaus, J.-M. and Masse, F. (2018), "A safety/security risk analysis approach of industrial control systems: a cyber bowtie – combining new version of attack tree with bowtie analysis", *Computers and Security*, Vol. 72, pp. 175-195.

Banerjee, A., Venkatasubramanian, K.K., Mukherjee, T. and Gupta, S.K. (2012), "Ensuring safety, security, and sustainability of mission-critical cyber-physical systems", *Proceedings of the Ieee*, Vol. 100 No. 1, pp. 283-299.

Bau, J. and Mitchell, J.C. (2011), "Security modeling and analysis", *IEEE Security and Privacy Magazine*, Vol. 9 No. 3, pp. 18.

BBC (2016), "Almost 6,000 online shops hit by hackers", [Online], available at: www.bbc.com/news/technology-37643754 (accessed 8 November 2018).

Bernsmed, K., Frøystad, C., Meland, P.H., Nesheim, D.A. and Rødseth, Ø.J. (2017), "Visualizing cyber security risks with bow-tie diagrams", *International Workshop on Graphical Models for Security*, *Springer*, pp. 38-56.

Bradbury, D. (2018), *Internet Shut down in Algeria to Stop Exam Cheats*, *Naked security*, available at: www.theguardian.com/world/2018/jun/21/algeria-shuts-internet-prevent-cheating-school-exams (accessed 25 June 2018).

Carver, J., Jaccheri, L., Morasca, S. and Shull, F. (2004), "Issues in using students in empirical studies in software engineering education", *Software Metrics Symposium, 2003 Proceedings. Ninth International*, *IEEE*, pp. 239-249.

Chen, Y. and He, W. (2013), "Security risks and protection in online learning: a survey", *The International Review of Research in Open and Distributed Learning*, Vol. 14.

Chockalingam, S., Hadžiosmanović, D., Pieters, W., Teixeira, A. and Van Gelder, P. (2016), "Integrated safety and security risk assessment methods: a survey of key characteristics and applications", *International Conference on Critical Information Infrastructures Security*, *Springer*, pp. 50-62.

Elahi, G., Yu, E. and Zannone, N. (2010), "A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities", *Requirements Engineering*, Vol. 15 No. 1, pp. 41-62.

El-Attar, M., Luqman, H., Karpati, P., Sindre, G. and Opdahl, A.L. (2015), "Extending the UML statecharts notation to model security aspects", *IEEE Transactions on Software Engineering*, Vol. 41 No. 7, pp. 661-690.

Fabian, B., Gürses, S., Heisel, M., Santen, T. and Schmidt, H. (2010), "A comparison of security requirements engineering methods", *Requirements Engineering*, Vol. 15 No. 1, pp. 7-40.

Falessi, D., Juristo, N., Wohlin, C., Turhan, B., Münch, J., Jedlitschka, A. and Oivo, M. (2018), "Empirical software engineering experts on the use of students and professionals in experiments", *Empirical Software Engineering*, Vol. 23 No. 1, pp. 452-489.

Firesmith, D.G. (2003), "Security use cases", *Journal of Object Technology*, Vol. 2 No. 3.

Green, D. (2018), "If you shopped at these 7 stores in the last year, your data might have been stolen", [Online]. Business Insider Nordic, available at: https://nordic.businessinsider.com/data-breaches-2018-4 (accessed 8 November 2018).

Höst, M., Wohlin, C. and Thelin, T. (2005), "Experimental context classification: incentives and experience of subjects", *Proceedings of the 27th international conference on Software engineering*, *ACM*, pp. 470-478.

Houmb, S.H., Islam, S., Knauss, E., Jürjens, J. and Schneider, K. (2010), "Eliciting security requirements and tracing them to design: an integration of common criteria, heuristics, and UMLsec", *Requirements Engineering*, Vol. 15 No. 1, pp. 63-93.

ISO/IEC (2011), *ISO/IEC 27005: 2011 Information Technology – Security Techniques–Information Security Risk Management*, ISO.

Jacobson, I. (1993), *Object-Oriented Software Engineering: A Use Case Driven Approach*, Pearson Education India.

Johnson, C. (2011), "Using assurance cases and Boolean logic driven Markov processes to formalise cyber security concerns for safety-critical interaction with global navigation satellite systems", *Electronic Communications of the EASST*, Vol. 45.

Jürjens, J. (2002), "UMLsec: extending UML for secure systems development", *International Conference on The Unified Modeling Language*, *Springer*, pp. 412-425.

Khakzad, N., Khan, F. and Amyotte, P. (2013), "Quantitative risk analysis of offshore drilling operations: a Bayesian approach", *Safety Science*, Vol. 57, pp. 108-117.

Kimminich, B. (2018), "OWASP juice shop tool project", [Online]. OWASP, available: www.owasp.org/index.php/OWASP_Juice_Shop_Project (accessed 8 November 2018).

Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., EL Emam, K. and Rosenberg, J. (2002), "Preliminary guidelines for empirical research in software engineering", *IEEE Transactions on Software Engineering*, Vol. 28 No. 8, pp. 721-734.

Kordy, B., Mauw, S., Radomirović, S. and Schweitzer, P. (2014), "Attack–defense trees", *Journal of Logic and Computation*, Vol. 24 No. 1, pp. 55-87.

Kriaa, S., Pietre-Cambacedes, L., Bouissou, M. and Halgand, Y. (2015), "A survey of approaches combining safety and security for industrial control systems", *Reliability Engineering and System Safety*, Vol. 139, pp. 156-178.

Kumar, R. and Stoelinga, M. (2017), "Quantitative security and safety analysis with attack-fault trees. High assurance systems engineering (HASE)", *IEEE 18th International Symposium on, 2017. IEEE*, pp. 25-32.

Lewis, S. and Smith, K. (2010), "Lessons learned from real world application of the bow-tie method", *6th Global Congress on Process Safety*, *American Institute of Chemical Engineers San Antonio, TX*, pp. 22-24.

Lodderstedt, T., Basin, D. and Doser, J. (2002), "SecureUML: a UML-based modeling language for model-driven security", *International Conference on the Unified Modeling Language*, *Springer*, pp. 426-441.

London, M. (2017), "5 Ways to cheat on online exams", [Online]. Inside Higher ED, available at: www.insidehighered.com/digital-learning/views/2017/09/20/creative-ways-students-try-cheat-online-exams (accessed 27 September 2018).

Lu, L., Liang, W., Zhang, L., Zhang, H., Lu, Z. and Shan, J. (2015), "A comprehensive risk evaluation method for natural gas pipelines by combining a risk matrix with a bow-tie model", *Journal of Natural Gas Science and Engineering*, Vol. 25, pp. 124-133.

Maggi, F., Quarta, D., Pogliani, M., Polino, M., Zanchettin, A.M. and Zanero, S. (2017), "Rogue robots: testing the limits of an industrial robot's security. Technical report", Trend Micro, Politecnico di Milano.

Marsh, S. (2017), "More university students are using tech to cheat in exams", [Online]. The Guardian, available at: www.theguardian.com/education/2017/apr/10/more-university-students-are-using-tech-to-in-exams (accessed 27 September 2018).

Matulevicius, R., Mayer, N. and Heymans, P. (2008), "Alignment of misuse cases with security risk management", *Availability, reliability and security, ARES 08. Third international conference on, 2008. IEEE*, pp. 1397-1404.

Mayer, N., Heymans, P. and Matulevicius, R. (2007), *Design of a Modelling Language for Information System Security Risk Management*, RCIS, pp. 121-132.

Meland, P.H. (2018a), "Bowtie experiment NTNU SINTEF 2018", [Online]. NTNU, available at: https://doi.org/10.21400/f685ryu2 (accessed 20 November 2018).

Meland, P.H. (2018b), "Misusecaseexperiments_SINTEF", [Online]. Zenodo, available at: https://doi.org/10.5281/zenodo.1492322 (accessed 20 November 2018).

Meland, P.H., Bernsmed, K., Frøystad, C. and Li, J. (2018), "An experimental evaluation of bow-tie analysis for cybersecurity requirements", in Katsikas, S.K., Cuppens, F., Cuppens, N., Lambrinoudakis, C., Kalloniatis, C., Mylopoulos, J., Antón, A. and Gritzalis, S. (Eds), *ESORICS 2018 International Workshops, CyberICPS 2018 and SECPRE 2018, September 06-07 2018*, Barcelona. Springer.

Mellado, D., Blanco, C., Sánchez, L.E. and Fernández-Medina, E. (2010), "A systematic review of security requirements engineering", *Computer Standards Interfaces*, Vol. 32 No. 4, pp. 153-165.

Mokhtari, K., Ren, J., Roberts, C. and Wang, J. (2011), "Application of a generic bow-tie based risk analysis framework on risk management of sea ports and offshore terminals", *Journal of Hazardous Materials*, Vol. 192 No. 2, pp. 465-475.

Moody, D.L. (2003), "The method evaluation model: a theoretical model for validating information systems design methods", *ECIS 2003 Proceedings*, Vol. 79.

Mouratidis, H., Giorgini, P. (2007), "Secure tropos: a security-oriented extension of the tropos methodology", *International Journal of Software Engineering and Knowledge Engineering*, Vol. 17, pp. 285-309.

Nolan, D.P. (2014), "Safety and security review for the process industries: application of HAZOP", *PHA, What-If and SVA Reviews*, Elsevier.

Pfleeger, S.L. (1994), "Design and analysis in software engineering: the language of case studies and formal experiments", *ACM SIGSOFT Software Engineering Notes*, Vol. 19 No. 4, pp. 16-20.

Piètre-Cambacédès, L. and Bouissou, M. (2013), "Cross-fertilization between safety and security engineering", *Reliability Engineering and System Safety*, Vol. 110, pp. 110-126.

Raspotnig, C., Karpati, P. and Katta, V. (2012), "A combined process for elicitation and analysis of safety and security requirements", *Enterprise, Business-Process and Information Systems Modeling*, Springer.

Røstad, L. (2008), "An extended misuse case notation: including vulnerabilities and the insider threat", *Access Control in Healthcare Information Systems*, Vol. 67.

Runeson, P. (2003), "Using students as experiment subjects – an analysis on graduate and freshmen student data", *Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering*, Citeseer, pp. 95-102.

Salman, I., Misirli, A.T. and Juristo, N. (2015), "Are students representatives of professionals in software engineering experiments? Software engineering (ICSE)", *IEEE/ACM 37th IEEE International Conference on, 2015*, *IEEE*, pp. 666-676.

Schmittner, C., Ma, Z. and Smith, P. (2014), "Fmvea for safety and security analysis of intelligent and cooperative vehicles", *International Conference on Computer Safety, Reliability, and Security*, *Springer*, pp. 282-288.

Schneier, B. (1999), "Attack trees", *Dr Dobb's Journal*, Vol. 24, pp. 21-29.

Shostack, A. (2008), "Experiences threat modeling at Microsoft", *Modeling security workshop. Department of Computing*, *Lancaster University*, *UK*.

Sindre, G. and Opdahl, A.L. (2001), *Capturing Security Requirments through Misuse Cases*, Norsk Informatikkonferanse 2001.

Sjøberg, D.I., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanovic, A., Liborg, N.-K. and Rekdal, A. C. (2005), "A survey of controlled experiments in software engineering", *IEEE Transactions on Software Engineering*, Vol. 31, pp. 733-753.

Souag, A., Mazo, R., Salinesi, C. and Comyn-Wattiau, I. (2016), "Reusable knowledge in security requirements engineering: a systematic mapping study", *Requirements Engineering*, Vol. 21 No. 2, pp. 251-283.

Stålhane, T. and Sindre, G. (2008), "Safety hazard identification by misuse cases: experimental comparison of text and diagrams", *International Conference on Model Driven Engineering Languages and Systems*, *Springer*, pp. 721-735.

Svahnberg, M., Aurum, A. and Wohlin, C. (2008), "Using students as subjects-an empirical evaluation", *Proceedings of the Second ACM-IEEE international symposium on Empirical Software Engineering and Measurement*, *ACM*, pp. 288-290.

Tichy, W.F. (1998), "Should computer scientists experiment more?", *Computer*, Vol. 31 No. 5, pp. 32-40.

Trbojevic, V.M. and Carr, B.J. (2000), "Risk based methodology for safety improvements in ports", *Journal of Hazardous Materials*, Vol. 71 No. 1-3, pp. 467-480.

Van Lamsweerde, A. (2004), "Elaborating security requirements by construction of intentional anti-models", *Proceedings of the 26th International Conference on Software Engineering*, *IEEE Computer Society*, pp. 148-157.

Vegendla, A., Søgaard, T.M. and Sindre, G. (2016), "Extending HARM to make test cases for penetration testing", *6th International Workshop on Information Systems Security Engineering*, *Ljubljana, Slovenia*, *Springer*, pp. 254-265.

Winther, R., Johnsen, O.-A. and Gran, B.A. (2001), "Security assessments of safety critical systems using HAZOPs", *International Conference on Computer Safety, Reliability, and Security*, *Springer*, pp. 14-24.

WMN (2014), "IMB: Shipping next playground for hackers", [Online], available at: https://worldmaritimenews.com/archives/134727/imb-shipping-next-playground-for-hackers/ (accessed 20 November 2018).

Zalewski, J., Drager, S., Mckeever, W. and Kornecki, A.J. (2012), *Towards Experimental Assessment of Security Threats in Protecting the Critical Infrastructure*, ENASE, pp. 207-212.

About the authors

Per Håkon Meland is a Senior Research Scientist at the independent research institute SINTEF in Norway. In 2002, he obtained an MSc in Computer Science at the Norwegian University of Science and Technology, where he is also a PhD Fellow in the intertwined fields of threat modelling and security economics. Per Håkon Meland is the corresponding author and can be contacted at: per.h.meland@sintef.no

Karin Bernsmed is a Senior Research Scientist at SINTEF. She has a PhD from the Norwegian University of Science and Technology, where she also holds an Associate Assistant Professor position. Her research interests include security threat and risk assessment, requirements engineering and design of secure and robust ICT systems.

Christian Frøystad is a Research Scientist at the independent research institute SINTEF in Norway. He obtained an MSc in Computer Science at the Norwegian University of Science and Technology in 2015.

Jingyue Li is an Associate Professor at the Department of Computer Science, Norwegian University of Science and Technology. He obtained a PhD from the Norwegian University of Science and Technology in 2006. His research interests lie in software engineering, software security and system safety.

Guttorm Sindre is a Professor at the Department of Computer Science, Norwegian University of Science and Technology, and is a Leader for the Excited Centre for Excellence in IT Education. He obtained a PhD from the Norwegian Institute of Technology in 1990. His research interests lie in requirements engineering, security requirements and IT education and didactics.