An experimental evaluation of bow-tie analysis for cybersecurity requirements

Per Håkon Meland[1,2], Karin Bernsmed[1], Christian Frøystad[1], Jingyue Li[2], and Guttorm Sindre[2]

[1] SINTEF Digital, Norway
{per.h.meland,karin.bernsmed,christian.froystad}@sintef.no
[2] Norwegian University of Science and Technology, Norway
{per.hakon.meland,jingyue.li,guttorm.sindre}@ntnu.no

.

Abstract. Bow-tie analysis includes a graphical representation for depicting threats and consequences related to unwanted events, and shows how preventive and reactive barriers can provide control over such situations. This kind of analysis has traditionally been used to elicit requirements for safety and reliability engineering, but as a consequence of the ever-increasing coupling between the cyber and physical world, security has become an additional concern. Through a controlled experiment, we provide evidence that the expressiveness of the bow-tie notation is suitable for this purpose as well. Our results show that a sample population of graduate students, inexperienced in security modelling, perform similarly as security experts when we have a well-defined scope and familiar target system/situation. We also demonstrate that misuse case diagrams should be regarded as more of a complementary than competing modelling technique.

Keywords: bow-tie analysis, requirements elicitation, controlled experiment, digital exams

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 cyberelement adversely affects the physical environment [2]. 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 [20]. 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 "the next playground for hackers" [42]. A common trait to all of these industries, is that there are already well-established practices for managing safety concerns. If these practices can be extended to also encompass security, we might have an easier path than introducing a set of security analysis techniques that are unfamiliar to them and must be done 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 [3]. Security modelling comes in many different forms and flavours [4], and there is not necessarily one single best or correct way of performing it [34]. 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 a number of other phenomena within software engineering disciplines, 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 in which the claims are true [26]. According to Tichy [39], "experimentation can accelerate progress by quickly eliminating fruitless approaches, erroneous assumptions, and fads. It also helps orient engineering and theory into promising directions".

The purpose of this paper is to present the result of an experiment related to 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 explore how well the same analysis technique performs in the context of security, and complements to existing security modelling techniques, such as misuse case diagrams [36]. The research hypothesis central to this work is that *the bow-tie notation has a suitable expressiveness for security as well as safety.* 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 assessment where we need to consider both safety and security in combination would make no sense.

This paper is structured as follows. We briefly show related work and explain the history and notation of bow-ties in Section 2. The same section also show how bow-tie diagrams compares with misuse case diagrams. In Section 3, we explain our research method and the details of the experiment 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 [11] 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. [41] 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. [28] make use of UML-based models within a combined safety and security assessment process to elicitate requirements. Kumar and Stoelinga [16] 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 on safety and security reviews for the process industries [25], but examples here only focus on safety incidents. FMVEA (Failure Mode, Vulnerabilities and Effect Analysis) [32] is safety and security co-analysis method extended from FMEA (Failure Mode and Effect Analysis), which is a safety analysis method. Like FMEA, FMVEA proposes to use the STRIDE model [35] to identify threat modes first, and then analyze the effect each threat mode. 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. [43], Piètre-Cambacédès and Bouissou [27], Chockalingam et al. [7], as well as Kriaa et al. [15].

2.2 Bow-tie history

Bow-tie analysis has since the 1970s been used by organisations world-wide for risk management purposes, but primarily to demonstrate control over health, safety and environmental (HSE) hazards [17]. For instance, Khakzad et al. show this application in safety risk analysis in offshore drilling [12], Trbojevic and Carr [40], as well as Mokhtari et al. [23], do the same for safety assessment in international maritime ports, and Lu et al. [19] 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 a barrier. As already described by Bernsmed et al. [4], 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. 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.

2.3 The Bow-tie modelling notation

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 [17]. Figure 1 gives an overview of the modelling elements that have been included in our experiment, based on [4]. First of all, the *Hazard* element represents the riskful 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 [1], 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, 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 since 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.

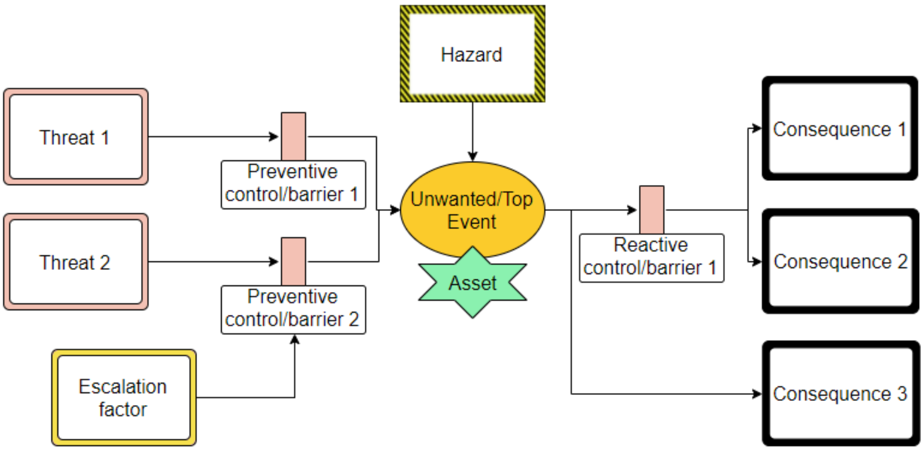


Fig. 1. The basic elements of the bow-tie notation with security extension.

2.4 Bow-tie and misuse case modelling

Misuse case modelling is a well-known technique for graphical security modelling, and can be summarized as an extension to regular UML use cases [10], adding misuse activities, which can be considered as threats, and mis-actors, who are malicious threat agents instantiating the misuse activities [36]. Misuse cases have been proven useful in different industrial cases when considering security [22] and eliciting requirements [36], and are therefore a good basis for comparison with bow-tie diagrams. Table 1 gives an overview of the main properties of both misuse case and bow-tie diagrams. Based on this comparison, 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 experiment.

Table 1. A comparison of misuse case and bow-tie diagrams.

Misuse case diagrams	**Bow-tie diagrams**
[Both] Defined by a simple to understand graphical notation with an open-ended method, allowing for a lot of creativity to the modeller.	
Originate from computer security and requirements engineering, based on UML use case diagrams.	Originate from the safety & reliability domain, related to fault analysis.
Developed to identify malicious actions (misuse) for a given scenario.	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.
Suitable for describing many different misuse activities in a single diagram.	Focus on a single unwanted top event per diagram.
Show actors (threat agents) related to misuse activities.	Do not represent actors, but in which riskful 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 vulnerabilities that a misuse activity can exploit.	Represent threats/causes that can lead to the top event.
Consequences are not part of the diagram.	Explicitly depict possible consequences following the top event.

3 Experiment method

In order to plan our experiment, we adopted and applied the guidelines by Kitchenham et al. [13], 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 [37], and as 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. Since there are no random assignments, this should be classified as a *quasi-experiment*, and as a formal experiment since we have a high level of control over the variables that can affect the truth of the hypothesis [26].

One of the sample populations consists of students, and therefore it has been important to make sure that they perceive a value from participation [5]. 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 [31]. The case in focus and experiment setup is described in the sections below.

3.1 Case: Digital exams

The security modelling assignment we chose is the 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 We [6] 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 due to new technical ways of cheating. According to the Guardian [21], there has been a 42% rise in cheating cases between 2012 and 2016 involving gadgets such as mini cameras and micro earbuds. London [18] gives an overview of further inventive and not-so-inventive ways that have been used for cheating on online exams. 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 of students participating in the exam in the same confined room and within the same time frame. This is a bit different to 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*[3] (SEB), which regulates access to websites, search engines, other applications and system calls, also referred to as *browser lockdown.*

3.2 Experiment setup

Our experiment engaged two types of populations as a basis for comparison; 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 experiment as a part of a classroom exercise in a course on secure software engineering, and were motivated to learn security modelling in order 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 on 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, and in particular bow-tie for

[3] This is an open source tool available and further documented at https://www.safeexambrowser.org/.

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 2, and depicts a number of actors and typical activities related to digital exams, as well as misuse case activities and associated threat actors. 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 that 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 in time try to *change the results* of the exam. During the examination day itself, the main legitimate actor is the *student* that needs to *setup* his/her computer, which also involves sub-activities such as *connecting to the network* and *installing the correct SEB software.* In order 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* actor that inherits all the activities from the legitimate student actor. With the misuse case notation, it is common to use a grey shading for such malicious “insiders” [29]. 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 bow-tie top event. This was demonstrated with *disrupt exam* as shown in Figure 3. 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* in order 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 riskful 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 2-3 persons per model, spending about 30 minutes 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[4] 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 4.

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.

[4] Freely available at https://github.com/KDPRO-SINTEF/BowtieTool

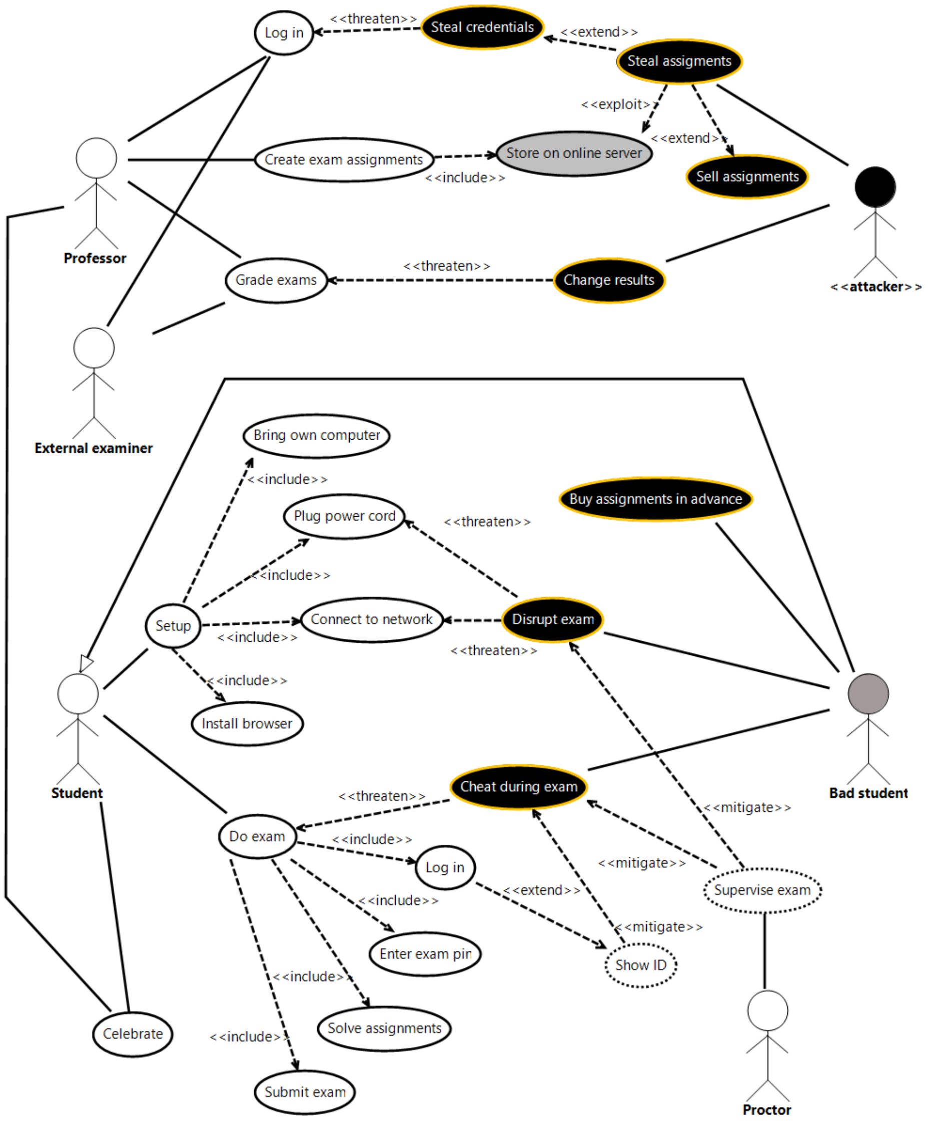


Fig. 2. Defining the scope with a misuse case diagram.

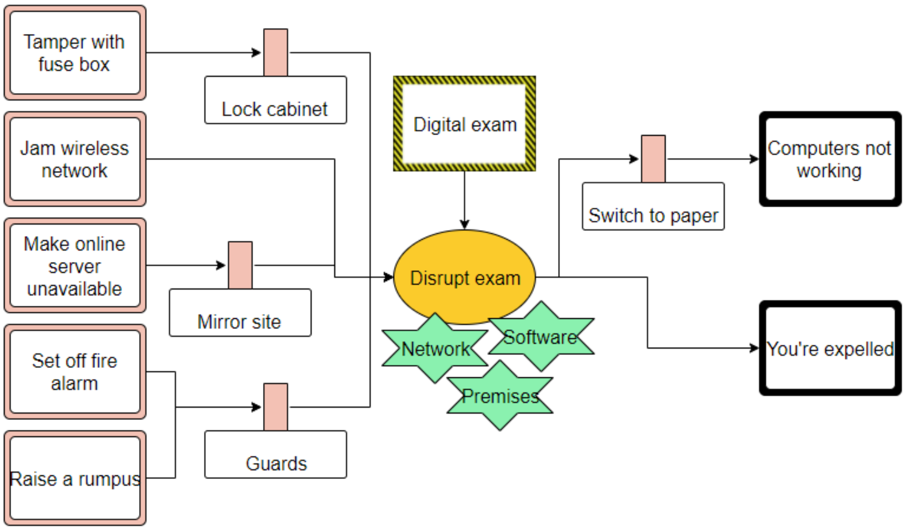
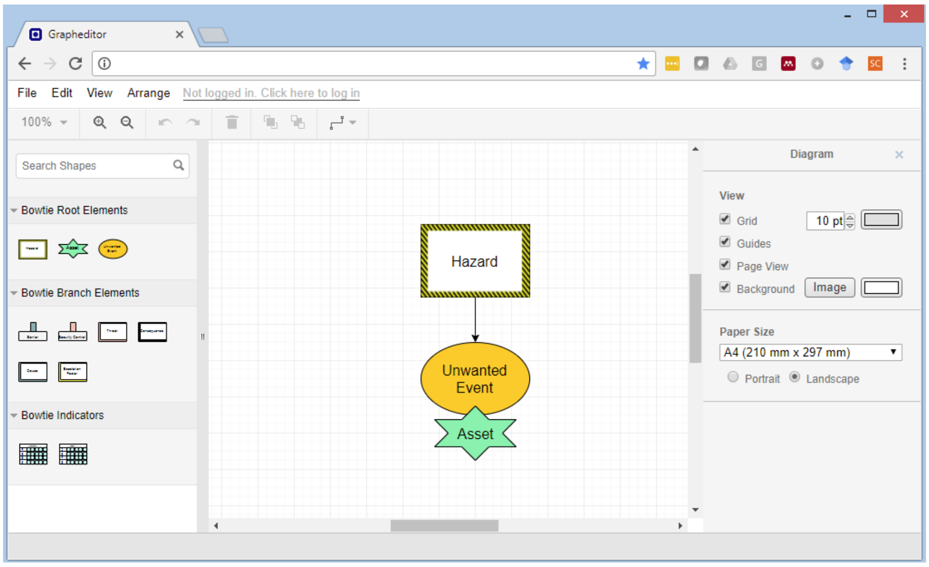


Fig. 3. Example model showed as a preparation.



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

Fig. 4. The online tool used for making the bow-tie diagrams.

4 Results

4.1 Models made by students

A total of 40 students were present in the experiment session, which resulted in 13 different models. Observations from the classroom indicated that approximately 30 students contributed to these models. This estimate is based on the average size of the groups and that we also know that not all models were submitted (this was voluntarily). The models were then analysed, and we created a small taxonomy of threats, controls/barriers and consequences in order to be able to compare them. Based on this, we developed a combined bow-tie diagram, shown in Figure 5 in Appendix A, which also indicates the frequency of the threat and consequence elements found in the models 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 involves smuggling in and making use of some 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 since 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.

Some additional threats can be found in Figure 5, but these were only present in one or two of the models. Additionally, we discarded three threats that were out of scope for this top event, namely *Retrieve exam answers beforehand*, *Disrupt exam* and *Blackmail professor*.

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). It is interesting to see that these are consequences for both successful cheating as well as consequences for the cheater if he/she gets caught.

The combined model 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 models (4/13) contained additional assets, so we added these to the combined model as well.

4.2 Models made by security experts

There were three security experts participating in this experiment, resulting in three independent bow-tie models. These were analysed in the same manner as the student models and aligned using the same taxonomy. The resulting combined model from the experts is shown in Figure 6 in Appendix A. There were only four threats that had an overlap between the expert models; *Access external information*, *Another person takes exam*, *Hack browser* and *Phone outsiders.* The three former were all present among the top threats from the student models as well, while the latter was not. We discarded one threat from the model, *Introduce vulnerability in SEB OSS project*, since this is something that must be done prior to the exam and hence out of scope for this top event. The expert and student models 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 2 shows a numerical comparison of the models created by the two populations. The last row shows how many distinct elements that are common between the combined models from each population. Since 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 model is mapped to the single threat *Digital communication with others* in the student model. Likewise, the preventive barrier *Strong authentication* in the expert model maps towards the less strict *Authentication* in the student model.

Table 2. A numerical summary of 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	

5 Discussion

5.1 Interpretation of 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 are all about 60% higher for the experts. This is to be expected, since 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. 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 considered to be 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 of 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.

5.2 Limitations and threat to validity

There are several factors to consider regarding the validity of this experiment. 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 sample 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. [8], 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. [31], Svahnberg et al. [38] and Höst et al. [9] show that there is little difference in performance between these groups, especially for graduate students [30].

Though the participation was voluntarily and anonymously, the students seemed motivated and we did not see any submitted models 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 and modelling was very limited. In real life, a thorough analysis would include defining a series of top events within the same riskful 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 of this study is that we did not perform any systematic user evaluation. Our evidence is thus solely based on the resulting models, aided by observations and comments received during the experiment. 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* [24] that combines measurable constructs such as effectiveness, perceived usefulness and ease of use, intention to use and actual usage. Another approach could also be to engage participants in interacting focus groups where they more freely discuss their opinions.

In our previous work [4], we have more informally evaluated situations that combine safety and security within the same bow-tie models. 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.

5.3 Further research directions

Both misuse case models and bow-tie diagrams are high-level modelling techniques, and are in their basic forms not concerned about attack sequences, relationships between threats, or attributes such as costs and likelihood. Attack(-Defense) trees [33, 14] 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 this experiment, the students and experts did not attempt to transform the barriers 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 of 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. In our study, the collaborating students were sitting closely together using the same computer, but it would be interesting to see how well such a web-based tool can facilitate online collaboration. The tool has already built-in functionality for sharing models 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.

6 Conclusion

Our research hypothesis has been that the bow-tie notation has a suitable expressiveness for security as well as safety, and our controlled experiment goes 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 have a tendency 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.

Acknowledgment

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. We would like to thank all participants in the experiment, as well as the group of NTNU students developing the bow-tie modelling tool that has supported our work greatly.

References

1. ISO/IEC 27005 Information technology – Security techniques – Information security risk management. Tech. rep. (2008), http://www.iso.org/iso/catalogue_detail?csnumber=56742
2. Banerjee, A., Venkatasubramanian, K.K., Mukherjee, T., Gupta, S.K.S.: Ensuring safety, security, and sustainability of mission-critical cyber–physical systems. Proceedings of the IEEE 100(1), 283–299 (2012)
3. Bau, J., Mitchell, J.C.: Security modeling and analysis. IEEE Security & Privacy 9(3), 18–25 (2011)
4. Bernsmed, K., Frøystad, C., Meland, P.H., Nesheim, D.A., Rødseth, Ø.J.: Visualizing cyber security risks with bow-tie diagrams. In: International Workshop on Graphical Models for Security. pp. 38–56. Springer (2017)
5. Carver, J., Jaccheri, L., Morasca, S., Shull, F.: Issues in using students in empirical studies in software engineering education. In: Software Metrics Symposium, 2003. Proceedings. Ninth International. pp. 239–249. IEEE (2004)
6. Chen, Y., He, W.: Security risks and protection in online learning: A survey. The International Review of Research in Open and Distributed Learning 14(5) (2013)
7. Chockalingam, S., Hadziosmanovic, D., Pieters, W., Teixeira, A., van Gelder, P.: Integrated safety and security risk assessment methods: A survey of key characteristics and applications. arXiv preprint arXiv:1707.02140 (2017)

8. Falessi, D., Juristo, N., Wohlin, C., Turhan, B., Münch, J., Jedlitschka, A., Oivo, M.: Empirical software engineering experts on the use of students and professionals in experiments. Empirical Software Engineering 23(1), 452–489 (Feb 2018)
9. Höst, M., Wohlin, C., Thelin, T.: Experimental context classification: incentives and experience of subjects. In: Proceedings of the 27th international conference on Software engineering. pp. 470–478. ACM (2005)
10. Jacobson, I.: Object-oriented software engineering: a use case driven approach. Pearson Education India (1993)
11. Johnson, C.: 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 45 (2011)
12. Khakzad, N., Khan, F., Amyotte, P.: Quantitative risk analysis of offshore drilling operations: a bayesian approach. Safety science 57, 108–117 (2013)
13. Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K., Rosenberg, J.: Preliminary guidelines for empirical research in software engineering. IEEE Transactions on software engineering 28(8), 721–734 (2002)
14. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Attack–defense trees. Journal of Logic and Computation 24(1), 55–87 (2014)
15. Kriaa, S., Pietre-Cambacedes, L., Bouissou, M., Halgand, Y.: A survey of approaches combining safety and security for industrial control systems. Reliability Engineering & System Safety 139, 156–178 (2015)
16. Kumar, R., Stoelinga, M.: Quantitative security and safety analysis with attack-fault trees. In: High Assurance Systems Engineering (HASE), 2017 IEEE 18th International Symposium on. pp. 25–32. IEEE (2017)
17. Lewis, S., Smith, K.: Lessons learned from real world application of the bow-tie method. In: American Institute of Chemical Engineers 6th Global Congress on Process Safety (2010)
18. London, M.: 5 ways to cheat on online exams (September 2017), https://www.insidehighered.com/digital-learning/views/2017/09/20/creative-ways-students-try-cheat-online-exams
19. Lu, L., Liang, W., Zhang, L., Zhang, H., Lu, Z., Shan, J.: 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 25, 124–133 (2015)
20. Maggi, F., Quarta, D., Pogliani, M., Polino, M., Zanchettin, A.M., Zanero, S.: Rogue robots: Testing the limits of an industrial robots security. Tech. rep., Technical report, Trend Micro, Politecnico di Milano (2017)
21. Marsh, S.: More university students are using tech to cheat in exams (April 2017), https://www.theguardian.com/education/2017/apr/10/more-university-students-are-using-tech-to-in-exams
22. Matulevicius, R., Mayer, N., Heymans, P.: Alignment of misuse cases with security risk management. In: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on. pp. 1397–1404. IEEE (2008)
23. Mokhtari, K., Ren, J., Roberts, C., Wang, J.: Application of a generic bow-tie based risk analysis framework on risk management of sea ports and offshore terminals. Journal of hazardous materials 192(2), 465–475 (2011)
24. Moody, D.L.: The method evaluation model: a theoretical model for validating information systems design methods. ECIS 2003 proceedings p. 79 (2003)
25. Nolan, D.P.: Safety and Security Review for the Process Industries: Application of HAZOP, PHA, What-IF and SVA Reviews. Elsevier (2014)

26. Pfleeger, S.L.: Design and analysis in software engineering: The language of case studies and formal experiments. SIGSOFT Softw. Eng. Notes 19(4), 16–20 (Oct 1994)
27. Piètre-Cambacédès, L., Bouissou, M.: Cross-fertilization between safety and security engineering. Reliability Engineering & System Safety 110, 110–126 (2013)
28. Raspotnig, C., Karpati, P., Katta, V.: A Combined Process for Elicitation and Analysis of Safety and Security Requirements, pp. 347–361. Springer Berlin Heidelberg, Berlin, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-31072-0_24
29. Røstad, L.: An extended misuse case notation: Including vulnerabilities and the insider threat. Access Control in Healthcare Information Systems p. 67 (2008)
30. Runeson, P.: Using students as experiment subjects–an analysis on graduate and freshmen student data. In: Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering. pp. 95–102. Citeseer (2003)
31. Salman, I., Misirli, A.T., Juristo, N.: Are students representatives of professionals in software engineering experiments? In: Proceedings of the 37th International Conference on Software Engineering-Volume 1. pp. 666–676. IEEE Press (2015)
32. Schmittner, C., Ma, Z., Smith, P.: Fmvea for safety and security analysis of intelligent and cooperative vehicles. In: Bondavalli, A., Ceccarelli, A., Ortmeier, F. (eds.) Computer Safety, Reliability, and Security. pp. 282–288. Springer International Publishing, Cham (2014)
33. Schneier, B.: Attack trees. Dr. Dobbs journal 24(12), 21–29 (1999)
34. Shostack, A.: Experiences threat modeling at microsoft. In: Modeling Security Workshop. Dept. of Computing, Lancaster University, UK (2008)
35. Shostack, A.: Threat modeling: Designing for security. John Wiley & Sons (2014)
36. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requirements engineering 10(1), 34–44 (2005)
37. Sjoeberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanovic, A., Liborg, N.K., Rekdal, A.C.: A survey of controlled experiments in software engineering. IEEE Transactions on Software Engineering 31(9), 733–753 (Sept 2005)
38. Svahnberg, M., Aurum, A., Wohlin, C.: Using students as subjects-an empirical evaluation. In: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement. pp. 288–290. ACM (2008)
39. Tichy, W.F.: Should computer scientists experiment more? Computer 31(5), 32–40 (1998)
40. Trbojevic, V.M., Carr, B.J.: Risk based methodology for safety improvements in ports. Journal of hazardous materials 71(1-3), 467–480 (2000)
41. Winther, R., Johnsen, O.A., Gran, B.A.: Security assessments of safety critical systems using hazops. In: International Conference on Computer Safety, Reliability, and Security. pp. 14–24. Springer (2001)
42. World Maritime News: IMB: Shipping Next Playground for Hackers (2014), http://worldmaritimenews.com/archives/134727/imb-shipping-next-playground-for-hackers/
43. Zalewski, J., Drager, S., McKeever, W., Kornecki, A.J.: Towards experimental assessment of security threats in protecting the critical infrastructure. In: ENASE 2012-Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering, Wroclaw, Poland (2012)

A Combined bow-tie diagrams

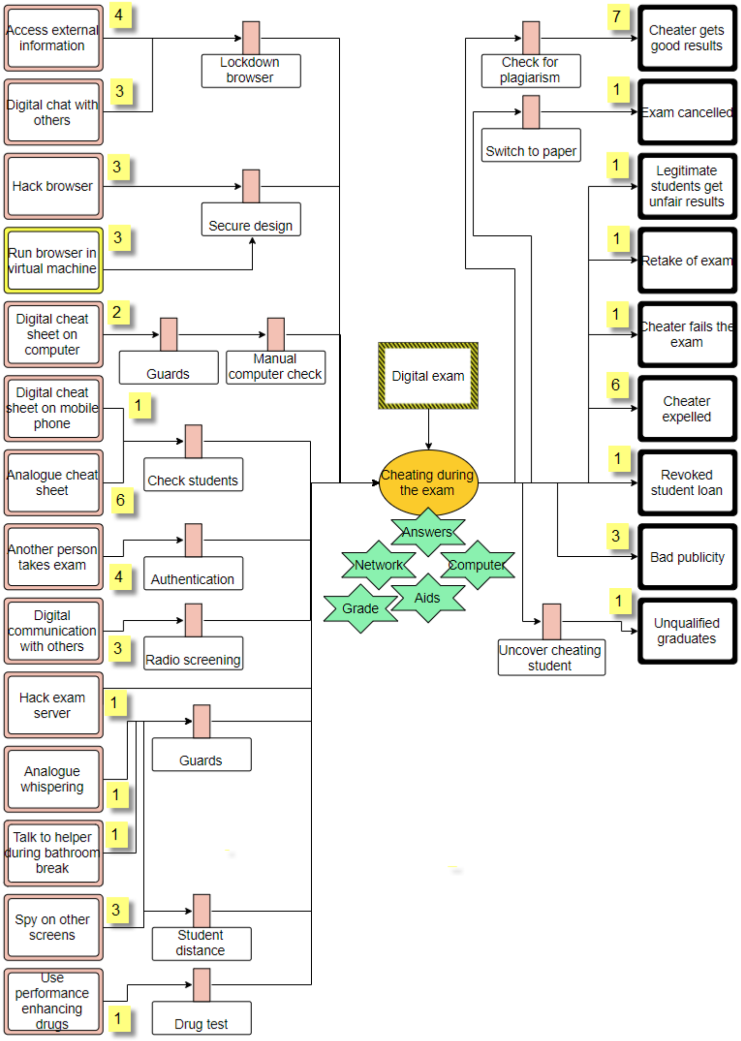


Fig. 5. A combination of the models made by the students.

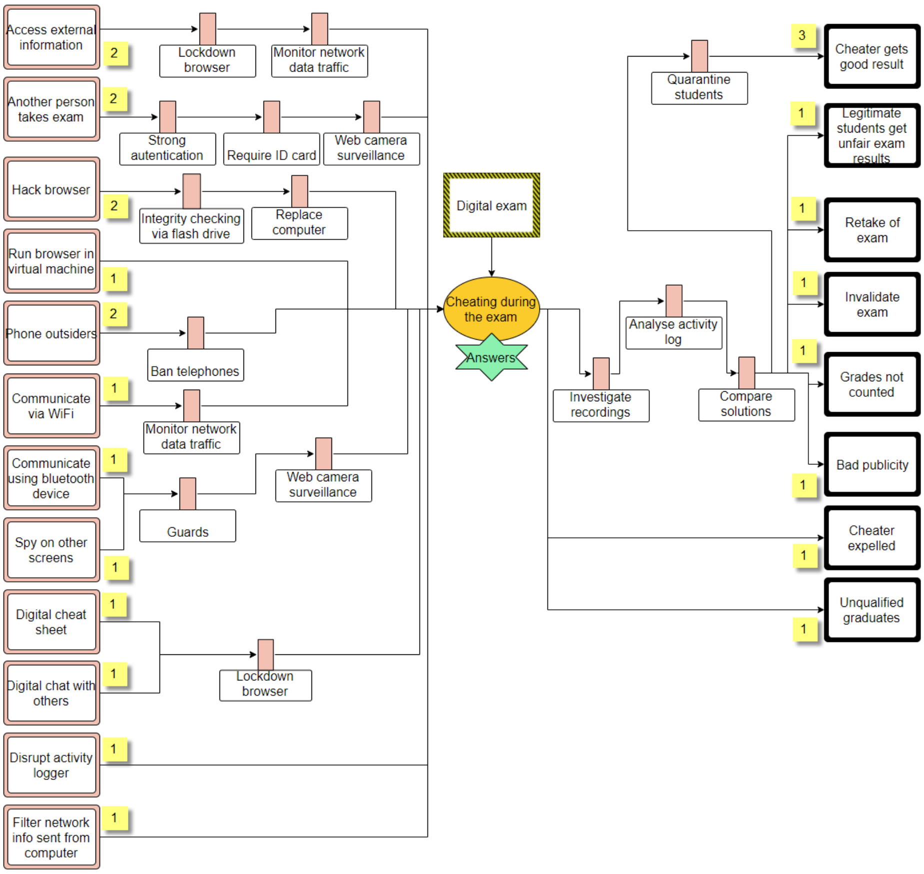


Fig. 6. A combination of the models made by the experts.