Misuse Cases and Abuse Cases in Eliciting Security Requirements

Chun Wei (Johnny), Sia
Department of Computer Science,
University of Auckland
csia005@ec.auckland.ac.nz
25 October 2005

Abstract: Misuse cases, the inverted version of a use case can be used to elicit security requirements. Abuse cases also are used in eliciting security requirements. Their notation appears to be similar. This paper presents a brief comparison between misuse cases and abuse cases. It is observed that misuse cases are able to model a wider range of mis-users and they also interact with use cases in interesting and helpful ways. Misuse cases do appear to be more developed compared to abuse cases, however both approaches have not been assessed in practical software development projects.

1 Introduction

Security requirements elicitation is important during the early stages of the software development life cycle because it determines whether the system is vulnerable to future attacks when it is exposed to the real world. Instead, security requirements are usually prepared after a product is finished and sold. Misuse and abuse cases offer the opportunity to elicit security requirements earlier; before the product is released into the real world.

Other published papers which refer to misuse cases and abuse cases tend to say misuse cases are the same as abuse cases. For example [4], "Also abuse cases and misuse cases have demonstrated how one can make explicit, and counteract, threatening scenarios". While this does not necessarily show that the author believes that abuse cases are the same as misuse cases, it does however reveal that the author believes they have the same purpose. In another paper [5], the authors write "… use cases [have] also been investigated in connection with security and safety requirements, in the form of misuse cases, also known as abuse cases". This quote certainly shows us that the authors of that paper believe misuse cases are actually the same as abuse cases. In [6] the author writes "A security 'misuse' case, a variation on a use case …

ultimately used to identify security requirements or security use cases. A similar concept has been described as an ‘abuse’ case”. This is another paper which agrees that misuse cases are similar to abuse cases. An interesting point here is that that paper contrasted use cases with abuse cases, and also contrasted misuse cases with security use cases; however the author did not contrast abuse cases with misuse cases.

This paper first presents a brief introduction of use cases, abuse cases and misuse cases (Section 2). Then it continues with a comparison between misuse cases and abuse cases (Section 3), followed by a discussion (Section 4). Finally Section 5 brings this paper to a close.

2 The Concepts and Notations

Based on the year of publication, abuse cases were introduced first. The first paper on abuse cases was authored by John McDermott [3] and Chris Fox and was published in 1999. A paper introducing misuse cases was later published in 2000 by Sindre [1] and Opdahl. The paper on misuse cases was published later; however it made no mention of abuse cases. Both papers make it clear that abuse cases and misuse cases are not a replacement for use cases. They both extend the standardized Unified Modeling Language (UML) notation for use cases.

2.1 Use cases

Use cases are the basis for abuse cases and misuse cases; therefore it is wise to spend some time with a brief introduction of the symbols. A use case diagram is a representation of actors and of a system’s use cases which describe possible scenarios for a single task or goal. Actors are stick figures. Use cases are ovals. Association lines connect actors with the use cases in which they participate [3]. These symbols are shown in Figure 1.

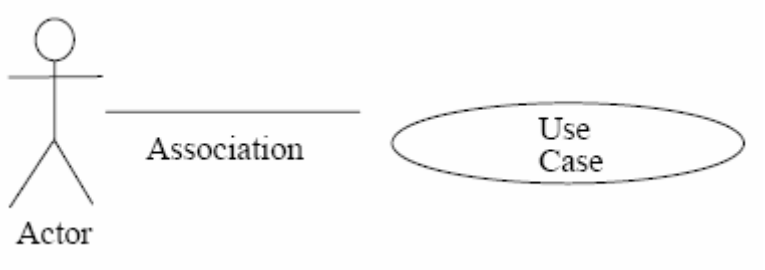

Figure 1. Use Case Diagram symbols

Use cases tend to focus more on what a system does rather then how the system does it. As such, they are not good at showing the opposite; that is, what a system should not do.

2.2 Abuse Cases

The term ‘abuse case’ is defined in [3] as “a specification of a type of complete interaction between a system and one or more actors, where the results of the interaction are harmful to the system, one of the actors, or one of the stakeholders in the system”. Abuse cases were adapted from a proven object-oriented modeling technique, use cases, to capture and analyze security requirements in a simple way. Abuse cases extend the UML notation to model abuse in systems.

John McDermott illustrates a use case diagram for an Internet-Based Information Security Laboratory, and also the abuse case diagram for it. The use case diagram and the abuse case diagram are given in Figure 2 and they both are meant to be drawn separate from one another. There is no new terminology or special symbols introduced for abuse case diagrams. They are drawn with the same symbols as a use case diagram. To distinguish between the two, the use case diagram and abuse case diagrams are kept separate. Hence abuse cases do not appear in the use case diagrams and vice versa.

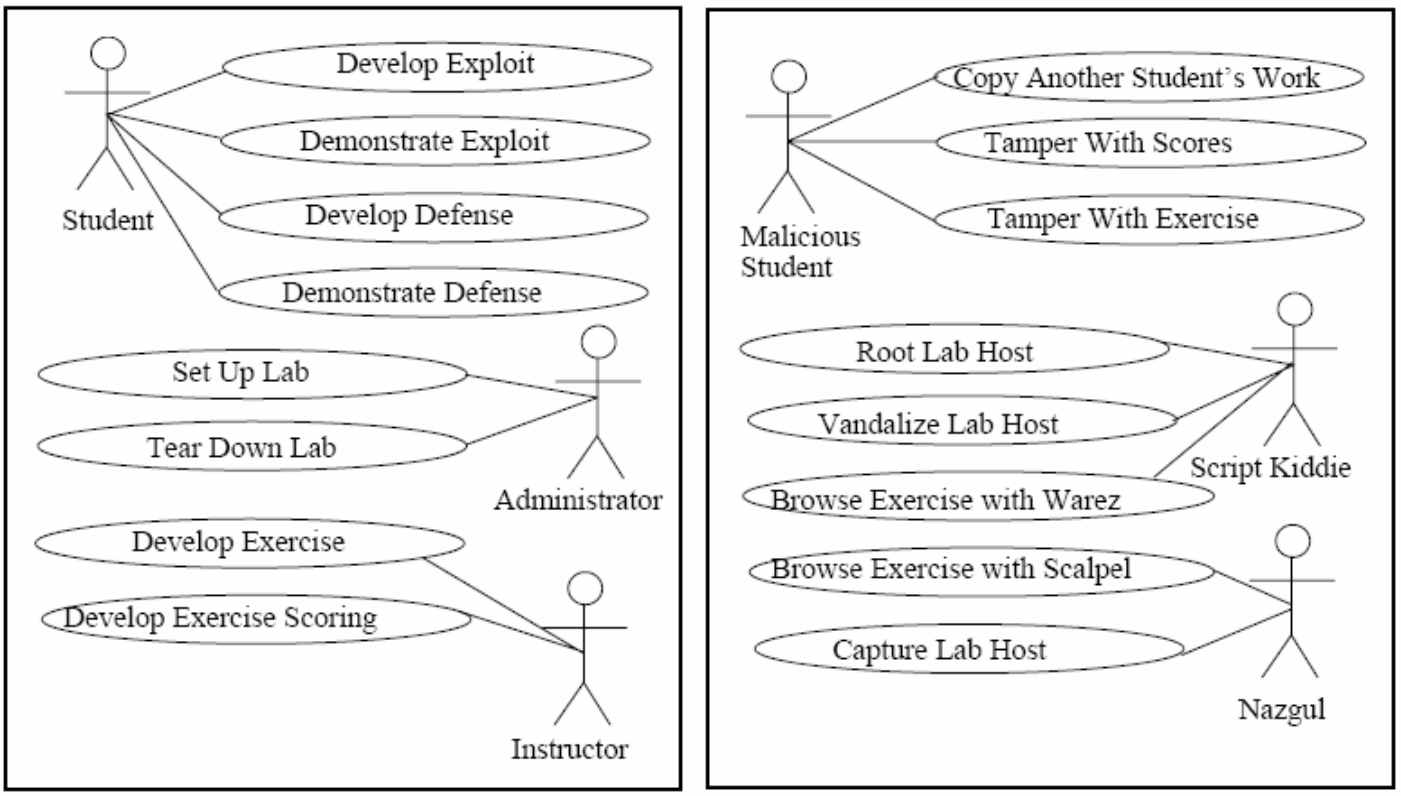

Figure 2. A use case diagram (shown on the left) and an abuse case diagram (shown on the right) for an Internet-Based Information Security Laboratory

2.3 Misuse Cases

Sindre [1] defined misuse cases as a "special kind of use case, describing behavior that the system/entity owner does not want to occur"; which adds on to the definition of a use case stated in the documentation of UML v.1.3 [2]. Sindre also defined some additional terminology. It was defined that a mis-actor is a special kind of actor who initiates misuse cases.

Sindre took the view that it may be interesting to look at the use and misuse together, hence they illustrate use cases and misuse cases in the same diagram. To avoid confusion between the two, misuse cases and any mis-actors are shown in an "inverted" format. This can be seen in Figure 3 where misuse cases use the same symbols as use cases except that the colors are inverted (i.e. white spaces are filled black).

In a more recent paper [7], Sindre re-defines some terminology from his previous paper and follows some suggestions made by Ian Alexander [8]. A misuser (which replaces mis-actor) is “an actor that initiates misuse cases, either intentionally or inadvertently”. The association between a misuse case and a use case can either be a “threatens” or a “mitigates” relationship [8]. A use case can mitigate a misuse case. This means that the use case reduces the chance of a misuse case succeeding. An example is “screen input” which reduces the chance of an outside crook spreading malicious code, as shown in Figure 3. A misuse case can threaten a use case. The use case is exploited or hindered by a misuse case. For example, an outside crook attempting to flood the system could prevent a customer from accessing customer registration.

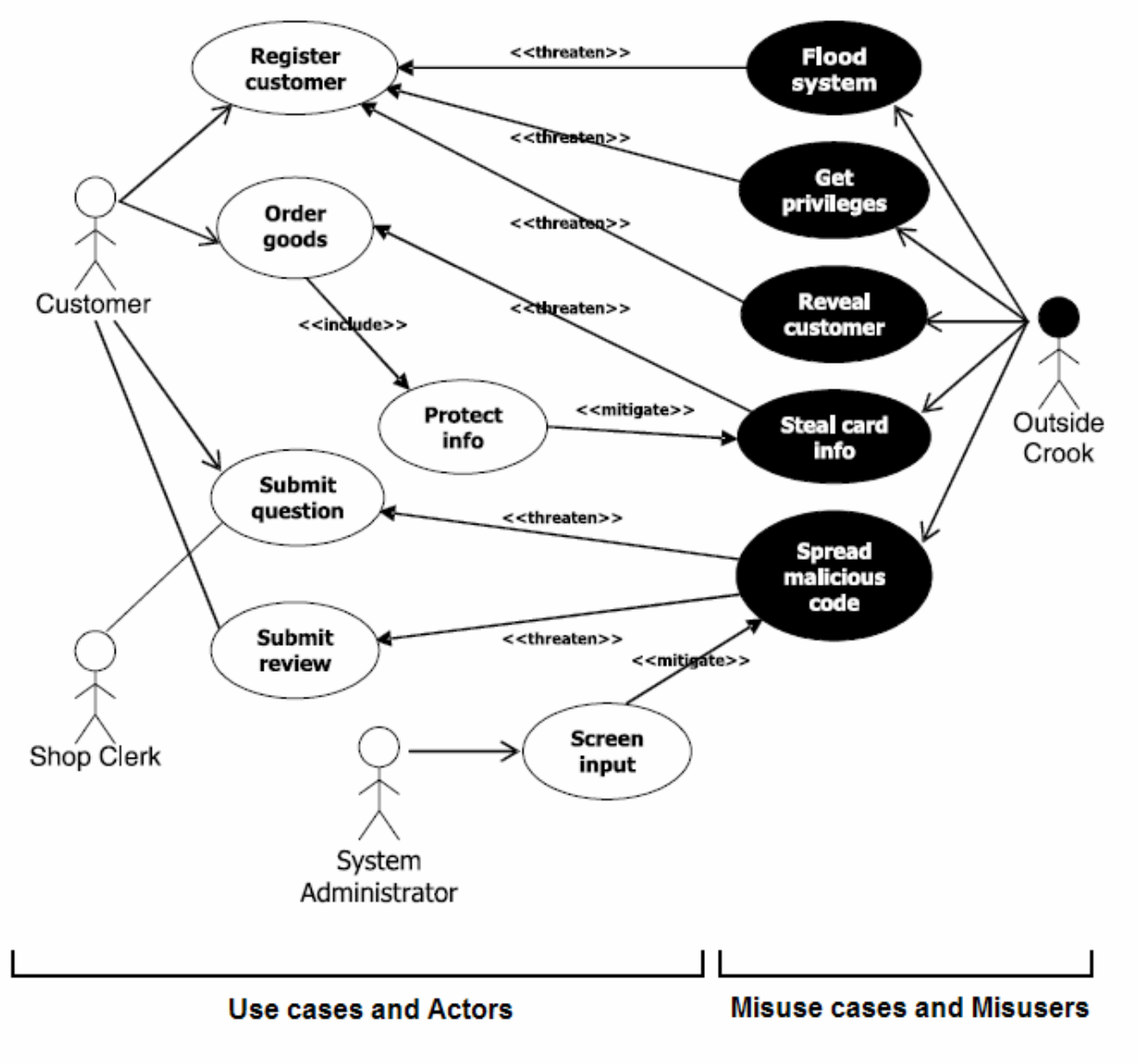

Figure 3. Example use and misuse cases for an electronic-store. Use cases and normal actors are displayed on the left, while misuse cases and misuses are displayed on the right hand side of the diagram.

The goal of the misuse case is to prevent a threat from occurring or to mitigate the impact of a threat if it does occur.

3 Comparison between Abuse cases and Misuse cases

This section contains a comparison between abuse cases and misuse cases. As abuse cases and misuse cases are always being developed on, it is complicated to compare every new improvement that has been added onto abuse cases or misuse cases. Hence the comparison is restricted down between the initial papers which introduced abuse cases [3] and misuse cases [1]. Also with the lack of systems built using either abuse cases or misuse cases, there are essentially no real world results available for comparison. In this section the focus are on some aspects which I found interesting - the definition, notation, process of creating an abuse/misuse diagram and the simplicity of abuse cases and misuse cases.

3.1 The definition

Abuse cases as seen in the previous section are defined as "… where the results of the interaction are harmful to the system …" whereas misuse cases are defined slightly differently as "behavior that the system/entity owner does not want to occur".

Consider the following scenario. An interaction results in a session key being revealed to an actor who should not see the session key.

According to the definition of an abuse case, this interaction is not an abuse case simply because no harm has been created. No actor has used the compromised key to reveal contents of a message or make unauthorized changes to stored data. Only when the actor posts the session key on a public website, then an abuse case takes place [3].

The definition of a misuse case however, refers to behavior. Even though no harm resulted, the fact that a session key was revealed to an actor who was not supposed to see it would result in a misuse case as it is an unwanted action.

3.2 Notation and level of detail

Both approaches refer to use cases; however abuse case diagrams are drawn separately from use cases. Misuse cases appear alongside use cases and there are associations between the misuse cases and the use cases. As abuse cases do not appear together with use cases, the author did not use different notation from use cases. Accordingly readers may get confused as to what they are looking at. If a reader did not read the words in Figure 2, it would be difficult to determine which diagram was which. Misuse cases employ a different color scheme to represent misuse cases and misusers while keeping the same symbols that a use case contains.

Abuse cases include a detailed description of their actors' resources, skills and objectives. The resources available to an actor include other persons, organizations, tools and systems that assist the actor and also the amount of time an actor has to devote to the abuse case. Skills are described in terms of the technical skills the actor has. Objectives are long-term goals that the actor potentially seeks over more than one abuse case. While Sindre did not mention much of misuse case descriptions in his original article [1], his more recent publication [7] discusses a lightweight misuse case description and an extensive misuse case description.

3.3 Illustrating misuse case diagrams and abuse case diagrams

Both [1] and [3] provide a list of steps which aid in constructing a misuse case diagram and an abuse case diagram respectively.

Both approaches start off by constructing a use case diagram for the scenario. In a misuse case diagram they introduce misuse cases and mis-actors while in an abuse case diagram the abuse cases and the actors for them are identified. It is shown that the first couple of steps in creating a misuse case diagram or an abuse case diagram are similar. However next in abuse cases, we have a step called "check granularity". This step can be described as checking whether the number of abuse cases in our diagram is appropriate. I.e. are there too few abuse cases or is there too many. This step is interesting as it serves to control the amount of complexity displayed in the diagram. However without having experience and not knowing which abuse cases to discard and which to include, this may be a hard step to complete. Misuse cases on the other

hand do not discuss the number of misuse cases that should appear in a misuse case diagram. It is imaginable that countless numbers of misuse cases could appear as creative people can think up many ways in which a system should not behave.

Table 1. Summary of steps used in illustrating a misuse case and an abuse case diagram

Misuse Cases	Abuse Cases
Construct a use case diagram	Construct a use case diagram
Introduce major mis-actors and misuse cases	Identify the actors for the abuse case Identify the abuse cases Define the abuse cases
	Check granularity
Investigate potential relations between misuse cases and use cases	
Introduce new use cases with the purpose to detect or prevent misuse cases	
Continue with a more detailed requirements documentation	Check completeness and minimality

The steps “investigate potential relations” and “introduce new use cases” are not applicable for use in abuse case diagrams as abuse case diagrams do not interplay the use cases and abuse cases together. Finally, the last step in a misuse case diagram specifies that we should continue with a more detailed requirements documentation. There are many considerations which cannot be described in a misuse case diagram like the motivation of a mis-actor, likelihood of various threats, cost of potential damage done. Meanwhile abuse cases check for completeness and minimality which reviews each abuse case’s description and checks whether each abuse case leads to harm and also checks whether a critical abuse case has been omitted.

3.4 Simplicity

The purpose of having abuse cases and misuses cases is to make eliciting security requirements easier to accomplish and understand. Traditional mathematical security models are described as being hard to understand [3]. Hence we shall look at the simplicity of abuse cases and misuse cases.

While abuse case diagrams (Figure 2) are easy to understand and create, misuse cases do make things a bit more complex by adding associations between misuse cases and use cases. There is a tradeoff between simplicity and the level of detail captured within the diagrams. While misuse cases are more complex than abuse cases, it adds interesting interactions between use cases and misuse cases which could lead it to be more useful in eliciting security requirements shown by new use cases that can mitigate threats.

Also as previously stated, while drawing up abuse case diagrams, we check the granularity of the diagram. It is possible here to reduce the number of abuse cases and remove similar abuse cases. This would lower the complexity of our finished abuse case diagram making it more simple.

4 Discussion

There are some similarities with abuse cases and misuse cases in that they share the same UML notation. However there are also differences between abuse cases and misuse cases. It was interesting to see that abuse cases do not model and omit cases where no harm has been done. Recall the scenario, where a session key was revealed to an actor who does nothing with it. That scenario is not modeled as an abuse case because there is no harm. But if that actor posts the session key on a website, then it becomes an abuse case [3]. But what if nobody visits the website, or suppose people do visit the website but they do nothing with the session key resulting in no harm. It is not clear whether this would still be an abuse case. Similarly in cases where an actor tries to cause harm but fails in doing so; this is also omitted from the abuse case diagram as no harm has resulted [3]. Perhaps this is done to reduce the number of abuse cases in the diagram, but it would be imprudent to not model a case just because it causes no harm as there is still someone trying to harm the system.

John McDermott [3] did not mention any differences between the normal use case actors and his abuse case actors. In misuse case diagrams, misusers are not limited to hostile actors, but even ‘bad luck’ and the ‘devil’ can be modeled [1]. This allows the diagram to model cases where security threats arise from unexpected equipment failure and sudden operator illness.

Misuse cases are illustrated together with the use cases and the associations between them are shown as “mitigates” or “threatens”. Putting the misuse cases and use cases in the same diagram can produce interesting outcomes. Ian Alexander [8, 9] proposes a way of turning security requirements elicitation into a game like chess where “a team’s best strategy consists of thinking ahead to the other team’s best move and acting to block it”. Abuse cases, because they are illustrated separately from use cases, have no association between them and hence are unable to explore what relationships occur between the two.

Both misuse cases and abuse cases allow the possibility of unlimited misuse cases and abuse cases; but they both do not discuss in great detail when and how one ought to stop adding misuse/abuse cases.

5 Conclusion

Abuse cases and misuse cases were mainly introduced because traditional mathematical security models were hard to understand. While abuse cases and misuse cases do not claim to be a substitute for mathematical security models, they do allow those who are unfamiliar with mathematical security models to more easily understand and elicit security requirements.

In a number of published papers, abuse cases and misuse cases were thought to be the same as they both employed the existing UML notation for use cases and had the same purpose to elicit security requirements. This paper exposes some differences between the two. Compared to actors in an abuse case, misusers in a misuse case are able to model a wider range of objects which are not limited only to human actors. Abuse case diagrams are drawn separate from use case diagrams whereas misuse cases include misuse cases and use cases in the same diagram. Both do not set out a method for deciding when to stop adding new abuse/misuse cases or determining which abuse/misuse case we would want to keep. Consequently increasing the number of abuse/misuse cases severely increases the complexity of the diagram. However as neither abuse cases nor misuse cases have been evaluated in practical software

development projects, it is unwise to say which approach is better without comparing the outcomes of real projects.

However in recent years, it does appear that more papers on misuse cases have been published compared to those relating to abuse cases. This may lead to more widening of the gap between abuse cases and misuse cases.

References

[1] Sindre G, Opdahl AL (2000) *Eliciting security requirements by misuse cases*. In: Proceedings of TOOLS Pacific 2000, Sydney, Australia

[2] Object Management Group (1999) *OMG Unified Modeling Language Specification*, version 1.3.

[3] McDermott J, Fox C (1999) *Using abuse case models for security requirements analysis*. In: Proceedings of the 15th annual computer security applications conference (ACSAC'99), Phoenix, Arizona

[4] Crook R et al (2002) *Security requirements engineering: when anti-requirements hit the fan*. In: Proceedings of the 10th anniversary IEEE international requirements engineering conference (RE'02), Essen, Germany

[5] Sindre G, Firesmith D, Opdahl AL (2003) *A reuse-based approach to determining security requirements*. In: Proceedings of the 9th international workshop on requirements engineering: foundation for software quality (REFSQ'03), Klagenfurt, Austria

[6] Mead, Nancy R (2003) *Requirements Engineering for Survivable Systems* (CMU/SEI-2003-TN-013). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

[7] Sindre G, Opdahl AL (2005) *Eliciting security requirements with misuse cases*, Requirements Engineering, Volume 10, Issue 1, Jan 2005, Pages 34 - 44

[8] Alexander IF (2002) *Initial industrial experience of misuse cases in trade-off analysis*. In: Proceedings of the 10th anniversary IEEE international requirements engineering conference (RE'02), Essen, Germany

[9] Alexander IF (2003) *Misuse cases, use cases with hostile intent*. IEEE Software 20(1):58–66