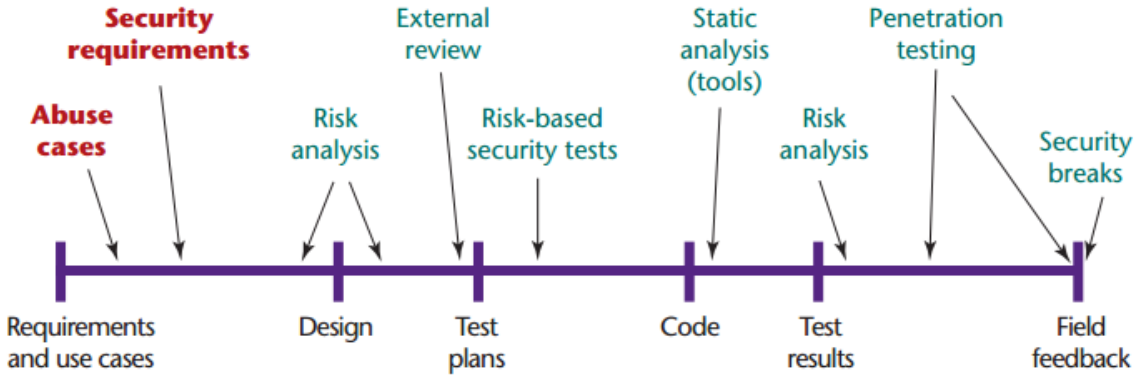
SE-4930: Developing Secure Software

Lab 3: Eliciting Misuse Cases

Report due: October 1, 2014 23:59 (One per team)

Introduction



Referring to Gary McGraw's secure software touchpoints, we can see that one of the first things we assess are the abuse cases and security requirements. With misuse cases, the goal is to decide and document a priori how a software system should act and respond to illegitimate usage by a potential adversary. This documentation then aids in the design and development of the system.

In a traditional system, we have UML actors and UML use cases created to define the normative behavior for the system. If we are interested in security, the model is modified slightly to include additional constructs. Misusers represent the adversaries who are intent on attacking the system. Misusers are actors that initiate a misuse case, either intentionally or inadvertently. Misuse cases represent threats to valid use cases. A misuse case is a sequence of actions, including variants, that a system or other entity can perform which will harm the system if allowed to complete. Security use cases may exist to mitigate threats. This is shown in the Figure 1.

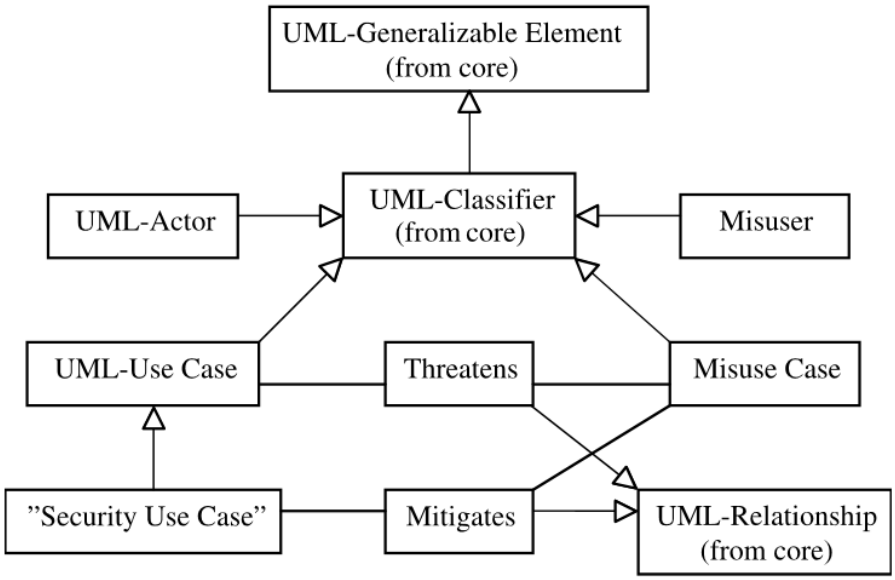


Figure 1: A metamodel for misuse cases.[1]

Misuse case diagram

To document misuse cases, we typically start with a use case diagram and append additional; information. Figure 2 shows a sample system which is responsible for storing grades for a professor. To start, the system has two use cases, record grades and print grade report. This is the use case shown on the left. As the use cases are expanded to include misuse cases, we see that there are two misusers in the system, shown in black on the right. These misusers attempt to exploit two misuse cases in the system. However, these misuse cases are mitigated by an additional use case. Note that by convention misuse cases have a stereotype applied to them and they are colored in black. The same applies for the abusers. Typically these are located on the right of the diagram, but not always.

[1] Guttorm Sindre and Andreas L. Opdahl. 2005. Eliciting security requirements with misuse cases.*Requir. Eng.* 10, 1 (January 2005), 34-44. DOI=10.1007/s00766-004-0194-4 http://dx.doi.org/10.1007/s00766-004-0194-4

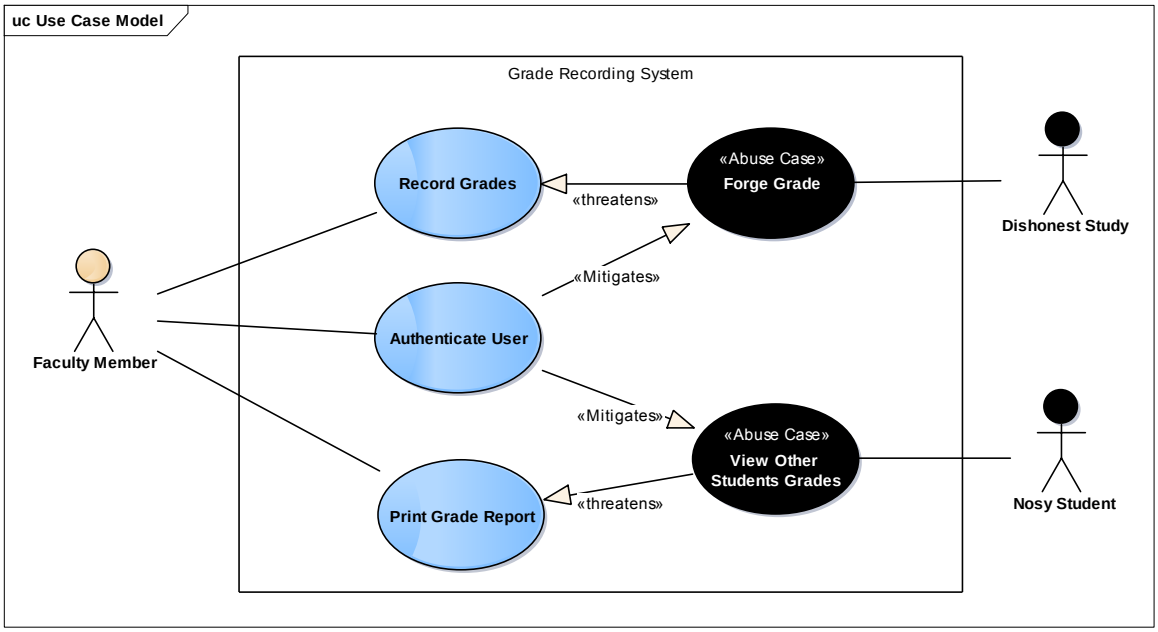


Figure 2: A sample misuse diagram for a grade recording system

Textual Specification for Misuse Cases

While the use case diagram is useful, it is important that the use case be specified in a textual format as well. This matches with the textual specifications for use cases and use case scenarios that you have seen in previous courses. As with use cases, we typically use templates to represent misuse cases as well.

The simplest approach uses additioanly information embedded within the use case template to capture descriptions of the misuse. This involves the additional of a field referred to as "Threats". Figure 3 shows a lightweight misuse case description embedded within a use case description. Note that the threats are defined in a separate section. If the use case behavior uses a user and system column to describe the behavior, a third column can be added to represent the threats, as is shown in Figure 4.

Name:	Register customer
Iteration:	Filled
Summary:	The customer registers for the e-shop, giving name, address, email, and phone
Basic path:	bp–1. The customer selects to register bp–2. The system provides the registration form bp–3. The customer completes the form and submits bp–4. The system acknowledges registration, returning a customer reference number
Alternative paths:	[...]
Exception paths:	**E1.** In action 3, the customer submits with mandatory information missing. Return to action 3 to provide more info **E2.** In action 3, the submitted info matches an already registered customer. The system notifies the user that registration is abandoned because the customer is already registered. This ends the use cases
Extension points:	[...]
Triggers:	[...]
Assumptions:	[...]
Preconditions:	[...]
Postconditions:	The customer is now registered, and will be enabled to order goods from the e-shop without providing contact info anew
Related business rules:	[...]
Threats:	**T1:** The customer is not registering with his own name and address, but with an assumed identity. Possible outcomes: T1–1. A non-existing person is registered as customer T1–2. An existing person is unwillingly and unknowingly registered as a customer T1–3. It is revealed to a third party that the named person is a customer of the e-shop (see exception path **E2** above)**T2:** [...]
Author:	John Davis
Date:	2001.05.23

Figure 3: A sample textual misuse case.[2]

gettingCash		
User intention	System response	Threats
Identify self		Identity spoofedIdentification spied on
	Verify identity Offer choices	ATM tampered with
Choose		
	Dispense cash	
Take cash		Customer is robbed

Figure 4: A sample three columns representation for the misuse case.[3]

Defining Security Requirements

Defining security requirements, based on the concepts from Sindre and Opdahl, involves a 5 step process. In the first step, the critical assets of the system are defined. In the second step, security goals for each asset are defined. Threats are then identified by identifying the stakeholders that may harm the system as well as how they may harm the system. This is where misuse case modeling occurs. In the fourth step, the risks are identified and analyzed using risk

[2] Guttorm Sindre and Andreas L. Opdahl. 2005. Eliciting security requirements with misuse cases.*Requir. Eng.* 10, 1 (January 2005), 34-44. DOI=10.1007/s00766-004-0194-4 http://dx.doi.org/10.1007/s00766-004-0194-4

[3] Guttorm Sindre and Andreas L. Opdahl. 2005. Eliciting security requirements with misuse cases.*Requir. Eng.* 10, 1 (January 2005), 34-44. DOI=10.1007/s00766-004-0194-4 http://dx.doi.org/10.1007/s00766-004-0194-4

analysis and costing. Lastly, the requirements for the problem are defined. In this step, typically use cases which represent mitigations to misuse cases are derived for the system. As with all software development, the five step process tends to be highly cyclical when a software system is developed.

Lab Specifics

In lab, you are going to work with your capstone senior design team (and / or an adopted team) to analyze the misuse cases for your system. At this point in time, you should have a good understanding of the use cases for your system, as well as have initial models constructed for the system. In this exercise, you will refine these models to reflect on the adversaries for your senior design project.

You should start by first identifying the assets in your system. And defining them. Once this is done, you'll need to identify the security goals for your system. When identifying the threats, you need to consider who might attack your system and what might they be trying to accomplish. The use case diagram that you have for your senior design project (or may need to create) will help with this activity. As you identify the risks and analyze them, figure out which ones are most concerning and determine a mitigation strategy for them.

Deliverables

Report

Each team is to submit a simple report in pdf format. The report should contain

1. The names of all team members, the date, and the assignment
2. A brief, one paragraph summary of the lab and what was learned
3. Definition of project assets
    a. Tabular format is fine, explaining the asset, what type it is, and its data classification (Confidentiality, Integrity, and Availability) as was discussed in class, and the goals for its protection.
4. Misuse case diagram
    a. The diagram should show the use cases for the system as well as abuse cases for the system.
    b. Initial mitigations for the misuse cases should be provided.
5. Misuse case textual definitions
    a. For each use case in the system, include a textual definition describing the threats to the system. Misuse cases may be expressed in either format provided above. (Note: This material generally should be in some form of a draft format for your capstone project.)
6. Things gone right and things gone wrong. Provide a brief, one paragraph summary, of the things that went right with this exercise and the things that went wrong with this exercise.