Misuse and Abuse Cases

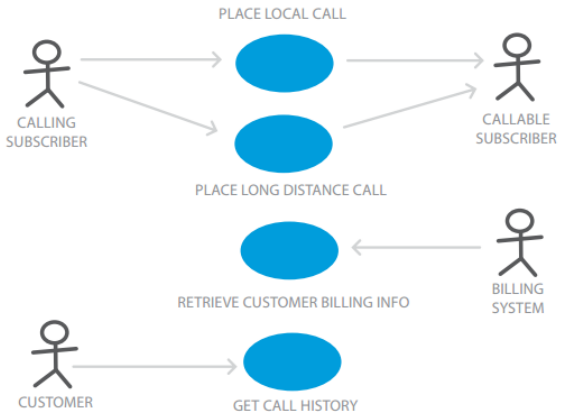
Laurie Williams

Slides enhanced by John Slankas

Computer Science NC STATE UNIVERSITY

Use Case

- The desired functionality of a system
- Intentional omission – what the system does not do



http://www.ivarjacobson.com/Use_Case2.0_ebook/

Computer Science NC STATE UNIVERSITY

Use Case Template

Use Case ID:	Enter a unique numeric identifier for the Use Case. e.g. UC-1.2.1
Use Case Name:	Enter a short name for the Use Case using an active verb phrase. e.g. Withdraw Cash
Created By:	Last Updated By:
Date Created:	Last Revision Date:
Actors:	[An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case (primary) and any other actors who will participate in completing the use case (secondary).]
Description:	[Provide a brief description of the reason for and outcome of this use case.]
Trigger:	[Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.]
Preconditions:	[List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each pre-condition. e.g. 1. Customer has active deposit account with ATM privileges 2. Customer has an activated ATM card.]
Postconditions:	[Describe the state of the system at the conclusion of the use case execution. Should include both minimal guarantees (what must happen even if the actor's goal is not achieved) and the success guarantees (what happens when the actor's goal is achieved. Number each post-condition. e.g. 1. Customer receives cash 2. Customer account balance is reduced by the amount of the withdrawal and transaction fees]
Normal Flow:	[Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. 1. Customer inserts ATM card 2. Customer enters PIN 3. System prompts customer to enter language performance English or Spanish 4. ….

http://austin.iiba.org/download/UseCase.dot

Use Case Template (continued)

Alternative Flows: [Alternative Flow 1 – Not in Network]	[Document legitimate branches from the main flow to handle special conditions (also known as extensions). For each alternative flow reference the branching step number of the normal flow and the condition which must be true in order for this extension to be executed. e.g. Alternative flows in the Withdraw Cash transaction:
Exceptions:	[Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. e.g. Exceptions to the Withdraw Case transaction 2a. In step 2 of the normal flow, if the customer enters and invalid PIN 1. Transaction is disapproved 2. Message to customer to re-enter PIN 3. Customer enters correct PIN 4. Use Case resumes on step 3 of normal flow]
Includes:	[List any other use cases that are included ("called") by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and "included" in all ATM Use Cases.]
Frequency of Use:	[How often will this Use Case be executed. This information is primarily useful for designers. e.g. enter values such as 50 per hour, 200 per day, once a week, once a year, on demand etc.]
Special Requirements:	[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]
Assumptions:	[List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description. e.g. For the Withdraw Cash Use Case, an assumption could be: The Bank Customer understands either English or Spanish language.]
Notes and Issues:	[List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved.

http://austin.iiba.org/download/UseCase.dot

Agile Alternative: User Story

- Agile practice
- Captures design of a software feature from an end-user perspective

As a <*role/type of user*>,

I want <*some goal*>

so that <*some reason/benefit*>

Computer Science
NC STATE UNIVERSITY

Exercise Part 1:

- Develop a use case diagram for an electronic voting system
  - 2+ actors
  - 4+ use cases

Computer Science
NC STATE UNIVERSITY

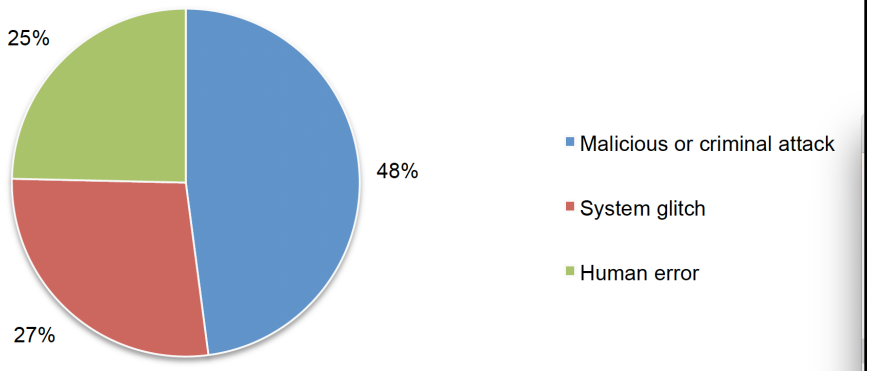
Misuse Case & Abuse Case

- A misuse case is "special kind of use case, describing behavior that the system/entity owner does not want to occur" [may not be intentional] Sindre & Opdahl
- An abuse case as a specification of a type of complete interaction between a system and one or more actors, where the results of the interaction are [intentionally] harmful to the system, one of the actors, or one of the stakeholders in the system. McDermott & Fox

Computer Science NC STATE UNIVERSITY

Root cause of breach: Misuses

Pie Chart 2. Distribution of the benchmark sample by root cause of the data breach
Consolidated view (n=383)



Poneman, 2016 Cost of Data Breach

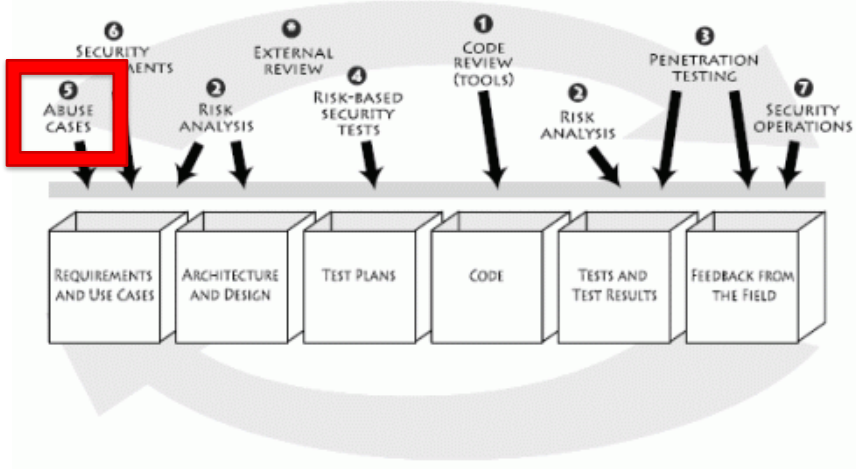
Computer Science NC STATE UNIVERSITY

Misuses

- Unintentional misuses are common
  - 44% of healthcare breaches
  - 80% of success DoD compromises
- Consider what legitimate users may do by accident … **can you protect them from themselves**?

Computer Science
NC STATE UNIVERSITY

Software Security Touchpoints



http://www.cigital.com/justiceleague/wp-content/uploads/2007/07/touchpoints.gif

Computer Science
NC STATE UNIVERSITY

Abuse cases … inverse of a use case

- Use case …. a function the system SHOULD do
- Abuse case ... a function the system SHOULD NOT allow an attacker to do

- For every new requirement, feature, or use case, someone should spend some time thinking about how that feature might be unintentionally misused or intentionally abused.

Computer Science
NC STATE UNIVERSITY

Abuse cases

- Tool for helping to think about your software the same way attackers do
- Pretend you are a bad guy … get into character
  - Ask yourself "What do I want?"
    - "I want to steal all the money."
    - "I want to learn the secret ways of the execs."
    - "I want to spy on my spouse."
- Goal of abuse case is to decide and document a priori how the software should react to illegitimate use

Computer Science
NC STATE UNIVERSITY

Abuse case

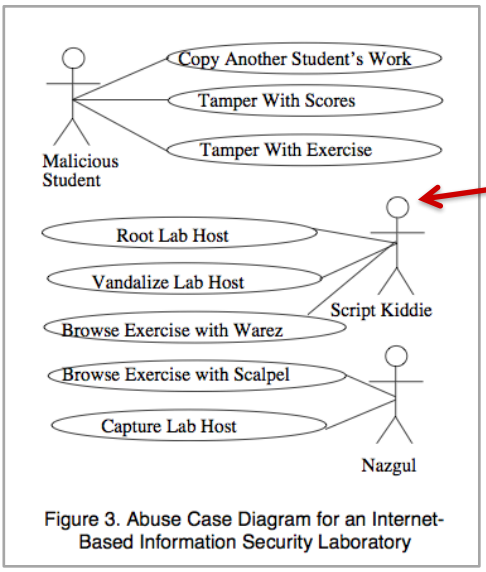


Figure 3. Abuse Case Diagram for an Internet-Based Information Security Laboratory

http://www.acsac.org/1999/papers/wed-b-1030-john.pdf

Computer Science NC STATE UNIVERSITY

UML Stereotypes

Use cases can mitigate abuse cases. The use case can be a countermeasure against an abuse case, i.e., the use case reduces the abuse case's chance of succeeding.

Example: "Steal the car", is mitigated by "lock the car" .

Abuse case threaten use case. The use case is exploited or hindered by an abuse case.

Example: "drive the car" is threatened by "steal the car"

Computer Science NC STATE UNIVERSITY

*Adapted from wwwagse.informatik.uni-kl.de/.../13-Talk-**Security**%20**Requirements**.ppt*

Abuse case

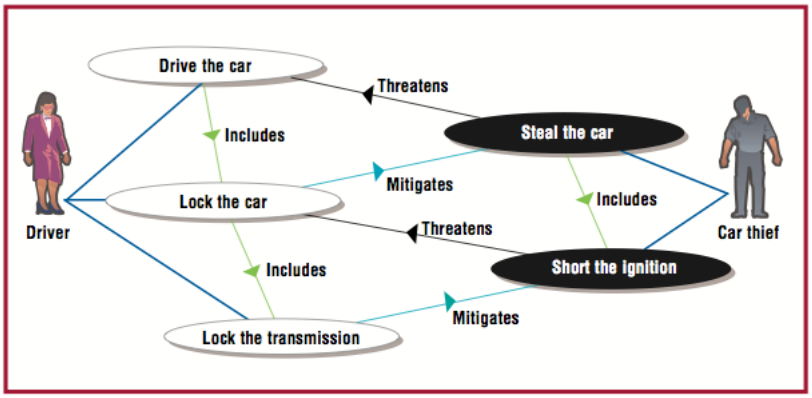


Figure 1. Use/misuse-case diagram of car security requirements. Use-case elements appear on the left; the misuse cases are on the right.

I. Alexander, *Misuse Cases: Use Cases with Hostile Intent*, IEEE Software, Jan/Feb 2003.

Computer Science
NC STATE UNIVERSITY

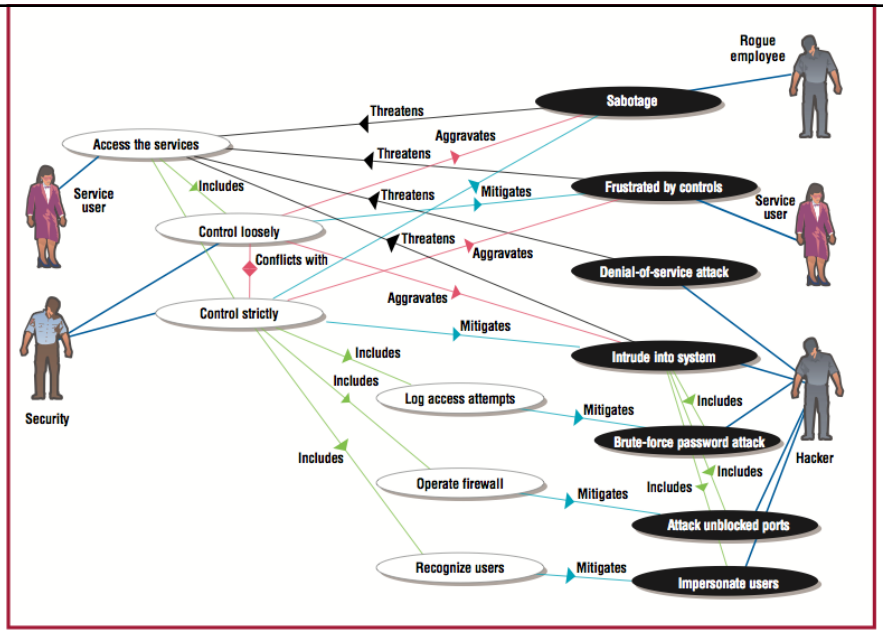
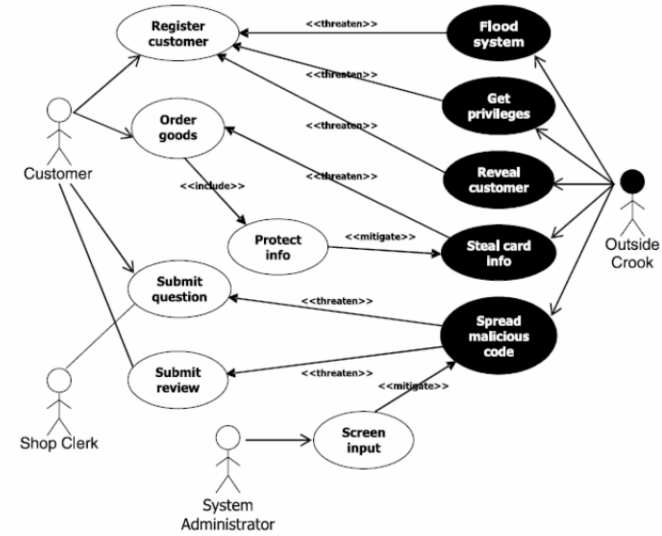


Figure 4. Use and misuse cases for Web portal security.

I. Alexander, *Misuse Cases: Use Cases with Hostile Intent*, IEEE Software, Jan/Feb 2003.

Computer Science
NC STATE UNIVERSITY



www.cs.auckland.ac.nz/compsci725s2c/archive/termpapers/csia.pdf

Computer Science NC STATE UNIVERSITY

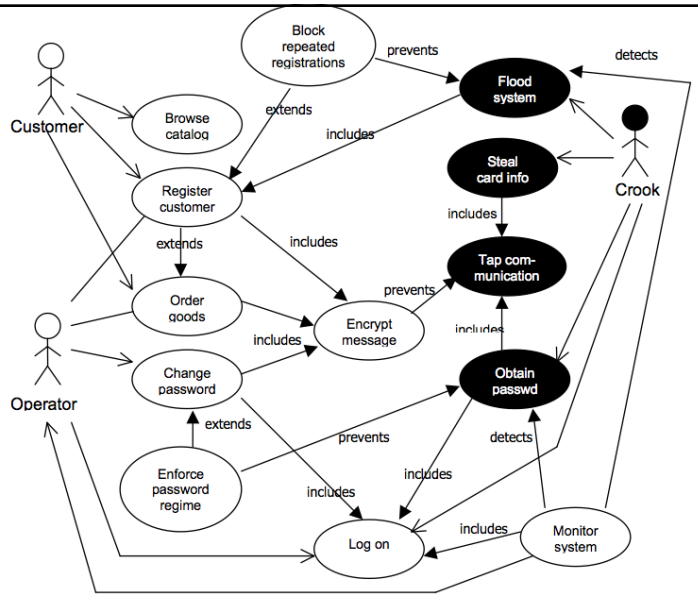


Figure 1: Misuse depicted as "inverted" use cases.

http://folk.uio.no/nik/2001/21-sindre.pdf

Computer Science NC STATE UNIVERSITY

Name: Obtain Password
Summary: A crook obtains and later misuses operator passwords for the e-shop by tapping messages sent through a compromised network host during operator log on.
Author: David Jones
Date: 2001.02.23.

Basic path:
bp0 A crook has hacked a network host computer and installed an IP packet sniffer (step bp0-1.) All sequences of messages sent through the compromised host and which contain strings like 'Logon', 'User name', 'Password', 'passwd' etc. are intercepted and analysed further (step bp0-2 and extension point e1.) In this way, the crook collects (likely) user names and passwords along with the IP addresses of the computers they are valid on (step bp0-3.) The crook - possibly much later - uses the user name and password to gain illegal operator access to the e-shop computer (step bp0-4.)

Alternative paths:
ap1 The crook has operator privileges on the network host. No hacking of the network computer is necessary (changes step bp0-1.)
ap2 The crook has not penetrated a network host, but instead intercepts messages sent through the telephone system from the e-shop operator's home (changes step bp0-1.)
ap3 Instead of home phone, the crook intercepts messages sent from the e-shop operator's portable devices (changes step bp0-1.)

Capture points:
cp1 The password does not work because it has been changed (in step bp0-4.)
cp2 The password does not work because it is time dependent (in step bp0-4.)
cp3 The password does not work because it is different for different IP addresses (in step bp0-4.)
cp4 Operator logon to the e-shop is only possible from certain IP addresses (in step bp0-4.)
cp5 Communication tapping (in step bp0-2) is not possible (perhaps because the communication is encrypted.)

Extension points:
ep1 Includes misuse case "Tap communication" (in step bp0-2.)

Table 1: Detailed misuse case description, part 1.

http://swt.cs.tu-berlin.de/lehre/saswt/ws0506/unterlagen/TemplatesforMisuseCaseDescription.pdf

Preconditions:
pc1 The system has a special user 'operator' with extended authorities.
pc2 The system allows the operator to log on over the network.

Assumptions:
as1 The operator uses the network to log on to the system as operator (for all paths.)
as2 The operator uses his home phone line to log on to the system as operator (for ap2.)
as3 The operator uses his home phone line to log on to the system as operator (for ap3.)

Worst case threat (postcondition):
wc1 The crook has operator authorities on the e-shop system for an unlimited time, i.e., she is never caught.

Capture guarantee (postcondition):
cg1 The crook never gets operator authorities on the e-shop system.

Related business rules:
br1 The role of e-shop system operator shall give full privileges on the e-shop system, the e-shop system computer and the associated local network host computers.
br2 Only the role of e-shop system operator shall give the privileges mentioned in br1.

Potential misuser profile: Highly skilled, potentially host administrator with criminal intent.

Stakeholders and threats:
sh1 e-shop
- reduced turnover if misuser uses operator access to sabotage system
- lost confidence if security problems get publicized (which may also be the misuser's intent)

sh2 customer
- loss of privacy if misuser uses operator access to find out about customer's shopping habits
- potential economic loss if misuser uses operator access to find credit card numbers

Scope: Entire business and business environment.
Abstraction level: Mis-user goal.
Precision level: Focussed.

Table 2: Detailed misuse case description, part 2.

http://folk.uio.no/nik/2001/21-sindre.pdf

Exercise Part 2:

- Extend your use case into a misuse and abuse case diagram
- Choose one misuse or abuse case and create detailed description

Computer Science
NC STATE UNIVERSITY

References

1. G. Sindre and A. L. Opdahl, "Templates for Misuse Case Description," 7th International Workshop on Requirements Engineering: Foundation for Software Quality, Interlaken, Switzerland, 2001.
2. John McDermott and Chris Fox (Dec 1999). "Using Abuse Case Models for Security Requirements Analysis". *Proceedings of the 15th Annual Computer Security Applications Conference, 1999. (ACSAC '99)*: 55–64.

Computer Science
NC STATE UNIVERSITY