Guideline for applying CHASSIS, draft

This document describes a guideline for applying the CHASSIS method for harm assessment, by combining safety and security assessments. The guideline is based on work presented in [1].

The main focus of the guideline is on Misuse Case (MUC), Misuse Sequence Diagram (MUSD) and Failure Sequence Diagram (FSD). For MUC both Textual MUC (T-MUC) and Diagrammatical MUC (D-MUC) are described, and how these two techniques relate to FSD and MUSD. For more details on MUC we refer to [2].

There are different approaches towards developing legacy systems and systems that are made from scratch. The main difference are that legacy systems inherit a design and architecture, as well as knowledge about how they operate and the environment are operated in, compared to systems developed from scratch. CHASSIS belongs to requirements activities, but there will be iteration with design activities. Decomposition of FSD/MUSD can be an example of such iteration.

Table of contents

Table of figures

Table of tables

Abbreviation

Abbreviation	Description
CHASSIS	Combined Harm Assessment of Safety and Security for Information Systems
D-MUC	Diagrammatical misuse case
D-UC	Diagrammatical use case
FHA	Functional Hazard Assessment
FSD	Failure sequence diagram
HAZOP	Hazard and Operability study
MUSD	Misuse sequence diagram
SAM	Safety Assessment Methodology
SecRAM	Security Risk Assessment Methodology
T-MUC	Textual misuse case
T-UC	Textual use case

Process overview

The CHASSIS process is shown in figure 1 below. The process is a part of the overall requirements engineering process, but focuses on safety and security requirements engineering. It consists of three main activities: (1) Eliciting Functional Requirements, (2) Eliciting Safety/Security Requirements and (3) Specifying Safety/Security Requirements. If activity (1) already has been undertaken in a project, the artifacts produced during this activity might have to be considered and adapted before used in (2).

The scope of CHASSIS is to support the elicitation of safety and security requirements, based on and thightly related to functional requirements. Safety and security aspects that should be elicited with CHASSIS are:

- Safety
  - hazards,
  - failures and
  - their mitigations for safety,
- Security
  - threats,
  - vulnerabilities and
  - their mitigations for security

However, other informations related to those aspects will also be elicted, in particular by T-MUC and the HAZOP table.

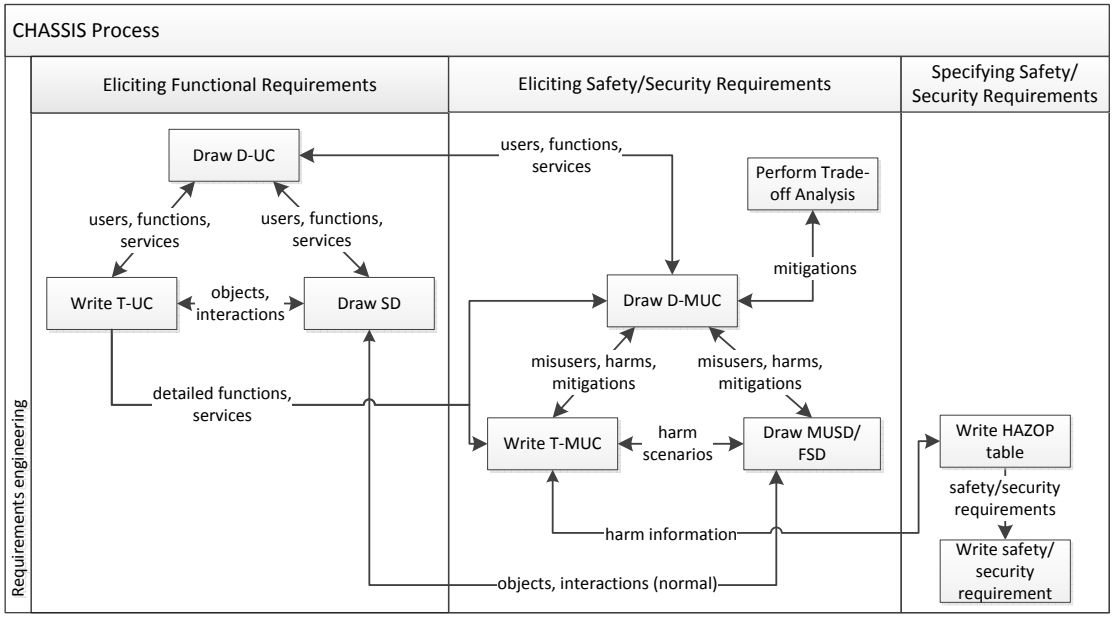

Figure 1 - an overview of the process of applying the CHASSIS method.

Use cases

For developing use cases we recommend this to be done according to the descriptions found in the book titled “Writing Effective Use Cases” by Alistar Cockburn from 2001, [3].

Diagrams
See above.

Template
See above.

Sequence diagrams
For developing sequence diagrams we recommend this to be done according to the descriptions found in the book titled "UML Distilled third edition – A brief guide to the standard object modelling lanuage" by Martin Fowler from 2004, [4].

Diagram
See above.

How UC and SD relate
Use cases in D-UC can be further described using T-UCs, and therefore a D-UC might refer to one or many T-UCs. An actor or user is an external element to the system, e.g., human (operator, maintenance personnel, etc.), another system communicating with the system under assessment, or external hardware parts (sensors, acutors, etc.) .The actors of a T-UC are present in the D-UC but not all actors in a D-UC are necessarily present in a T-UC. SD and T-UC are in a tight connection but an SD might detail further an artificial actor like a system or syb-system and thus have more actors than a T-UC. SDs can detail a specific UC described in a T-UC by placing it in a concrete situation and context. T-UC can be updated based on the SD coming from the interaction analysis.

A. Misuse case – safety requirements elicitation
This section provides guidance of how to organize the identification of hazards, faulty components/systems and possible mitigations in a structured and systematic manner by use of diagrammatical and textual misuse cases.

Identifying initial threats, misusers and mitigations with MUC
Input: Use case diagrams and textual descriptions
Participants: System, domain and safety experts,
Organization: Meeting with facilitator, secretary and participants. Using whiteboard and projector

A.1. Preparations
Build up understanding of the function by all participants:

1. Go through D-UC to get an overview of system functions and actors involved
2. Read through T-UC in order to understand the details of a particular system function and the actors involved.
3. Go through related SDs if available and needed. It might happen in parallel with step 2.
4. Ask for further explanation by domain expert (if needed)

A.2. Structured/guided brainstorming activity for hazards using D-MUCs

Organized as a structured meeting with system, domain and safety (security can join in for parts) experts.

A.2.1 Applying HAZOP guidewords for identifying hazards to functions

1. Apply one guideword to the name of use case (either in front, middle or after name) resulting in a guide phrase
2. Brainstorm for misuse of the function based on the guide phrase. This step might result in more than one misuse case per guide phrase.

A.2.2 Identify misuser

1. Brainstorm and discuss possible misuser (faulty system-user/components/systems) for the identified hazard in a structured way
    a. Consider all human users of the system (e.g. different operators, maintenance personnel)
    b. Consider all the external systems which the system in dependent on (e.g. other computer systems, electricity system, networks, sensors)
    c. Consider the internal parts of the system which might fail
    d. Consider known misusers from earlier sources available (e.g. list from Eurocontrol SAM guideline)
2. Draw identified misusers in the diagram (ref. step *Draw D-MUC* below) and possibly write them down in more detail in the T-MUC (ref. step *Write T-MUC* below), as they usually are discussed during this activity in more detail, e.g. assumptions are made and the misusers are characterized.

A.2.3 Identify and evaluate mitigations

1. Discuss possible mitigations to the individual misuse cases
2. Consider different mitigations against both misuse case and misuser

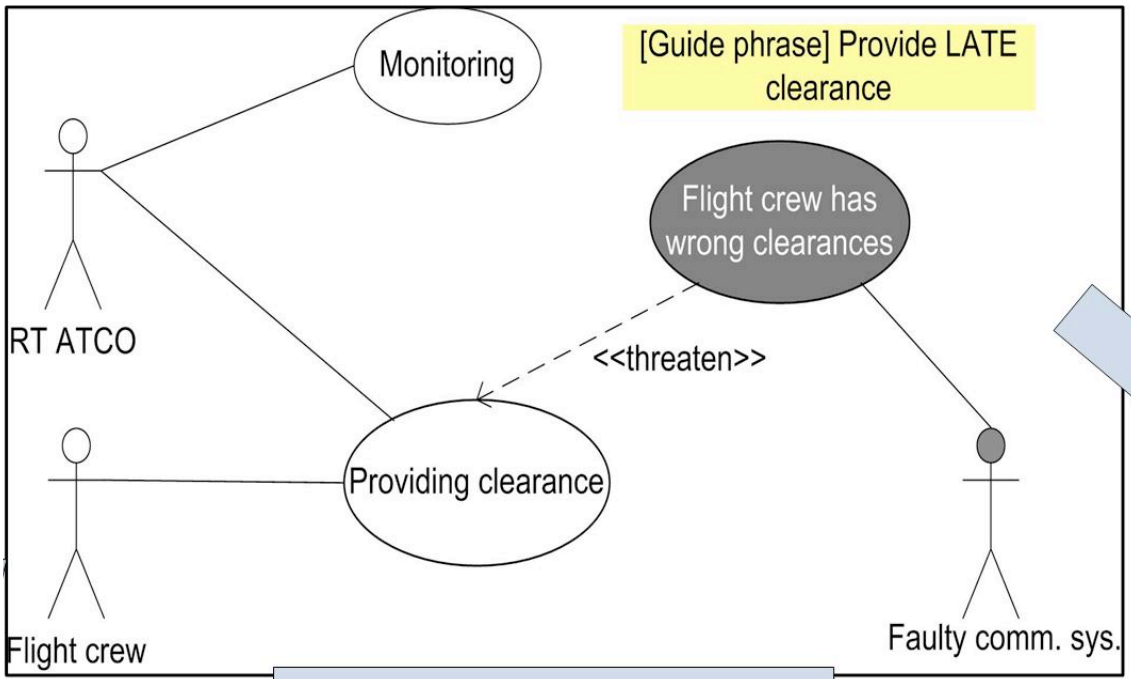

Figure 2 - Example of a D-MUC for safety [1]

A.3. How to create D-MUC during the brainstorming session?

Use an existing D-UC or create a new D-UC of system under analysis

A3.1. Draw D-MUC

1. Add the combined *guideword* and *use case name* as a *guide phrase* in *note* in D-MUC.
2. Draw a *black oval* as *misuse case* right to the *use case* under assessment and write a *phrase* in the *misuse case* with a few words (short and descriptive).
3. Draw an *arrow* from the *misuse case* to the *use case* and name it *threatens*.
4. Draw a *misactor* outside the *system boundary* and relate her/him to *misuse case* by a *line*.
5. Draw a (*safety) use case* below the *misuse case* it mitigates and specify the *security function* as *name*.
6. Draw an *arrow* from the (*safety) use case* to the *misuse case* and *name* it *mitigates*.

4. Create T-MUC

Secretary can start writing some parts of template while diagrams are made, but basic and alternative path should be written or at least throughly checked and agreed upon by participants.

4.1 Write T-MUC

Table A - Example of T-MUC for safety based on [2]

Fields name	**Explanation**
Name, summary, author and date	These fields retain the same meaning as in regular use cases
Basic path	This field describes the actions that the misuser(s) and the system go through to harm the proposed system
Alternative path	This field describes ways to harm the proposed system that are not accounted for by the basic path, but are still sufficiently similar to be described as variants of the basic path
Mitigation points	This field identifies those actions in a basic or alternative path where misuse can be mitigated. Several ways to mitigate misuse of a particular action can be described in the same field and each of them may be further described in a separate safety use case. As for extension points, the misuse case must eventually have a mitigate relationship to a corresponding safety use case. However, the detailed description of safety use cases is optional, because it is often closer to design, requiring detailed analysis of risks and implementation costs that go beyond use and misuse cases
Extension points	In some cases, a misuse case may be extended with optional paths whose details are described in a separate extension misuse case. This field lists the actions in the main or alternative paths where optional paths may be inserted. As for extension points in regular use cases, the misuse case must have an extend relationship to the misuse case that

	contains the optional path
Trigger	This field describes the states or events in the system or its environment that may initiate the misuse case. For some misuse cases, the trigger is just the predicate True, indicating a permanently present danger
Assumptions	This field describes the states in the system's environment that make the misuse case possible
Preconditions	This field describes the system states that make the misuse case possible
Mitigation guarantee	This field describes the guaranteed outcome of mitigating a misuse case. If the mitigation points are not yet specified in detail, the mitigation guarantee describes the level of security required from the mitigating safety use cases that will be designed later. When the mitigation points in the misuse case have been detailed by safety use cases, this field describes the strongest possible safety guarantee that can be made, regardless of how the misuse case is mitigated
Related business rules	Typically, business rules will be violated by the misuse. This field contains links to such rules, maybe along with links to rules that enable the hazard or that limit how it could be mitigated or eliminated
Misactor profile	This field describes whatever can be assumed about the misactor, for example, whether the misactor acts inadvertently; whether the misuser is an insider or outsider; and how technically skilled the misuser must be
Scope	This field indicates whether the proposed system in a misuse case is, e.g., an entire business, a system of both users and computers, or just a software system
Iteration	As for regular use cases, it is useful to allow both initial and detailed descriptions of misuse cases. This field indicates the misuse case's iteration level, usually taken from the set of iteration levels used for the use cases in the project
Level	As for regular use cases, misuse cases can be specified at a general or specific abstraction level. This field indicates whether the misuse case is, e.g., a summary, a user goal, or a sub-function, following [3]
Stakeholders and risks	This field specifies the major risks for each stakeholder involved in the misuse case. On an abstract level, risks can be described textually, e.g., ''the system is unavailable for several hours'' or ''a competitor gets hold of sensitive medical data about an applicant''. On a concrete level, the likelihood and cost of each misuse variant can be estimated, where the cost includes potential losses, should the threats come true
Technology and data variations	A misuser may carry out a misuse case from a variety of technical platforms, such as a PC or a WAP phone and, since only a few equipment-related actions will differ in each case, it is unnecessary to specify two separate paths. Instead, this field lists the candidate types of equipment and explains

	how they differ in particular actions
Terminology and explanations	This field contains explanations of technical terms and other issues

Mandatory fields – as a minimum the following fields should be filled:

- Name, author and date
- Basic path
- Mitigation points
- Trigger
- Assumptions
- Preconditions

Optional fields – the remaining fields can be filled as seen necessary or according to the level of details required.

B. Failure sequence diagrams – safety requirements elicitation

This section provides guidance of how to organize and further explore hazards, misusers and possible mitigations in a structured and systematic manner by use of failure sequence diagrams and textual misuse cases.

Explore hazards and mitigations with FSD

Input: Sequence diagrams and textual use case descriptions
Participants: System, domain and safety experts
Organization: Meeting with facilitator, secretary and participants. Using whiteboard (for drawing FSD) and/or projector (can be used for both drawing (MFSD) and writing (T-MUC)).

B.1. Preparations

Build up understanding of the system interaction by all participants:

1. Go through SD to get an overview of system interactions and main components
2. Read through T-UC in order to understand the details of the system interactions and other relevant information.
3. Read through T-MUC in order to understand the details of the system interactions and other relevant information.
4. Ask for further explanation by domain expert (if needed)

Some of the steps might have been done at the MUC session and thus not need to be repeated (though a reminder might be useful). T-MUC and FSD might be developed in any order and also in parallel. Based on our experience, starting with the graphical FSD is easier for the participants. While developing the FSD, some parts of the T-MUC can be filled in while the finalization of both artifacts might need more than one iteration.

B.2. Guided activity for developing failure scenarios using FSD

Organized as a meeting with system, domain and safety (security can join in for parts) experts

B.2.1 Explore how hazards can be combined into a successful failure scenario

1. Use D-MUC as overview and brainstorm for the composition of failures that can lead to a hazard.

2. Use T-MUC basic and alternative paths as starting point, and discuss the steps involved in the failure propagation and components from SD involved in each of the steps.

B.2.2 Explore mitigations for failures

1. Use D-MUC as overview and brainstorm for the composition of mitigations that can successfully mitigate a hazard.
2. Use T-MUC mitigation points for discussing the steps involved in mitigating the hazard and which components from SD involved in each of the steps.

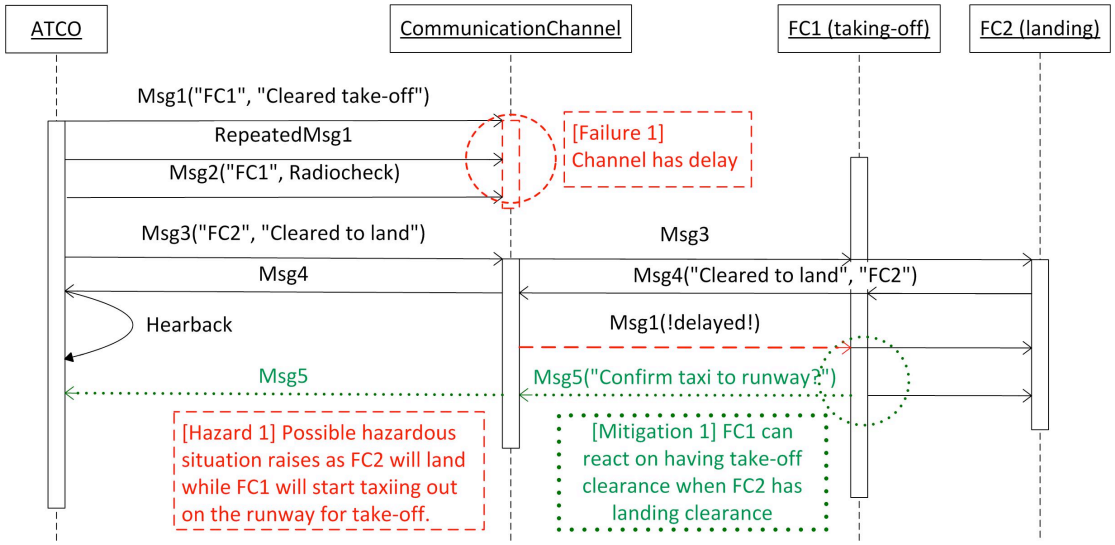

Figure 3 - Example of a FSD for safety from [1]

B.3. How to create FSD during the brainstorming session?

Use an existing SD or create a new SD of system under analysis

B.3.1. Draw FSD

1. Take one *misuse case* from the D-MUC
2. Include *misuser* from T-MUC as *mis-actor/-component/-object*
3. Follow the steps of T-MUC *basic path* and draw failure propagating by system function as *red/dashed messages*

3.1. Consider to include *alternative path* of a *failure* by use of *alt operand* and *messages*

4. Consider to include *repeatable failures* by *loop operand*
5. Enter the *failure* at *life line* of *component/object* (*red dashed circles*) and include *note* describing and refer back to the misuse in D-MUC (*misuse case*) or T-MUC (*basic path*)
6. Consider the mitigations from *mitigation path* by

6.1. Current control in system (exception handling, redundancy, etc.) and draw as *green/dotted circles* around the *failures*

6.2. New *safety functionality* to be implemented by system and draw in as *green/dotted messages*

7. Discuss most efficient mitigation from the FSD based on removal, reduction or control philosophy and conclude on how to mitigate

Iterations between T-MUC and FSD

... TBD

Updating D-MUC after finalizing T-MUC and FSD

... TBD

Update T-MUC

Update T-MUC with new information of hazards, mitigations and propagation of failures. If there are extensive changes, consider creating a new T-MUC for the FSD.

Update D-MUC

After updating new T-MUC, return to original D-MUC:

1. Draw newly identified hazards as misuse cases
2. Draw the relations between new hazards and
    a. Use cases – draw arrow from MUC to UC
    b. Other hazards (by etxension)
    c. Mitigations (if mitigated by existing mitigations)
3. Draw newly identified mitigations as safety use cases
4. Draw the relations between … Relate mitigations to
    a. Hazards that it mitigates
    b. Relate to use cases that it extends or is included by
5. Perform trade-off analysis

Trade-off analysis

1. Compare all mitigations, which are either connected to the same
    a. Use case (e.g. extend)
    b. Hazard
2. Consider all the mitigations and identify and mark whether mitigations are in [5]:
    a. Antagonism – mark with red arrow between the mitigations
    b. Mutual reinforcement – mark with green arrow that they
    c. Conditional dependent – mark with arrow going from the dependant
    d. Indepentent – no marking
3. Evaluate

Re-assess with MUC

The new use cases for mitigation should be subject to a re-assessment, starting with applying guidewords and brainstorming for new threats (ref. SAM and ED-153 for safety).

C. Misuse cases – security requirements elicitation

This section provides guidance of how to organize the identification of threats, misusers and possible mitigations in a structured and systematic manner by use of diagrammatical and textual misuse cases.

Identifying initial threats, misusers and mitigations with MUC

Input: Use case diagrams and textual descriptions
Participants: System, domain and security experts
Organization: Meeting with facilitator, secretary and participants. Using whiteboard and projector.

C.1. Preparations

Build up understanding of the function by all participants:

1. Go through D-UC to get an overview of system functions and actors involved
2. Read through T-UC in order to understand the details of a particular system function and the actors involved.
3. Ask for further explanation by domain expert (if needed)

C.2. Structured/guided brainstorming activity for threats

Organized as a meeting with system, domain and security (safety can join in for parts) experts. Structured process

C.2.1. Applying HAZOP guidewords for identifying threats to functions

1. Apply one guideword to the name of use case (either in front, middle or after name)
2. Brainstorm for misuse of the function (use description from FHA in SAM guideline for how to brainstorm? Length of brainstorm, who to involve, etc.).

C.2.2. Identify misactor

1. Discuss possible misactors for the identified threat.
2. Consider different misactors (use list from Eurocontrol SecRAM?)

C.2.3. Identify mitigation

1. Discuss a possible mitigation to misuse case
2. Consider different mitigations against both misuse case and misactor (e.g., misactor is virus)

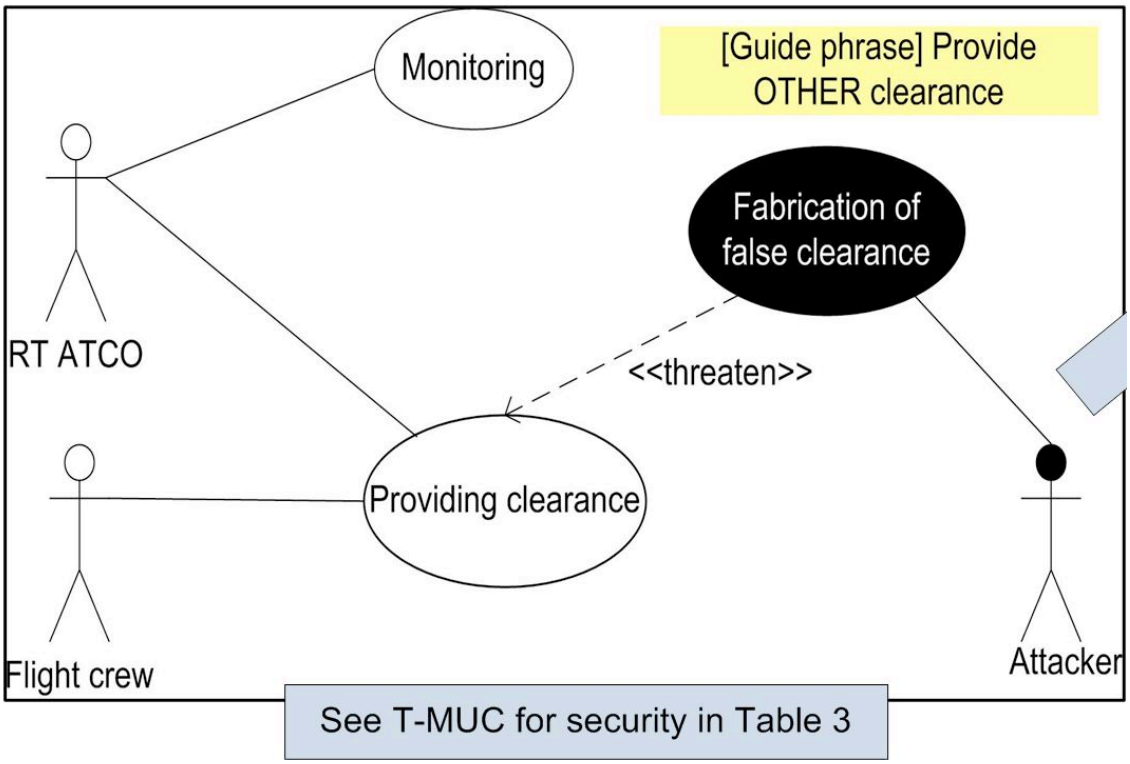

Figure 4 - Example of a D-MUC for security from [1]

C.3. Create D-MUC

Use an existing D-UC or create a new D-UC of system under analysis

C.3.1. Draw D-MUC

1. Add the combined *guideword* and *use case name* as a *guide phrase* in *note* in D-MUC.
2. Draw a *black oval* as *misuse case* right to the *use case* under assessment and write a *phrase* in the *misuse case* with a few words (short and descriptive).
3. Draw an *arrow* from the *misuse case* to the *use case* and name it *threatens*.
4. Draw a *misactor* outside the *system boundary* and relate her/him to *misuse case* by a *line*.
5. Draw a (*security) use case* below the *misuse case* it mitigates and specify the *security function* as *name*.
6. Draw an *arrow* from the (*security) use case* to the *misuse case* and *name* it *mitigates*.

C.4. Create T-MUC

Secretary can start writing some parts of template while diagrams are made, but basic and alternative path should be written or at least throughly checked and agreed upon by participants (check this with REFSQ paper).

C.4.1. Write T-MUC

Table B - Example of T-MUC for security from [2]

Fields name	**Explanation**
Name, summary,	These fields retain the same meaning as in regular use cases

author and date	
Basic path	This field describes the actions that the misuser(s) and the system go through to harm the proposed system
Alternative path	This field describes ways to harm the proposed system that are not accounted for by the basic path, but are still sufficiently similar to be described as variants of the basic path
Mitigation points	This field identifies those actions in a basic or alternative path where misuse can be mitigated. Several ways to mitigate misuse of a particular action can be described in the same field and each of them may be further described in a separate security use case. As for extension points, the misuse case must eventually have a mitigate relationship to a corresponding security use case. However, the detailed description of security use cases is optional, because it is often closer to design, requiring detailed analysis of risks and implementation costs that go beyond use and misuse cases
Extension points	In some cases, a misuse case may be extended with optional paths whose details are described in a separate extension misuse case. This field lists the actions in the main or alternative paths where optional paths may be inserted. As for extension points in regular use cases, the misuse case must have an extend relationship to the misuse case that contains the optional path
Trigger	This field describes the states or events in the system or its environment that may initiate the misuse case. For some misuse cases, the trigger is just the predicate True, indicating a permanently present danger
Assumptions	This field describes the states in the system's environment that make the misuse case possible
Preconditions	This field describes the system states that make the misuse case possible
Mitigation guarantee	This field describes the guaranteed outcome of mitigating a misuse case. If the mitigation points are not yet specified in detail, the mitigation guarantee describes the level of security required from the mitigating security use cases that will be designed later. When the mitigation points in the misuse case have been detailed by security use cases, this field describes the strongest possible security guarantee that can be made, regardless of how the misuse case is mitigated
Related business rules	Typically, business rules will be violated by the misuse. This field contains links to such rules, maybe along with links to rules that enable the threat or that limit how it could be mitigated or eliminated
Misuser profile	This field describes whatever can be assumed about the misuser, for example, whether the misuser acts intentionally or inadvertently; whether the misuser is an insider or

	outsider; and how technically skilled the misuser must be
Scope	This field indicates whether the proposed system in a misuse case is, e.g., an entire business, a system of both users and computers, or just a software system
Iteration	As for regular use cases, it is useful to allow both initial and detailed descriptions of misuse cases. This field indicates the misuse case's iteration level, usually taken from the set of iteration levels used for the use cases in the project
Level	As for regular use cases, misuse cases can be specified at a general or specific abstraction level. This field indicates whether the misuse case is, e.g., a summary, a user goal, or a sub-function, following [3]
Stakeholders and risks	This field specifies the major risks for each stakeholder involved in the misuse case. On an abstract level, risks can be described textually, e.g., ''the system is unavailable for several hours'' or ''a competitor gets hold of sensitive medical data about an applicant''. On a concrete level, the likelihood and cost of each misuse variant can be estimated, where the cost includes potential losses, should the threats come true
Technology and data variations	A misuser may carry out a misuse case from a variety of technical platforms, such as a PC or a WAP phone and, since only a few equipment-related actions will differ in each case, it is unnecessary to specify two separate paths. Instead, this field lists the candidate types of equipment and explains how they differ in particular actions
Terminology and explanations	This field contains explanations of technical terms and other issues

Mandatory fields – as a minimum the following fields should be filled:

- Name, author and date
- Basic path
- Mitigation points
- Trigger
- Assumptions
- Preconditions

Optional fields – the remaining fields can be filled as seen necessary or according to the level of details required.

D. Misuse sequence diagrams – security requirements elicitation

This section provides guidance of how to organize the further explore threats, misusers and possible mitigations in a structured and systematic manner by use of misuse sequence diagrams and textual misuse cases.

Explore threats and mitigations with MUSD

Input: Sequence diagrams and textual use case descriptions

Participants: System, domain and security experts
Organization: Meeting with facilitator, secretary and participants. Using whiteboard (for drawing MUSD) and/or projector (can be used for both drawing (MUSD) and writing (T-MUC)).

D.1. Preparations

Build up understanding of the system interaction by all participants:

1. Go through SD to get an overview of system interactions and main components
2. Read through T-UC in order to understand the details of the system interactions and other relevant information.
3. Read through T-MUC in order to understand the details of the system interactions and other relevant information.
4. Ask for further explanation by domain expert (if needed)

D.2. Guided activity for developing threat scenarios using MUSD

Organized as a meeting with system, domain and security (safety can join in for parts) experts

D.2.1. Explore how threats can be combined into a successful attack scenario

1. Use D-MUC as overview and brainstorm for the composition of threats that can achieve a successful attack.
2. Use T-MUC basic and alternative paths as starting point, and discuss the steps involved in the attack and components from SD involved in each of the steps.

D.2.2. Explore mitigations for threats

1. Use D-MUC as overview and brainstorm for the composition of mitigations that can successfully stops an attack.
2. Use T-MUC mitigation points for discussing the steps involved in stopping the attack and which components from SD involved in each of the steps.

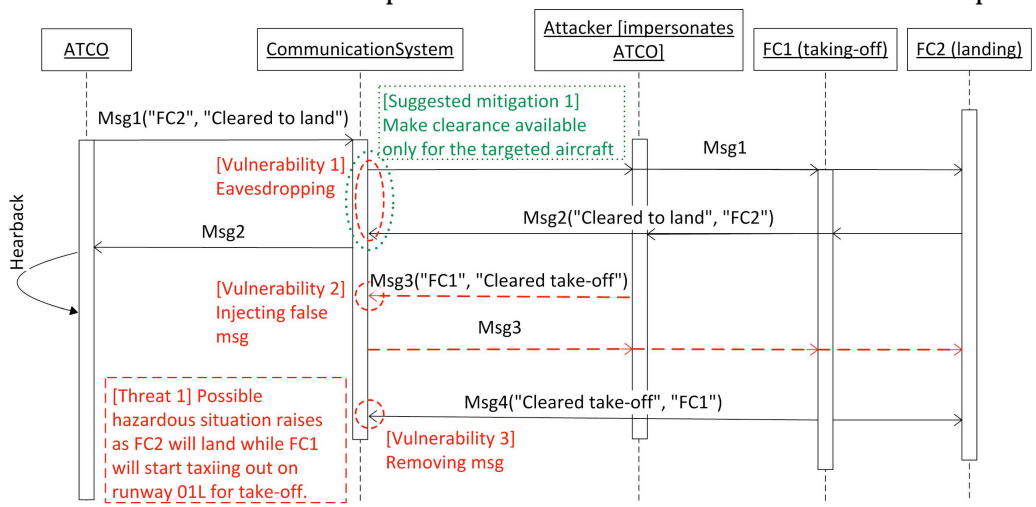

Figure 5 - Example of a MUSD for security from [1]

D.3. Create MUSD

Use an existing SD or create a new SD of system under analysis

D.3.1. Draw MUSD

1. Take one *misuse case* from the D-MUC
2. Include *misuser* from T-MUC as *mis-actor/-component/-object*
3. Follow the steps of T-MUC *basic path* and draw as *messages*

3.1. Attacker using legal system function, draw it as *black message*

3.2. Attacker using illegal system function (hacked/cracked), draw it as *red/dashed message*

4. Consider to include *alternative path* of attacker by use of *alt operand* and *messages*
5. Consider to include more attempts of attacker by *loop operand*
6. Enter the misuse as *vulnerabilities* at *life line* of *component/object* (*red dashed circles*) and include *note* describing and refer back to the misuse in D-MUC (*misuse case*) or T-MUC (*basic path*)
7. Consider the mitigations from *mitigation path* by

7.1. Current control in system (firewall, logging etc.) and draw as *green/dotted circles* around *vulnerability*

7.2. New security functionality to be implemented by system and draw in as *green/dotted messages*

8. Discuss most efficient mitigation from the MUSD based on removal, reduction or control philosophy and conclude on how to mitigate

Update T-MUC

Update T-MUC with new information of vulnerabilities, mitigations and steps of attacker. If there are extensive changes, consider creating a new T-MUC for the MUSD.

Update D-MUC

After updating/creating new T-MUC, return to original D-MUC:

1. Add new threats identified
2. Relate threats to
    a. Use cases
    b. Other threats (by etxension)
    c. Mitigations (if mitigated by existing mitigations)
3. Add new mitigations identified
4. Relate mitigations to
    a. Threats that it mitigates
    b. Relate to use cases that it extends or is included by
5. Perform trade-off analysis

Trade-off analysis

1. Compare new mitigations to other mitigations, which are either connected to the same
    a. Use case
    b. Threat
2. Identify and mark whether mitigations are in [5]:
    a. Antagonism – mark with red arrow between the mitigations
    b. Mutual reinforcement – mark with green arrow that they

c. Conditional dependent – mark with arrow going from the dependant
d. Indepentent – no marking

Re-assess with MUC

The new use cases for mitigation should be subject to a re-assessment, starting with applying guidewords and brainstorming for new threats (ref. SAM and ED-153 for safety).

Trade-off analysis between security and safety artifacts after an iteration

1. Draw an updated D-MUC including all the hazards, threats and mitigation (from safety and security)

2. Create a consistent and complete set of mitigations covering both safety and security issues

3. Change the other artifacts affected (i.e. T-MUCs and FSDs, MUSDs) accordingly

4. Create updated regular D-UCs, T-UCs, Sds and start the process from the beginning considering the new parts.

Finishing the process

When no new mitigations appear, the CHASSIS process can be finished by creating the HAZOP tables based on the T-MUCs produced and creating the (consistent and complete) safety and security reuirements based on the HAZOp tables.

References

1. C. Raspotnig, P. Karpati, V. Katta. A Combined Process for Elicitation and Analysis of Safety and Security Requirements. In *Enterprise, Business-Process and Information System: Vol. 113*. (pp. 347-361). Springer Berlin Heidelberg. 2012
2. G. Sindre, A. L. Opdahl. Eliciting security requirements with misuse cases. In *Journal Requirement Engineering,* 10 (1), 34–44. 2005
3. A. Cockburn. Writing Effective Use Cases. Addison-Wesley. 2001
4. M. Fowler. UML Distilled third edition – A brief guide to the standard object modelling lanuage. Addison-Wesley. 2004
5. L. Pietre-Cambacedes, M. Bouissou. Modeling safety and security interdepedencies with BDMP (Boolean logic Driven Markov Processes). In *Proceeding of the 2010 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 2852-2861). IEEE Computer Society. 2010