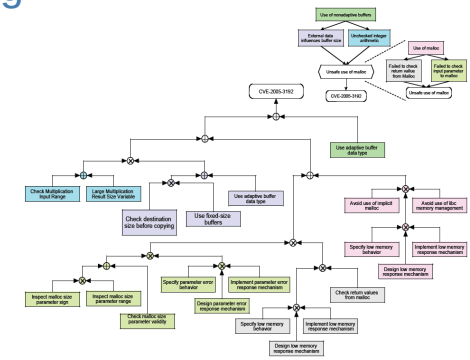
Security Modeling

Anna Vapen

TDDC90 Software Security
2013-12-03



LiU

1

Outline

- What is security modeling?
- Threat modeling
- Misuse cases
- Attack trees
- Vulnerability modeling

- ...and some practical exercises!

LiU

2

What is security modeling?

- Draw security problems as diagrams and pictures
- Useful for:
  - Finding suitable security requirements
  - Finding possible attacks and their mitigations
  - Finding causes to vulnerabilities
  - ...and much more!

- Commonly used in SDL and several risk management methods

LiU

3

Threat Modeling

Threat modeling

- Done early: During design and specification
- Typically part of risk management!
- Broad concept – can be done in many ways

- Threat modeling steps:
  - Identify critical assets
  - Decompose the system to be assessed
  - Identify possible points of attack
  - Identify threats
  - Categorize and prioritize the threats

LiU

4

Threat Modeling (contd.)

Threat modeling

- Treat modeling usually contains:
  - Assets (examples below)
    - Personal information, passwords
    - Confidential data
    - Financial data
    - Source code
  - Entry points (examples below)
    - On the network level, each service is an entry point
    - Vulnerabilities in web servers, protocols, databases…
  - Threat scenarios

LiU 5

Use Cases

Use- and misuse cases

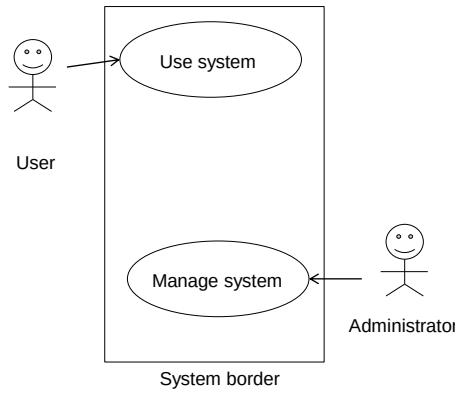
- Helps defining how a system should work
- Parts of a use case:
  - Usage story: Describes actors in the system and their actions
  - Use case diagram
  - Traceability features (who made the use case, when was it made, which actor is most important, what actions are essential etc.)
- In the diagram and story:
  - Actors, e.g. user, system administrator
  - Actions, e.g. log in, encrypt message, delete account

LiU 6

Simple Use Case

Use- and misuse cases

Use case diagram



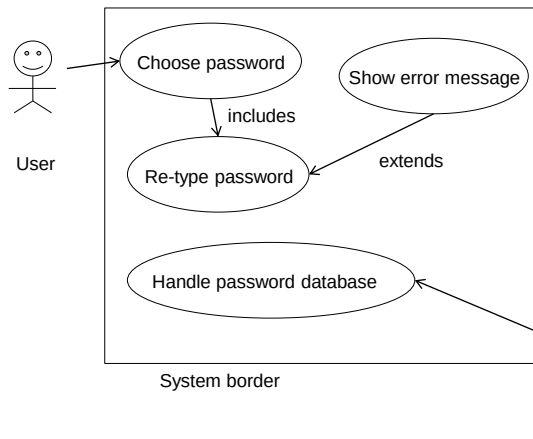
Usage story:
The user uses the system. The administrator manages the system.

LiU

(Not so) Simple Use Case

Use- and misuse cases

Use case diagram



Usage story:
The user chooses a password for the system. The selection of password includes re-typing the password. If the password in wrongly typed the second time, an error message is shown. The administrator handles the password database.

LiU

Simple Misuse cases

Syntax by Sindre and Opdahl

Use- and misuse cases

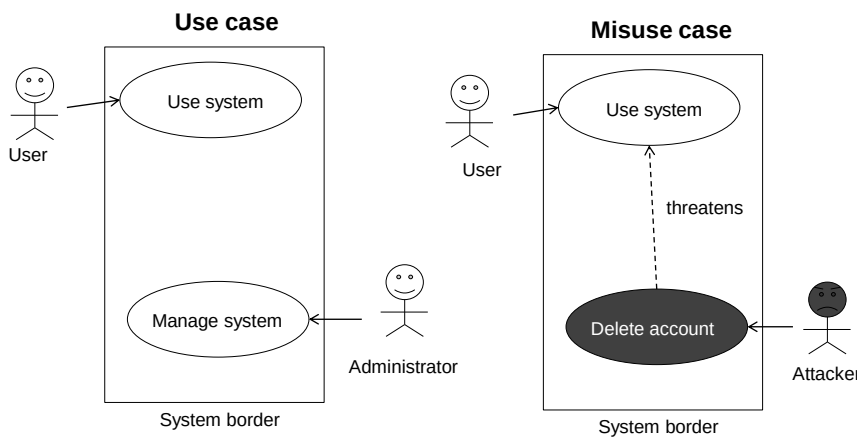
- Same as in use cases:
  - Positive actors and actions
  - Include and extend
- This is new for misuse cases:
  - Negative actors (attackers and other “misusers”) and actions
  - Threatens, detects and prevents

LiU

9

From Use Case to Misuse Case

Use- and misuse cases

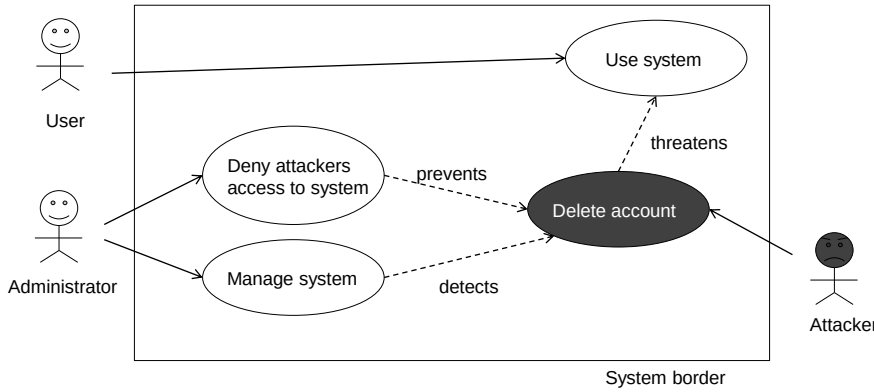


LiU

10

Misuse case: Simple Syntax

Use- and misuse cases



Extended Misuse cases

Syntax by Røstad

Use- and misuse cases

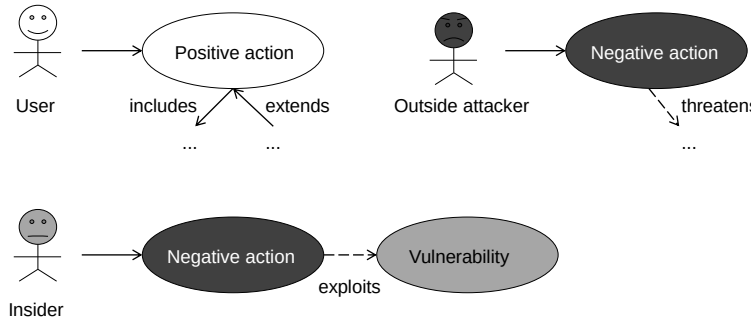
- Same as in use cases:
  - Positive actors and actions
  - Arrow labels: Include and extend
- Same as in simple misuse cases:
  - Negative actors (**outside** attackers) and actions
  - Threatens, ~~prevents~~ **mitigates** (detect is not used)
- New for extended misuse cases:
  - Insiders (e.g. good users gone bad, attackers with guest accounts)
  - Exploit and vulnerabilities

Extended Misuse cases (contd.)

Syntax by Røstad

Use- and misuse cases

- Let's look at our actors!



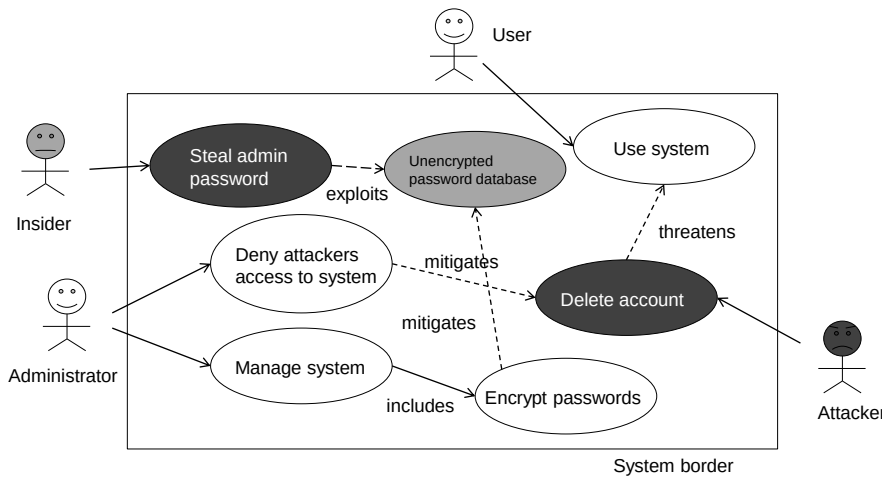
Note: Insiders can also do positive things.
Users and other positive actors can do things that lead to vulnerabilities!

LiU

13

Misuse case: Extended Syntax

Use- and misuse cases



Note: Important to model **each** high-risk attack and its **mitigations**

LiU

14

Exercise: Misuse Cases

Extended Syntax

Use- and misuse cases

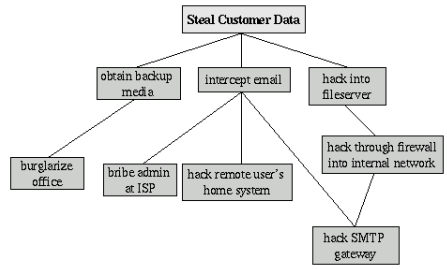
- In groups of 2-3 students
- Let's model the exploitation from the Pong-lab!
- You are the insider (already have a limited account)
- Your goal is to get root access
- ...you are also the administrator fixing the vulnerability

- *10 minutes modeling starts now!* ☺

Attack Trees

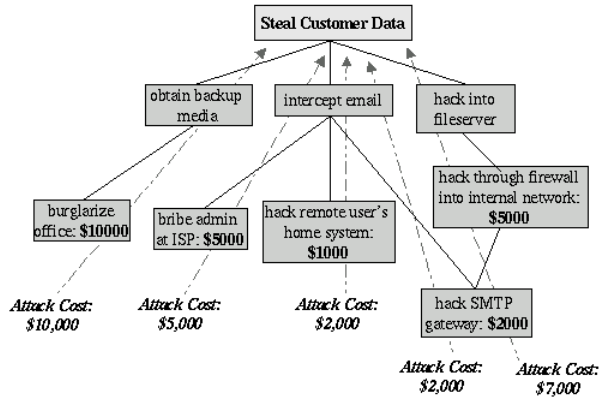
Attack trees

- Attack tree: visualization of possible attacks against a target
- Top node: Attack goal
- Sub-nodes: How to reach the attack goal



Attack Trees (contd.)

Attack trees



LiU

17

Exercise: Attack trees

(Simple, no labels on edges)

Attack trees

- In groups of 2-3 students
- Let's model the attack against Pong
- *5 minutes fast brainstorming and modeling!*

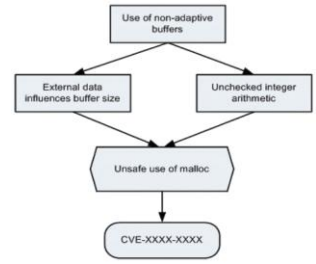
LiU

18

Vulnerability Modeling

Vulnerability modeling

- Method included in S³P
- Model vulnerabilities
  - Initial analysis
  - Identifying vulnerability causes
  - Constructing Vulnerability Cause Graphs (VCGs)



LiU

19

Example: CVE-2005-3192

Vulnerability modeling

- Heap-based buffer overflow in Xpdf 3.01 (CVE-2005-3192)
- Xpdf: open-source viewer for PDF files
- User-supplied integers are not checked for overflow or underflow in buffer size calculations
- A remote attacker may execute arbitrary code in the context of a user running the application

LiU

20

Example (contd.)

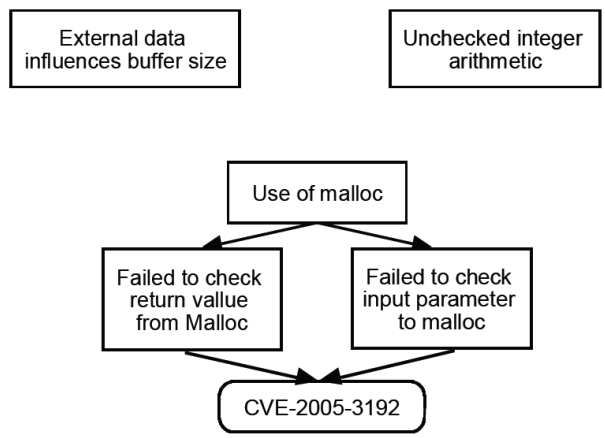
Vulnerability modeling

```
StreamPredictor::Stream Predictor(Stream *str, int predictor, int
    width, int nComps, int nBits) {
    nVals = width * nComps;
    pixBytes = (nComps * nBits + 7) >> 3;
    rowBytes = ((nVals * nBits + 7) >> 3) + pixBytes;
    predLine = malloc(rowBytes);
    memset(predLine, 0, rowBytes);
    predIdx = rowBytes;
}
```

LiU 21

Modeling CVE-2005-3192

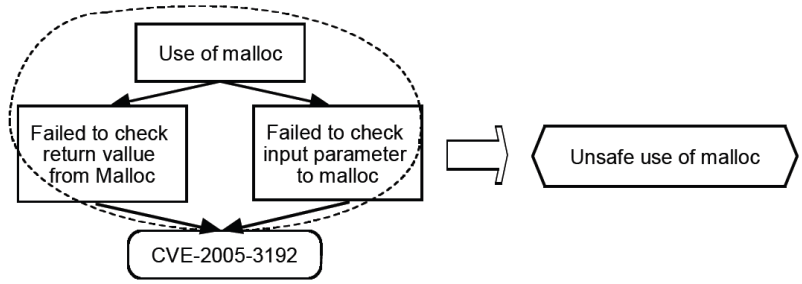
Vulnerability modeling



LiU 22

Modeling CVE-2005-3192

Vulnerability modeling

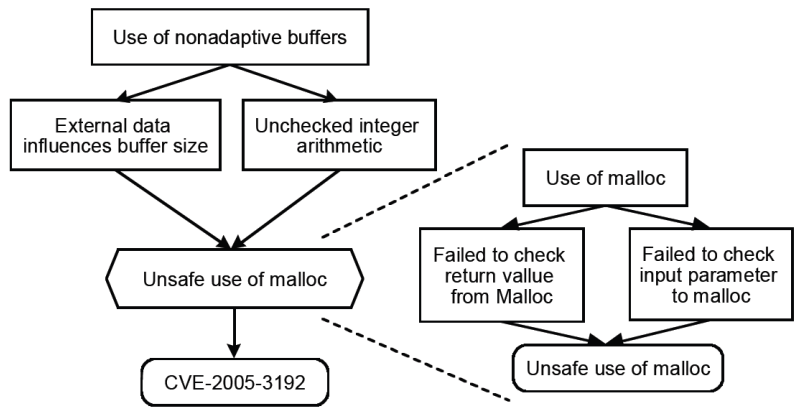


LiU

23

Modeling CVE-2005-3192

Vulnerability modeling



LiU

24

Example: CVE-2005-2558

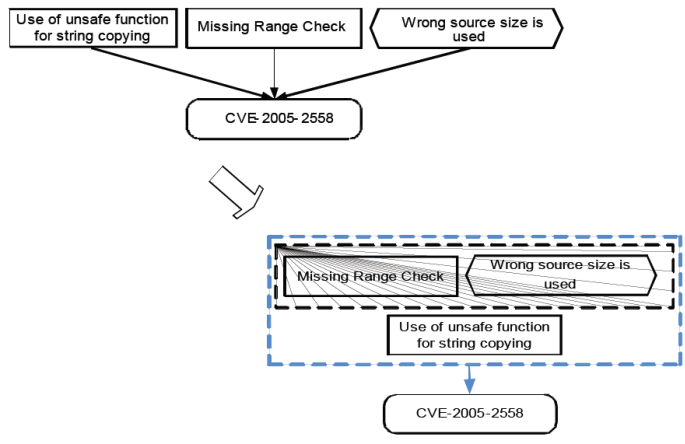
Vulnerability modeling

- Buffer overflow in MySQL
  - Stack-based buffer overflow in the init_syms function in MySQL 4.0 (several versions) allows remote authenticated users who can create user-defined functions to execute arbitrary code via a long function_name field.
- Analysis result
  - Unsafe string copy function
  - Function name (64 characters) copied into a buffer (size = 34)
  - Buffer size defined by MAX_FIELD_NAME instead of NAME_LEN

LiU 25

Modeling CVE-2005-2558

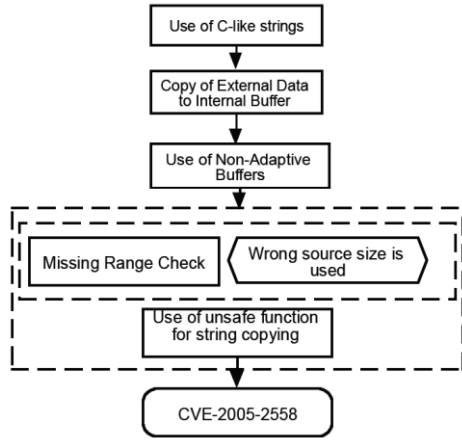
Vulnerability modeling



LiU 26

Modeling CVE-2005-2558

Vulnerability modeling

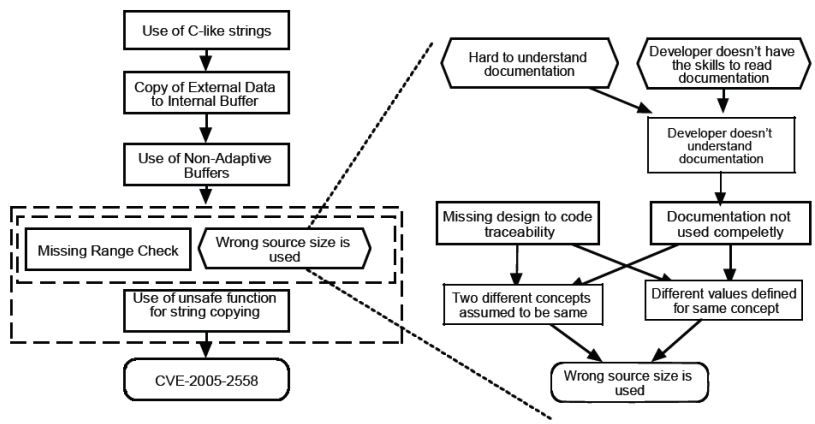


LiU

27

Modeling CVE-2005-2558

Vulnerability modeling

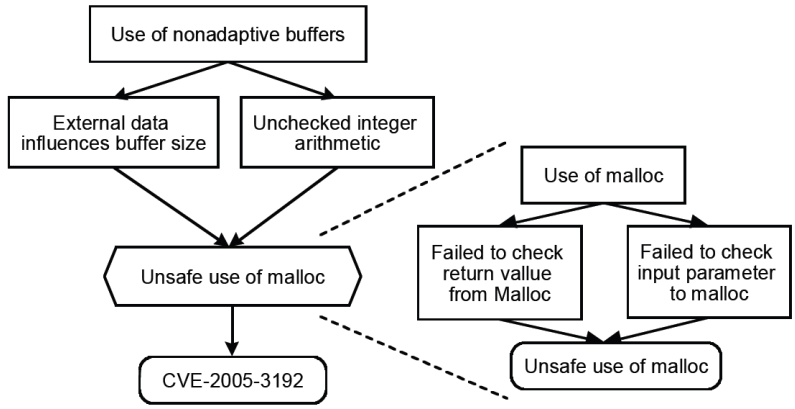


LiU

28

Modeling CVE-2005-3192

Vulnerability modeling



LiU

29

Exercise: VCGs

Vulnerability modeling

- In groups of 2-3 students
- Let's model...

  the buffer overflow in Pong!

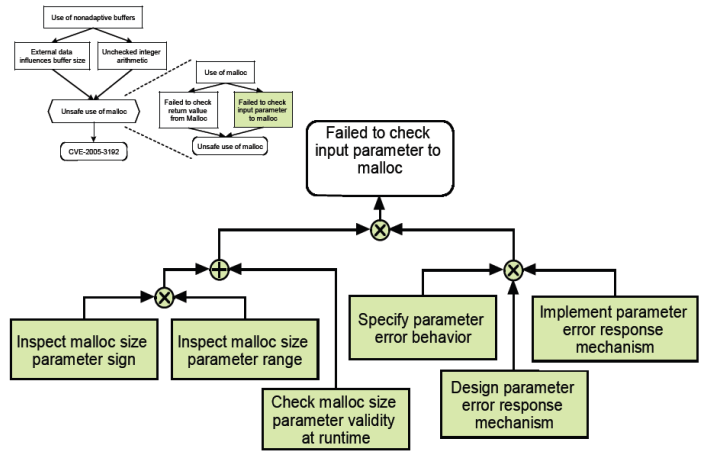
5 minutes modeling starts now!

LiU

30

SAG

Vulnerability modeling

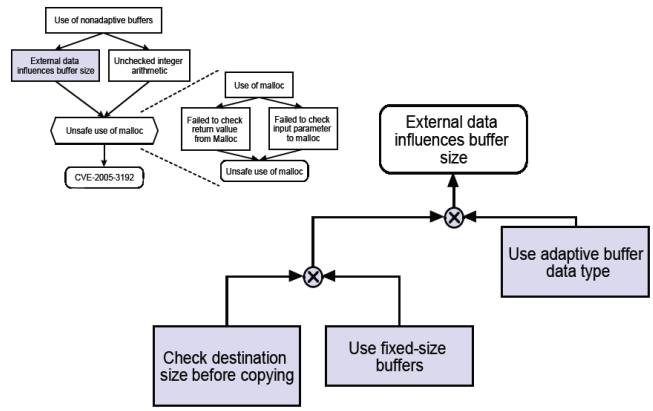


LiU

31

SAG

Vulnerability modeling

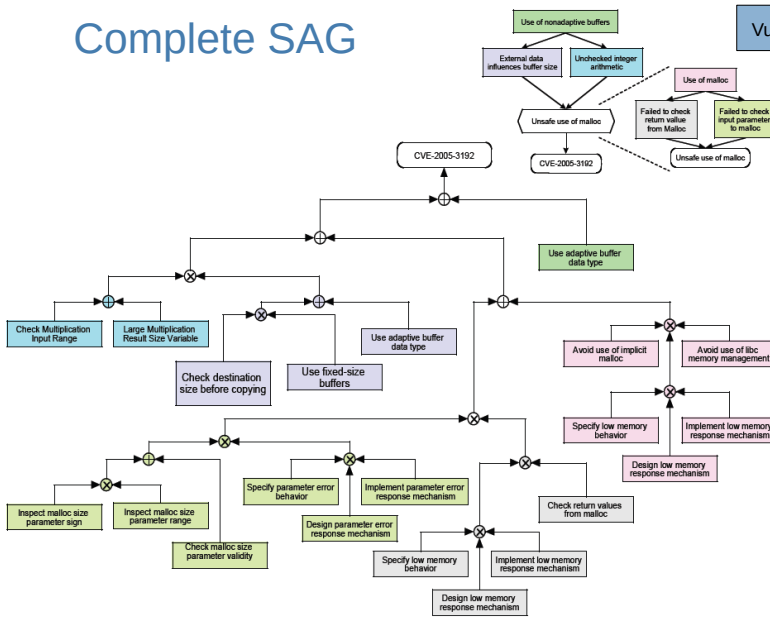


LiU

32

Complete SAG

Vulnerability modeling



LiU

33

Summary

- Security modeling: Draw security scenarios
  - Pictures and diagrams helping you to fix security problems
- Modeling techniques seen today:
  - Threat modeling (broad concept)
  - Misuse cases
  - Attack trees
  - Vulnerability modeling: VCGs and SAGs

LiU

34

Practical Information

- Only **two** lectures left!
  - Common Criteria guest lecture
  - Course wrap-up
- All other lecture slots in the schedule is extra time (not used)

- Labs must be passed at 16th December → finish earlier!
- Muddy card evaluation!
  - What do you like with this course? (Labs, lectures, people, contents, reading material, website, structure...)
  - Should something be improved? (We listen and apply fixes!)

LiU

35