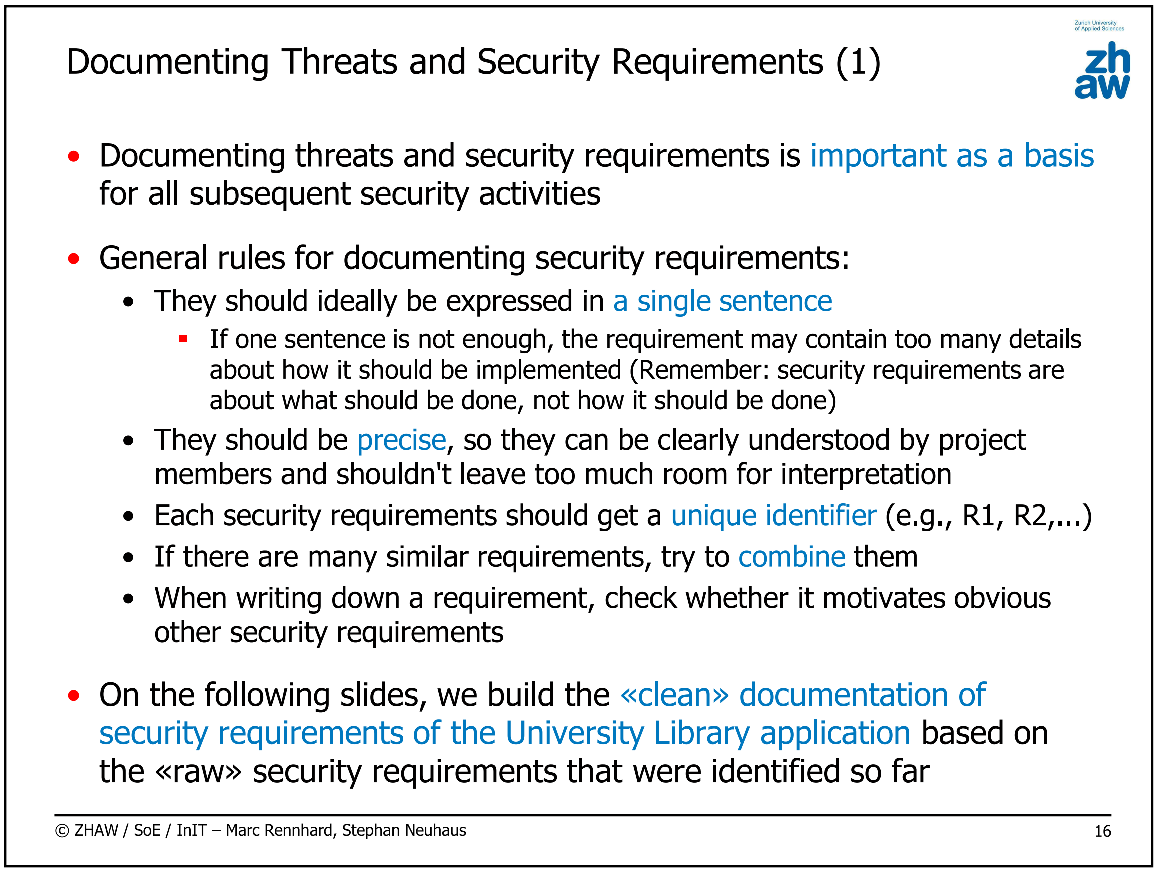


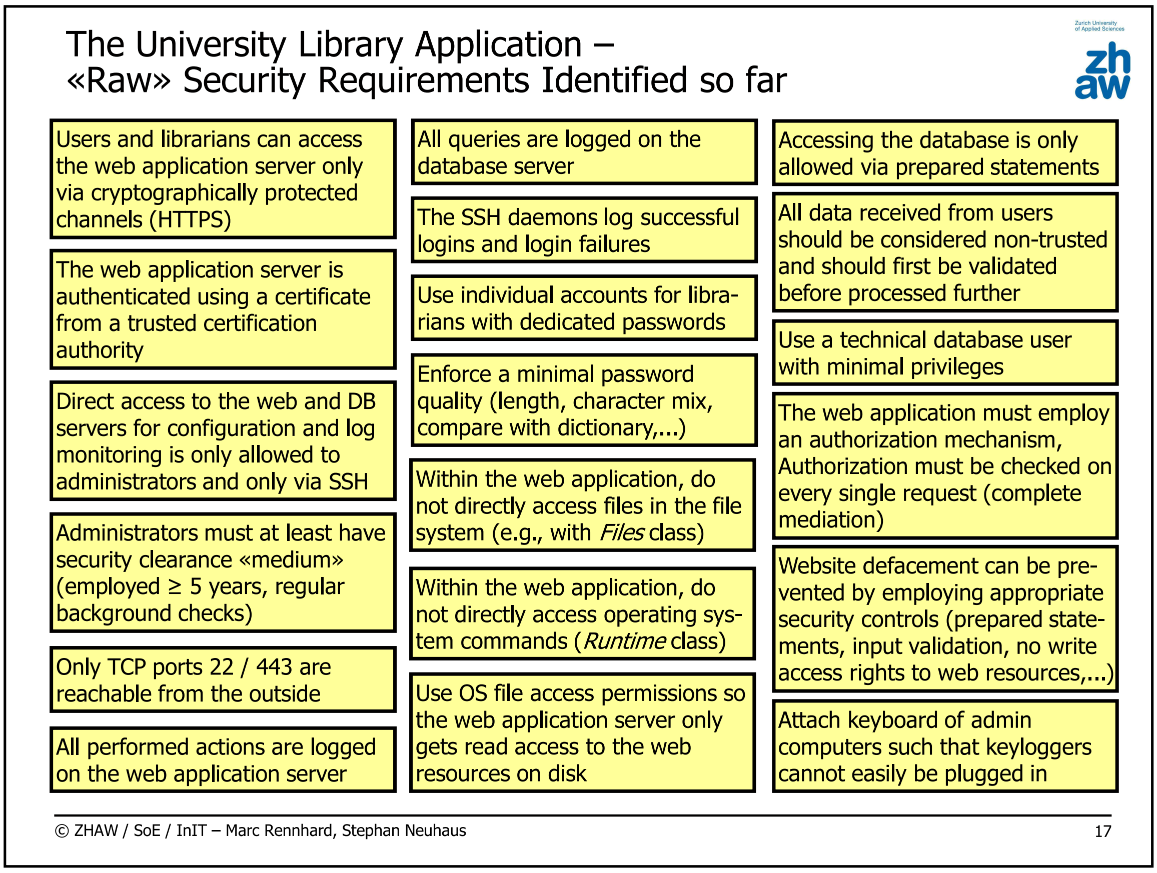
Documenting Threats and Security Requirements (1)

- Documenting threats and security requirements is important as a basis for all subsequent security activities
- General rules for documenting security requirements:
  - They should ideally be expressed in a single sentence
    - If one sentence is not enough, the requirement may contain too many details about how it should be implemented (Remember: security requirements are about what should be done, not how it should be done)
  - They should be precise, so they can be clearly understood by project members and shouldn't leave too much room for interpretation
  - Each security requirements should get a unique identifier (e.g., R1, R2,...)
  - If there are many similar requirements, try to combine them
  - When writing down a requirement, check whether it motivates obvious other security requirements
- On the following slides, we build the «clean» documentation of security requirements of the University Library application based on the «raw» security requirements that were identified so far

Documenting Security Requirements

There is no widely used or accepted standard about how security requirements should be documented. In general, you can use virtually any method / template (e.g. wikis, spread sheets, custom developed tools etc.) you like, but what's important is that you make sure that this is then used consistently during the entire project and that security requirements are truly documented. Because if some of them are not documented, they will likely be forgotten and no countermeasures will be implemented.

Remember that security requirements engineering is usually an iterative process, so documentation must typically be updated during each iteration. With too much documentation, it can easily happen that not everything will be maintained and updated consistently. So it's important to focus on the most important details when documenting security requirements and leave out what's not needed.



The University Library Application – «Raw» Security Requirements Identified so far

Users and librarians can access the web application server only via cryptographically protected channels (HTTPS)

The web application server is authenticated using a certificate from a trusted certification authority

Direct access to the web and DB servers for configuration and log monitoring is only allowed to administrators and only via SSH

Administrators must at least have security clearance «medium» (employed ≥ 5 years, regular background checks)

Only TCP ports 22 / 443 are reachable from the outside

All performed actions are logged on the web application server

All queries are logged on the database server

The SSH daemons log successful logins and login failures

Use individual accounts for libra-rians with dedicated passwords

Enforce a minimal password quality (length, character mix, compare with dictionary,...)

Within the web application, do not directly access files in the file system (e.g., with *Files* class)

Within the web application, do not directly access operating sys-tem commands (*Runtime* class)

Use OS file access permissions so the web application server only gets read access to the web resources on disk

Accessing the database is only allowed via prepared statements

All data received from users should be considered non-trusted and should first be validated before processed further

Use a technical database user with minimal privileges

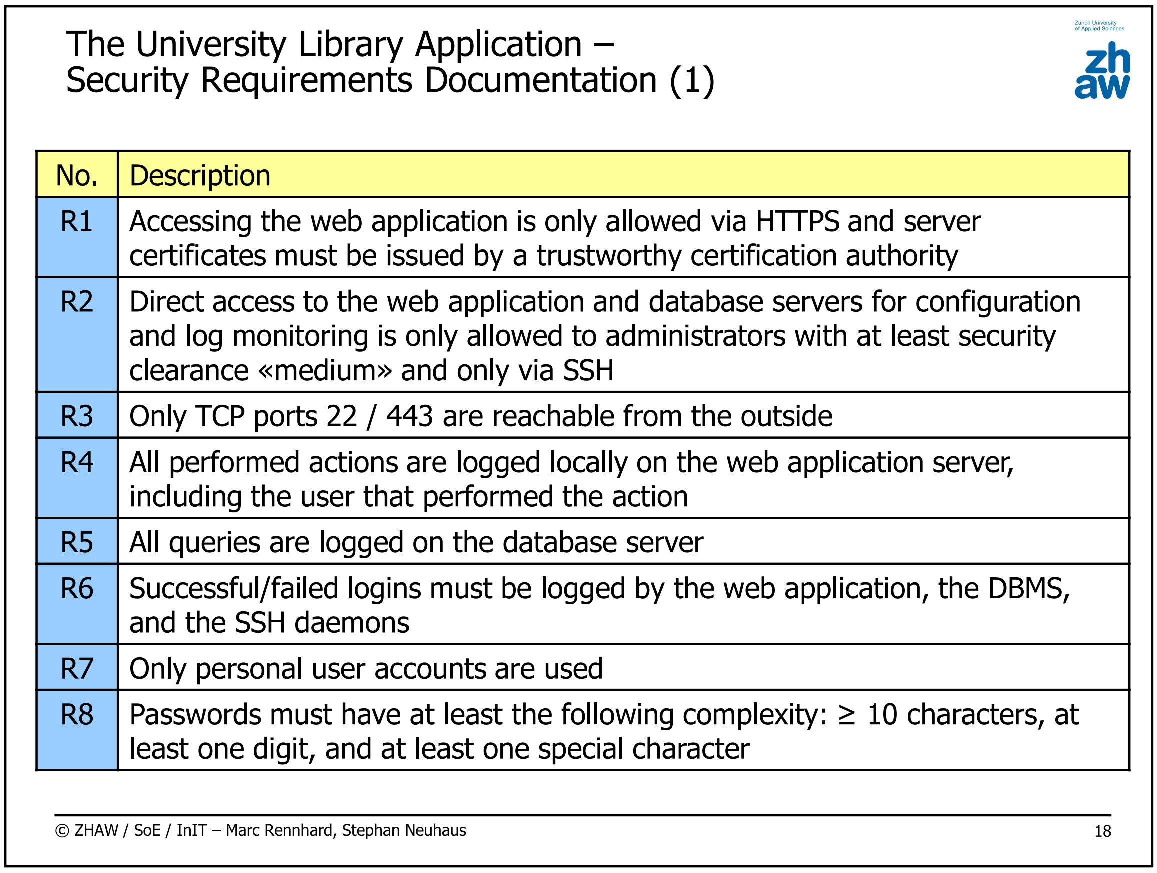
The web application must employ an authorization mechanism, Authorization must be checked on every single request (complete mediation)

Website defacement can be pre-vented by employing appropriate security controls (prepared state-ments, input validation, no write access rights to web resources,...)

Attach keyboard of admin computers such that keyloggers cannot easily be plugged in

«Raw» Security Requirements identified so far

This slide contains all security requirements that were identified so far throughout this chapter. So far, we didn't follow a specific way to document the security requirements and they were basically just written down without thinking too much about them and consequently, they are currently a bit «rough around the edges» and there may be redundancy among them. Therefore, they are first reviewed and rearranged a bit before they are inserted into the «clean» documentation of security requirements that follows on the next two slides.



The University Library Application – Security Requirements Documentation (1)

No.	Description
R1	Accessing the web application is only allowed via HTTPS and server certificates must be issued by a trustworthy certification authority
R2	Direct access to the web application and database servers for configuration and log monitoring is only allowed to administrators with at least security clearance «medium» and only via SSH
R3	Only TCP ports 22 / 443 are reachable from the outside
R4	All performed actions are logged locally on the web application server, including the user that performed the action
R5	All queries are logged on the database server
R6	Successful/failed logins must be logged by the web application, the DBMS, and the SSH daemons
R7	Only personal user accounts are used
R8	Passwords must have at least the following complexity: ≥ 10 characters, at least one digit, and at least one special character

«Clean» Documentation of Security Requirements

The table illustrated above (and continued on the next slide) shows how a reasonably good documentation of security requirements may look like. As can be seen, not very much is required and a table with a description of the security requirements where each gets a unique identifier is usually enough. The following remarks describe how this documentation was built based on the «raw» security requirements on the previous slide.

Remarks

- R1: This includes the requirements “Users and librarians can access the web application server only via cryptographically protected channels (e.g. HTTPS / TCP port 443)” and “The web application server is authenticated using a certificate from a trusted certification authority”.
- R2: This includes the requirements “Direct access to the web and database servers for configuration and log monitoring is only allowed to administrators and only via SSH” and “Administrators must at least have security clearance «medium» (employed ≥ 5 years, regular background checks)”.
- R4: This was extended to emphasize that actions must be associated with a specific user because this is requested by one of the security goals.
- R6: This is a generalization of “The SSH daemons log successful logins and login failures” to emphasize that all logins and login attempts should be logged by all components.
- R7: This is a generalization of “Use individual accounts for librarians with dedicated password”.
- R8: This requirement was elaborated to specify what is considered a good password. Otherwise, it is unclear what a “good password” is.

The University Library Application – Security Requirements Documentation (2)

No.	Description
R9	Within the web application, do not perform any direct accesses to the underlying operating system (files, operating system commands,...)
R10	All processes should only get the minimal access rights necessary
R11	Accessing the database is only allowed via prepared statements
R12	All data received from users should be considered non-trusted and should first be validated before processed further
R13	Use a technical database user with minimal privileges
R14	The web application must employ an authorization mechanism, authorization must be checked on every single request (complete mediation)
R15	Cryptographically protected channels must employ at least 128-bit keys for symmetric encryption and must not use RC4 for symmetric encryption or MD5 for hash/MAC computation
R16	Attach keyboard of admin computers such that keyloggers cannot easily be plugged in

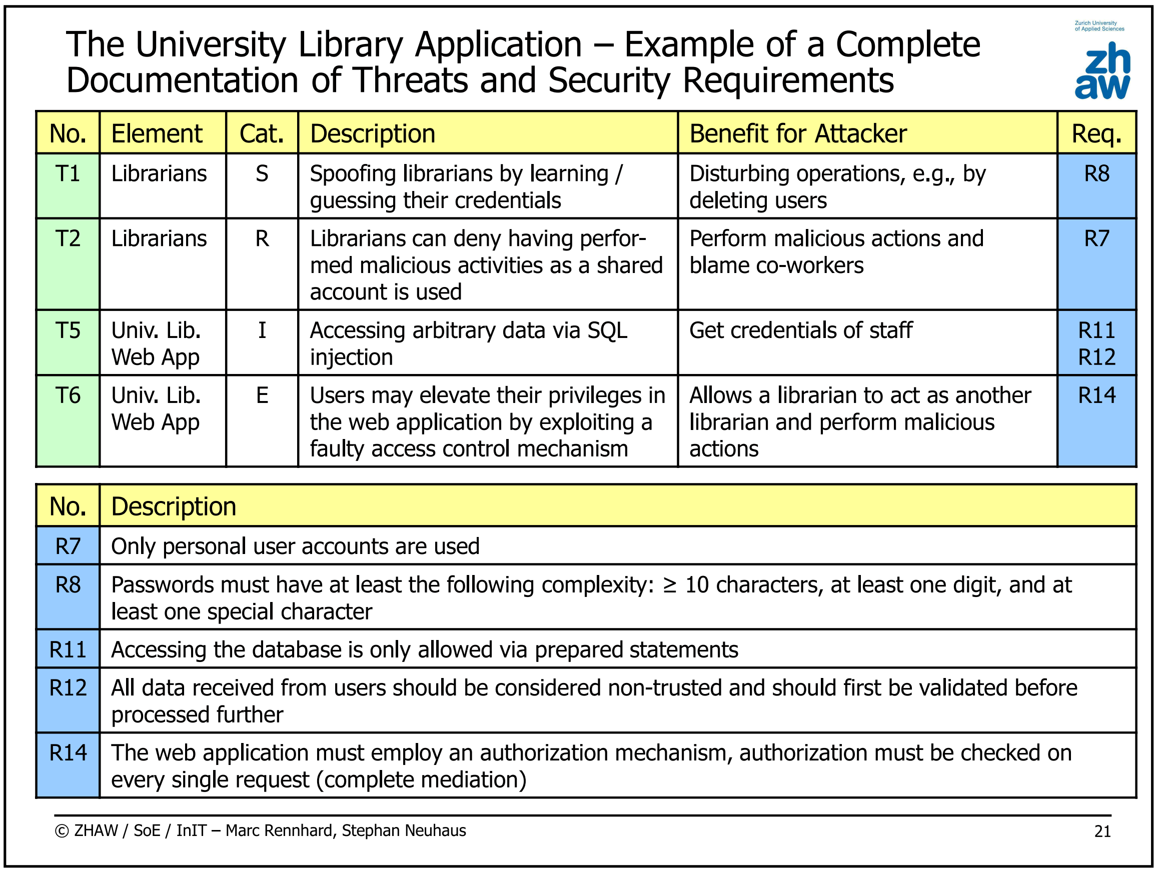
Remarks

- R9: This is a generalization of "Within the web application, do not directly access files in the file system (e.g., with *Files* class)" and "Within the web application, do not directly access operating system commands (with the *Runtime* class)".
- R10: This includes and is a generalization of "Use OS file access permissions so the web application server only gets read access to the web resources on disk", "Website defacement can be prevented by employing appropriate security controls (prepared statements, input validation, no write access rights to web resources...)".
- R11: This also partly includes "Website defacement can be prevented by employing appropriate security controls (prepared statements, input validation, no write access rights to web resources,...)".
- R12: This also partly includes "Website defacement can be prevented by employing appropriate security controls (prepared statements, input validation, no write access rights to web resources...)".
- R15: This requirement was added because other requirements that concern cryptographic channels do not make any statements about key lengths and allowed algorithms. To avoid repeatedly adding this information to other requirements (here R1 and R6), a specific requirement is added

Documenting Threats and Security Requirements (2)

- Beyond documenting the actual security requirements, one should also document the threats and map the requirements to the threats
  - This helps to understand the necessity of the individual security requirements and to check that all threats are covered

- A complete documentation should therefore include:
  - A complete list of threats that are considered realistic
    - Should provide enough information so that the threats are at least understandable by project members
    - One way to do this is as follows (for each threat): Describe the attack and give an example of the benefit for the attacker when the attack is executed
  - A complete list of security requirements, concise and precise
  - A mappings that shows which requirements are used to mitigate which threats
  - All related artifacts (data flow diagrams, attack trees,...)



The University Library Application – Example of a Complete Documentation of Threats and Security Requirements

No.	Element	Cat.	Description	Benefit for Attacker	Req.
T1	Librarians	S	Spoofing librarians by learning / guessing their credentials	Disturbing operations, e.g., by deleting users	R8
T2	Librarians	R	Librarians can deny having perfor-med malicious activities as a shared account is used	Perform malicious actions and blame co-workers	R7
T5	Univ. Lib. Web App	I	Accessing arbitrary data via SQL injection	Get credentials of staff	R11 R12
T6	Univ. Lib. Web App	E	Users may elevate their privileges in the web application by exploiting a faulty access control mechanism	Allows a librarian to act as another librarian and perform malicious actions	R14

No.	Description
R7	Only personal user accounts are used
R8	Passwords must have at least the following complexity: ≥ 10 characters, at least one digit, and at least one special character
R11	Accessing the database is only allowed via prepared statements
R12	All data received from users should be considered non-trusted and should first be validated before processed further
R14	The web application must employ an authorization mechanism, authorization must be checked on every single request (complete mediation)

Documentation of Threats

The documentation of the threats can be different from the one above. But when using STRIDE as the basis, it's reasonable to include the DFD elements and the STRIDE categories as well. On the other hand, a minimal threat documentation could also just consist of threat numbers, descriptions and corresponding security requirements.

Of course, in reality, all threats and security requirements would be listed. The tables above just show an excerpt due to space restrictions.

Summary

- Security Requirements Engineering means determining security requirements early in the development lifecycle
  - It's a very important and fundamental security activity as it provides the basis for the following security activities during the development lifecycle
- Threat modeling is closely associated with security requirements engineering
  - Security requirements engineering without threat modeling is hardly going to produce good results
  - The goal of threat modeling is to identify security design flaws, i.e., to find out whether the already specified security requirements or security controls are appropriate
- Security requirements engineering is basically a creative and «not very scientific» process, but several methods help to perform threat modeling
  - E.g., data flow diagrams, STRIDE, attack trees