**Security Process, Misuse Cases & Attack Trees (4): Mandatory**

Q 1: Which process models in security engineering do you know? What are typical activities of a security analysis?

A: Security Engineering for the V-Model.

Security Analysis:

1. Structure analysis
2. Protection requirements analysis
3. Threat analysis
4. Risk analysis

Q 2: Which optimizations are suggested by the BSI?

A: The Federal Oﬃce for Information Security (German: Bundesamt für Sicherheit in der Informationstechnik, BSI)

Q 3: How to classify protection goals based on the categories communication content and communication circumstances?

A:

**Conﬁdentiality** ensures that nobody apart from the communicants can discover the content of the communication

**Hiding** ensures the conﬁdentiality of the transfer of conﬁdential user data. This means that nobody apart from the communicants can discover the existence of conﬁdential communication

**Anonymity** ensures that a user can use a resource or service without disclosing his/her identity, not even the communicants can discover the identity of each other

**Unobservability** ensures that a user can use a resource or service without others being able to observe that the resource or service is being used. Parties not involved in the communication can observe neither the sending nor the receiving of messages

**Integrity** ensures that modiﬁcations of communicated content (including the sender’s name, if one is provided) are detected by the recipient(s)

**Accountability** ensures that sender and recipients of information cannot successfully deny having sent or received the information. This means that communication takes place in a provable way

**Availability** ensures that communicated messages are available when the user wants to use them

**Reachability** ensures that a peer entity (user, machine, etc.) either can or cannot be contacted depending on user interests.

**Legal** Enforceability ensures that a user can be held liable to fulﬁll his/her legal responsibilities within a reasonable period of time.

Q 4: Which correlations between protection goals do you know?

A: **Tasks for a Protection Requirement Analysis**

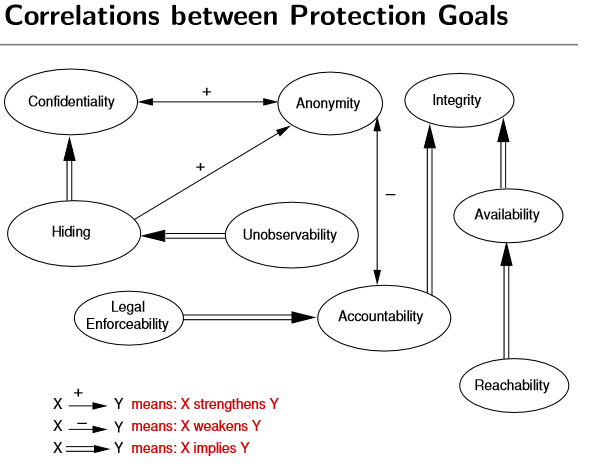
Ü Check the feasibility of the collected protection goals

Ü Suggest possible compromises in case of conﬂicts (e.g. pseudonymity – false name of a person or do not disclose his original/legal identity)

**Basic Theory**

Ü Multi Lateral Security

Ü Correlations and monotony behaviour of protection goals



Q 5: What are the most important artifacts for security analyses?

A:

**1 Structure Analysis**

* Network topology model
* Table with attributes for each system component
* Initial requirements speciﬁcation

**2 Protection Requirements Analysis**

* Use case model to describe actors and system functionality
* Identiﬁying damage scenarios, protection goals and conﬂicts
* Evaluation of the required level of protection for scenarios

**3 Threat Analysis**

* Misuse case model to decribe misactors, attacker behaviour and countermeasures
* Attack tree model to reﬁne attacker goals

**4 Risk Analysis**

* Risk assessment to ﬁnd out the required level of protection
* Annotated attack trees, e.g. using costs and probabilities

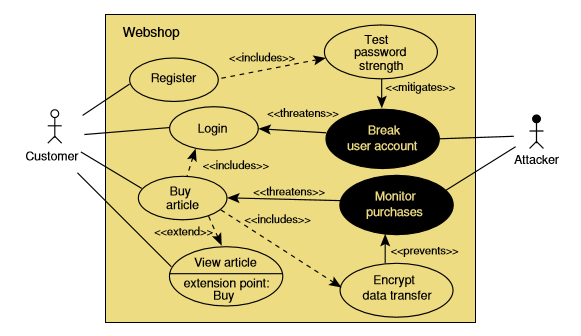
Q 6: What is the basic notation of misuse cases?

A:

Misuse cases model attacker behaviour

Attacker is also called Misactor

Relationships: threatens, mitigates & prevents



+ Provides a detailed analysis of attack scenarios

+ Is supported by a distinct methodology for describing functional and non-functional requirements, e.g. by dealing with external threats

+ Similar to the popular UML notation

Q 7: How to use attack trees to reﬁne attacker goals?

A:

**Objectives**

How can attacks be graphically modelled?

What is the probability of a successful attack?

How can an attack goal be reﬁned by subgoals?

|  |  |
| --- | --- |
| F:\First Semester\Software Security_6 Credit\Exam Preparation\Attack Tree.PNG | F:\First Semester\Software Security_6 Credit\Exam Preparation\Attack Tree _refinement.PNG |

Q 8: Please illustrate misuse cases and attack trees for a given example. How to annotate an attack tree by costs and probabilities?

A:

How to propagate attributes?

1 A parent node evaluates to FEASIBLE if ...

* all its sub-nodes also evaluated to FEASIBLE for an AND relationship or
* one of its sub-nodes is also evaluated to FEASIBLE for a OR relationship

2 The costs of a parent node are ...

* the sum of the costs of all sub-nodes for an AND relationship or
* the costs of the most cost-eﬀective sub-node for a OR relationship

3 The probability of a parent node is ...

* the product of the probabilities of all sub-nodes for an AND relationship or
* the maximum of the probabilities of all sub-nodes for a OR relationship

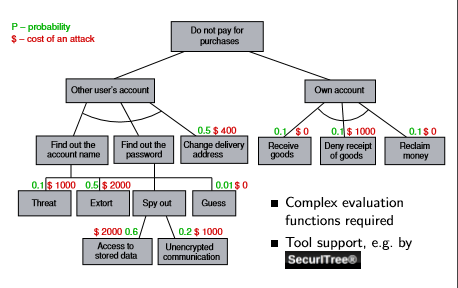
A: How to attribute and propagate the attack trees:

|  |  |
| --- | --- |
| F:\First Semester\Software Security_6 Credit\Exam Preparation\Attack Tree F 1.PNG | F:\First Semester\Software Security_6 Credit\Exam Preparation\Attack Tree F 2.PNG |

A: How to annotate an attack tree by costs and probabilities:

|  |  |
| --- | --- |
| F:\First Semester\Software Security_6 Credit\Exam Preparation\Prob. Tree 1.PNG | F:\First Semester\Software Security_6 Credit\Exam Preparation\Prob. Tree 2.PNG |

A: How to annotate an attack tree by costs and probabilities:



A: illustrate misuse cases

|  |  |
| --- | --- |
| F:\First Semester\Software Security_6 Credit\Exam Preparation\Misuse Case.PNG | F:\First Semester\Software Security_6 Credit\Exam Preparation\Misuse Case 2.PNG |

**Multi-Level Security (5): probably**

Q 1: Which access control strategies are distinguished?

are the core of a security policy model

1. Discretionary Access Control (DAC) - restricting access to objects based on the identity of subjects

|  |  |
| --- | --- |
| F:\First Semester\Software Security_6 Credit\Exam Preparation\DAC.PNG | F:\First Semester\Software Security_6 Credit\Exam Preparation\DAC 1.PNG |

2. Role Based Access Control (RBAC) - access control mechanism deﬁned around roles and privileges

|  |  |
| --- | --- |
| F:\First Semester\Software Security_6 Credit\Exam Preparation\RABC.PNG | F:\First Semester\Software Security_6 Credit\Exam Preparation\RABC 2.PNG |

3. Mandatory Access Control (MAC) - the operating system constrains the ability of a subject to access to an object

|  |  |
| --- | --- |
| F:\First Semester\Software Security_6 Credit\Exam Preparation\MAC.PNG | F:\First Semester\Software Security_6 Credit\Exam Preparation\MAC 2.PNG |

Q 2: Which protection goal is implemented by the Bell-LaPadula model (BLP)?

A:

Bell-LaPadula-Model (BLP, 1969/73)

Ü Protection goal: Conﬁdentiality of data

Q 3: What are the most important rules of BLP?

A:

Ü Rules: No-Read-Up (Simple Security Property) and No-Write-Down (\*-Property)

|  |  |
| --- | --- |
| F:\First Semester\Software Security_6 Credit\Exam Preparation\BLP no read up.PNG | F:\First Semester\Software Security_6 Credit\Exam Preparation\BLP no write down.PNG |

Ü No Read Up: Subjects are not allowed to read an object of a higher security class

Ü No Write Down: Subjects are not allowed to write an object of a lower security class

Q 4: How are security classes represented and how are these classes ordered? What is a sensitivity level and what is a compartment set?

A:

Ü Security classes form a lattice structure and are only partially ordered

... Security classes are represented as pairs (A,C), where

A: Sensitivity level (security label)

C: Compartments (set of security categories)

|  |  |
| --- | --- |
| Sensitivity level  0 unclassiﬁed  1 conﬁdential  2 secret  3 top secret | Compartments (e.g.)  D doctor  N nurse  P patient  A admin staﬀ |

Determinations for BLP

1 Sensitivity levels are totally ordered

top secret(3) > secret(2) > conﬁdential(1) > unclassiﬁed(0)

2 Compartments are sets which can only be partially ordered

{}⊆{D}⊆{D,N}⊆ ...

Q 5: How to make BLP more ﬂexible? What do you know about the high watermark principle?

* Standard software often doesn’t run without errors, because temporary ﬁles created before an upgrade can’t be written afterwards
* WriteUp must be blind, because conﬁrming the successful writing would open an information channel about the state of the higher class
* How to create excerpts [meaning - take (a short extract) from a text] from documents for publication (downgrading)?
* How do you classify certain documents that are only to be protected in combination?
* Conclusion: The assumption of the Tranquility Property is often too strong in practice!

The **high-water mark** for [access control](https://en.wikipedia.org/wiki/Access_control) was introduced by [Clark Weissmann](https://en.wikipedia.org/w/index.php?title=Clark_Weissmann&action=edit&redlink=1) in 1969.[[1]](https://en.wikipedia.org/wiki/High-water_mark_(computer_security)#cite_note-1) It pre-dates the [Bell–LaPadula](https://en.wikipedia.org/wiki/Bell%E2%80%93LaPadula_model) security model, whose first volume appeared in 1972.

Under high-water mark, any object less than the user's security level can be opened, but the object is relabeled to reflect the highest security level currently open, hence the name.

The practical effect of the high-water mark was a gradual movement of all objects towards the highest security level in the system. If user A is writing a CONFIDENTIAL document, and checks the unclassified dictionary, the dictionary becomes CONFIDENTIAL. Then, when user B is writing a SECRET report and checks the spelling of a word, the dictionary becomes SECRET. Finally, if user C is assigned to assemble the daily intelligence briefing at the TOP SECRET level, reference to the dictionary makes the dictionary TOP SECRET, too.

Q 6: What is the diﬀerence between a strong and a week tranquility property?

**Strong**

* Subjects and objects do not change security classes during the lifetime of the system

**Weak**

* Security classes are only modiﬁed in conformance with the speciﬁed security policy model
  + ⇒ High Watermark Principle
* e.g. process starts with low security class and is upgraded when accessing objects of higher security classes
* An upgrade is only possible if the security model is not violated

Q 7: How to bypass BLP with the help of covered channels?

A:

If a Trojan horse enters a high security class level and transmits information to lower security class level by bypassing the protection mechanism, a hidden channel is present

**Such information ﬂows can be implemented via ...**

Ü Storage Channels:

* Process of high security class transmits conﬁdential information by actions with the hard disk drive
* e.g. modify the position of the hard disk’s read head or open, close or lock ﬁles
* Process of low security class is able to monitor these actions

Ü Timing Channels:

* Information transmission by measuring the runtime of processes

Q 8: What are the diﬀerences between the BIBA and BLP model?

**Bell-LaPadula-Model (BLP, 1969/73)**

Ü Protection goal: Conﬁdentiality of data

Ü Rules: No-Read-Up and No-Write-Down

**Biba-Model (1977)**

Ü Protection goal: Integrity of data

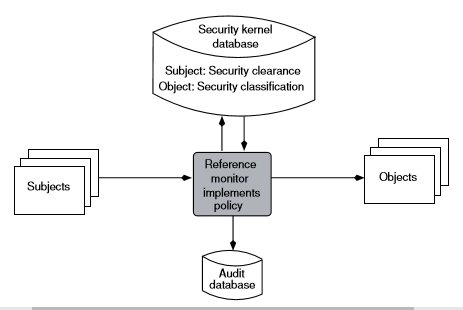
Ü Biba is dual of BLP, i.e.

Rules: No-Read-Down and No-Write-Up

Q 9: What are the design principles of a Trusted Computing Base (TCB)?

A:

* TCB covers all hardware and software components for implementing the security concepts
* Implementable as a reference monitor or as a part of an operating system core



Design Principles

* TCB should be as small as possible and consist of just a few components
* Reference monitor and databases have to be protected from unauthorized access
* Verify the correctness of all TCB components if possible
* Every access to the system must be controlled by the TCB!

**Information Flow Control (6) probably**

Q 1: Why is the generalization of the Bell-LaPadula model by Dorothy Denning useful for information ﬂow control?

Q 2: What do you know about implicit information ﬂows inside of a program and how can we analyse the code using Denning’s operators (maximum and minimum)?

Q 3: Where in a Java program implicit information ﬂows can arise?

Q 4: How to specify a security policy for conﬁdentiality or integrity using JiF?

Q 5: How are implicit information ﬂows handled in JiF?

Q 6: What is the meaning of an empty security label in Jif?

Q 7: What does the JIF compiler check for an assignment?

Q 8: What is problematic with a method call? How the JiF-Compiler is able to solve this problem?

Q 9: What is the meaning of begin and end-labels in JiF? How is it possible to support JiF-refactorings?