EITP20: Secure Systems Engineering — Final Exam Study Questions
Lund University

Karl Hallsby

Last Edited: September 15, 2020

Contents

1 Design Process

1. Which are the steps involved in the overall design process of a secure system?

   Solution:

   1. Threat Analysis
   2. Security Requirements
   3. Security Architectures
   4. Security Design
   5. Security Evaluation

   (a) Describe the relationship between the different steps?

   Solution: In the Threat Analysis, the system is first scoped, to ensure that we limit our possibilities a little bit. Then, we attempt to find potential threats to the scoped system by performing one or more techniques ("Attacker can steal sensitive personal information.").

   Once we find the potential threats, we use those along with any information/requirements given by the client to develop Security Requirements ("We require the use of encryption on personal data.").

   The Security Architecture is the phase where we determine the kind of infrastructure we will need, the communication flows that are possible, find the interactions in the system, etc. This allows us to place our requirements onto the system as it exists already, and allow us to design for the system AND its longevity ("The personal info server is kept physically separate from others, only only communicates over mutually authenticated means using an appropriate cryptographic key.").

   The Security Design is the process of actually choosing the standard, protocols, and hardware used to build the system ("The server will use keys generated by a Hardware Security Module, to ensure randomness.").

   The Security Evaluation is done by going back to the Security Requirements and comparing the solutions you built in the Security Design and flows you developed in the Security Architecture to see if the requirements were satisfied. Other considerations can be put here too, including business ones ("Using a too difficult encryption algorithm will cause slowdowns of data requests/uses of the personal information.").

2. In which way does a use-case description assist in the secure system design process?

   Solution: It gives a general idea of *how* the system will be used. This is important, because you can design a secure system in multiple ways to counter multiple threats. But, given the use-case of the client, there may be restrictions on what you can design for. It will help you derive the Security Requirements for the system, helping you Architect, Design, and Evaluate later in the process.

3. What is a threat analysis?

   Solution: The process of finding potential threats to a system. These can be from inside the client's group, outside, or anywhere else. Using one or more methods, the applier can find current vulnerabilities, and future problems with the system. These are used after being given the use-case description, so that potential attack vectors can be limited to some extent.

4. What is the main purpose of performing a threat analysis?

Solution: It helps find the potential vulnerabilities in a system that **HAVE NOT** been found already. This will give a broader overview of the system, allowing for a more general, and hopefully, more all-encompassing secure system design.

5. Can you give examples of threat analysis approaches?

Solution:

1. Schneier Attack Trees
2. Microsoft's STRIDE Analysis
   (a) Spoofing
   (b) Tampering
   (c) Repudiation
   (d) Information Disclosure
   (e) Denial of Service
   (f) Elevation of Privilege
3. MITRE's TARA Analysis
   (a) Crown Jewel Analysis
   (b) Cyber Risk Remediation Analysis
   (c) Cyber Threat Susceptability Analysis

6. What is the purpose of a security requirements list?

Solution: To ensure that you (the secure system designer) and the stakeholders/clients/users of your system design can agree on what must be done. Agreement here will ensure the system is secure from both perspectives and that all parites are on the same page when it comes to the issue of security.

(a) Can you list input sources for deriving security requirements?

Solution:

- The use-case description.
- The client's system needs.
- The client's limitations on cost and performance.
- The results from the threat analysis.
- Further discussions and interaction between the designer and the client.

(b) Which are the duties of the security engineering in the security requirements gathering process?

Solution:

- To identify the threats and use them to develop the Security Requirements.
- To analyze the Security Requirements and correctly design to handle them.
- To breakdown the Security Requirements into manageable pieces that allow for simpler system design.

7. Can you elaborate on the main differences between security requirements and other system requirements?

> **Solution:** Security Requirements have a mixture of ***BOTH*** functional and non-functional requirements. By having both, it can be dfficult to verify that the original requirements are fulfilled.
>
> Some of these requirements may be ways to "handle" or "work" with the system, such as processes, rather than properties of the system. A common example of this is requiring that people can only access data by giving a password, or correctly setting user permissions.
>
> There is always the possibility that the system may be secure now, but not in the future. This means that the system might not have the expected security properties, even if all the evaluations and tests indicated it did. This means that Security Requirements **MUST** be continually updated.

8. Please list at least three different "types" of security architectures?

> **Solution:** The three that we discussed the most in-class and used in our reports were:
>
> 1. Logical Security Architecture
> 2. Physical Security Architecture
> 3. Security Service Management Architecture

   (a) Explain the main differences between the different listed types?

> **Solution:** Logical Security Architecture is taking the view of the **information to be secured** and designing around that. This involves finding the entities/actors in our system, then identifying the functions that the system must fulfill/perform. Next, we identify security domains/levels in the system that are in play. These determine where we need what kind of security and how much we trust each entity and function in each domain.
>
> Physical Security Architecture is done by representing the **security data model structures** and showing the "builder's" view of the system. Here, we map the security services/functions/entities identified in the logical architecture to physical security mechanisms. Essentially, if we specify that there is supposed to be a unique random number that acts as an ID for something, then we might say that a Hardware Security Module is required to generate those numbers. We do not specify any specific HSM to use, but just state that one must be used to ensure that the numbers are genuinely unique.
>
> Security Service Management Architecture takes a step back from designing the system for the security of the system in mind, and rather consider the maintainability of the system's security aspects. This is technically done at every layer of the SABSA Security Architecture model, but it is also an explicit job. Here, we determine how maintainable a system is given the security decisions made.

9. Which are the main steps preceeding the actual security design step?

> **Solution:** The performance of a Security Architecture, and ensuring that the Architecture is followed when designing the system. Additionally, the Security Requirements make a difference on the choice of standards/protocols/hardware to use in the system.

10. Explain the main design choices that need to be done at the security design process.

Solution:

- To find suitable security "elements" that can implement our chosen Secure Architecture.
- To find suitable security "elements" that can meet both our and the provided Security Requirements.
- To find suitable security "elements" that can fulfill the expected system performance and cost expectations.
- To make design choices that allow us to handle future, currently unknown, security weaknesses.

11. List and explain different type of security evaluations that typically are done "in-house"

Solution: Typically, if there is little threat if the system's security is compromised, any and all testing can be done in-house. However, if the system contains more valuable information, or has higher risks of being compromised and may be dangerous, typically external experts are used.

12. List and explain different type of security evaluations that typically are done by external experts.

Solution: Pen-tests, Protocol Analyses, Common Criteria Evaluations, etc. can all be done by external experts. In addition, the external experts will likely have certifications in various aspects of secure system design.

They can Attack (Red), Build (Yellow), or Defend (Blue) from attacks and designs to attempt to subvert the system.

13. What is a pen test and what is the purpose of such test?

Solution: To deliberately attempt to attack and compromise the system. This helps find security vulnerabilities that may have been missed, or that the system doesn't counter against. It is also a way to test the recovery infrastructure, and test the people that are tasked with working with the system and ensure that they know what to do.

14. What is a protocol analysis tool and what is the purpose of using such tool?

Solution: To see if the communciation flows used in the system are actually secure. These can be used to ensure that when entities are communicating,t hey will actually be doing so in a safe and secure manner. This also helps formally prove the security of a system.

15. What is the Common Criteria (CC) standard?
    (a) List the 7 evaluation levels defined in CC and explain the main differences between the levels?

Solution:

EAL 1 Functionally Tested
EAL 2 Structurally Tested

EAL 3 Methodically Tested and Checked
EAL 4 Methodically Designed, Tested, and Reviewed
EAL 5 Semi-Formally Designed and Tested
EAL 6 Semi-Formally Verified, Designed, and Tested
EAL 7 Formally Verified, Designed, and Tested

(b) List the three different system documents part of a CC and describe their main purpose.

Solution:

1. Protection Profile (PP)
   - Security needs for a particular class of devices.
   - The product that is being designed for will use one or more Protection Profiles to evaluate against.
2. Security Target (ST)
   - The specification of the actual security properties that we will be evaluating.
   - This is usually published to give a clear understanding of the scope of the evaluation.
3. Security Functional Requirements (SFR)
   - Specification of the security functions of the target product that is provided for individuals

16. What is CISSP?

Solution: CISSP is the Certified Information Systems Security Professional certification and certification program. This essentially says that you are skilled in designing secure information systems.

2 Threat Analysis and Security Requirements

1. List three different typical security threats to an IT system?

Solution:

1. Network-based threats
2. Physical threats
3. Software vulnerabilities

2. What is the first step in an attack tree threat analysis process?

Solution: There is a "Step Zero" that stipulates you have a *good* system description.

The first real step is to identify potential goals that the attacker could have when attacking your system as it currently exists.

3. Make an attack tree based analysis of a BankID system.

Solution: There is no single solution for this, everyone's will be different.

For the Non-Swedes reading this, BankID is a way of verifying banking instructions and transactions through your phone as a second factor of authentication. To use it, you initiate some banking operation, which must be signed. Then, your phone will open the BankID app and ask you to input a secure 6-digit (minimum) code to "Digitally Sign" the transaction. Once done, the app closes and the trasaction is recorded and completed at the appropriate time.

4. List three well established threat assessment methodologies

Solution:

1. Shneier Attack Trees
2. Microsoft's STRIDE Analysis
3. MITRE's TARA Analysis

5. Spell out the acronym STRIDE

Solution:

S Spoofing
T Tampering
R Repudiation
I Information Disclosure
D Denial of Service
E Elevation of Privilege

(a) Explain the meaning of the six different concepts in STRIDE

Solution:

Spoofing Pretending to be someone or something you are not. Getting the correct service to communicate with someone it thinks is also the correct thing, when it really isn't.
Tampering Modifying something (file, config, etc.) somewhere (disk, network, memory, etc.).
Repudiation Claiming you did/didn't do something when you didn't/did. Essentially, the system claims you did something that you didn't do (can be honest or dishonest). The real question to answer here is what evidence do we have to trace this?
Information Disclosure Providing information to someone **NOT** authorized to view it.
Denial of Service Absorbing resources required for regular system function for malicious reasons.
Elevation of Privilege Allowing someone to do something they should **NOT** be allowed to do.

(b) Give examples of attacks for the six different concepts in STRIDE

Solution:

Spoofing Man-in-the-Middle. Impersonating the intended recipient.

Tampering Changing an Excel Spreadsheet. Changing a configuration file somewhere to change the functions of the system.
Repudiation Writing a file, then deleting the kernel log about the file requests.
Information Disclosure Forwarding an email to someone that should not be able to see the email.
Denial of Service Using a botnet to deny network service to some online provider.
Elevation of Privilege You are supposed to be a regular user, but you need to elevate to administrator to run your usual programs, and IT allows people to do so.

6. Which are the three basic steps in STRIDE?

Solution:

1. Identify the main entities/actors in the system.
   - These are the people and computers that interact with the system.
2. Identify the main entities' interactions.
   - How do the people interact with the computers in the system?
   - How do the computers interact with other computers in the system?
3. For each entity, perform a STRIDE analysis on each of the following items:

   Process A Process is a program or set of programs that achieve or do something.
   External Entity An External Entity is one that is not **EXPLICITLY** part of **YOUR** system design.
   Data Flow A Data Flow is how information can be moved through the system.
   Data Store A Data Store is how information can be stored throughout the system.

7. Which are the two main activities in a MITRE TARA security analysis

Solution: There is a zeroth step, the Crown Jewel Analysis for the identification of things in the system that are important to protect.

1. Cyber Threat Susceptibility Analysis (CTSA)
2. Cyber Risk Remediation Analysis (CRRA)

(a) Which are the main input sources to these two analysis activities?

Solution: The question asks for 2, but I will give 3.

1. Common Attack Pattern Enumeration and Classification (CAPEC)
2. Common Weakness Enumeration (CWE)
3. Common Vulnerabilities and Exposures (CVE)

(b) The output of these two activities are stored in special databases. What is the name of these two databases?

Solution:

1. Various vulnerability and exploit databases that detail various attacks and how they can/may be executed. These are known as **Attack TTP Catalogs**.
2. Countermeasure databases that explain how to mitigate the vulnerabilities and exploits that are found in the previous databases. These are known as **Countermeasure Catalogs**.

8. Describe briefly the different steps performed during a TARA CTSA.

Solution:

1. Establish the Assessment Scope.
   - Typically this is partly done during the Crown Jewel Analysis.
   - This is used to make sure you don't evaluate parts of the existing or new system that you are not concerned aobut.
2. Identify the candidate Tactics, Techniques, and Procedures (TTP).
   - Here, you collect ***ALL*** potential TTPs that could affect your properly scoped system.
3. Eliminate Implausible TTPs.
   - If you perform Step 2 correctly, you will have WAY too many TTPs to mitigate for you system to ever work.
   - You must use your best judgment to remove the ones that are implausible to occur in your system.
4. Apply some scoring model.
   - MITRE has a scoring model provided that can be quite useful.
   - These will be the scores that you use in the Threat Matrix to find the best TTPs to design against.
5. Construct a Threat Matrix.
   - The scores used in the Threat Matrix are based off the scoring model that is created in the previous step.
   - This will show you the which TTPs that remain are the most important to design against.

9. Spell out the acronyms CAPEC, CWE and CVE.

Solution:

C Common
A Attack
P Pattern
E Enumeration
C Classification

C Common
W Weakness
E Enumeration

C Common
V Vulnerabilities
E Exposures

(a) What does CAPEC contain and how it is used in a TARA analysis?

Solution: CAPEC contains general attack patterns and threats (TTPs) to system that are broken down into a family hierarchy. This also contains the potential threat of a TTP and how difficult it is to mitigate. It is used to find TTPs, their potential score, and to help construct the Threat Matrix.

(b) What does CWE contain and how it is used in a TARA analysis?

Solution: CWE describes software weaknesses. It is best used when finishing the choice of software to implement.

(c) What does CVE contain and how it is used in a TARA analysis?

Solution: CVE contains known software vulnerabilities. It is best used when the software implementation decisions have been made.

10. Describe briefly the different steps performed during a TARA CRRA.

Solution: There is the obvious requirement that the TARA CTSA step has been performed.

1. Select which TTPs to mitigate.
   - You may not be able to mitigate **ALL** TTPs.
   - For example, you can attempt to mitigate Social Engineering attacks, but these are ***NOTORIOUSLY, INCREDIBLY*** difficult to protect against.
2. Identify plausible countermeasures.
   - There may be several possible countermeasures for your particular TTP.
   - You must use your best judgment to ensure you counter as much of the TTP's threat as possible.
3. Assess countermeasure merits.
   - You can perform a Cost-Benefit Analysis here to mathematically determine the best countermeasures.
   - You will get a *Utility* score that estimates how beneficial your countermeasure is and how effectively it counters the threat the TTP provides.
   - You will also get a *Cost* score that estimates how costly your countermeasure is to implement and let run.
   - To find the "best bang for your buck" solution, you must take *Utility*/*Cost*.
4. Identify "Optimal" Countermeasure Solution
   - This is what you think would be the best set of countermeasures to implement to secure the system.
5. Prepare recommendations.
   - These will be used in conjunction with original project description to build your Security Requirements.

11. Where can one find TTP mitigation solutions?

Solution: There are a large variety of databases available, but we used the CAPEC database throughout the course to find countermeasures.

12. Which are the four different mitigation types?

Solution:

1. Detect: **FIND** the problem that is occurring.
2. Limit: **LIMIT** the potential damage a harmful action can do.
3. Neutralize: The harmful action has already begun, but not yet finished. How do you **DEAL** with it?
4. Recover: The harmful action may have finished, or you may have interrupted it. How do you **COME BACK** from this harmful action, to make it seem like it never happened?

13. Which are the steps used to obtain a final ranking table for countermeasures?

Solution: The main step is to perform a Cost-Benefit analysis. Here you identify and create a score of benefits and potential costs each countermeasure has, and find countermeasures that maximize the benefits, while minimizing the costs.

14. How do one select final TARA recommendations based on a countermeasure ranking table?

Solution: Typically, you would select TARA countermeasures that have as high a U/C ratio as possible, so they maximize utility, while minimizing cost. Although, these can also be somewhat arbitrarily chosen if there are specific items that you must counter.

15. Which are the three mandatory parts of a TARA TTP recommendation?

Solution:

1. The Action/Device/Procedure/Technique that is recommended. Which countermeasure(s) should be applied?
2. The reason why that particular countermeasure is required. Which TTP(s) does this countermeasure mitigate?
3. The implication/effect if the countermeasure is *not* applied. What is the potential impact(s) to the system's/mission's capabilities resulting from the compromise of a cyber asset?

16. Which are the different input sources to the security requirements derivation process?

Solution:

1. Client

- Use-case
- Needs
- Costs
- Performance
- Other Business considerations

2. Engineer (Us)
   - Threats
   - Analysis
   - Breakdowns
   - Discussions with client

17. Which are the main outputs from the attack tree, the STRIDE and the TARA process respectively which are used to derive high-level security requirements?

Solution:

- Attack Tree:
  - **Methods of preventing** the potential attacks identified, through the trees, that can be used to compromise the system.
- STRIDE:
  - Key system elements.
  - Potential methods of compromising each letter of STRIDE. Each of the letters gets its own table that gives potential attacks and mitigations that must be provided.
- TARA:
  - TTPs from CTSA
  - The CTSA Threat Matrix
  - Countermeasures from CRRA
  - The CRRA U/C matrix

18. Give example of high level security requirements for a BankID system.

Solution: The BankID discussed here is the same as the one discussed above.
There is **no** general solution for this. Everyone's will be different.

19. Give example of low level security requirements for a BankID system.

Solution: The BankID discussed here is the same as the one discussed above.
There is **no** general solution for this. Everyone's will be different.

20. Describe a four step approach for security requirements identification and documentation.

Solution:

1. Identify the generic security requirements.
2. Review the previously identified requirements against business, usability, performance, cost, and other requirements. Revise if any requirement breaks something from the client's requirements.
3. Map generic security requirements onto system components. Typically, this is already partly done during the Threat Analysis. Now, it is transformed into a *Language of Requirement*, rather than Language of Threat.
4. Break down the component requirements into more specific design/implementation-specific requirements.

3 Security Architectures

1. Describe what constitutes a security architecture and give some examples.

Solution: A Security Architecture has graphical and textual representations of a security system. It includes relations, trust levels, trust relationships, and interfaces between different parts of the system. It also has security and system boundaries to delimit different parts of the system that have various properties.

2. The Sherwood Applied Business Security Architecture (SABSA) consist of 5 layers and one cross layer.
   (a) Describe the different layer views and list the names of the different layers.

Solution:

1. Contextual Security Architecture is the view the business has of the system. It is quite similar to the Security Requirements.
2. Conceptual Security Architecture is the highest-level view that an engineer can have of the system. This includes the general breakdown into secure portions of a system.
3. Logical Security Architecture is the next level, that deals with data and its logical flows. Here, the way that data and its logical propagation are safeguarded is first developed.
4. Physical Security Architecture is the 3rd design level. Here, the specific high-level security requirements and previous security architecture levels are combined to choose what needs to be done to secure the data.
5. Component Security Architecture is the lowest level of the SABSA Security Architecture. Here, the actual protocols, standards, and hardware are chosen.
6. Security Service Management Architecture is the "cross layer". It is concerned with how to maintain the system and its security.

   (b) Give examples of questions the different SABSA architecture views are supposed to answer.

Solution:

1. Contextual Security Architecture
   - What does the business need from the system?
   - Why do we need to mitigate against these risks and threats?

- How do we protect the processes in this system?
- Who are going to be the ones using the system?
- Where is this system going to be geographically located and where is this product going to be used?
- When does the client require this system and for how long?

2. Conceptual Security Architecture
   - What does the client need to have protected?
   - Why are these risks that need to be mitigated and do these assets need protection?
   - How do we provide protection, in very high-level technical and maanagement security terms/strategies?
   - Who are the people/organizations involved in the security management and the assumed trust relationships?
   - Where is protection needed in terms of security domains?
   - When is the relevant time scope(s) of the system's protection?
3. Logical Security Architecture
   - What is the actual information being secured?
   - Why shall this security policy be applied to the system?
   - How are the actual security services in the system put together?
   - Who are the entities in the system and how can they interact?
   - Where are the security domains and the relationships between the domains?
   - When is the security processing cycle?
4. Physical Security Architecture
   - What is the data model and security-related data structures?
   - Why are these the rules that drive logical decisions in the system?
   - How do these security mechanisms work to provice security and what physical machines or modules are needed?
   - Who are the users, the applications they use, and the security interface?
   - Where is the required security infrastructure required to provide the security?
   - When is the dependency in the system present in the form of execution control structures?
5. Component Security Architecture
   - What are the data field specifications, address specifications, etc.
   - Why are we using these security standards and best practices?
   - How are the entities/modules, tools, etc. used and put together?
   - Who are the users' identities, privileges, functions, actions, Access Control Lists?
   - Where are the computing processes, nodes addresses, and inter-process protocols?
   - When are the security step timings and sequencing?
6. Security Service Management Architecture
   - What is the operational continuity and information processing of the system?
   - Why do we need to minimize operational risks and mitigate failures/disruptions?
   - How do we perform these specialized security-related operations?
   - Who do we support? All users and their applications?
   - Where do we perform maintainance from and on?
   - When do we schedule things and execute our time-table of security-related operations?

3. Which are the three different types of security services in a logical security architecture?

Solution:

1. Prevention services
2. Detection, Notification, Assurance, and Event Collection services
3. Recovery and Restoration services

(a) List and explain examples of services part of the non-prevention type of security services.

Solution:

- Detection, Notification, Assurance, and Event Collection services
  - Log review
  - Message integrity verification
  - Security monitoring
  - Audit trails
  - Security Training/awareness
  - Intrusion Detection
- Recovery and Restoration services
  - Incident response
  - Data replication
  - Data backup
  - Disaster recovery
  - Crisis management

4. Describe each of the different prevention security services in more details.

(a) Give example of at least four different Entity Security Services and how they contribute to security prevention.

Solution:

1. Unique Entity naming
2. Entity registration
3. Entity credentials certifications
4. Directory services
5. Entity authorization
6. User authentication
7. Device authentication

(b) Give example of at least four different Communication Security Services and how they contribute to security prevention.

Solution:

1. Session authentication
2. Message origin authentication
3. Non-repudiation
4. Message reply protection
5. Traffic flow confidentiality
6. Security administration

7. User support
8. Physical security services
9. Environmental security services

(c) Give example of at least four different Application-level Security Services and how they contribute to security prevention.

Solution:

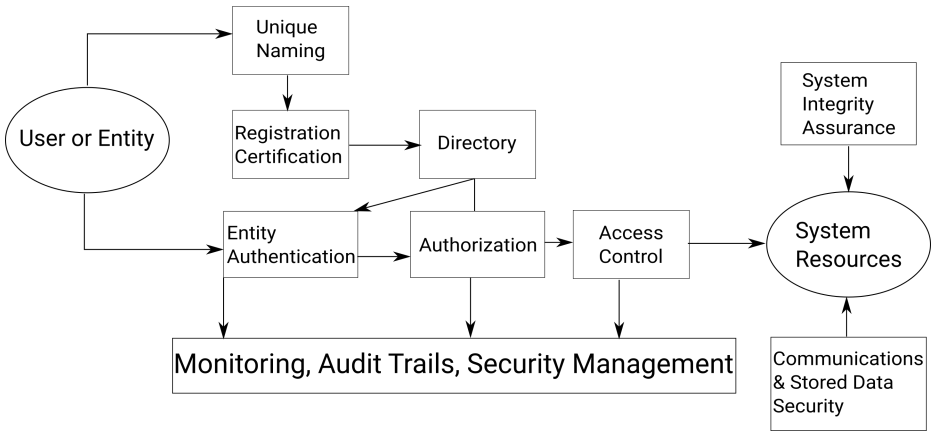
1. Entity authorization
2. Logical access control
3. Audit trails
4. Stored data integrity protection
5. Stored data confidentiality
6. Software integrity protection
7. Software licensing management
8. System config protection
9. Data replication
10. Data backups
11. Software replication
12. Software backups
13. Trusted time
14. Secure user interface

(d) Give example of at least four Security Management Services and how they contribute to security prevention.

Solution:

1. Security policy management
2. Security training and awareness
3. Security ops management
4. Security provisioning
5. Security monitoring
6. Security measurements and metrics
7. Security administration
8. User support
9. Physical security devices
10. Environmental security services

5. Draw a picture showing the relations between major different security services from a systems point of view.



Solution:

6. A logical security architecture can be created using a six steps methodology:
   (a) Describe each of the different steps

   Solution:

   1. Identify the main entities. Here, the users and the physical components in the system(s) are identified. Who is doing something and what is executing the user's actions?
   2. Identify the major functions in the system. Each computing entity has some function(s) that it needs to fulfill. What does each hardware entity need to do?
   3. Identify the security domains and their associations. Each entity and function exists within a security domain, i.e. how much they are trusted. In addition, how do these security domains interact with each other?
   4. Identify security services already meeting the Security Requirements. There may be some properties of the system that already meets the requirements as laid out in the Security Requirements. Identify these. There may also be security services that are required to meed the Security Requirements, identify these too.
   5. Map security services to the functional entities. Take the security services from the previous step, and match them together with the entities in the system.
   6. Create a graphical view of the logical architecture. It is easier to see how these security services get mapped to the entities and security domains if it is done in a graphical view and their interactions are shown. Each computing entity should be a box, each flow of communication an arrow, each user shown, and the computers/users should be grouped together into their security domains.

   (b) What is the end result?

   Solution: A general logical view of the system to see what data needs to be secured and how that should be done. This will also specify the security services that should be used, the security domains, the users, the computing entities, among others.

   (c) Give an example of a logical security architecture.

   Solution: See Slides 25–31 of Lecture 3.

7. What is a Physical Security Architecture?

Solution: A Physical Security Architecture is the general specification of how to implement the security given by the Logical Security Architecture. The expected outcome is a specification of the security solutions offered in the system, including the platform, software, and hardware security functions.

8. A physical security architecture when using the SABSA includes making a mapping to physical security mechanisms.

   (a) Describe what is meant by a "Naming and Registration" mechanism and give examples.

   Solution: The Naming and Registration mechanism is concerned with how we identify the entities within our system. Additionally, when they have been given an identifier to recognize by, how we can register them with a system to ensure that when an entity claims to be something with a certain name, we can verify that.

   This typically includes:

   - Unique Naming
     - Using a common naming standard.
     - Using a common naming procedure.
     - Using a directory system to separate different names.
   - Entity Registration
     - Using a standard registration policy.
     - Having a standard registration procedure.
     - Using a registration authority system.
   - Public Key Certification
   - Directory Services
     - Having a directory system.
     - Having a directory access protocol.
     - Having a directory object and attribute syntax rules.
     - Allowing for directory replication.

   (b) Describe what is meant by a "Storage and Runtime" mechanism and give examples.

   Solution: This is concerned with how we store data securely and use it securely during runtime.

   - Data confidentiality
   - Data integrity
   - Software integrity
   - Secure Execution
     - Secure hardware modules
     - Secure execution enclaves
     - Secure virtual machines
     - Protected execution
   - Software licensing Protection
   - Data and software replication

   (c) Describe what is meant by a "Physical Security" mechanism and give examples.

Solution: This is concerned with the physical security of a location.

- Physical Security
  - Locks on doors
  - Locked server rooms
  - Physical protection of cables
  - Personnel identification
- Environmental Security
- Personnel Security
  - Background checks
  - Vetting procedures
  - Training Courses

(d) Describe what is meant by an "Authentication and Session" protection mechanism and give examples.

Solution: When a registered name wants to access our system, how do we ensure that they are who they say they are and make sure that they have not been compromised?

- Entity Authentication
  - Login procedure
  - User passwords/tokens
  - Authentication protocol
- Session Authentication
- Message Integrity
  - Message Authentication Codes (MACs)
  - Hashing
- Message Confidentiality
- Message Reply Protection
- Non-repudiation

(e) Describe what is meant by a "User Interface and Naming" mechanism and give examples.

Solution: Any time a user is interfacing with the system, that is a major vulnerability. The user has the ability to doanything they want there, and they may do something that compromises the security of the system. So, we need to minimize the number of actions a user can perform on a system to ensure they don't do something that harms the system.

- User Interface (UI) Security
  - GUI login
  - GUI Security messages
  - Single Sign-On
- Service Management
- Operations Management
- Secure Provisioning and Administration

(f) Describe what is meant by a "Authorization and Access Control" mechanism and give examples.

Solution: If an entity is registered, authenticated, and shown to be secure, then what actions do we let them perform, and how can they do them?

- Authorization
  - Roles
  - Attributes
- Audit Trail
  - Event logs
  - Logging integrity protection
  - Log browsing
  - Log analysis tools
- Logical Access Control
  - Access Control Lists (ACLs)
  - Central Access Manager (CAM)
  - Access Control Agents
  - Database Management System
  - Filesystem management

(g) Describe what is meant by a "Monitoring and Incident" mechanism and give examples.

Solution: When something **does** go wrong, how can we make sure we find it and know about it?

- Secure Monitoring
  - User activity logs
  - Application event logs
  - Operator activity logs
  - Management event logs
  - Event log browsing
  - Event log analysis
  - Reporting
- Security Measurements and Metrics
  - Cryptographic tests
  - Inspection tools
  - Penetration Testing
  - Statistical Tests
- Alarm Management
- Intrusion Detection
- Incident Response
- Disaster Recovery
  - Data and software backups
  - Data restoration procedures
  - Hardware redundancy
  - Recovery plan
  - Recovery procedure

9. What must in addition to the security mechanisms be specified in the physical security architecture?

Solution: The security of the network and the system's platforms must be considered as well.

10. Give examples of a platform security solution that can be used to build solutions meeting a logical security service and can be used to protect the chosen physical security mechanism?

Solution: If a logical security service requires secure, unique cryptographic keys be generated, then a platform security solution would be to use a Hardware Security Module to generate them.

If a logical security service requires something be executed in a hardware-based secure execution environment, then the processor's built-in secure execution environment can be used.

11. For the SSO logical security architecture given at the lecture, perform the following:
    (a) Identify the main physical security mechanisms needed in the corresponding physical security architecture.

Solution:

- Each user must have a unique naming, likely drawn from their legal name.
- Each user must have a device(s) registered to them, which must be tracked.
- Each device will have keys to be able to use the system.
- The user has a password to be able to user the Single-Sign On service.
- The OIDC client must perform an authentication action on the user.
- The OIDC client must then confirm the authenticity of the user with the OIDC provider.
- This must them go to the Hosting Servers to perform the actual computation.
- Each of these servers must have replicas ready to take over.
- Each of these servers must also have data backups, in case of failure.
- They must be stored somewhere separate from the physical location of the user, likely close to the services that the SSO service provides.
- The physical security of this location must be tested.
- The UI for signing in should be as simple as possible.
- Once the user signs in, they should not have to do it again. If the system requires them to "sign in" again, then there needs to be a button for SSO.
- All login attempts should be logged, with user- and device-identifying information recorded too.
- There needs to be monitoring to ensure inappropriate IP addresses are banned (Users obviously not from your group).

    (b) Identify the main platform security components needed to fulfill the architecture

Solution:

- The system must have a user database somewhere.
- The user's passwords are only stored in their final hashed value.
- The servers and databases must have replicas for failover, and backups for complete failure.
- Any actions these servers take must be logged with entity-identifying information.
- Unlikely we need HSMs to generate keys, use built-in cryptographic functions on devices.
- Run the minimum of services required for functioning.
- The servers that run identification should likely run in a hardware-based secure execution environment.

- Each of these servers could be VMs as well, to allow for on-demand scaling and to prevent one issue from propagating.

i. Suggest concrete platform security mechanisms to use for the physical realization.

4 Security Design

1. Can you list four different principles upon which a security design should be based?

Solution:

1. Use well-proven techniques, standards, libraries, hardware modules, and solutions.
2. Keep system complexity as low as possible.
3. Minimize the utilization of trusted components.
4. Only develop your own solutions when really needed.
5. Use open designs whenever possible.

(a) Give a motivation for each of the listed design principles

Solution:

1. Well-proven solutions are constantly tested, so they have more verification of their security.
2. By reducing the number of things that can go wrong, the system can be more secure. The reduced number of vulnerabilities means we can focus more on the ones that remain.
3. The greater the reliance on trusted components, the more trust that must be placed in the hardware of the system.
4. Developing novel and unique solutions requires a lot of design and engineering time. Additionally, if they are not well-designed, they can cause more problems than they solve.
5. By using open designs, there is a lower risk of licensing issues.

2. Describe the process steps to perform when going from a security architecture to a design specification.

Solution:

1. Start with the Security Architecture.
2. Map the desired security services to security standards.
3. Identify gaps that are present in the System Design/Architecture.
    - Complement the existing Security Design/Architecture with new Designs if needed.
4. Perform a Security Evaluation.
5. Document the Security Design.
6. Identify implementation efforts (missing hardware/software).

3. What is typically the role of unique identities in security systems?

Solution: To create a unique mapping from name to key, allowing for reliable authentication. This unique name can be used for discrete access control and authorization decisions in the system.

(a) Give example of three different widely used identity types?

Solution:

1. ITU
2. IEEE's MAC Addresses
3. IETF and URIs

4. What is the problem from privacy perspective with using fixed identities?

Solution: If the identity has been compromised (on the back-end), it can be traced back to a user/device.

(a) Describe two different methods for avoiding the identity privacy problem in a system design

Solution:

1. Randomly Generated Pseudonym. There is a middle-man that receives the unique identifier and assigns a random value to that, which is changed each time, and allows for identification of that user/device for that session.
2. Encrypted Identity that is not visible outside trusted container boundaries.

5. What is a digital certificate and how it is used?

Solution: A digital certificate is a way to securely distribute cryptographic keys while ensuring that they come from the correct person.

(a) Which are the most important data fields in an X.509 certificate?

Solution:

- The algorithm used in the signature
- The Certificate Issuer's name
- The period of validity
- The Receipient's public key information

6. Which is the most used authentication principle over HTTP?

Solution: The most common priniciple is the Challenge principle. The only way that the correct user can authenticate with the service is by solving a challenge only the real user should know.

(a) Describe the different steps in an HTTP basic authentication.

Solution:

1. The client performs a `GET` request.
2. The server responds with `Error 401: Unauthorized`, which asks the user to input information.
3. The client returns their credentials.
4. The server checks their credentials.
5. The server either returns `Error 200: OK` or `Error 403: Forbidden`.

(b) Under which circumstances can basic authentication be used?

Solution: Basic authentication can ***ONLY*** be used over already-secured TLS channels.

(c) Typically, how is basic authentication treated at the server side and what is the main reason for using this type of storage?

Solution: The main reason is that it is fairly quick to perform and easy to implement. The user inputs their username and password, which is securely transformed, and the server stores the hashed version of the password and a randomly generated number as salt. The database can combine the password and salt, hashing them, and checking in the user database, to verify the password.

7. Describe the principles behind hardware token-based authentication.

Solution: By giving ***ONLY*** the correct user a hardware token, only they can correctly give the information required by the system and provided by the token. This is combined with a simpler end-user local authentication procedure. This is also quite user-friendly, as the token should always be with the user too.

This is typically categorized as a 2nd-factor authentication step.

8. What is the rationale behind two factor authentication?

Solution: By providing only the correct user with the correct information, even if one form of authentication is compromised, the attacker must also compromise another. This second compromise should be a different attack vector, reducing the likelihood that one attack compromises both at the same time.

9. What are the main differences between:

(a) TLS server authentication?

Solution: In this implementation, the hosting server is the one that performs the authentication. The end-user device only provides credentials in a secure form that the server then verifies. This is done at before every session.

(b) TLS client certificate authentication?

Solution: Here, a Public Key Infrastructure must be setup and maintained. This has the benefit that the end-user does **not** have provide credentials. However, the PKI must be maintained and customized, which is quite complex, expensive, and difficult on the client-side.

(c) TLS pre-share key authentication?

Solution: The server and client pre-share a key, then perform a challenge-based authentication to ensure that each other say are who they say they are.

10. Explain the main principle behind an object security scheme.

Solution: Instead of ensuring that the communication session between the client and server is secure, we instead secure the things sent between them. This has the benefit of securing the data throughout the entire transport process, including intermediate buffers and storage.

(a) How does it differ from a session protection scheme like TLS or IPsec?

Solution: It is more difficult to verify who is sending the information to whom, but the information is still secure. MACs and hashes do not provide enough power for non-repudiation. MACs and hashes **must** be combined with Digital Signatures and secure logs to create non-repudiation.
There is also different types of packets that can be sent to setup the object security scheme.

11. What does RBAC and ABAC stand for with respect to access control systems?

Solution:

RBAC Role-Baesd Access Control

- Users are grouped into roles.
- These roles have certain permissions, and can perform their allowed actions only in selected places.
- Roles can have a hierarchical structure, allowing for inheritance schemes to be used.
- Removing access to information can be done on a per-user basis after this too.

ABAC Attribute-Based Access Control

- Users are given attributes about themselves.
- An Access Control Policy is setup with rules about what attributes can do what.
- Each object being stored has attributes about it, describing what it is, what can be done with it, etc.
- The current environment conditions are also considered, the time for instance.
- These are all combined with into a logical Access Control Mechanism, which grants access/permission to files based on the attributes and rules of a system.

(a) Explain the main differences between RBAC and ABAC.

Solution: RBAC groups users into Roles, which are then given permissions as a whole.
ABAC keeps users separate, but gives each attributes. Each object is also given a set of attributes. A set of rules is created for each user attribute and object attribute to allow access. These are then combined with the current environment's conditions to make an access decision.

12. What is the purpose with an access token?

Solution: These tokens contain information about the authorization decisions made by the access controller for the user.

(a) What does a SAML assertion contain?

Solution:

1. AssertionID
2. IssueInstant
3. Issuer
4. MajorVersion
5. MinorVersion
6. Conditions
   - NotBefore
   - NotOnOrAfter
   - Audience Restriction
7. Authentication Statement
   - Subject (User's Identity)
   - Authentication Instant
   - Authentication Mechanism
8. Attribute Statement
   - Subject
   - Asserted Attributes
9. Digital Signature

13. How can a Hardware Security Module (HSM) assist in protection of cloud data storage?

Solution: The HSM will store the actual secret key, that it generated. It also encrypts this key, which is then returned to the user, and contains an index to the actual secret key in the HSM.
This allows the customer to authenticate against the HSM, then perform actions using the information the HSM returns.

14. List a couple of widely used commercial server anti-virus tools.

Solution: These are mostly used on Microsoft Servers.

- BitDefender

- Comodo
- Avira
- Kaspersky Labs' tool

15. How can the security of a Docker container be enhanced beyond using default configurations?

Solution: It can be enhanced with DockerSec's AppArmor profile, which creates a set of path rules about what the process can and cannot do.

16. How can a Web design be made to make "clickjacking" less likely?

Solution: Clickjacking is the process of making buttons do things other than what they are supposed to do, or to look different than what they are supposed to. Preventing resources embedded in the page from changing must be ensured.

17. What is the main difference between key provisioning of a public key system compare to a symmetric key-based systems?

Solution: The biggest difference is the number of keys required to be distributed in a PKI. There are many more keys in-use at any given time.

Symmetric keys also compromise 2 or more hosts at once, since both sides use the same key for encryption and decryption. Thus, they must be protected more.

(a) What are the key issues to consider when making a public key issuing design?

Solution: Ensuring that the distribution of the public key does not reduce the security of the private key too much. The public key must also be identifieable to an entity that is known in the system.

(b) Which are the key issues to consider when making a symmetric key issuing design?

Solution: A symmetric key system allows the same key to be used for encryption and decryption. Thus, if a single key is compromised, 2 or more hosts will also be compromised. Thus, the security of the key must be much higher than others.

18. List the three main different intrusion detection principles and explain how they work on high level.

Solution:

1. Signature-Based Detection work by finding signatures of breaches or malicious software. Every piece of software has some unique properties to it, and those can be identifiable.
2. Statistical Anomaly-Based Detection works by representing normal data flows as a statistical model. If this normal model does not align with the current statistical properties of the system, then it must be compromised.

3. Stateful Protocol Analysis Detection works by having the machine be in one of several states. If the machine does not move to the next state correctly, come from the correct state, etc., it will classify it as a compromised system.

19. What is a syslog and how is it typically used?

Solution: A syslog, or System Log is a standardized way to collect information about events and actions taken. It is important to note that a system log format ***does not*** include any security mechanisms.

This is specified by NIST SP 800-92.

20. What is the main risks with a debug interface (like JTAG) and how should it be treated in a product design to avoid these risks?

Solution: The debug interface provides more detailed system information than is actually needed to run the application, because it is meant to find issues in the program. Since more information is presented by the JTAG interface, it is possible to learn information that the attacker should not typically be able to access. Additionally, if the debugging interface allows for changes to be made in place, then the execution of the program could be altered.

21. Which are the three different most severe attacks threats against smart card designs and which are the typical countermeasures?

Solution: Attacks:

1. Side-Channel Analysis.
2. Invasive Attacks.
3. Advanced Fault Exploitation.

Countermeasures:

1. Adding noise to the system.
2. Actively shielding electronic components.
3. Step counters, code integrity verification, etc.

22. Give three examples of widely used NIST security standards and what they specify?

Solution:

1. AES for encryption
2. SHA3 hash function
3. HMAC for computing Message Authentication Codes

23. What does IETF stand for?

Solution:

I Internet
E Engineering
T Task
F Force

(a) Give example of two well-knows IETF security standards and explain what they specify?

Solution:

1. TLS specifies a way to secure the communications used between a client's web browser and the connecting webserver. It has session authentication and allows for the secure transmission of communications.
2. SSH specifies the protocol used to open a secure remote shell session on a computer. It specifies the types of identify verification that can be used, how the challenge-based protocol works, etc.

24. Which type of security standards are done by IEEE and 3GPP respectively?

Solution: IEEE and 3GPP both create wireless communication standards. IEEE has created and maintains the 802.11 Data-Link Layer protocols for implementing WiFi, along with others. They have also created many other standards that are used today.

3GPP created the 3G, 4G, and 5G protocols for wireless cellular communications.

25. What is the difference between public industry bodies and industry standards?

Solution: Industry standards are practices, processes, and protocols that are already in use by industry. Public industry bodies are groups, that may contain industrial groups that help create industry standards.

26. What is meant by a cancellable biometric protection scheme?

Solution: The person's stored biometric data can be securely deleted at any time they wish.

(a) Describe how to achieve a cancellable biometric matching system

Solution: This is going to be different for everyone. However, it will likely include a way to verify the person that wishes to cancel their stored biometric data. It will also need to specify a way to securely remove the stored data from everywhere, which may need to be done in a certain timeframe.

(b) Which alternative biometrics protection approaches can be used?

Solution: This is more of a research question, so there will be a variety of solutions, depending on what you find.

5 Security Evaluation

1. What is the purpose with a security review and when should it be performed?

   Solution: A security review is intended to perform a general "sanity" check that the chosen Security Architecture makes sense. It also helps identify potential gaps in the Security Architecture that might have been missed. It can also provide a business evaluation of the chosen architecture to find business consequences. Lastly, we take the perspective of an attacker attempting to break into the system.

   Generally, this is done after the Architecture and Design is completed, but performing it regularly during these steps also works.

2. At which four main levels do you typically perform a security evaluation?

   Solution:

   1. Security Architecture Review
   2. Design against Requirements Review
   3. Design Review
   4. Implementation Review

   (a) At which occasion should they be done?

   Solution:

   1. After the Security Architecture design selections.
   2. Before the Security Design Specification.
   3. After the Security Design Specification.
   4. Before the system's release.

3. Mention three different aspects to consider at an architecture "sanity check" review.

   Solution:

   1. System purpose vs. Security
   2. Usability Check
   3. Administrative Burden

   (a) For each aspect, list what should be considered?

   Solution:

   1.
      - Does the chosen Security Architecture change any system functions?
      - Are the security functions adding any new system properties?
      - Can these new security functions be accepted the end-users?
   2.
      - Will the security functions have any direct usability considerations?
      - If they do, are they acceptable?

3.
- Is it possible to modify the system to make it easier to use while maintaining security?
- Which routines and security policies are necessary to make the architecture functional?
- How costly is it to maintain the routines and policies?
- How much will the system be dependent on manual routines?

4. Mention three different aspects to consider at an architecture business review.

Solution:

1. Security Control
2. Customer Relations
3. Business Risks and Opportunities

(a) For each aspect, list what should be considered?

Solution:

1.
- Who is in control of the system's security funcctions?
- Will the security control give the party controlling these an advantage?
- If so, make sure the security controls are kept within the business' domains.

2.
- Will the suggested Security Architecture influence any existing or future customer relations?
- If so, this ***MUST*** be highlighted and discussed with the client.

3.
- Will the architecture influence the current business in a direct or indirect way?
- If so, can we make sure the business implications are favorable?

5. What do you perform during a security requirements review?

Solution: The Security Requirements are revisted to ensure the current System Architecture addresses **ALL** the mandator requirements. Any that are missing are added and the Security Design is updated to reflect this.

6. Mention four different aspects to consider at a design review.

Solution:

1. Standards Review
2. Data Representation
3. Protocols
4. Review of New Designs

(a) For each aspect, list what should be considered?

Solution:

1.
   - Are standards used when possible?
   - Are the chosen standards appropriate for this use-case?
   - Are our own design choices compliant with the relevant standards?
2.
   - Is the data format used appropriate for the system's functions?
   - Is the data representation in a protected format when needed?
   - Is the data accessible whenever it is needed?
   - Is the footprint required for securing this data acceptable?
3.
   - Are all new protocol designs evaluated and verified?
   - Are the protocol's performance figures acceptable?
4.
   - In the new designs, do the parts have security soundness?
   - Is ti possible to verify (theoretically or formally) any new design choices?
   - Is the performance of these new design choices acceptable?
   - Has the new design been reviewed by independent experts?

7. Describe a process for issuing and security testing of a software product.

Solution: The software product goes from the owner to the requirements board with the desired result. The requirements board uses these to generate the software's requirements, which is handed off to the engineers. From there, it is designed, implemented, tested, delivered, reviewed, redesigned, etc. until it meets the client's criteria. From there, it moves onto penetration testing, to ensure that it works. Once the software is validated through penetration testing, it is given to the client as a release candidate.

   (a) What is the role in the process for the security officer, the security architect, the security master and security penetration tester respectively?

Solution:

- The Security Officer is the one that generates the first round of security requirements for the product.
- The Security Architect is the one who finds potential vulnerabilities, creates an architecture for them, and ensures that they are handled.
- The Security Master ensures that the security protocols addressed in the Security Requirements are met during developement.
- The Penetration Testers attempt to break into the program by attacking and stealing information that should not be stolen from the program, to determine additional vulnerabilities or items that were missed.

8. Give examples of things to consider during a performance review of a design?

Solution:

- Execution speed of the design and the algorithms chosen.
- The memory usage of the chosen algorithms and their required parameters and keys.
- Is there a need for special hardware?
  - If so, what is the execution speed/memory usage of this hardware?
- Does any part of this Security Design have real-time performance impacts?
- Are there any performance bottlenecks in the design?

9. What is the purpose of trying to "measure" the security of a design?

Solution: To put a quantifiable metric on a solution. By using numbers, we know what we need to design for, which design is the most effective, what changes must be made, etc. We can also track improvements to the system, among other statistics that could be made.

10. A simple security measurement takes three basic security characteristics into account:
    (a) Which three characteristics are then measured and how do you combine the measurements to get an overall measurement of a system?

Solution:

1. Confidentiality
2. Integrity
3. Availability

Each of these is given a score according to a function that is determined from the weighting of the items in that portion of the evaluation tree. These are combined according to another weighted function to produce a final result.

$$\begin{gathered}\langle f_1(\text{Confidentiality}), f_2(\text{Integrity}), f_3(\text{Availability})\rangle \\ g(f_1, f_2, f_3) = w_1 f_1 + w_2 f_2 + w_3 f_3\end{gathered} \tag{5.1}$$

11. Consider the smart card security system in Figure 5.1. Assume the side-channel and physical channel break are equal important and the overall smart card security is the minimum strength of the two nodes. Calculate the smart card security score using the weighted weakest link approach.

Solution:

$$\begin{aligned} WWL &= \sqrt{WS \times WW} \\ WS(\text{of Parent Node}) &= \sum_{k=1}^{n} S_k W_k \\ WW &= \min\left(\frac{S_1}{NW_1}, \frac{S_2}{NW_2}, \ldots, \frac{S_n}{NW_n}\right) \\ NW_k &= \frac{W_k}{\max(W_1, W_2, \ldots, W_n)} \\ W_{\text{Total}} &= \sum_{k=1}^{n} W_k = 1 \end{aligned} \tag{5.2}$$

12. How can the sensitivity for a certain security component be calculated?

Solution:

$$SI_{\text{Component}} = \frac{\partial S(\text{Root})}{\partial S(\text{Component})} \tag{5.3}$$

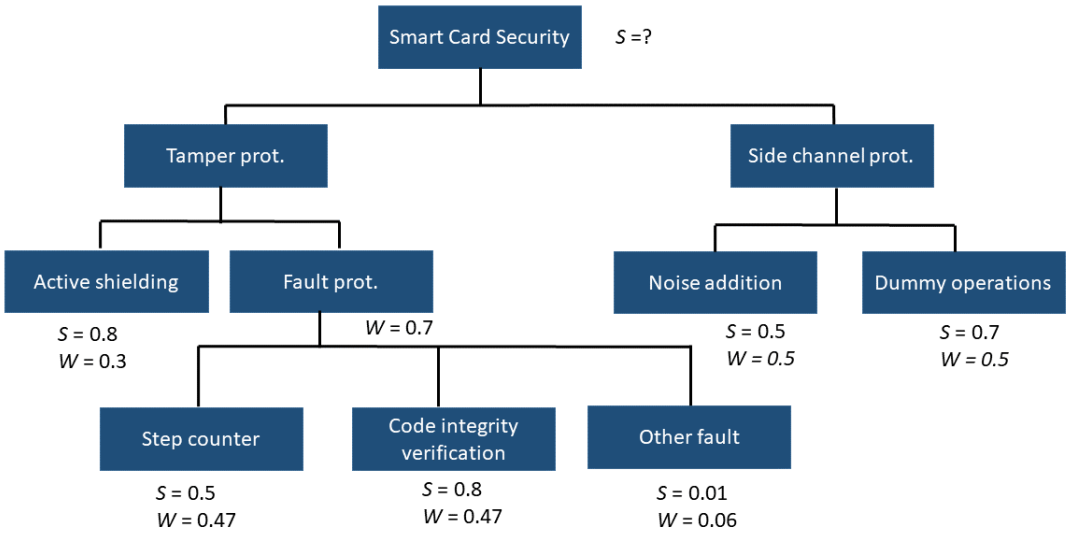


Figure 5.1: Smart Card Security System

13. What is a CVE database? Which organizations maintain global CVEs?

Solution:

C Common
V Vulnerabilities
E Exposures

A CVE is a database of software vulnerabilities and exposure problems that have been recorded and written in a standard language to ensure the software is secured properly.

NIST maintains a CVE, and MITRE does too.

14. What are the three basic categories for which CVE scoring is based

Solution: CVEs are usually scored using the CVSS Scoring Principle.

1. Basic Metric Group
2. Temporal Metric Group
3. Environmental Metric Group

(a) Briefly explain each of the three categories and which security aspects are considered for each of them?

Solution:

1. These are items that are constant over any amount of time and across any user environment. Typically, they are the ways to exploit a vulnerability or exposure or the effects of the vulnerability or exposure.
2. These are vulnerabilities that may change over time. These are typically going to change over time as the ability to exploit this documented code is patched and reviewed.
3. These vulnerabilities are dependent on the particular user's environment. These are unique to every case and cannot be used everywhere.

15. A communication product have three different categories of weaknesses, buffer overflow, TPM weaknesses, and authentication weakness with the CVE list below. Calculate an overall vulnerability score for the product (use the NIST CVE database to obtain the individual scores).
    (a) Buffer overflow: CVE–2019–2304, CVE–2019–2242, CVE–2019–10572
    (b) TPM: CVE–2019–16863, CVE–2018–6622
    (c) Authentication: CVE–2019–3768, CVE–2019–5108,CVE–2019–17627, CVE–2018–5389
16. Explain the terms TOE, PP, ST and EAL used in CC evaluations.

Solution:

TOE Target of Evaluation is the thing that is being evaluated.
PP Protection Profile is an implementation-independent set of security requirements for a category of TOEs.
ST Security Target is a sety of security requirements and specifications that are used in the evaluation of your TOE.
EAL Evaluation Assurance Level is the resulting assurance level that the device/system functions securely as it should.

17. What is the purpose with the PPs?

Solution: Protection Profiles (PPs) can provide a starting place to perform an evaluation on your chosen TOE. These are a collections of many similar TOEs that are used to generate generic system requirements and specifications for other TOEs that fall into the same PP category.

18. What is the main differences between the different EAL levels?

Solution: The biggest difference between the levels is the amount of work that must be put into them to reach a certain level. The higher the level, the more work required. They start with just functioning requirements, then move to semi-formally proving it, then formally proving it.

6 Protocol Analysis

1. What is mutual authentication?

Solution: Mutual authentication is where each communicating end ensures that the person they are talking to is the actual person they are intending to talk to.

2. Which attacks are typically applicable to an authentication protocol?

Solution:

- Impersonation Attacks: One end of the communication pretends to be another principle/actor.
- Man-in-the-Middle Attacks: There is an actor between the 2 valid communicators, potentially alteing messages, $A \leftrightarrow I \leftrightarrow B$.
- Replay Attacks: Parts of previous messages are reused to form a new one.
- Reflection Attacks: A sender send their information to the intended receiver expecting a response, and the attack send the initiators message right back to them.
- Unknown Key-Share Attacks: Different actors in the system have diffeerent views of their peers.
- Oracle Attacks: Reponses in the protocol's messages are used as encryption and decryption services, essentially using the protocol to the attacker's advantage.
- Type Flaw Attacks: Use the absence of proper message type checking and substitute a message with a different message field.
- Chosen Protocol Attack: A brand new protocol is created to interact with the original protocol, creating a security hole.
- Internal Action Flaws: Perform an attack on the algorithm itself. For example, an Invalid-Curve Attack on an Elliptic Curve algorithm.
- Cryptanalysis: Perform dictionary attacks, attempt to factor numbers, etc. This can be done online or offline.

3. What is a man-in-the-middle attack? How could you prevent it?

Solution: A man-in-the-middle attack is where there are 2 parties that are honestly trying to communicate with each other. However, the messages are not secured in a way that prevents tampering, so the attacker masquerades as the other party, receiving the valid message, forwards it onto the intended recipient after changing the source entity. This way both honest actors are communicating with each other (there may or may not be modifications to the messages), but it is done *through* the man-in-the-middle. This attacker may see the information, may alter it, etc. but the bigger problem is that this attack is almost undetectable unless something else is done with the protocol.

The best way to prevent it would be to find discontinuities in the message flow. So, if a message is supposed to take a certain amount of time from one host to the other, but it takes much longer, i.e. through a third host longer, then there may be a MITM attack. This could also be prevented by ensuring the messages are sent with the contents and recipients encrypted and/or protected by other means, preventing a potential MITM from changing information without it being known.

4. What is a replay attack? How could you prevent it?

Solution: A replay attack is where portions of previous messages are reused in new messages. For example, this means that if a session key is transmitted in plaintext in a protocol without any

additional verification, then the attacker could reuse an old session key and send it back during another instance of the same protocol running. This will mean the attacker and the honest host will communicate with each other, rather than the host and the server.

This can usually be prevented by attacking timestamp information, to ensure that the information provided is fresh.

5. What is a reflection attack? How could you prevent it?

Solution: A reflection attack is one where an attacker takes a message that someone just sent and send the message right back to the original sender. If the sender is expecting a response to their initial message, then the protocol might continue to function, using the original message as the response, compromising the security of the protocol itself.

This is harder to prevent, but attaching the intended recipient to the message, and having the actual recipient compare them is the only way to prevent these.

6. What is a labeled multiset rewriting rule? What are l, r, a?

Solution: A labeled multiset rewriting rule is a tuple where a set of states on the left are mutated into a set of states on the right. This is done with an action fact that is added to the trace, allowing for tracing of the actions that have already taken place.

$$l \xrightarrow{a} r \tag{6.1}$$

l Left side. This is the set of states that the rule must have in order to run.
a Action Fact. This is the fact that is recorded in the trace when this multiset rule is executed.
r Right side. This is the set of states that the rule puts back into the multiset after execution.

7. What is a state agent fact $St\_R\_s(A, id, \ldots)$?

Solution: A state agent fact is a fact regarding the state of a single agent in a protocol.

$$St\_R\_s(A, id, \ldots) \tag{6.2}$$

R The role that this agent plays in the protocol. Are they an initiator or a responder?
s What step is this? 1, 2, 76, etc.?
A What is this agent's name? This is typically a public piece of information. This also identifies the actual agent throughout the protocol's execution.
id This is a unique ID that is given to each agent in the protocol execution to allow for easier tracing.
$\ldots$ An agent state fact can have as many additional pieces of information necessary. This can be thought of as the information that the host must carry around throughout the protocol in order to let the protocol work.

8. What are `In` and `Out` facts? What are `Send` and `Recv` action facts? When do you have them?

Solution: The `In` and `Out` facts are ones that communicate messages through the network. Both of these only take in one (1) piece of information, the message. The `In` fact can only be used on the l portion of Equation (6.1). The `Out` fact can only be used on the r portion of Equation (6.1).

The `Send` and `Recv` are action facts the record *who* sent *what*. Additional information could also be recorded, if it is contained within a tuple. These are contained in the a portion of Equation (6.1). `Send` is used when an actor `Out`s a message to the network. `Recv` is used when another actor `Ins` a message from the network.

9. What is a protocol rule? What is an action fact?

Solution: A protocol rule formalizes the *roles* of the protocol. They define the sending and receiving of messages for use in agent state facts to keep track of each role's activities.

An action fact is a fact that is recorded in the trace as a set of actions taken during each protocol rule's execution.

10. What is fresh rule? What is $Fr()$ fact?

Solution: A fresh rule is one that takes in an empty set of states from the multiset and outputs a new set of states to the multiset. It creates new, fresh terms/facts. It does not have any preconditions (the empty initial set) and is the only rule allowed to create new information.

$$[] \rightarrow [Fr(N)] \tag{6.3}$$

The $Fr()$ fact indicates that the parameter is fresh, unique, and completely random. These facts are assumed to be unguessable.

11. What is infrastructure rule? How do you write the key generation for PKI? How can you generate private/public keys and publish public keys using Fr, Ltk, Out, PK facts?

Solution: An infrastructure rule is used to formalize the generation of cryptographic information, typically before the protocol begins execution of communications.

$$[Fr(sk)] \xrightarrow{GenPKI(A,sk,pk(sk))} [Ltk(A, sk), Pk(A, pk(sk)), Out(pk(sk))] \tag{6.4}$$

The generation of the private secret key is essentially a random number, which is modeled by a $Fr(sk)$ fact. The secret key is a long-term key fact about the agent who generated it, and is "registered" as the secret key for this agent by the $Ltk()$ fact. The public key is found by running the public key fact/function $Pk()$ which "registers" a given public key to the given agent. The public key is then shared through the network by the $Out()$ fact.

12. What is an initialization rule? How do you write the initialization rule for a given protocol (e.g. a public key-based protocol)? What is Create action fact?

Solution: An initialization rule creates an agent/actor for a given role in a protocol. They are typically written like so;

$$[Fr(idA), Ltk(A, skA), Pk(B, pkB)] \xrightarrow{Create(A,idA,R)} [St\_R\_1(A, idA, skA, pkB), Ltk(A, skA), Pk(B, pkB)] \quad (6.5)$$

The Create action fact records the explicit creation of an agent in a given role, with their generated id.

13. What is the meaning of well-formedness? How could you write protocol rules that are well-formed?

Solution: Well-formedness is a set of requirements needed to make an executable protocol. To be well-formed, a protocol rule must increment the step count for a role by *just* 1. In addition, all information present on r in Equation (6.1) must be derivable from all information given by public valies, the input agent state fact(s), any fresh fact, and any `In` fact. To be shown mathematically, we need

$$St\_R\_r(A, id, k_1, \ldots, k_n) \in l$$
$$St\_R\_s(A, id, k'_1, \ldots, k'_m) \in r \text{ where } m \geq n$$

To be well-formed, all terms

$$\{k'_1, k'_2, \ldots, k'_m\} \cup \{t | \texttt{Out}(t) \in r\}$$

are derivable from the terms

$$PV \cup \{k_1, k_2, \ldots, k_n\} \cup \{u | \texttt{Fr}(u) \in l\} \cup \{v | \texttt{In}(v) \in l\}$$

AND

$$s = r + 1$$

14. How can you write protocol rules for a given protocol?

Solution: Realistically, any which way you want. But, you want also make it readable for others, so you will want to have a good set of adversary rules that the attacker must follow, a good set of infrastructure rules to model the initial protocol setup, and a good set of protocol rules for the protocol to follow. All the protocols rules must be well-formed to be executed. An agent in each role must have an initialization rule, and each preshared key used should have an infrastructure rule.

15. Assume that you are given a public key-based protocol. How could you write the initialization and protocol rules for it? How could you prepare the protocol and split the roles?

Solution:

$$[Fr(sk)] \xrightarrow{SecretKey(X,sk)} [Ltk(X, sk), Pk(X, pk(sk)), Out(pk(sk))] \quad (6.6a)$$

$$[Fr(idA), Ltk(A, skA), Pk(B, pkB)] \xrightarrow{Create(A,idA,I)} [St\_I\_1(A, idA, skA, pkB), Ltk(A, skA), Pk(B, pkB)] \quad (6.6b)$$

$$[Fr(idB), Ltk(B, skB), Pk(A, pkA)] \xrightarrow{Create(B,idB,R)} [St\_R\_1(B, idB, skB, pkA), Ltk(B, skB), Pk(A, pkA)] \quad (6.6c)$$

Now, Equation (6.6a) creates the private keys (sk) and public keys (pk) on demand, whenever a new user is required. Then, the initiator is created in Equation (6.6b), who needs the public key of B to move onto the next stages. It is also given an ID of idA. Last, Equation (6.6c), the reponder is created, given their keys, and given the public key of A.

16. What is protocol instrumentation? What is a claim event $Claim\_claimtype(A, t)$?

Solution: Protocol instrumentation is an attempt to formalize security properties of protocols independent of any single protocol. To do this, there are claim events that are placed in the a portion of Equation (6.1) to indicate a claim of some type. These are then recorded in the trace, and are there only for proving properties of the protocol. They cannot be observed, modified, or generated by the adversary.

A claim event $Claim\_claimtype(A, t)$ makes the claim that something of *claimtype* is occurring. In this case, A is the one executing this claim over the data t.

17. What is secrecy?

Solution: Secrecy is a claim event in protocol instrumentation for ensuring that some data t cannot be learned by an attacker, even if they get their hands on the message. This does have some constraints, mainly that the secret keys not be leaked, meaning the agents are honest during hte protocol's execution.

Formally, a term t is *secret* for an agent A in role R if and only if whenever A executes R and **believes** to be communicating with honest agents t will not be inferable form the adversary's knowledge.

18. How is the role instrumentation for secrecy? What is $Claim\_secret(A, M)$? Where do you place the hexagon for secret (M) in role instrumentation for secrecy?

Solution: The role instrumentation for secrecy is shown below.

$$\forall A\ t\ i.\ (Claim\_secret(A, t)@i) \Rightarrow \neg(\exists j.\ K(t)@j) \vee (\exists B\ k.\ Rev(B)@k \wedge Honest(B)@i) \quad (6.7)$$

This means that for all traces tr, for all agents A, for all data messages t, at all times i, we can make the claim of secrecy for A, t if: There does not exist a time where the adversary has the message $(\neg\exists j.\ K(t)@j)$ **OR** there exists another agent B at time k whose keys are revealed when they are supposed to be honest.

Typically, the hexagon is placed at the end of communications for that agent. So, you can make the claim for each agent running in a protocol, but it may not be true everywhere.

19. What is a compromised agent? When an agent is honest? What are `Honest` and `Rev` action facts?

> **Solution:** A compromised agent is one whose secret keys have been leaked to the network.
>
> An agent being honest means that their secret keys have not been leaked.
>
> The `Honest` action fact makes an assertion about the state of another agent's keys and the `Rev` action fact states whose secret keys have been revealed. `Honest` is the assertion by one agent that another has *not* leaked their secret keys. The `Rev` action fact is the act of an agent revealing their secret keys.

20. How can you verify if secrecy claims hold for a given protocol? (See examples in slides 31–34 of lecture 9. See also an exercise here).

> **Solution:** You must run through the protocol and attempt to find any cases where information may be given to the attacker, or find any cases where the secret keys for the agents in the protocol are leaked through the execution of the protocol.
>
> For the secrecy of something to *not* hold, the data supposed to be secret may not have to be altered. For example, if
>
> $$A \to B : A, \{N_A\}_{pk(B)}$$
> $$B \to A : B, \{N_A\}_{pk(A)}$$
> $$Secret(A, N_A)$$
>
> then an attacker could take the first message $A, \{N_A\}_{pk(B)}$, replace the A with K, creating $K, \{N_A\}_{pk(B)}$. Now, B would respond to K with the message $B, \{N_A\}_{pk(K)}$. This means that the secrecy of N_A does not hold for A, because K also has it now.

21. What is forward secrecy?

> **Solution:** Forward secrecy is the property that the secrecy of a message t is still secret, even while communications are ongoing, except for if the adversary has previously compromised an agent that is required to be honest. Formally, this is the set of all traces tr that satisfy
>
> $$\forall A\ M\ i.\ (Claim\_secret(A,t)@i) \Rightarrow \neg(\exists j.\ K(t)@j) \vee (\exists B\ k.\ Rev(B)@k \wedge Honest(B)@i \wedge k < i) \quad (6.8)$$

22. How can you find out that a given protocol provides forward secrecy? (See examples in slides 36–37 of lecture 9).

> **Solution:** The only way is to "run" the protocol and see if the adversary is able to compromise a session key given they have already compromised one or more agents in the protocol. If the adversary can compute a single-use session key given they know one or more long-term private keys, then forward secrecy is not possible.

23. How is the role instrumentation for authentication? What are *Claim_commit* and *Claim_running* events? Where do you place Commit and Running hexagons when A wants to agree with B? Where do you place them when B wants to agree with A?

Solution: The role instrumentation for authentication is a bit more complex, because there are multiple types of authentication possible. The placement of the hexagons when A wants to agree with B is the "running" portions go on action facts that exist on B and A performs the commit. When B wants to agree with A, the "running" portions go on A's action facts and B commits to them. Listed from weakest to strongest assertions, there is Aliveness, Weak Agreement, Non-Injective Agreement, and Injective Agreement.

Aliveness simply says that the agent a running in role X has aliveness with another agent that a **thinks** is b running in role Y if whenever a completes an execution of the protocol with b, then b was previously running the protocol. This is weak because there is no timeframe on when b was last running the protocol, nor on who b really is.

$$\forall a\ b\ i.\ Claim\_alive(a, b, \langle\rangle)@i \Rightarrow (\exists id\ R\ j.\ Create(b, id, R)@j) \vee (\exists X\ r.\ Rev(X)@r \wedge Honest(X)@i) \quad (6.9)$$

Weak Agreement says nearly the same thing as aliveness, but makes the stipulation that when a was executing the protocol with b, b was running the protocol at the same time with someone b **thinks** is a.

$$\begin{aligned}\forall a\ b\ i.\ &Claim\_weakAgreeCommit(a, b, \langle\rangle)@i \Rightarrow \\ &(\exists j.\ Claim\_weakAgreeRun(b, a, \langle\rangle)@j) \\ &\vee (\exists X\ r.\ Rev(X)@r \wedge Honest(X)@i)\end{aligned} \quad (6.10)$$

Non-Injective Agreement guaranteed that a and b were executing the protocol with each other, a was acting as role X and b was acting as role Y, and that they both agreed on some message t.

$$\begin{aligned}\forall a\ b\ i.\ &Claim\_nonInjCommit(a, b, \langle 'I', 'R', t\rangle)@i \Rightarrow \\ &(\exists j.\ Claim\_nonInjRun(b, a, \langle 'I', 'R', t\rangle)@j) \\ &\vee (\exists X\ r.\ Rev(X)@r \wedge Honest(X)@i)\end{aligned} \quad (6.11)$$

Injective Agreement is the same as Non-Injective Agreement, except for the fact that the execution of the protocol between a and b must be unique, i.e. the run we are referring to in a is the **exact same** run as in b.

$$\begin{aligned}\forall a\ b\ i.\ &Claim\_injCommit(a, b, \langle 'I', 'R', t\rangle)@i \Rightarrow \\ &(\exists j.\ Claim\_injRun(b, a, \langle 'I', 'R', t\rangle)@j \\ &\wedge \neg(\exists a_2\ b_2\ i_2.\ Claim\_injCommit(a_2, b_2, \langle 'I', 'R', t\rangle)@i_2 \vee \neg(i_2 = i))) \\ &\vee (\exists X\ r.\ Rev(X)@r \wedge Honest(X)@i)\end{aligned} \quad (6.12)$$

24. How do you model a protocol using Tamarin? How do you write a labeled multiset-rewriting rule in Tamarin?

Solution: Tamarin uses labeled multiset rewriting rules, like the ones shown in Equation (6.1). To model Equation (6.1) in Tamarin, it is written as shown below.

```
rule Basic:  [l] --[a]-> [r]
```

25. What are linear and persistent facts in Tamarin? When do you use `!` or `~` or `$` in Tamarin?

Solution: By default, all facts in Tamarin are linear. This means that when they are used in a protocol rule during execution, they can only be used once on the left-hand side l. Persistent facts are ones that can be used any number of times, and they shouldn't be consumed.

Persistent facts are denoted with the `!`.
Fresh information is denoted with the `~`, and should only be used when the agent **KNOWS** that the value should be fresh.
The `$` denotes the information after it is publicly available. Typically, this is used with agent names, because the ability to identify agents in inherent in the system.

26. What does $\langle x, y \rangle$ mean in Tamarin?

Solution: This is a tuple. All of the information within the angles is "concatenated together" to and is then handled as a single unit. If a function only takes one argument, but you want to give 2 pieces of information, the angles allow you to "put them together" and pass it to that function.