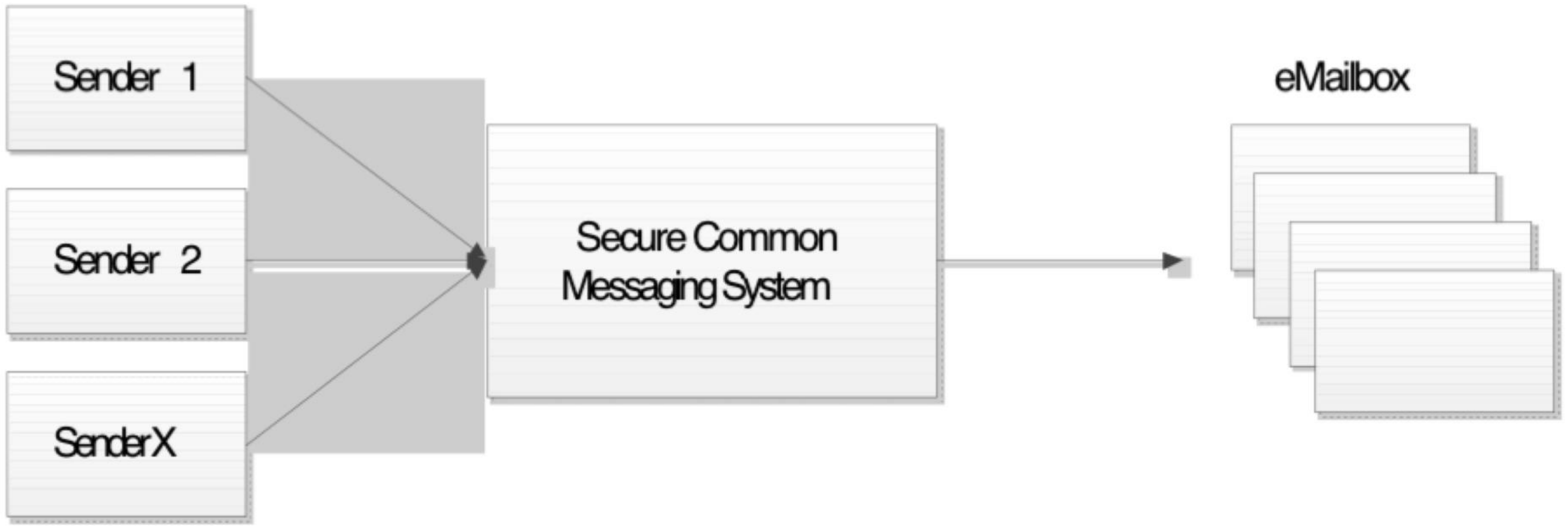
TDT4237 Xam 2013 walk-through

Task 1- 40%



Above is a high-level sketch for a system which purpose is to provide secure messaging services, to send messages from government to citizens. Depending on the sender, these messages may contain personal and sensitive data, so it is important for the system to be secure. The messages should only be readable by the intended receiver and it is very important also to protect the integrity of the messages.

Task 1 contd.

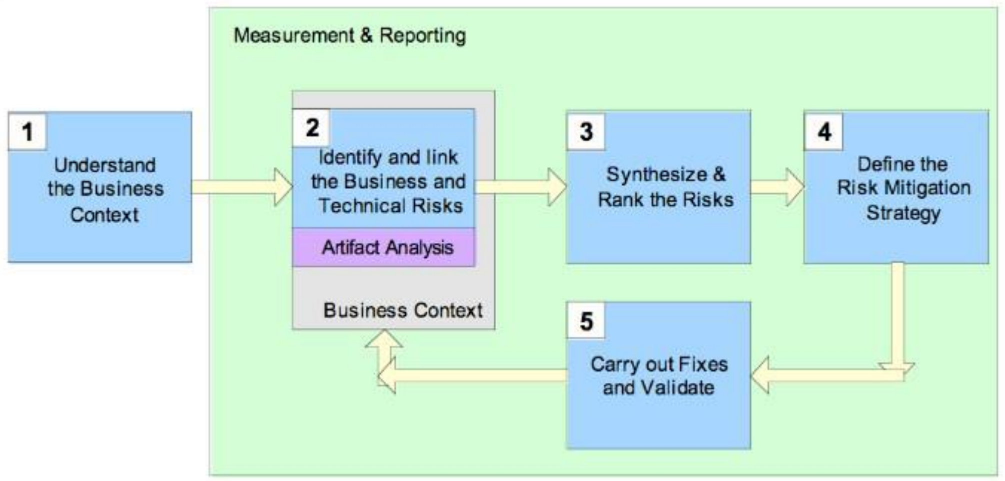
The central component in the system has been given the name Secure Common Messaging System (SCMS). The purpose of the SCMS is to enable secure communication of messages from government to citizens:

- By providing addressing services for citizens. So government need only know who they are sending a message to, but not to which address or mailbox. A citizen may have one or several mailboxes.
- By providing services that ensures that government can know, with reasonable certainty, that messages have been delivered to, and opened by, the citizen.
- The SCMS should not be able to see the content of messages from government to citizens.

The task of your project is to create security requirements for the SCMS using the Risk Management Framework.

Solution expectations:

Based on the questions from the exam day we should expect the students to have different understandings of what this is – a transparent messaging system (what I was thinking) or something web-based. Or something else. All are ok. It's how they figure out the security requirements, and if the requirements fits their understanding of the case, that matters (most – it is not ok if they miss important aspects of the system because they have not read the task carefully).
To create security requirements the students need to complete steps 1-4 of the RMF. The risk mitigation strategy represents the security requirements. We've covered this in class. On the exercises they have only used the RMF to create a risk-based test plan – not requirements. But, it is part of exams from previous years that they have access to.


Measurement & Reporting
1
Understand the Business Context
2
Identify and link the Business and Technical Risks
Artifact Analysis
Business Context
3
Synthesize & Rank the Risks
4
Define the Risk Mitigation Strategy
5
Carry out Fixes and Validate

From a question I have gotten on email after the exam, it seems some students believe that they only need to complete business risks to deduce requirements, since the requirements should not be too specific – i.e. a good requirement is “The transmission of messages through the SCMS should be encrypted” but not “..it should be encrypted using the AES algorithm with a 256-bit key”. I don’t agree – I don’t see how they can get good requirement from business risks.

Note: this task counts for 40% of the total score. We’ve had similar tasks previous years, so it should be pretty straight-forward for the students to understand what we are looking for. But, they have had less time on this task in previous years (accounted for less of the total). This year, since the exercises also counts on the final score, I’ve tried to give them fewer tasks and hence more time per task on the exam. That also means that we are expecting better/more thorough answers. As always we are looking for both demonstration of method knowledge and meaningful content. The requirements need to make sense, have to be explained, and though the list need not be complete – we should expect substance.

Task 2 (30%)

a) When creating a software system, when and for what would you use threat modeling?

Solution

The guest lecture on this mentioned many threat modeling dialects, but we have focused on attack trees, misuse cases and (somewhat) data flow diagrams. They have used misuse cases and/or attack trees on exercise 2 as a help when designing new functionality, so I expect them to answer something related to that.

In summary - threat modeling may be used to:

Deduce security requirements from functional requirements (misuse cases)

Analyze threats towards a system (attack trees and misuse cases)

Do risk analysis (attack trees)

Plan security testing (attack trees)

Map a system’s attack surface (data flow diagrams)

Task 2 contd.

b) How does an XSS-attack work? Explain using an example.

Solution

The goal of a cross-site scripting attack is to inject code so that this code is run by a user visiting a legitimate site. There are two main types of cross-site scripting attacks:
Reflected XSS – when the code is pasted into a URL and the user is tricked into visiting the site using this USL. This attack only works on a per-user basis but may be more targeted towards specific users.
Stored XSS – when the code is injected into the site, e.g. in a comment field, and run when a user visits that part of the site. This attack affects all users of a site.
An example of a cross site scripting attack is when javascript code is injected that can read from the document object and (for example) write the content of a user's session cookie to a log ol on a different site, effectively giving the attacker the ability to reuse that cookie information.

Task 2 contd.

- c) What is a one-time pad?

A one time pad is, if used correctly, an encryption key that makes it impossible to break the resulting encrypted text. The concept is that you use for encryption a key that is:
The exact same length as the plaintext that shall be encrypted
Truly random – i.e. not guessable of predictable
Only used once and then discarded
Secret

Task 2 contd.

- d) What is the attack surface of a system?
- *The attack surface of a system is an overview of any place an attacker can get access to, and launch an attack towards, a system.*

Task 2 contd.

e) What is the purpose of creating a risk-based test plan?

The purpose of creating a risk-based test plan is to:
Prioritize your efforts – deciding what is most important to test, given limited resources
Ensuring that the tests you perform are specific and relevant to the system you are testing

Task 2 contd.

f) What are the main pros and cons of manual code review vs code review with a tool?

Manual code review is:
Con: time consuming and expensive
Pro: the best way of identifying subtle vulnerabilities, such as race conditions, in code
Pro: specific to your system/code

Code review with a tool is:
Con: prone to false positives
Con: but false negatives are worse - getting a "green flag" does not mean that you don't have any issues in your code – only that what this tool tests for is not a problem in your code
Pro: resource-efficient – can be done many times with little effort
Pro: often integrates with IDE – can be used by developers

Task 3 (30%)

Authentication and password management is a key feature of many systems, but also something easily done wrong. For this task we will use an online ticket ordering system as our case. In this system a user has to create an account in order to be able to order tickets.

Create a password policy for the ticket ordering system.

Explain how you would:
Store passwords.
Perform password checks.
Allow users to reset their password.

You may provide pseudo-code examples to explain your thinking.

Solution expectations

Note: the parts (a and b) do not count 50% each. A total score is given on this task depending on the total quality of the answer.

Password policies have been discussed in class. We've talked about the different pros and cons of short but complex password vs long passphrases. The system (a bookstore) the students have been working on in the exercises also had a very poor password policy (none at all – a password of "a" was accepted), so this is something they had to fix in exercise 2.

Here we are looking for a password policy that includes requirements on the password quality (length, symbols etc). It may also include how often a password have to be reset. Many policies are possible, but we are also looking for a good explanation for the chosen policy. The students probably have to make some assumptions about the system to explain their policy. An answer that is simply a policy and no explanation is not a good answer.
This they also had to do in the exercises.

Solution expectations

Password resetting was implemented by some, but not all groups. Many solutions are possible here, expected is maybe using and e-mail link. Quality of the answer depends on how they reason and explain their choices. NOTE: the text for this task says that you may use pseudocode. Many asked about this on the exam – if it was a requirement. I answered that it is not. The text clearly says "may". So, pseudocode is a bonus if it helps explain, but the lack of pseudocode is not a negative. Also note that in part 2 of the exercises, when designing and adding new functionality to the application, we specifically asked the students to use threat modeling. So, I am hoping to see some threat modeling here.

Solution expectations contd.

We are expecting the students to know about secure password storage (hashed, using a random salt). The original exercise code used a static salt value.

Those who didn't catch the problem with the salt did receive feedback on this and it was also published in the vulnerability list for the application that the students were given before exercise 3 (hacking another group's fixed application).
They should also know how to validate a password.Demonstrate knowledge of best practice. Important: the students should not suggest any kind of decryption of what is stored in the password database (not possible with hashing, I know – but be aware that some may suggest to "encrypt" and store the passwords – not really understanding the difference between hashing and encryption).